

A decorative graphic on the right side of the page. It features three sets of concentric circles in shades of blue. Two thin blue lines originate from the top left and extend diagonally towards the circles. The circles are positioned at the top right, middle right, and bottom right of the page.

Transformation (SAP) after mapping has been done.

PRGX Transform

After the loading is done, the next step will be to transform the data and then run the process package and then test the website created after the process package.

**agapar01
1/7/2013**

Table of Contents

Transformation (SAP) after mapping has been done.	I
Data Extraction Best Practices	1
Source System Impact.....	1
Create Staging Database.....	4
Creating Tables in Staging databases.....	4
VendorMaster Transform	9
OLE DB Source.....	10
Derived Column	12
Data Conversion.....	14
Lookup.....	14
Destinations	16
OLE DB Destination	17
Saving the Package.....	19
Executing the Package	20
Data Viewers	20
Payments.....	23
Lookup 1.....	24
Look Up	27
OLE DB Destination	28
InvoiceHeader	29
OLE DB Source.....	30
Derived Column	30
Data Conversion.....	31
Lookup 1.....	31
Lookup.....	33
OLE DB Destination	36
PaymentLink.....	37
OLE DB Source.....	37
Derived Column	39
Data Conversion.....	39
Lookup1.....	40
Lookup.....	42
OLE DB Conversion.....	44

Checking for Duplicates and Other Checks	46
SQL Common Errors	47
G. Functions on indexed columns in predicates.....	47
Consol.....	49
Moving Data from Staging to Consol	49
Companies.....	55
Payments.....	62
Derived Column	64
OLE DB Destination	64
Common Data conversion Errors.....	66
Date in form dd.mm.yyyy Error	66
Text Qualifiers' Errors	69
Creating Consol and ASP Database for APTrax	70
Going Live.....	80
Avoiding Misleading Reports	81
Uses of Process Package reports	81
Running Statistics for loaded Files	83
Backup Database using SQL Server Management Studio.....	84
Restore Database.....	85
Best Practices.....	88
Extracting Data from Sources	88
Appendices.....	89
Companies Table	89
InvoiceHeader Table	90
Payments Table	91
PaymentLink Table	92
VendorMaster Table	93
Working with Common Expressions	95
Database Copy	97
Quick Reference of SQL Commands	101
SAP Tables	102
Website testing (http://Servername/APTraxNameof Site).....	103
Session filters	103
Vendor.....	104

AP Stats	106
Cash Management	108
Compliance	110
Tax.....	113
Master Data	114
Duplicates	116
Identify Dupes.....	117
Investigate Dupes.....	118
Statement Management.....	120
Claims Management	121
Workflow Reports	122

Before diving into transformation, we will take a brief look at data extraction best practises. Extraction of source data is arguably the most important part of an ETL process, because everything afterward depends on its success. It is of great interest to everyone with any involvement with transformation.

Data Extraction Best Practices

Data extraction is the process of moving data off a source system, potentially to a staging environment, or into the transformation phase of the ETL. An extraction process may pull data from a variety of sources, including files or databases systems.

The data you extract will likely be kept in your staging database for validation and user drill-through reporting purposes. There likely is some extra information from the source system that isn't strictly required, but could improve the meaningfulness of the extracted data.

Source System Impact

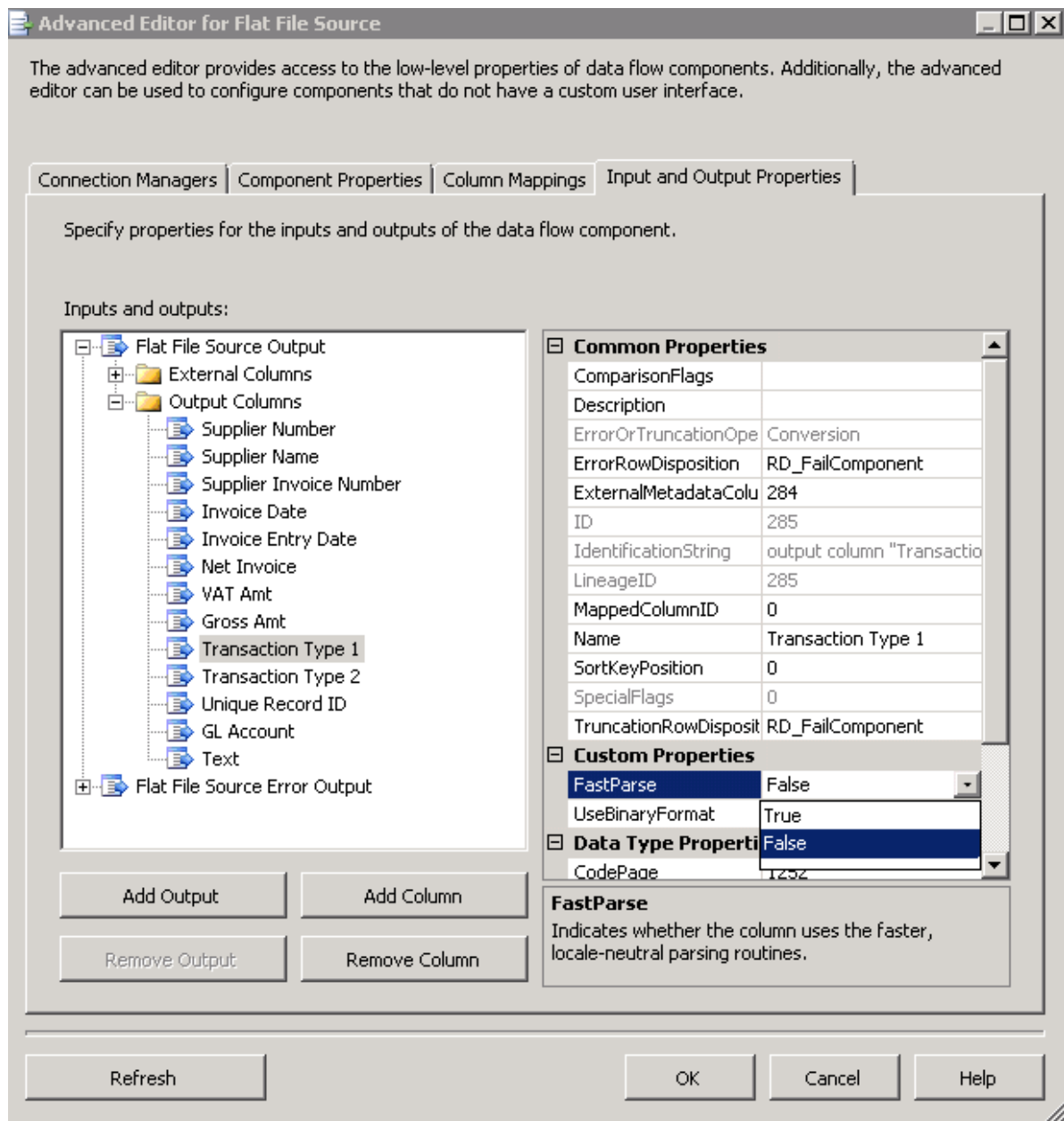
A good ETL process will be kind to its source systems. For these sources to be heavily loaded transactional systems, and maybe a legacy mainframe system to boot, is not uncommon. Here are some generally principles for creating an extraction process with minimal impact on the source system:

- Bring the data over from the source with minimal transformations. Don't use expressions or functions that execute on the source system. The less the source system has to think about your query, the better. The exception is if the source uses a data type that your extraction process doesn't understand. In this case, you may need to cast the data to another type.
- Keep selection criteria simple. You may need to find a balance between the cost of transferring extra rows, and the cost of filtering them out in the extraction query. Functions or expressions that must think about every single row in the source table would not be good.
- Do not use joins.

Flat-File Source

Flat files are a very common extraction source. This is because many source systems will push data to flat files that can be consumed by another process in a loosely coupled manner. The flat-file source adapter is a powerful mechanism to bring data into the data flow.

Flat files come in many different structures and formats. SSIS supports delimited, fixed-width, and ragged-right, where the last column may have an undetermined number of characters. These can be set in the advanced properties of the connection. Two important features of the flat-file source give the source adapter extra flexibility and great performance. First the flat-file supports various code page formats. This can be found in the general tab of the Flat file connection Manager Editor. Second property of the flat file adapter is the FastParse property available for date, time, and integer source columns.



The FastParse property of output columns (found in advanced Editor, “Input and Output Properties” tab) reduces the conversion overhead of each column as it is imported into the dataflow buffers, and overall gives generous performance benefits. By using the Fast Parse option, you are giving up the capability to translate the value to the local specific data, and you are forced to have date column formatted in YYYYMMDD or YYYY-MM-DD format. But, if you are dealing with large files, you will want to work through these limitations to take advantage of the setting. In essence, you are telling the connection that the data can be trusted in the source column to be in a standard format. One easy practical method is to let all dates be Varchar and then use change them using the Conversion task.

OLE DB Source

The PLE DB source adapter provides the most flexibility, because it allows several mechanisms to customize the source query, including parameterization and variable bound queries.

Excel Source

The limitation is that they are only supported in a 32-bit execution mode.

One thing that may cause you issues when extracting from Excel sources is that all the text columns are pulled back as Unicode (DT_WSTR) data types in SSIS. Even if you try to modify the advanced input and output properties of these columns, the adapter will not accept a change to these types. Therefore another approach is required to maintain any native string (DT_STR) types.

To change the data type of your columns, you must use the Data Convert Transformation, and add new columns for each Unicode column that needs conversion back to native strings.

Anytime you need to change the data type of a pipeline column, SSIS requires a new column to be added to the buffers, because the data type of an existing column cannot be changed in place. Therefore, new columns are added to the pipeline with adjusted data types. The extra columns make the rows wider and the buffers less efficient.

Create Staging Database

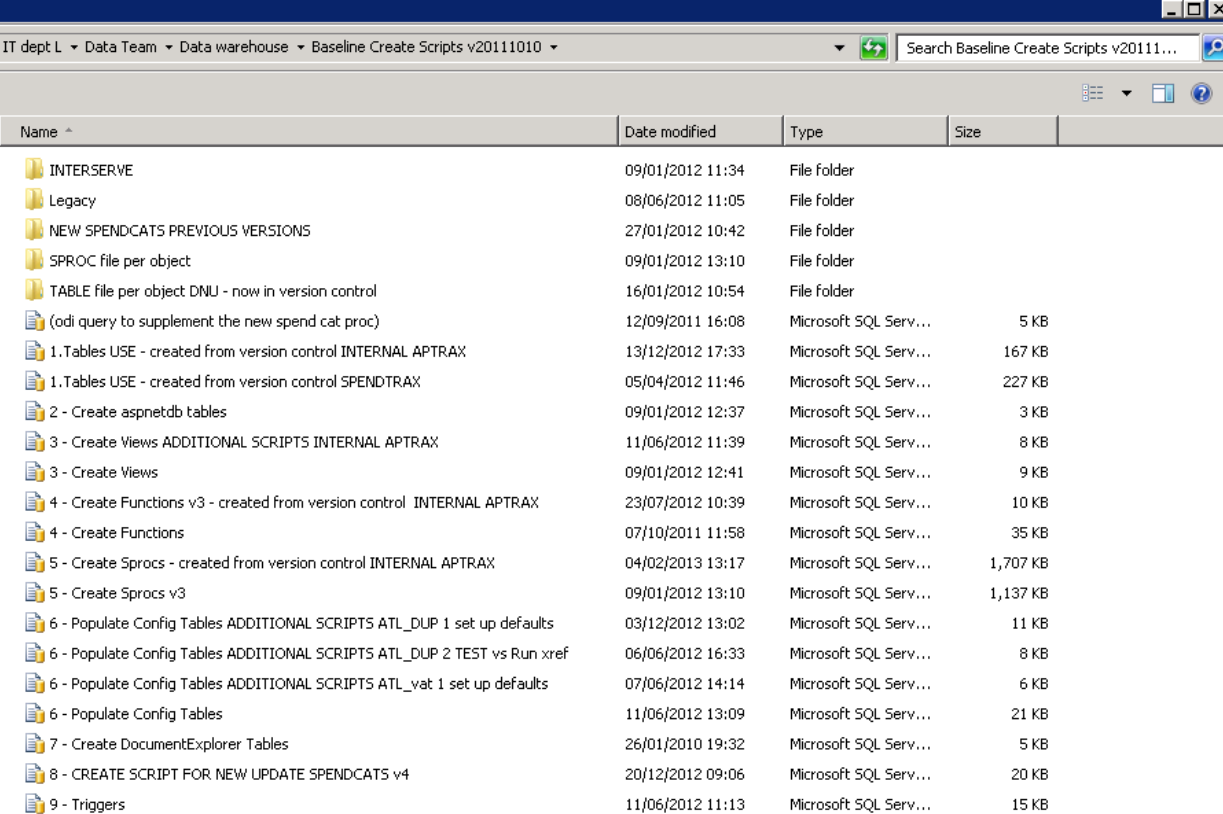
- In SQL Server Management Studio (SSMS) go to Object Explorer and click on databases.
- Right Hand Click and Select New Databases. For the naming convention it is like the following <<CompanyName>>Staging e.g. RoyalMailStaging.
- For Owner, leave as <default>
- Click **Add** to create your database.

Creating Tables in Staging databases

The tables to be created are VendorMaster, Payments, InvoiceHeader, PaymentLink and sometimes Companies. Companies table is created when the Company has several accounting divisions.

To create identical tables used by the Company, you need to script out similar tables from the baseline scripts.

[\\192.168.3.16\library\IT dept L\Data Team\Data warehouse\Baseline Create Scripts v20111010](http://192.168.3.16/library/IT dept L/Data Team/Data warehouse/Baseline Create Scripts v20111010)



Name	Date modified	Type	Size
INTERSERVE	09/01/2012 11:34	File folder	
Legacy	08/06/2012 11:05	File folder	
NEW SPENDCATS PREVIOUS VERSIONS	27/01/2012 10:42	File folder	
SPROC file per object	09/01/2012 13:10	File folder	
TABLE file per object DNU - now in version control	16/01/2012 10:54	File folder	
(odi query to supplement the new spend cat proc)	12/09/2011 16:08	Microsoft SQL Serv...	5 KB
1.Tables USE - created from version control INTERNAL APTRAX	13/12/2012 17:33	Microsoft SQL Serv...	167 KB
1.Tables USE - created from version control SPENDTRAX	05/04/2012 11:46	Microsoft SQL Serv...	227 KB
2 - Create aspnetdb tables	09/01/2012 12:37	Microsoft SQL Serv...	3 KB
3 - Create Views ADDITIONAL SCRIPTS INTERNAL APTRAX	11/06/2012 11:39	Microsoft SQL Serv...	8 KB
3 - Create Views	09/01/2012 12:41	Microsoft SQL Serv...	9 KB
4 - Create Functions v3 - created from version control INTERNAL APTRAX	23/07/2012 10:39	Microsoft SQL Serv...	10 KB
4 - Create Functions	07/10/2011 11:58	Microsoft SQL Serv...	35 KB
5 - Create Sprocs - created from version control INTERNAL APTRAX	04/02/2013 13:17	Microsoft SQL Serv...	1,707 KB
5 - Create Sprocs v3	09/01/2012 13:10	Microsoft SQL Serv...	1,137 KB
6 - Populate Config Tables ADDITIONAL SCRIPTS ATL_DUP 1 set up defaults	03/12/2012 13:02	Microsoft SQL Serv...	11 KB
6 - Populate Config Tables ADDITIONAL SCRIPTS ATL_DUP 2 TEST vs Run xref	06/06/2012 16:33	Microsoft SQL Serv...	8 KB
6 - Populate Config Tables ADDITIONAL SCRIPTS ATL_vat 1 set up defaults	07/06/2012 14:14	Microsoft SQL Serv...	6 KB
6 - Populate Config Tables	11/06/2012 13:09	Microsoft SQL Serv...	21 KB
7 - Create DocumentExplorer Tables	26/01/2010 19:32	Microsoft SQL Serv...	5 KB
8 - CREATE SCRIPT FOR NEW UPDATE SPENDCATS v4	20/12/2012 09:06	Microsoft SQL Serv...	20 KB
9 - Triggers	11/06/2012 11:13	Microsoft SQL Serv...	15 KB

Paste the URL above in RUN box on the remote desktop to get like above. Next you double-click **Table file per object DNU – now in version control**.

You then look for the table you want to script out.

If you have Management Studio open, you can then double-click the required table script. When the new query window opens you have 2 ways to create the tables you want.

- On the new query window that opens, you will see at the top **USE [HomeBaseConsol] GO**. The first option you will comment out these 2 lines and then select your database from the available databases drop down window. The second option you simply change the name of the database inside the brackets to your own. i.e. USE[HomeBaseConsol] to **USE[RoyalMailStaging]**. NB. We are using consol tables not staging tables so as to avoid mismatch between the tables. APTrax uses these tables as they are. Please do not try to change these tables in any way.
- Click **Execute** to execute the command.

Repeat the above procedure for the 4 tables InvoiceHeader, Payments, PaymentLink and Companies, if the companies table is required.

- When you finish creating the tables, go to the top of the window and close the query. This can be done by right hand clicking the query name and then selecting close.

After all the tables have been created, we can now move on with the transform of the various tables. The order of the transform does not matter as long as the PaymentLink is done last since it is populated from the payments and InvoiceHeader. The SQL part of the transform will be done after the process has been highlighted.

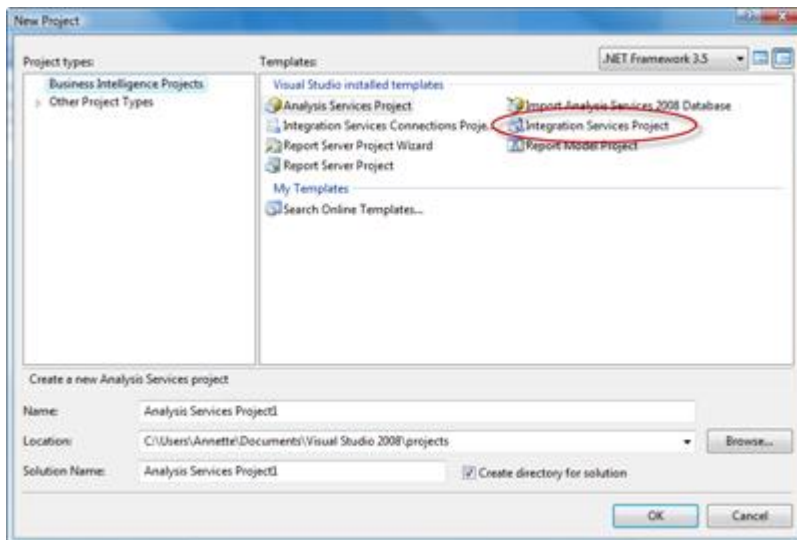
Integration services

When you first open BIDS, you're presented with the interface as shown below.



The SSIS interface in BIDS

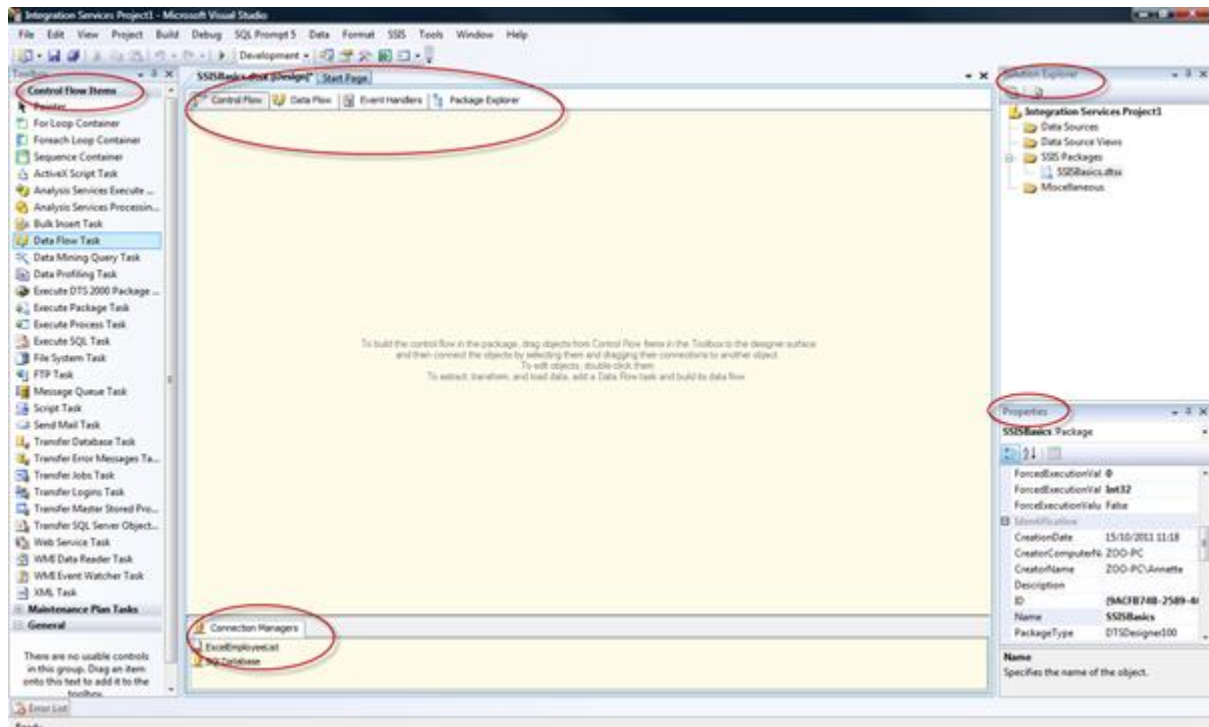
To create an SSIS package, point to the File menu, point to New, and click Project. This launches the New Project dialog box, as shown below.



The New Project dialog box in BIDS

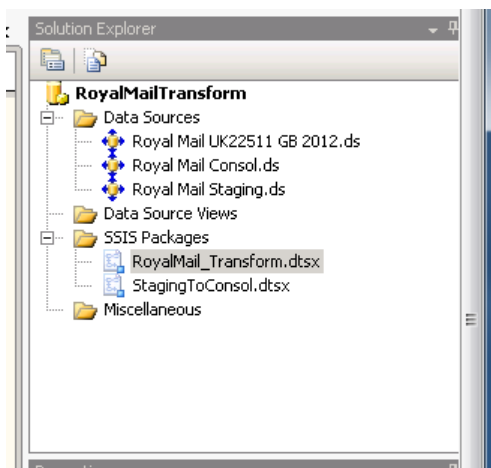
In the New Project dialog box, select the Integration Services Project template. Then, provide a name for the project in the Name text box. Next, in the Location text box, specify the folder where your project files should be saved, and then provide a name for the solution in the Solution Name text box. SSIS Projects are saved in ukchellut01, **Projects** Folder.

After you've entered the project and solution information, click OK. Your new package will open in the SSIS window, as shown below.



Creating a new SSIS package in BIDS

Notice that the SSIS interface is divided into the following five sections (windows): Control flow, Connection Managers, Solution Manager, Properties and SSIS Designer.



After creating the project and you'll see a default **Package.dtsx** package file in the Solution Explorer. Right-click the **Package.dtsx** file in the Solution Explorer and select **Rename**. Rename the file e.g. RoyalMail_Transform.dtsx. When you're asked if you'd like to rename the package object as well, click **yes**. If the package isn't opened yet, double-click it to open it in the Package Designer.

Now that you have the package ready, you need to create a shared connection that can be used across multiple packages. In **Solution Explorer**, right-click **Data Sources** and select **New Data Source**. This opens the Data Source Wizard. Select the "Create a Data Source based on an existing or new connection" radio box and click **New**, which opens the window to create a new Connection Manager.

There are many ways you could have created the connection. For example, you could have created it as you're creating each source and destination. Find what works best for you.

Control Flow

- Click Toolbox and drag Data Flow Task to the design window.
- Click Data flow task box and right-click it and then select rename. Rename it VendorMaster. When you start on payments, you can name it payments and so on, to signify the process you will be doing.
- Double-click the VendorMaster. This will open a window on dataflow.
- Click on toolbox and under data flow sources drag **OLE DB Source** to the design window.

The data flow is made up of 3 components sources, transformations and destinations. Sources, extract data out of flat files, OLE DB Databases and other locations. Transforms, process the data once it has been pulled out, and destinations, write the data to the final location.

A source is where you specify the location of your source data. Most sources will point to the Connection Manager in SSIS. By pointing to the connection manager, you can reuse connections throughout your package, because you only need change the connection in one place.

After dragging OLE DB Source to the Data Flow area; you need to do the following:

Derived Column

Click on toolbox and under Data Flow Transformations drag Derived column to the design window.

Data Conversions

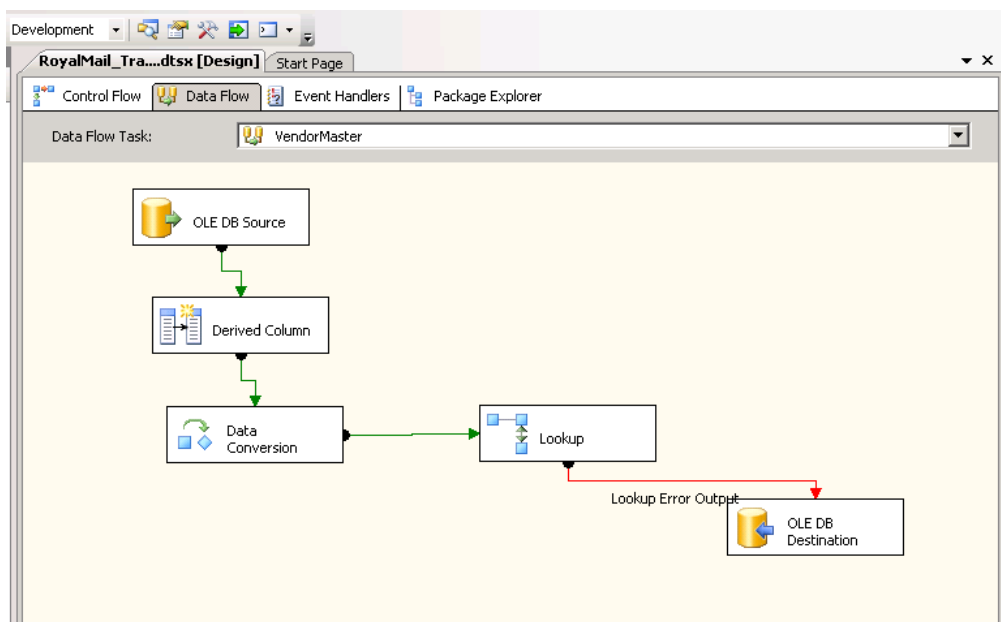
Click on toolbox and under Data Flow Transformations drag Data Conversion.

Lookup

Click on toolbox and under Data Flow transformations drag Lookup to the design window.

OLE DB Destinations

Click on toolbox and under Data Flow Destinations drag **OLE DB Destinations** to the design window. Click on OLE DB Source and then drag the green arrow to the Derived column. The Derived Column's green arrow to the Data Conversion box, Data Conversion green arrow to lookup. On lookup you then drag the RED arrow to the OLE DB Destination. It should then look like this:



Red Arrow

If you drag the red arrow, then rows that fail to transform will can be directed to that target. (You will need to redirect rows after failure of a defined condition) After you have the transform connected, you can double-click it to configure the transform. In our case above, we can define to look for duplicates; when it does have a failure to find any duplicates; it proceeds along the red path.

Green Arrow (precedence constraint)

If everything is successful, this is the path that will be followed. This precedence constraint controls the execution order of the various tasks in your package and can be used by dragging the green arrow to the next task that you want to chain together.

Blue Arrow: Only on completion of task, will it then proceed to the next step.

Any colour with FX logo.

The precedence constraint can be different colours to represent different commands. They can also have an FX logo to represent an expression. Placing expressions on precedence constraints gives you more advanced ways to control the execution of each package.

TIP:

Experts advise to never keep the default names of the tasks. You should rename them to something that you can recognize in the log later. This name shows up in your logs later when a problem occurs and can also help you self-document your package.

VendorMaster Transform

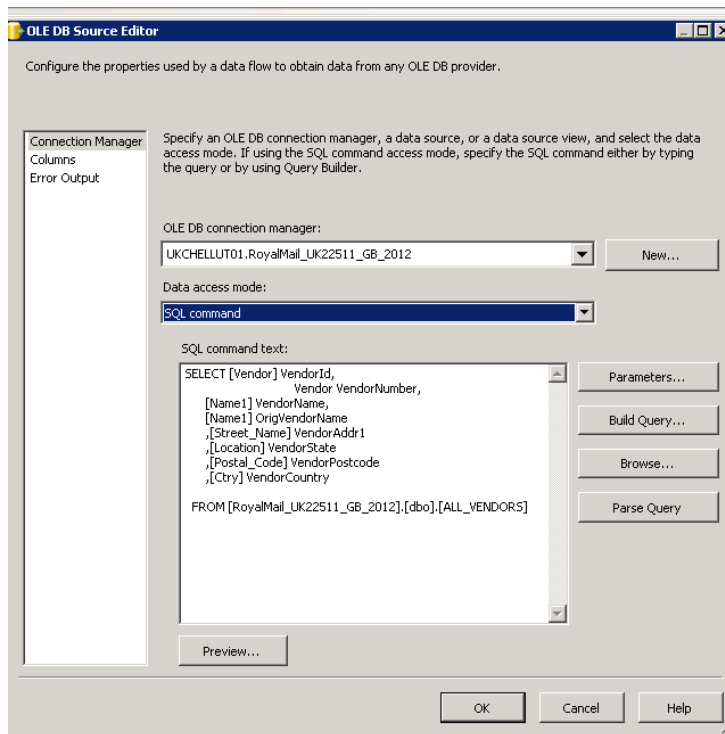
In general the vendor contact details are required here. To find out which files have the required details, we get that from the mapping documents. The mapping document is coloured coded. The mapping document also has the valid date range on top.

The blue shaded boxes in the output column are required. The blue field names in Raw Data are nice to have and the ghosted field names in output field name column are not required. The SQL queries will need to be written to output the required fields. This query will be required in the transform package. More details about the mapping instructions and the commonly used SQL queries will be covered later.

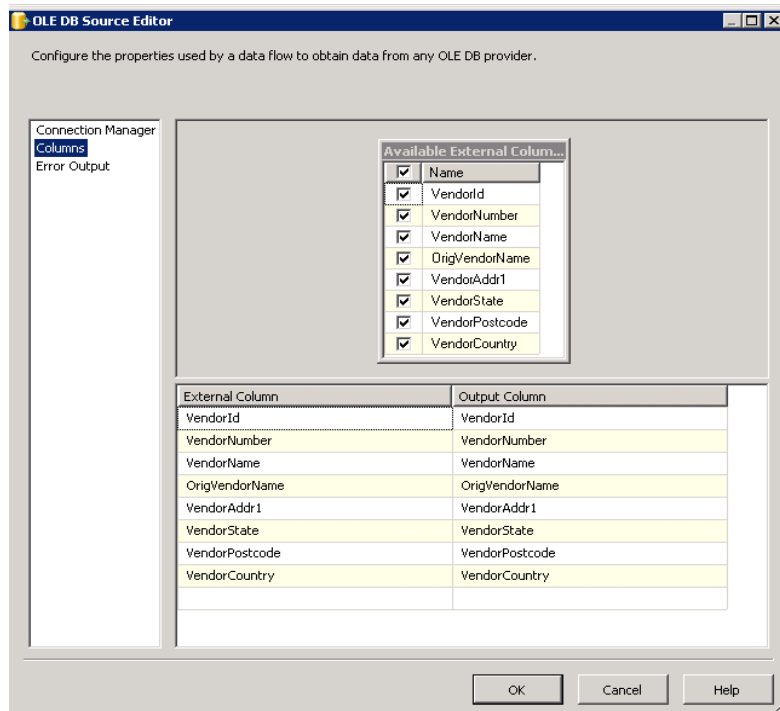
A VendorMaster script is in appendix.

OLE DB Source

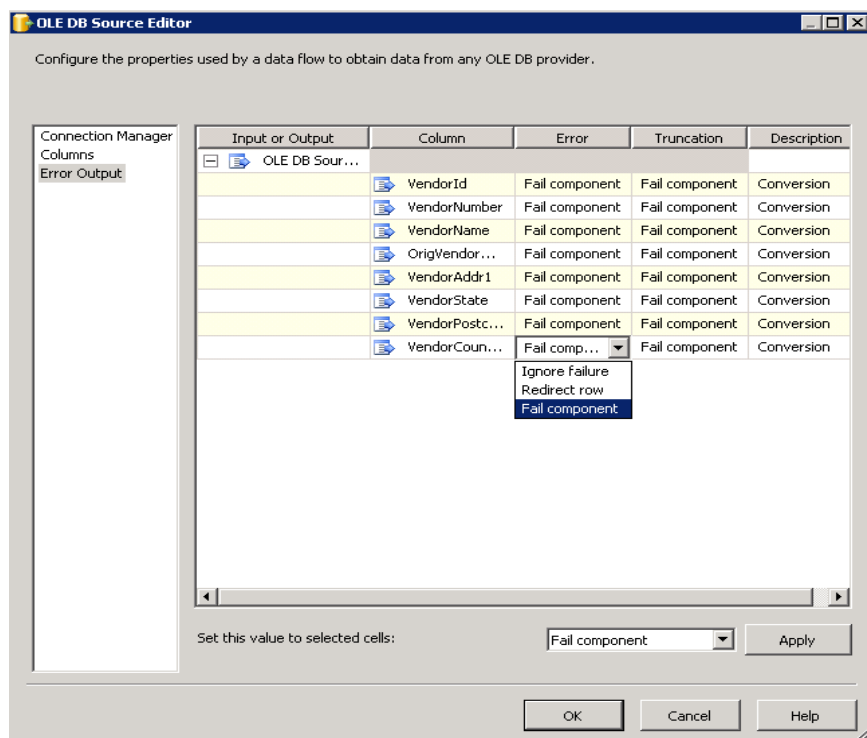
The OLE DB Source is the most common type of source, and can point to any OLE DB compliant Data Source such as SQL Server, Oracle, or DB2. To configure the OLE DB source, double-click the source.



- On connection manager we need to link with our database. If it is new we select new and then select the name of our database.
- Data access mode: It is an option for setting how we can retrieve the data. The options here are Table/View or SQL Command, or you can pull either from variables. Once you select the Data Access Mode, you will need the table or view, or you can type a query. It is a best practise for multiple reasons that we retrieve data from a query. This query can also be a stored procedure.
- SQL Command text: Copy your VendorMaster query and then paste it here. Remember you cannot copy and paste between servers. Nothing goes onto the clipboard. After pasting the query click **Parse Query** to make sure it has the right SQL syntax. Click **Preview** to see how it comes out. If you have run it before in SSIS you should not have any problems.
- Click on columns. These are how the columns will appear. If there are any columns that you do not want to appear you can deselect them in "Available external columns". You will not have this problem if you have written the query and not used Table or view / Table name or view name variable under Data access mode. You can also assign the name you wish to send down the Data Flow in the Output column. In short, it is better from a performance perspective to have typed the query in Connection manager page versus selecting a table. Selecting a table to pull data from essentially selects all columns and all rows from the target table, transporting all of that data across the network. Then, going to the Columns page and un-checking the unnecessary columns applies a client-side filter on the data, which is not nearly as efficient as selecting only the necessary columns in the SQL query.



- Error Output: We can specify how to handle rows that have errors. The default is Fail component, which means the transform stops (fail) when there is an error. You can change it to **ignore failure** or **redirect row**. If you choose to ignore failures, the column for that row will be set to NULL. If you choose to redirect the row, the row will be sent down the red path in the data flow coming out of the OLE DB source. The truncation column specifies what to do if a data truncation error occurs. A truncation error would occur for example, if you try to place 100 characters of data into a column in the Data Flow that only supports 50. You have the same options available for your Truncation as you do for the Error option. The redirection is normally done on debugging, but you need to remember to reset to default after finishing the debug process, otherwise you get unexpected results.



Derived Column

Transformations are broken down into two main categories: synchronous and asynchronous. Synchronous transformations are components like the Derived column and Data Conversation transforms where rows flow into memory buffers in the transform and the same buffers come out. No rows are held and typically these transforms perform very quickly with minimal impact to your Data Flow.

The derived column contains all the columns coming from the OLE DB Source. You can add new columns or modify some of the columns in derived column. If you select one of the columns flowing into the transformation, you can use an expression to modify the value in some way. The expression column is a blank slate upon which you write the code that controls value. You can hard code a value, such as a space, by enclosing a space inside double-quotes.

You may wish to use this transformation, for example, to multiply the quantity of orders by the cost of the order to derive the total cost of the order. You can also use it to find the current date or to fill in the blanks in the data by using the ISNULL function. This is one of the top five transforms that you'll find yourself using to alleviate the need for TSQL scripting in the package.

To configure, drag the column or variable into the Expression column. Then add any functions to it. A list of functions can be found in the top-right corner of the Derived Column transformation Editor. You must then specify, in the Derived Column drop-down box, if you want the output to replace an existing column in the Data Flow or to create a new column. As you can see below, the second derived column is doing an in place update of the quantity column. The expression states that if the Quantity column is null, then convert it to 0, and otherwise keep the existing data in the Quantity column. If you create a new column, specify the name in the Derived Column Name column.

Derived Column Name	Derived Column	Expression	Data Type
TotalPrice	<add as new column>	[Quantity]*[ActualCost]	Numeric[DT_Numer..
Quantity	Replace 'Quantity'	ISNULL([Quantity])=True?0: [Quantity]	Four byte signed int...

Derived Column Name	Derived Column	Expression	Data Type	Length	Precision	Scale
InsertDate	<add as new column>	GETDATE()	database timestamp [DT_DBTIMESTAMP]			
LedgerId	<add as new column>	1	four-byte signed integer [DT_I4]			
CompanyRef	<add as new column>	"Multi"	Unicode string [DT_WSTR]	5		

In most cases the derived column will look like this.

InsertDate

This is the date when this was done. It helps to trace back changes and errors.

LedgerID

It is the unique identifier for Ledger / ledgers used by the organisation. (A ledger is the principal account file for recording and totalling monetary transactions by account, with debits and credits in separate columns and a beginning balance and ending balance for each account). **If there is one ledger and multiple business units share the same vendors, then the fact table joins the dimensions tables on LedgerID. If there are multiple ledgers the join is also on LedgerID.**

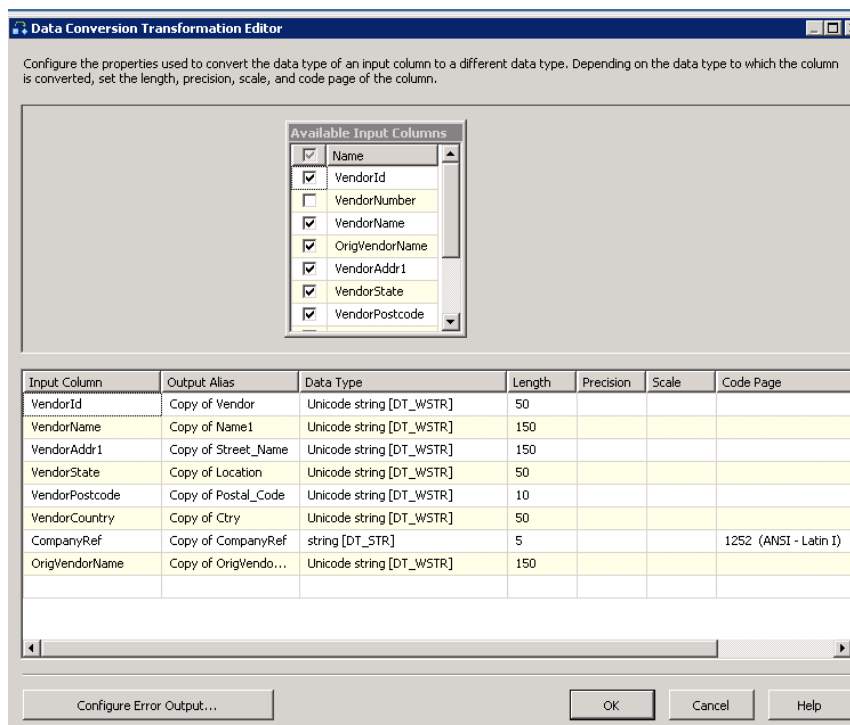
CompanyRef

With the combination of LedgerID and CompanyRef, they have to be unique in each table. The InvoiceHeader has LedgerID, CompanyRef and Invoiceld, Payments table has LedgerID, CompanyRef and PaymentId. PaymentLink has LedgerID, CompanyRef, PaymentId, Invoiceld and PayItem. The VendorMaster has LedgerID, Company Ref and VendorId.

Data Conversion

The data conversion Task helps us convert data from one type to another. A very common problem is with original excel sources. SSIS supports Excel data types. The problem is, the default excel format is General, so most columns in Excel are set to General. SSIS translates this general format as a Unicode data type. In SQL Server, Unicode translates to Nvarchar which is probably not what you want. If you have Unicode data type in SSIS and you try to insert into a Varchar column, it will potentially fail. The answer is then to place a Data Conversion Transform between the source and the destination in order to change the Excel data types.

In our transforms the columns that we use in our input column are the columns that are causing Unicode conversion errors. The output alias gives us copies of input columns after they have been converted. You will need to map the converted data types, which appears as “copy of data” and map it to the destination. There is a drop down window to the right of the data type window where you can select the data type.



Lookup

Lookup allows you to include columns from a reference table to the output. Useful to also correlate sources, by providing the capability to augment columns from a relational table to the pipeline based on matching source key columns. It can also be used as a filter mechanism to identify matches and non-matches.

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
 Connection
 Columns
 Advanced
 Error Output

Cache mode

☒ Full cache
☐ Partial cache
☐ No cache

Connection type

☐ Cache connection manager
☒ OLE DB connection manager

Specify how to handle rows with no matching entries

Redirect rows to error output

OK Cancel Help

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
 Columns
 Advanced
 Error Output

Specify a data source to use. You can select a table in a data source view, a table in a database connection, or the results of an SQL query.

OLE DB connection manager:

UKCHELLUT01.RoyalMailStaging New...

☐ Use a table or a view:

New...

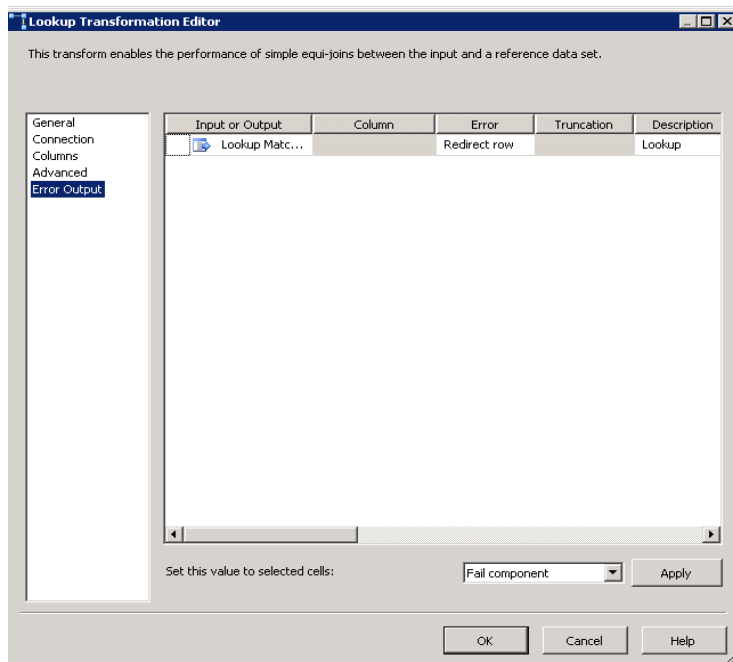
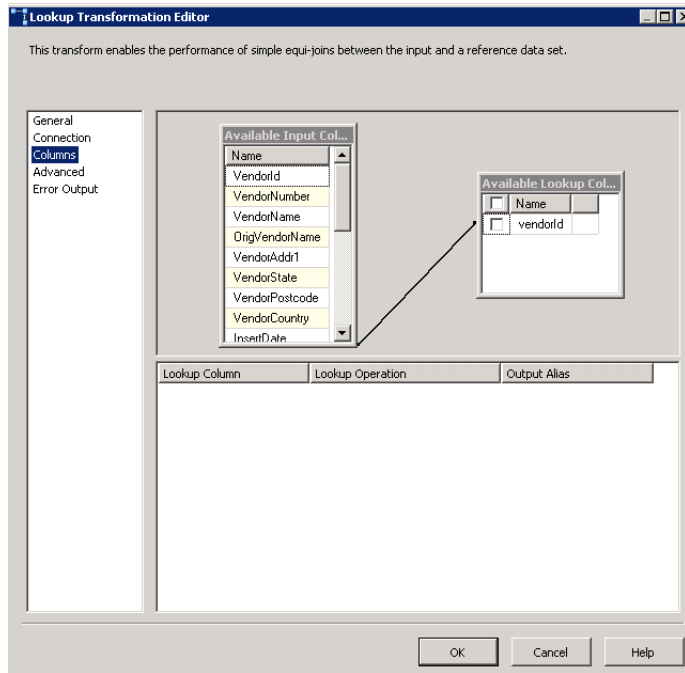
☒ Use results of an SQL query:

SELECT vendorId FROM vendorMaster

Build Query...
 Browse...
 Parse Query

Preview...

OK Cancel Help

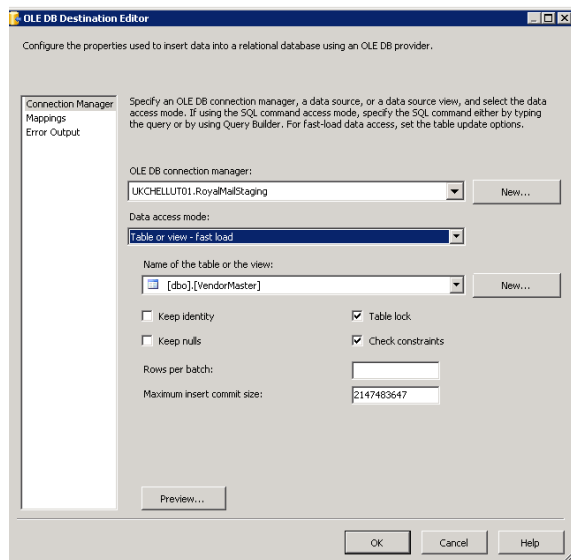


Destinations

Inside the Data Flow, the destinations accept the data from the Data Sources and from the transformations. The architecture can send the data to nearly any OLE DB-compliant Data Source, flat file, or Analysis Services, to name just a few. Like sources, destinations are managed through the Connection Manager. The configuration difference between sources and destinations is that in destinations, you have a mappings page, where you specify how the inputted data from the Data Flow maps to the destination.

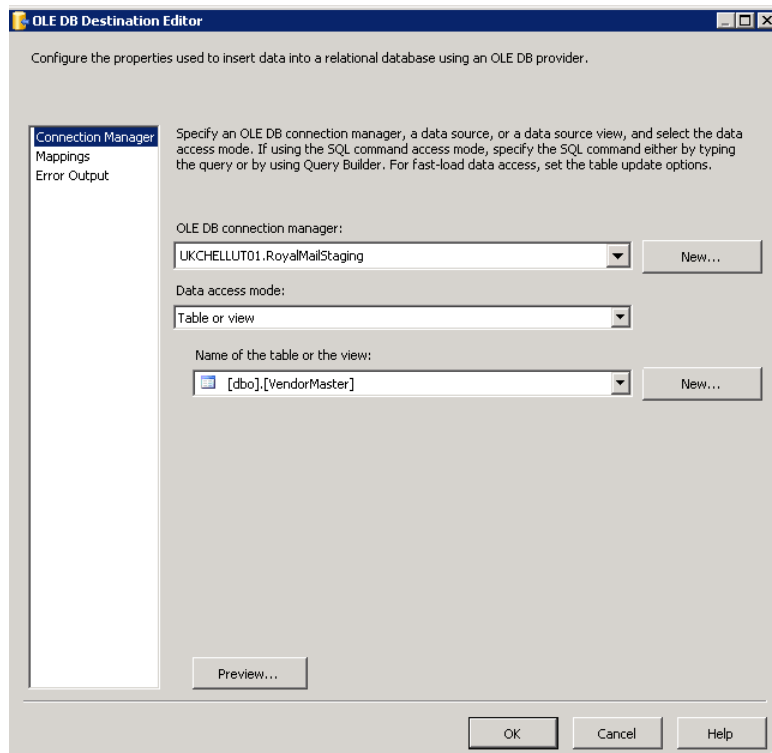
OLE DB Destination

The most commonly used destination is OLE DB Destination. A dynamic option it does have is the Data Access Mode. If you select Table or View –Fast Load, or its variable equivalent, you will have a number of options, such as Table Lock. This fast Load option is available only for SQL Server database instances.

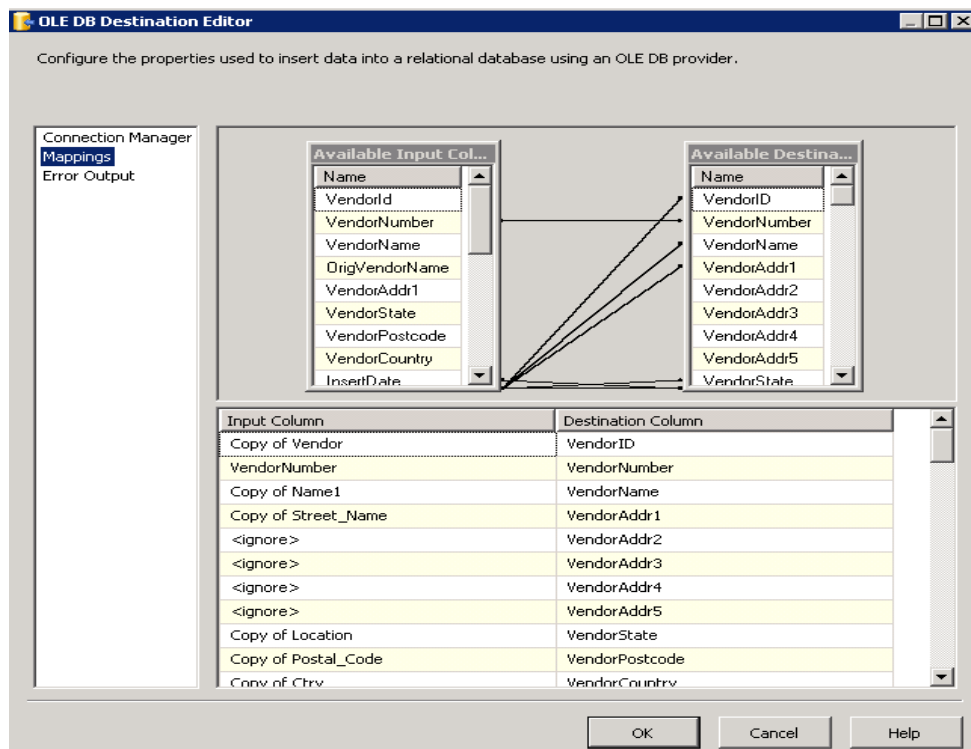


A few options of note here are the Row per Batch option, which specifies how many rows are in each batch sent to the destination, and another option is the Maximum Insert Commit Size, which specifies how large the batch size will be prior to issuing a commit statement. The Table Lock will place a lock on the destination table to speed up the load. As you can imagine, this will only cause grief for users if they're trying to read from the table at the same time. The other important option is Keep Identity. The option allows you to insert into a column that has the identity property set on it. Generally speaking, you can gain performance by setting the Max insert Commit Size to a number like 10 000 but that number may vary based on how wide the columns are.

Now the question is, what is the difference between fast load and the normal load (table or view option) for the OLE DB Destination? The fast load option specifies that SSIS will load in bulk into the OLE DB Destination's target table. Because this is a bulk operation, error handling via a redirection or ignoring of the data errors is not allowed. If you require this level of error handling, you need to turn off bulk loading of the data by selecting Table or View for the Data access mode option. Doing so will allow you to redirect your errors down the red line but will cause a slowdown of the load by a factor of at least four.



The destination connection manager points us to the tables that we have created before. These are the tables that we scripted out earlier. The name of the table or view is given under the name of table. If you would prefer to script it out then you would write SQL command under Data access mode and then paste the query for scripting the table on the SQL command screen.

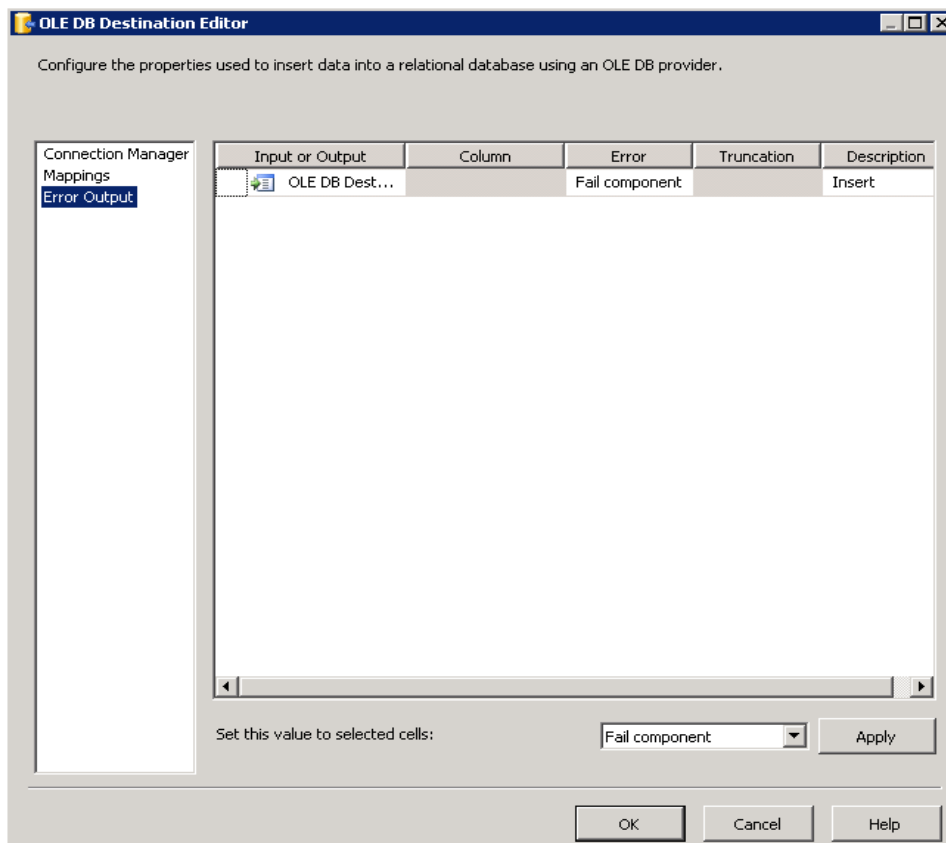


On mappings, we can now map our original table names to the input of our VendorMaster table. If we have converted the columns we will now need to use Copy of.... If we do not have an equivalent input column name (no column name from source) then we can ignore it by leaving it as **<ignore>**

In mappings the columns are mapped automatically based on column names but don't necessarily have to be lined up. You can choose to ignore given columns, such as when you're inserting into a table that has an identity column and don't wish to inherit the value from the source table. We have to make sure all our columns have been mapped correctly; otherwise we might get unexpected results.

You won't be able to configure the destination until it is connected to the Data Flow. To do this, select the source or a transformation and drag the green arrow to the destination. If you want to output the bad data to a destination, you would drag the red arrow to that destination. If you try to configure the destination before attaching it to the transformation or source, you would get an error.

For the error output, it is as expected:



Where an error appears, double-click it to view the errors.

Saving the Package

To save the package, click the **Save** icon in the top menu, or select File >> **Save Selected items**. By clicking Save, you're saving the .DTSX file to the project, but you have not saved it to the server yet. To do that, you'll have to deploy the solution or package.

Executing the Package

When you want to execute a package, you can click the Play icon on the toolbar, press F5, or choose Debug >>Start. You can also execute packages by right-clicking the package in Solution Explorer and selecting Execute Package. This technique may be a better habit to get into because clicking the Play button will initiate a build, and if some properties are selected, it will cause each package to open prior to your package execution. This puts the design environment into execution mode, opens several new windows, enables several new menu and toolbar items, and begins to execute the package. When the package finishes running, BIDS doesn't immediately go back to design mode, but rather stays in execution mode to allow you to inspect any runtime variables or to view any execution output. This also means that you can't make any changes to the objects within the package, but can modify variables and objects' read/write properties.

To get back to design mode, you must click Stop icon on the debugging toolbar, press Shift+F5, or choose Debug>>Stop Debugging.

You can see the progress under the Progress tab or in the Output window. In the Control Flow you'll see the two tasks go from yellow to green (hopefully). If they all turn green, then the package execution was successful. If your package failed, you can look in the Output window to see why. The Output window should be open by default, but in case it's not, you can open it by clicking View >> Other Windows >> Output. You can also see a graphical version of the Output window in the Progress tab (it can also be called the Execution results tab if you package is stopped)

You can go to the Data Flow tab to see how many records were copied over. You can notice the number of records displayed in the path as SSIS moves from transform to transform. Changes you make in this mode will not be made available until you run the package again. You will also not be able to add new tasks or enter some editors.

Data Viewers

They allow us to view data at points in time at runtime. If you place a data viewer before and after the aggregate transform, you can see the data flowing into the transform at runtime and how it looks after the transform happens. Once you deploy your package and run it on the server as a job or with the source, the data viewers do not show because they are only a debug feature. Anytime the package is executed outside the designer, the data viewers won't show.

There are four types of data viewers:

- Grid: Shows a snapshot of the data in grid format at that point in time.
- Histogram: Shows the distribution of numeric data in a histogram graph.
- Scatter Plot: Shows the distribution of numeric data using an x- and y-axis.
- Column Chart: Displays the occurrence count of discrete values in a selected column.

To place a data viewer in your pipeline, right-click one of the paths (red or green arrows leaving a transform or source) and select Data Viewers. The Configure "Data Flow Path Editor" dialog box will appear. Click Add to enter the Configure Data Viewer dialog box, and select the type of data viewer you wish to use and optionally give it a name if needed. You can go the other tab that's named after the type of data viewer you are using to select what columns will be used in the data viewer.

Once you run the package, you will see the data viewers open and populate with data when the package runs the transform that it's attached to. The package will not proceed until you click the > button. You can also copy the data into a viewer like Excel or Notepad for further investigation by clicking Copy Data. The data viewer will show up to 10 000 rows by default, so you may have to click the > button multiple times in order to go through all the data.

As you add more data viewers, you may want to remove them eventually to speed up your development execution. You may remove them by right-clicking the path that has the data viewer and selecting Data Viewers. You could then select the data viewer to remove and click Delete. You can also delete all the data viewers and breakpoints at one time by selecting Delete All Breakpoints from the Debug menu.

Understanding Data types

In SSIS, you simply have to pay attention to the data types, whether the data is coming from your Data Flow, being stored in variables, or being included in expressions. You have to pay attention because the syntax checker will complain. If something in your Data Flow allows incompatible data types, your packages will raise either warnings or errors (if implicit conversions are made). This will happen even if the conversion is between Unicode and non-Unicode sets. Bad data type decisions can have a serious impact on performance.

The following table provides a matrix between SSIS data types and a typical SQL Server set of data types. You will need this to interpret between data stream contents and data types in an expression. The .NET Managed types are important only if you are using script component, CLR, or .NET-based coding to manipulate your Data Flows.

SSIS Data Type	SQL Server Data Type
DT_WSTR	nvarchar, nchar
DT_STR	Varchar, char
DT_DBTIMESTAMP	Smalldatetime, datetime
DT_TIME2	time
DT_BYTES	binary, varbinary
DT_NUMERIC	numeric
DT_DECIMAL	decimal
DT_I2, DT_I4, DT_I8	smallint, int, bigint
DT_R4, DT_R8	Real, float

A common issue in SSIS packages is the improper selection of an SSIS date data type. For some reason DT_DBDATE and DT_DATE seem to be common selections for data types in Data Flow transforms. Improper use of these types can result in overflow errors or the removal of the time element from the date values. Prgx scripts for running statistics do not like to run DT_DBDATE. On type of data they output data type as **40 – Unknown**, while minimum and maximum values are given some funny characters. The idea is that SSIS data types provide a larger net for processing incoming values than you may have in your destination data source.

Wrong Data Types and sizes can affect performance

Data Flow transforms are done in memory. This can be good because it eliminates the most time-consuming IO operations. However, because SSIS uses memory buffers to accomplish this, the

number of rows that can be loaded into the buffer is directly related to the width of the row. The narrower the row means the more rows that can be processed at a given moment. If you are defining the data types of a large input source, pick your data carefully, so that you are not using the default 50 characters per column for a text file, or the suggested data types of the connection manager when you do not need this extra safety cushion. The tables in appendix will give you an idea of how wide eventually the given columns will be pushed into. E.g. VendorAddr Nvarchar (255), Postcode Nvarchar (15), Vendor Tel/ Fax Varchar (50), InvoiceGrossAmount decimal (15,2).

Data conversion is a fact of life, and you will have to pay for it somewhere in the ETL process. The following are just guidelines:

- Convert only when necessary. There is no need to convert all columns from a data source that are going to be dropped from the data stream.
- Convert to the nearest type for your destination source using the mapping files.
- Convert using the closet size and precision. There is no need to import all columns as 50-character data columns if you are working with a fixed or reliable file format with columns that do not require as much space.
- Evaluate the option to convert after the fact. Don't forget that SSIS is still an ETL tool and sometimes it is more efficient to stage the data and convert the data using set based methods.

The bottom line is that data type issues can be critical in high volume scenarios, so plan with these guidelines in mind.

Unicode and Non-Unicode Conversion Issues

SSIS uses Unicode as the default in SSIS packages and in importing. Not only is this the default import behaviour, but all the string functions in SSIS expect Unicode strings as input. At the very least, using Unicode requires an additional step that is frequently missed, resulting in errors.

Note that you may experience some data being replaced by NULLs when importing Excel files using the Excel Connection Manager. This typically occurs when numeric and text data is stored within one column. One solution is to update the extended properties section of the connect string to look this: Extended Properties="EXCEL 8.0;HDR=YES;IMEX=1".

"HDR=Yes;" indicates that the first row contains columnnames, not data. "HDR=No;" indicates the opposite.

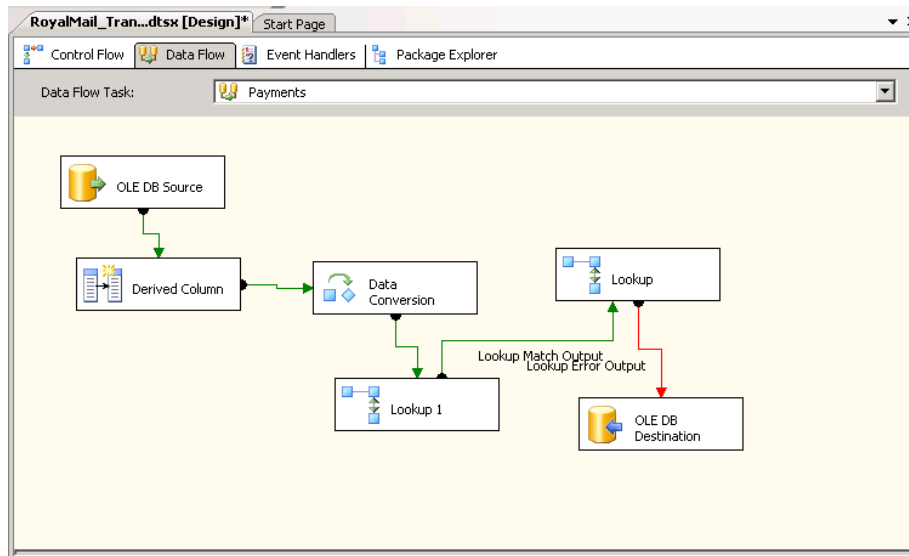
"IMEX=1;" tells the driver to always read "intermixed" (numbers, dates, strings etc) data columns as text. Note that this option might affect excel sheet write access negative.

SQL syntax "SELECT [Column Name One], [Column Name Two] FROM [Sheet One\$]". I.e. excel worksheet name followed by a "\$" and wrapped in "[" "]" brackets.

Check out the [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines\Excel] located registry REG_DWORD "TypeGuessRows". That's the key to not letting Excel use only the first 8 rows to guess the columns data type. Set this value to 0 to scan all rows. This might hurt performance. Please also note that adding the IMEX=1 option might cause the IMEX feature to set in after just 8 rows. Use IMEX=0 instead to be sure to force the registry TypeGuessRows=0 (scan all rows) to work.

Payments

We will use similar methods to transform payments. There may be some minor differences.

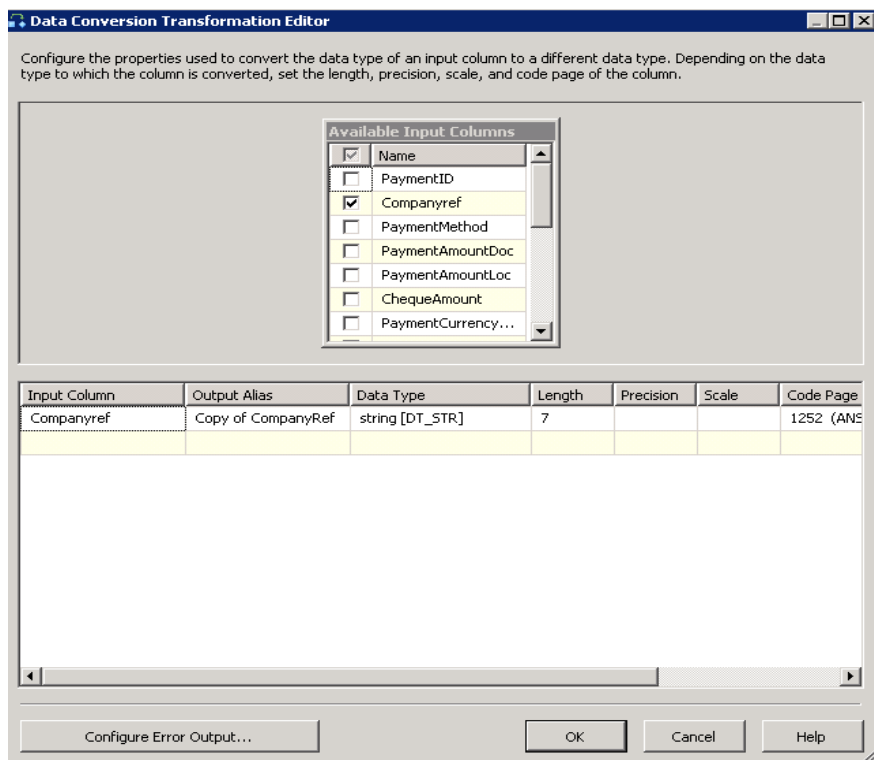
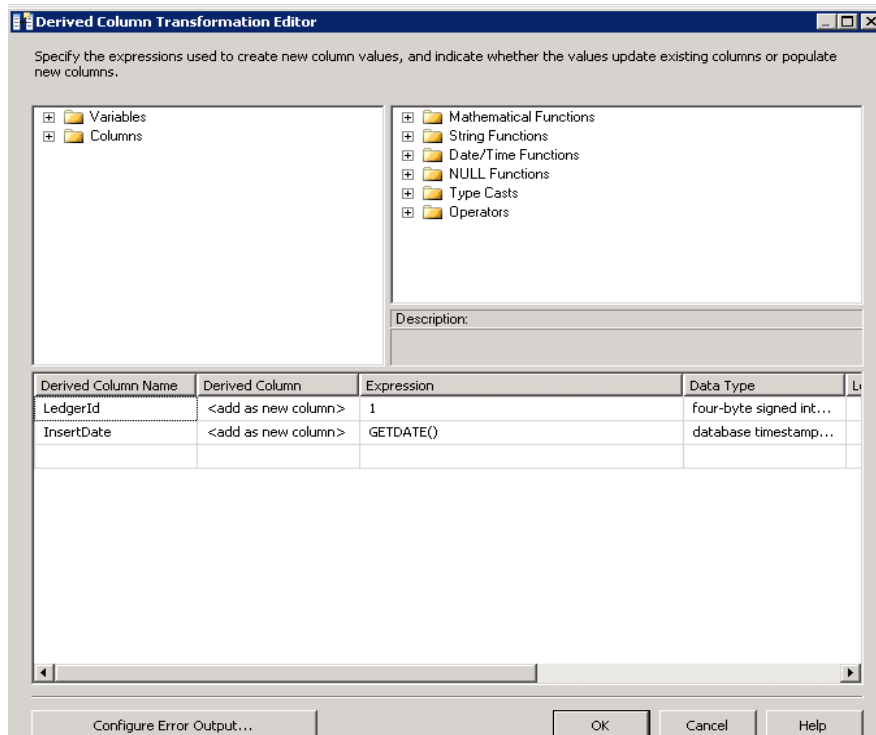


OLE DB Source

The OLE DB Source Editor dialog box is shown. It has a left pane with 'Connection Manager', 'Columns', and 'Error Output'. The main area contains the following fields and buttons:

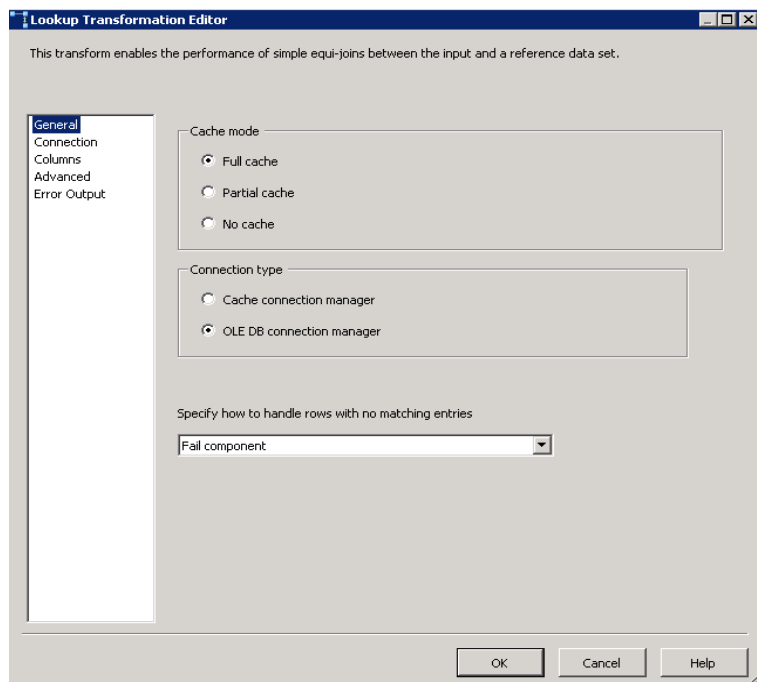
- OLE DB connection manager: UKCHELLUT01.RoyalMail_UK22511_GB_2012 (with a 'New...' button)
- Data access mode: SQL command (dropdown)
- SQL command text:

```
/*SELECT a.VBLNR + ' ' + a.LIFNR + ' ' + CONVERT
(varchar,a.LAUFID) + ' ' + a.LAUFID
PaymentID,
a.[LIFNR] VendorID,
a.ZBLNR Companyref,
a.RZAWR PaymentMethod,
a.[VBLNR] PaymentRef,
a.ZALDT Paymentdate,
(a.RWBTR*-1) PaymentAmountDoc,
(a.RBETR*-1) PaymentAmountLoc,
c.[CHECT] CheckNumber,
c.[PRIDT] CheckPrintDate,
isnull(c.[ RWBTR],0.00)*-1 ChequeAmount,
a.WAERS PaymentCurrencyCode,
c.[XRAQL] ReversalFlag,
c.[VOIDD] CheckVoidedDate,
c.[VOIDR] VoidReasonCode
```
- Buttons: Parameters..., Build Query..., Browse..., Parse Query, Preview...
- Bottom buttons: OK, Cancel, Help

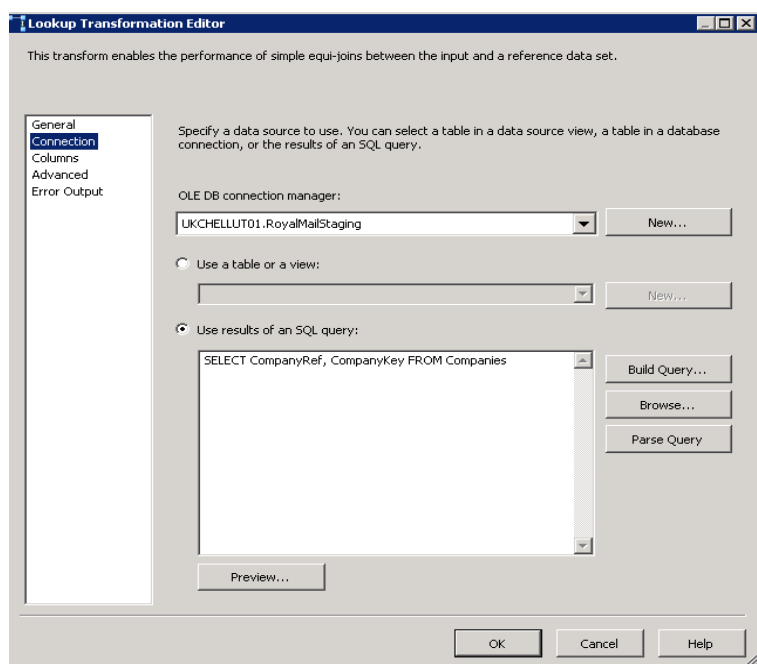


Lookup 1

The lookup Transformation allows you to map to the rows that are missing information by looking the record up against a table you loaded earlier. Open up the transformation editor for the Lookup Transform, and in General page, ensure that the Full cache property is set, and that you have the OLE DB Connection Manager property set for the Connection Type. Change the “No Matching Entries” drop-down box to “Fail Component”.



In the Connection page, select the staging database i.e. “RoyalMailStaging” as the Connection Manager that contains your lookup table. Use the SQL query to get the required (missing) information.



SELECT CompanyRef, CompanyKey

FROM Companies

Next, go to the Columns page and drag PaymentId from the left Available Input Columns to the right CompanyRef column in the Available Lookup Columns table. This will create an arrow between the two tables as shown. Then check the CompanyKey column that you wish to output. This will transfer their information to the bottom grid. If you wanted to replace the given column, you would then change the Add as New Column option to Replace. There are many options here. In our case we are adding a new column named CompanyKey.

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
Columns
Advanced
Error Output

Available Input Columns

Name
PaymentID
CompanyRef
PaymentMethod
PaymentAmountDoc
PaymentAmountLoc
ChequeAmount
PaymentCurrencyCode
ReversalFlag
VniidReasonCode

Available Lookup Columns

<input checked="" type="checkbox"/>	Name
<input type="checkbox"/>	CompanyRef
<input checked="" type="checkbox"/>	CompanyKey

Lookup Column	Lookup Operation	Output Alias
CompanyKey	<add as new column>	CompanyKey

OK Cancel Help

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

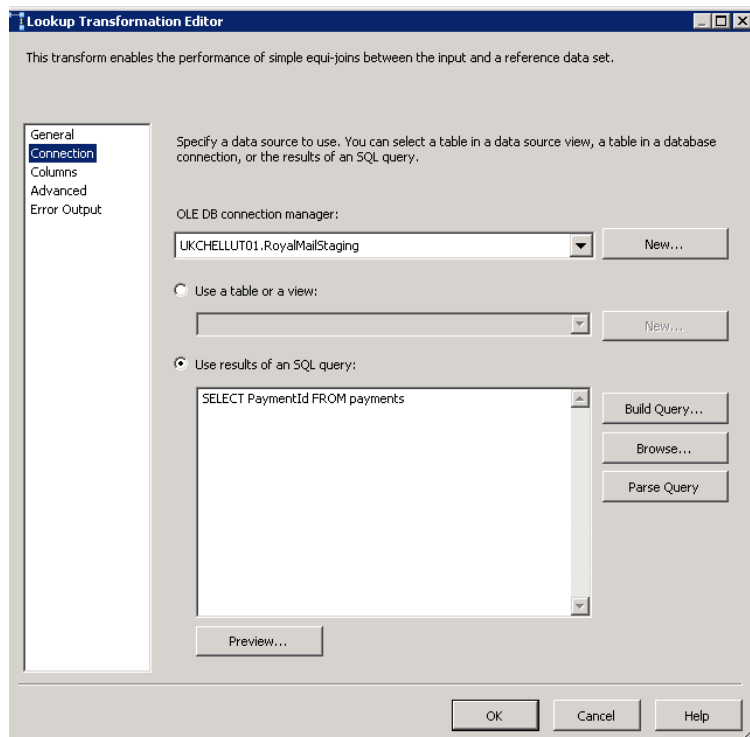
General
Connection
Columns
Advanced
Error Output

Input or Output	Column	Error	Truncation	Description
Lookup Matc...		Fail component		Lookup
CompanyKey		Fail component		Copy Column

Set this value to selected cells: Fail component Apply

OK Cancel Help

Look Up



Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
Columns
Advanced
Error Output

Specify a data source to use. You can select a table in a data source view, a table in a database connection, or the results of an SQL query.

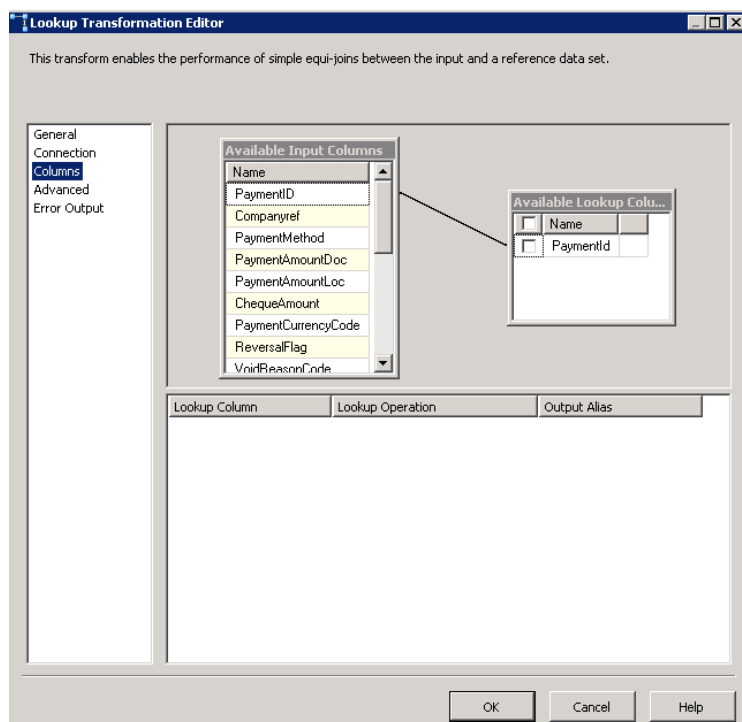
OLE DB connection manager:
UKCHELLUT01.RoyalMailStaging New...

☐ Use a table or a view:
New...

☒ Use results of an SQL query:
SELECT PaymentId FROM payments Build Query...
Browse...
Parse Query

Preview...

OK Cancel Help



Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
Columns
Advanced
Error Output

Available Input Columns

Name
PaymentID
Companyref
PaymentMethod
PaymentAmountDoc
PaymentAmountLoc
ChequeAmount
PaymentCurrencyCode
ReversalFlag
VoidReasonCode

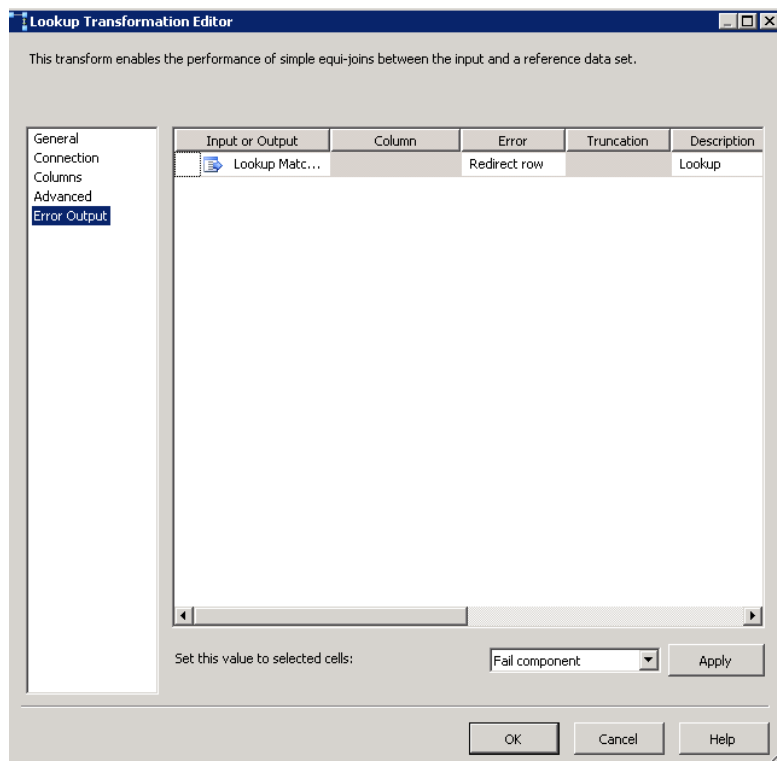
Available Lookup Columns

<input type="checkbox"/> Name
<input type="checkbox"/> PaymentId

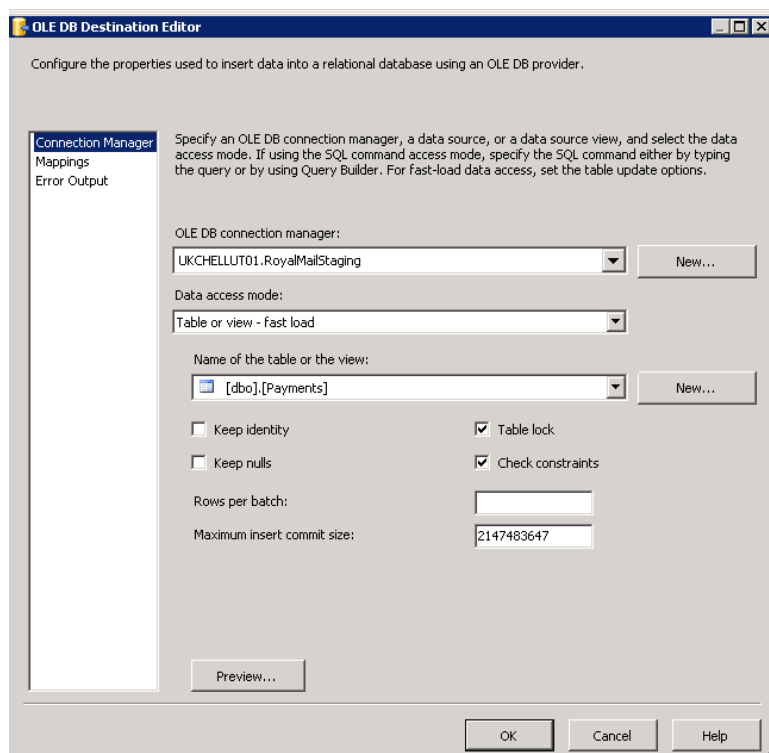
Lookup Column **Lookup Operation** **Output Alias**

Lookup Column	Lookup Operation	Output Alias
---------------	------------------	--------------

OK Cancel Help



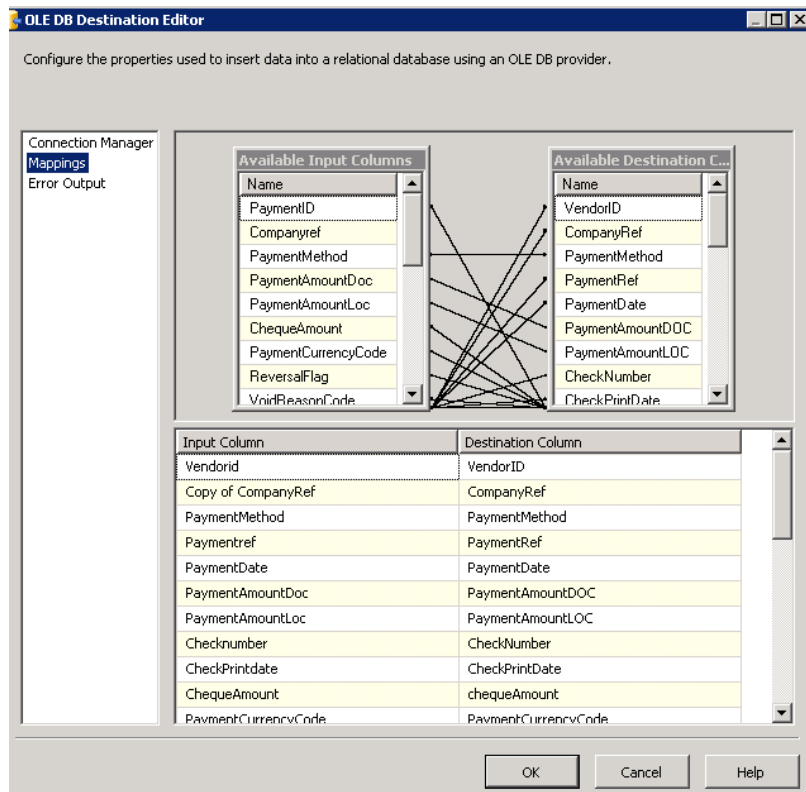
OLE DB Destination



As mentioned earlier they are several options here, 1 is to use a table or view – fast load. Then you select the payments table that was scripted out earlier.

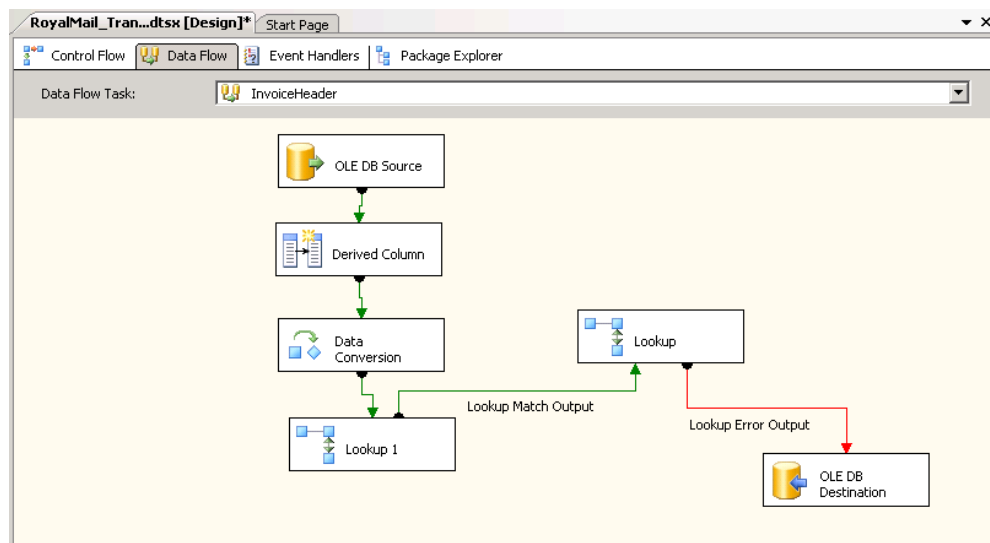
2nd For the Use a Table or View option, select the **New** button next to the drop-down box. The default DDL for creating the table will use the destination's name and the data types may not be exactly what you'd like.

You can make changes to your table so that it can be like the destination table. **Please note** make no changes to the destination table. The scripts for the destination tables are in the appendix.



You may have to map some of the columns, because the column names are different. Go to the mappings page and map each column to its new name. Click OK to close the editor.

InvoiceHeader



OLE DB Source

OLE DB Source Editor

Configure the properties used by a data flow to obtain data from any OLE DB provider.

Connection Manager
Columns
Error Output

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.

OLE DB connection manager:
UKCHELLUT01.RoyalMail_UK22511_GB_2012 New...

Data access mode:
SQL command

SQL command text:

```

/****
--SELECT * FROM (
--SELECT COUNT (*), InvoiceId FROM ( *
SELECT
    InvoiceId ,
    VendorId,
    VendorInvoiceRef,
    InvoiceDate,
    InvoiceRef,
    InvoiceTypeRef,
    InvoiceTypeRef2,
    InvoiceTypeRef3,
    MAX(InvoiceComment) InvoiceComment,
    SUM(InvoiceGrossAmountDoc)
    InvoiceGrossAmountDoc ,
    SUM(InvoiceNetAmountDoc)

```

Parameters...
Build Query...
Browse...
Parse Query

Preview...

OK Cancel Help

Derived Column

Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

☒ Variables
☒ Columns

☒ Mathematical Functions
☒ String Functions
☒ Date/Time Functions
☒ NULL Functions
☒ Type Casts
☒ Operators

Description:

Derived Column Name	Derived Column	Expression	Data Type	
LedgerId	<add as new column>	1	four-byte signed int...	
InsertDate	<add as new column>	GETDATE()	database timestamp...	

Configure Error Output... OK Cancel Help

Data Conversion

Data Conversion Transformation Editor

Configure the properties used to convert the data type of an input column to a different data type. Depending on the data type to which the column is converted, set the length, precision, scale, and code page of the column.

Available Input Columns

<input checked="" type="checkbox"/>	Name
<input type="checkbox"/>	InvoiceId
<input checked="" type="checkbox"/>	VendorID
<input type="checkbox"/>	Companyref
<input checked="" type="checkbox"/>	VendorInvoiceRef
<input type="checkbox"/>	Invoicedate
<input type="checkbox"/>	Invoiceprocess...
<input type="checkbox"/>	InvoiceRef

Input Column	Output Alias	Data Type	Length	Precision	Scale
VendorID	Copy of VendorId	Unicode string [DT_WSTR]	50		
VendorInvoiceRef	Copy of VendorInvoiceRef	Unicode string [DT_WSTR]	50		
InvoiceComment	Copy of InvoiceComment	Unicode string [DT_WSTR]	50		

Configure Error Output... OK Cancel Help

Lookup 1

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General

Connection
Columns
Advanced
Error Output

Cache mode

☒ Full cache
☐ Partial cache
☐ No cache

Connection type

☐ Cache connection manager
☒ OLE DB connection manager

Specify how to handle rows with no matching entries

Fail component

OK Cancel Help

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
 Columns
 Advanced
 Error Output

Specify a data source to use. You can select a table in a data source view, a table in a database connection, or the results of an SQL query.

OLE DB connection manager:
 UKCHELLUT01.RoyalMailStaging New...

☐ Use a table or a view:
New...

☒ Use results of an SQL query:
 Select CompanyRef, CompanyKey From Companies Build Query...
Browse...
Parse Query

Preview...

OK Cancel Help

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
Columns
 Advanced
 Error Output

Available Input Columns

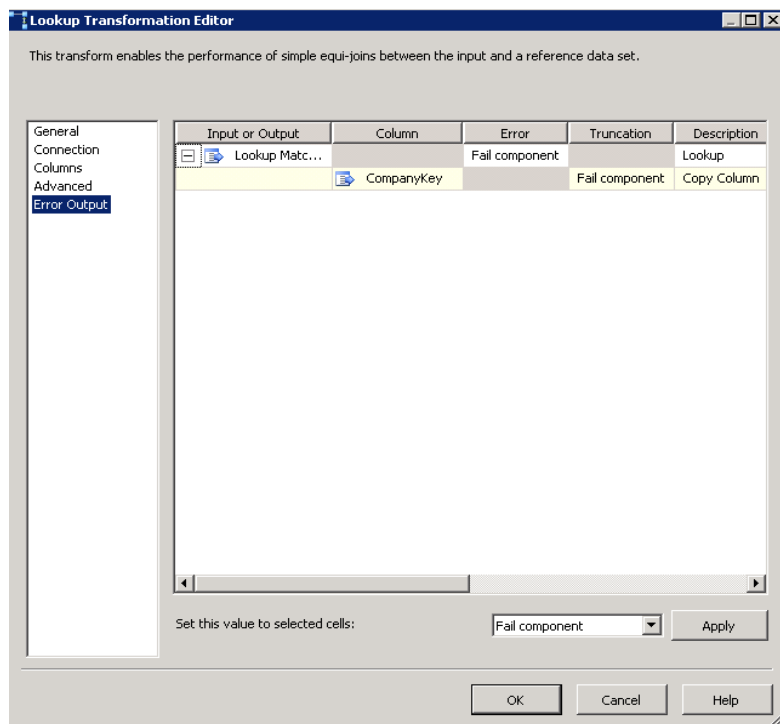
Name
InvoiceId
VendorId
CompanyRef
VendorInvoiceRef
Invoicedate
Invoiceprocessdate
InvoiceRef
Invoicetyperef
Invoicetunerel?

Available Lookup Columns

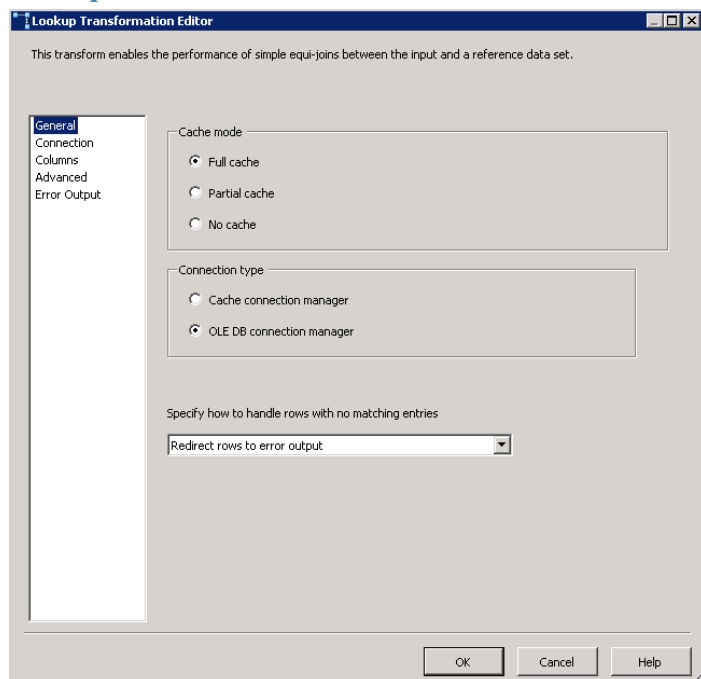
<input checked="" type="checkbox"/>	Name
<input type="checkbox"/>	CompanyRef
<input checked="" type="checkbox"/>	CompanyKey

Lookup Column	Lookup Operation	Output Alias
CompanyKey	<add as new column>	CompanyKey

OK Cancel Help



Lookup



Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
 Columns
 Advanced
 Error Output

Specify a data source to use. You can select a table in a data source view, a table in a database connection, or the results of an SQL query.

OLE DB connection manager:
 UKCHELLUT01.RoyalMailStaging New...

☐ Use a table or a view:
 New...

☒ Use results of an SQL query:
 SELECT InvoiceId FROM Invoiceheader
 Build Query...
 Browse...
 Parse Query

Preview...

OK Cancel Help

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

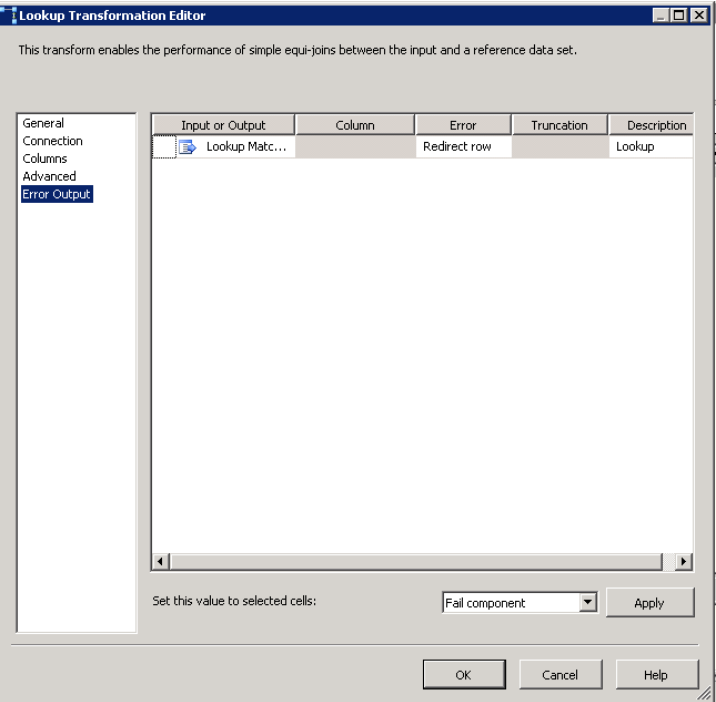
General
Columns
 Connection
 Advanced
 Error Output

Available Input Columns:
 Name
 InvoiceId
 VendorID
 Companyref
 VendorInvoiceRef
 InvoiceDate
 Invoiceprocessdate
 InvoiceRef
 InvoicetypeRef
 InvoiceStatusRef?

Available Lookup Columns:
☐ Name
☐ InvoiceId

Lookup Column	Lookup Operation	Output Alias
---------------	------------------	--------------

OK Cancel Help



OLE DB Destination

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

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:
UKCHELLUT01.RoyalMailStaging New...

Data access mode:
Table or view - fast load

Name of the table or the view:
[dbo].[InvoiceHeader] New...

☐ Keep identity ☒ Table lock
☐ Keep nulls ☒ Check constraints

Rows per batch:

Maximum insert commit size:

Preview...

OK

Cancel

Help

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 Columns

Name
InvoiceId
VendorID
Companyref
VendorInvoiceRef
Invoicedate
Invoiceprocessdate
InvoiceRef
InvoiceTypeRef
InvoiceTypeRef2

Available Destination Columns

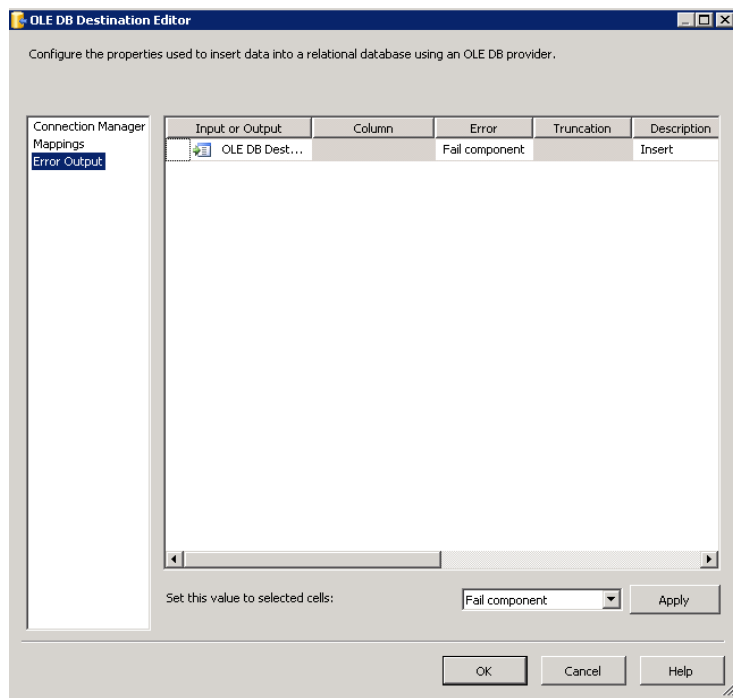
Name
InvoiceID
VendorID
VendorInvoiceRef
InvoiceDate
InvoiceRef
InvoiceTypeRef
InvoiceTypeRef2
InvoiceComment
InvoiceNetAmn

Input Column	Destination Column
InvoiceId	InvoiceID
Copy of VendorId	VendorID
Copy of VendorInvoiceRef	VendorInvoiceRef
Invoicedate	InvoiceDate
InvoiceRef	InvoiceRef
InvoiceTypeRef	InvoiceTypeRef
InvoiceTypeRef2	InvoiceTypeRef2
Copy of InvoiceComment	InvoiceComment
InvoiceNetAmountDoc	InvoiceNetAmountDOC
<Ignore>	InvoiceDiscountAmountDOC
<Ignore>	InvoiceFreightAmountDOC

OK

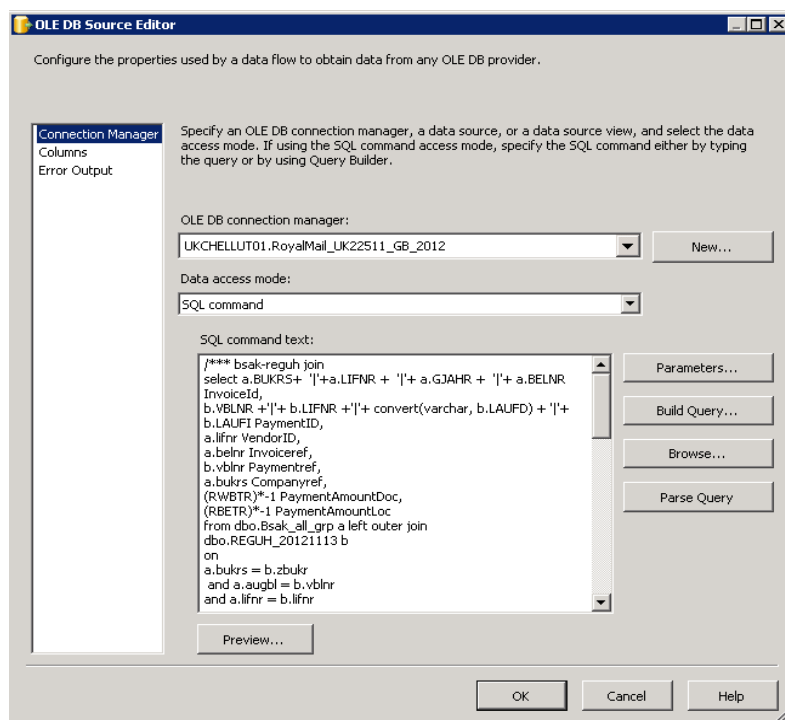
Cancel

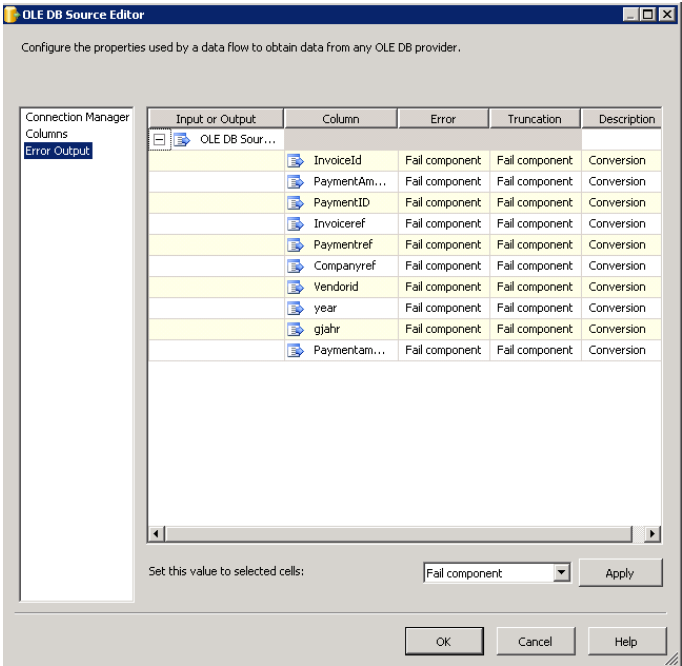
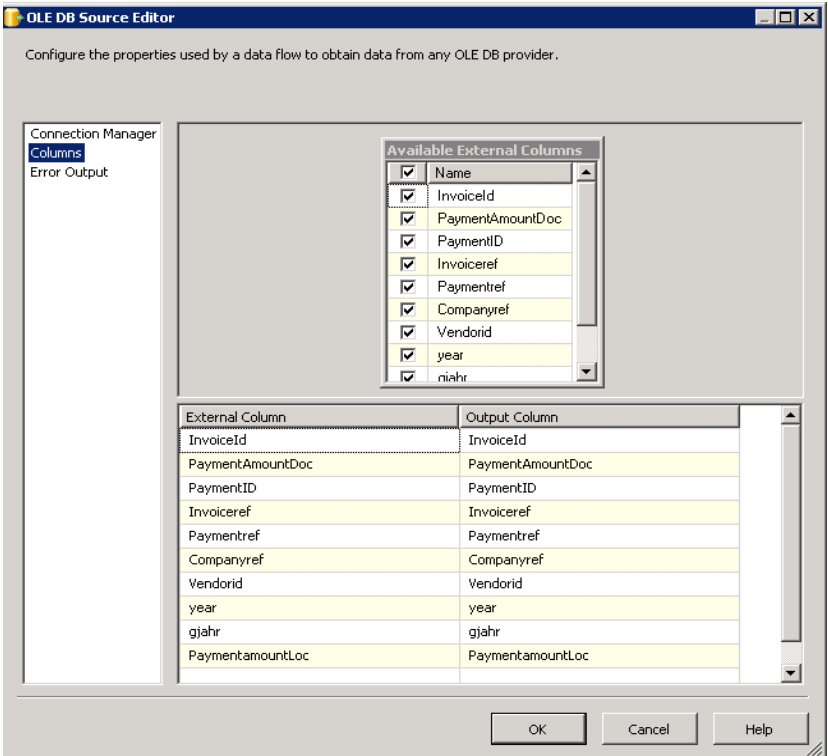
Help



PaymentLink

OLE DB Source





Derived Column

Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

☒ Variables
☒ Columns

☒ Mathematical Functions
☒ String Functions
☒ Date/Time Functions
☒ NULL Functions
☒ Type Casts
☒ Operators

Description:

Derived Column Name	Derived Column	Expression	Data Type	Length
LedgerId	<add as new column>	1	four-byte signed integer [DT_I4]	
InsertDate	<add as new column>	GETDATE()	database timestamp [DT_DBTIMESTAMP]	

Configure Error Output... OK Cancel Help

Data Conversion

Data Conversion Transformation Editor

Configure the properties used to convert the data type of an input column to a different data type. Depending on the data type to which the column is converted, set the length, precision, scale, and code page of the column.

Available Input Columns

☒ Name
☐ InvoiceId
☐ PaymentAmountDoc
☐ PaymentID
☐ Invoiceref
☐ Paymentref
☐ Companyref
☒ VendorId

Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Page
VendorId	Copy of VendorId	Unicode string [DT_WSTR]	50			

Configure Error Output... OK Cancel Help

Lookup1

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
Columns
Advanced
Error Output

Cache mode

☒ Full cache
☐ Partial cache
☐ No cache

Connection type

☐ Cache connection manager
☒ OLE DB connection manager

Specify how to handle rows with no matching entries

Fail component

OK Cancel Help

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
Columns
Advanced
Error Output

Specify a data source to use. You can select a table in a data source view, a table in a database connection, or the results of an SQL query.

OLE DB connection manager:

UKCHELLUT01.RoyalMailStaging New...

☐ Use a table or a view:

New...

☒ Use results of an SQL query:

SELECT CompanyRef, CompanyKey FROM Companies

Build Query...
Browse...
Parse Query

Preview...

OK Cancel Help

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
Columns
Advanced
Error Output

Available Input Column...

Name
Invoiceld
PaymentAmountDoc
PaymentID
Invoiceref
Paymentref
Companyref
Vendorid
year
niahr

Available Lookup Columns

<input checked="" type="checkbox"/>	Name	
<input type="checkbox"/>	CompanyRef	
<input checked="" type="checkbox"/>	CompanyKey	

Lookup Column	Lookup Operation	Output Alias
CompanyKey	<add as new column>	CompanyKey

OK

Cancel

Help

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
Columns
Advanced
Error Output

Input or Output	Column	Error	Truncation	Description
Lookup Matc...		Fail component		Lookup
CompanyKey		Fail component		Copy Column

Set this value to selected cells:

Fail component

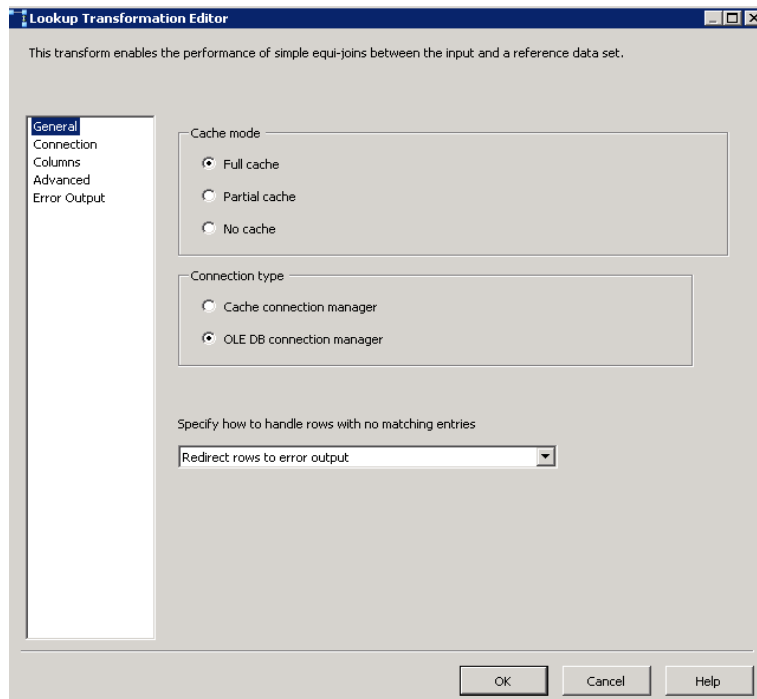
Apply

OK

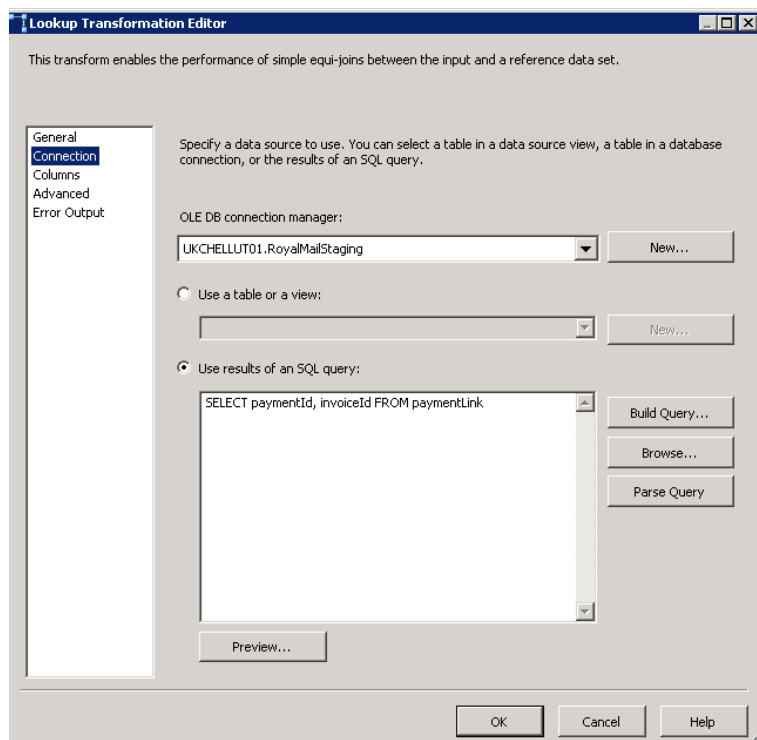
Cancel

Help

Lookup



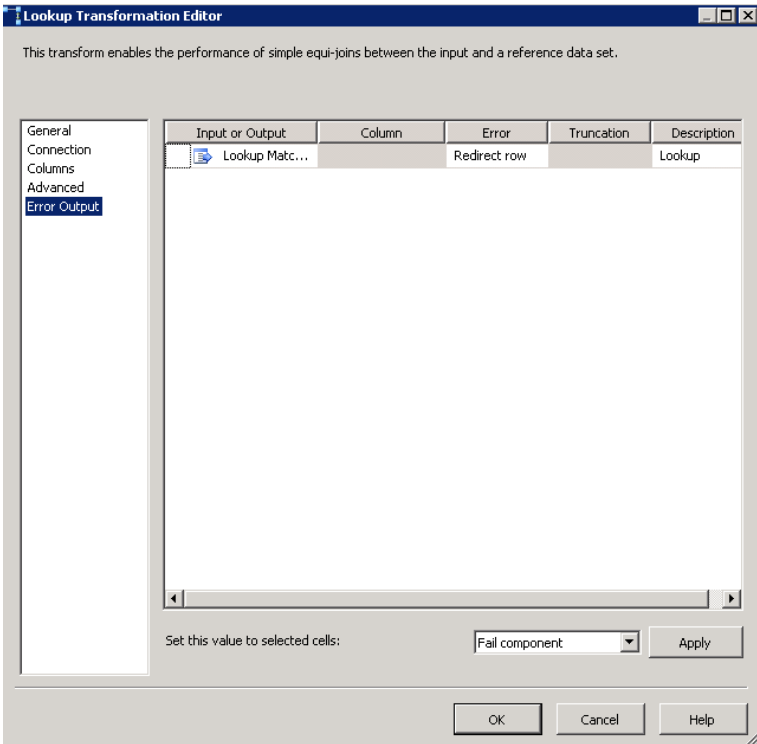
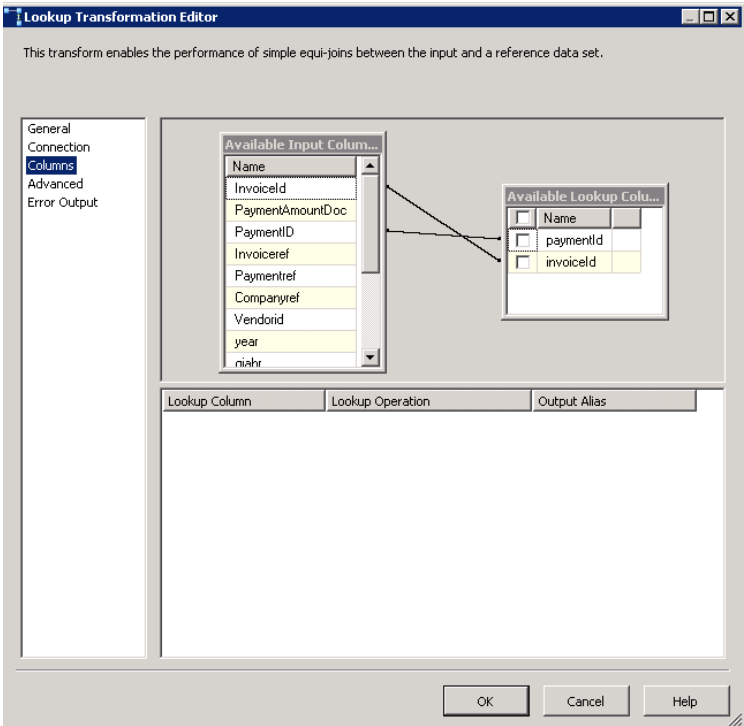
The screenshot shows the 'Lookup Transformation Editor' window with the 'General' tab selected. The window title is 'Lookup Transformation Editor'. Below the title bar, a description states: 'This transform enables the performance of simple equi-joins between the input and a reference data set.' On the left, a vertical pane contains a list of tabs: 'General', 'Connection', 'Columns', 'Advanced', and 'Error Output'. The 'General' tab is active. The main area contains three sections: 1. 'Cache mode' with three radio buttons: 'Full cache' (selected), 'Partial cache', and 'No cache'. 2. 'Connection type' with two radio buttons: 'Cache connection manager' and 'OLE DB connection manager' (selected). 3. 'Specify how to handle rows with no matching entries' with a dropdown menu showing 'Redirect rows to error output'. At the bottom right are 'OK', 'Cancel', and 'Help' buttons.



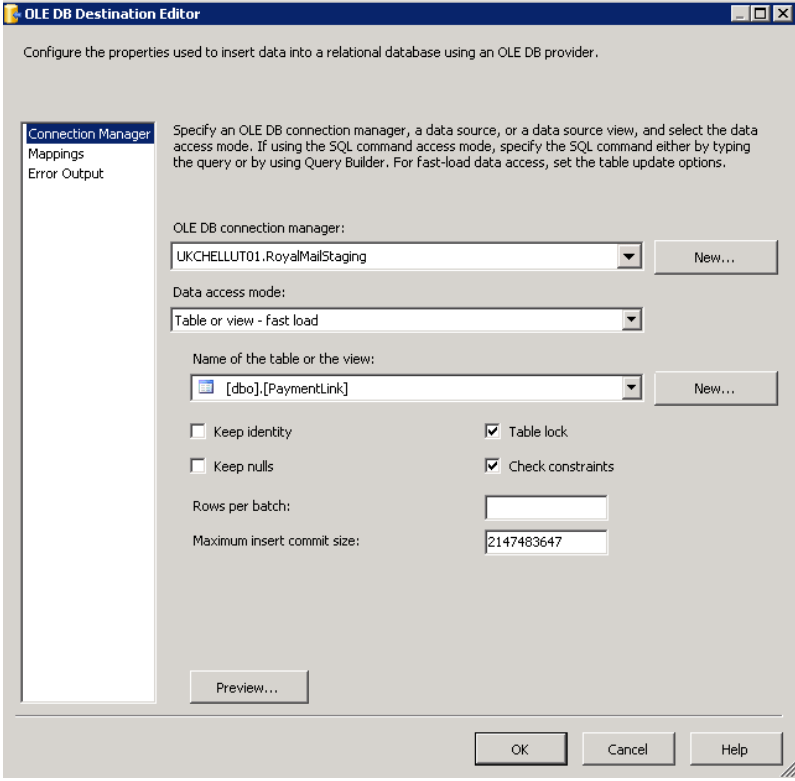
The screenshot shows the 'Lookup Transformation Editor' window with the 'Connection' tab selected. The window title is 'Lookup Transformation Editor'. Below the title bar, a description states: 'This transform enables the performance of simple equi-joins between the input and a reference data set.' On the left, a vertical pane contains a list of tabs: 'General', 'Connection', 'Columns', 'Advanced', and 'Error Output'. The 'Connection' tab is active. The main area contains the following sections: 1. A text instruction: 'Specify a data source to use. You can select a table in a data source view, a table in a database connection, or the results of an SQL query.' 2. 'OLE DB connection manager:' with a dropdown menu showing 'UKCHELLUT01.RoyalMailStaging' and a 'New...' button. 3. 'Use a table or a view:' with a radio button (not selected), a dropdown menu, and a 'New...' button. 4. 'Use results of an SQL query:' with a radio button (selected), a text area containing the query 'SELECT paymentId, invoiceId FROM paymentLink', and buttons for 'Build Query...', 'Browse...', and 'Parse Query'. 5. A 'Preview...' button at the bottom left of the main area. At the bottom right are 'OK', 'Cancel', and 'Help' buttons.

PaymentLink

SELECT PaymentID, InvoiceID FROM



OLE DB Conversion



The OLE DB Destination Editor dialog box is used to configure the properties for inserting data into a relational database using an OLE DB provider. It features a left-hand pane with three tabs: 'Connection Manager' (selected), 'Mappings', and 'Error Output'. The main area contains several configuration options: a dropdown for 'OLE DB connection manager' (set to 'UKCHELLUT01.RoyalMailStaging'), a 'Data access mode' dropdown (set to 'Table or view - fast load'), and a 'Name of the table or the view' dropdown (set to '[dbo].[PaymentLink]'). Below these are checkboxes for 'Keep identity', 'Keep nulls', 'Table lock', and 'Check constraints'. At the bottom, there are input fields for 'Rows per batch' and '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

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:
UKCHELLUT01.RoyalMailStaging New...

Data access mode:
Table or view - fast load

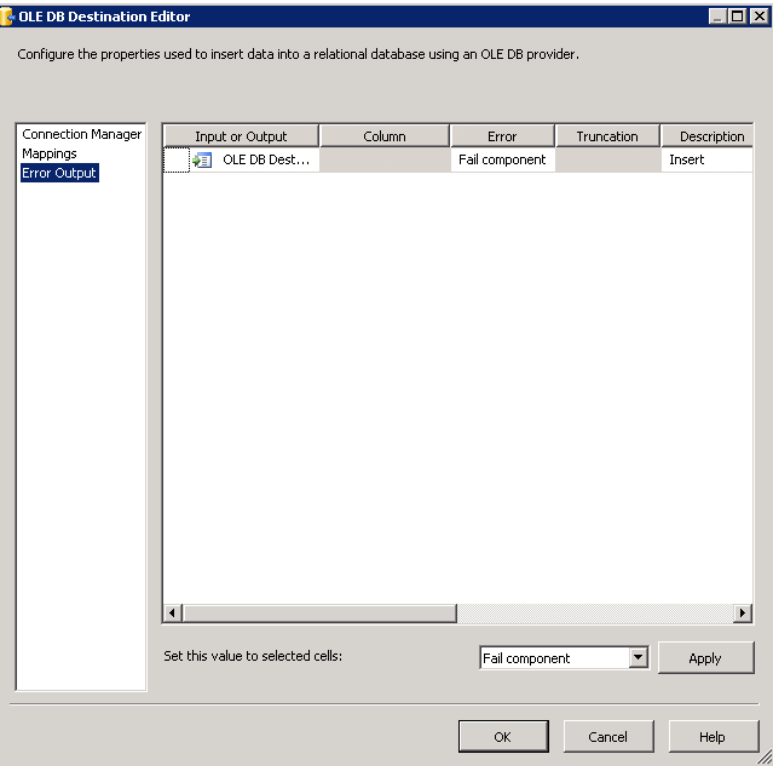
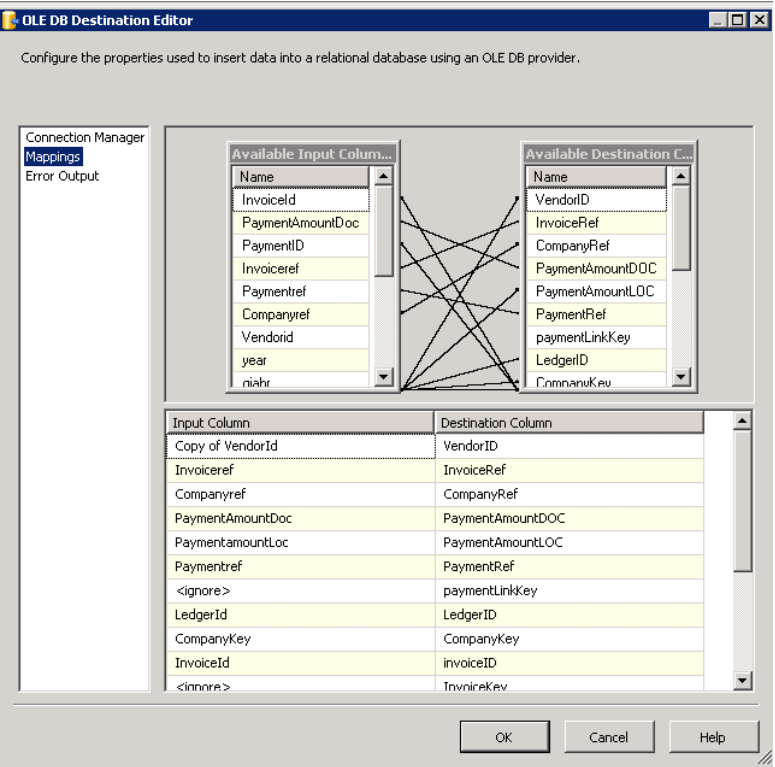
Name of the table or the view:
[dbo].[PaymentLink] New...

☐ Keep identity ☒ Table lock
☐ Keep nulls ☒ Check constraints

Rows per batch:
Maximum insert commit size:

Preview...

OK Cancel Help



To improve the processing speed of SQL

- Avoid the OR Operator. In the Where statement you can get away by using IN. In most cases SQL will not use an index if the condition in a WHERE clause contains the OR operator.
- Avoid unnecessary use of the UNION operator
- Avoid the NOT Operator. If the condition in a WHERE clause contains the NOT operator, SQL will generally not use an index. If possible replace with a comparison operator.
- Use the BETWEEN operator
- Add Redundant conditions to Joins
- Avoid the HAVING clause. In the SELECT statement, conditions can be specified in two places, in the WHERE and the HAVING clauses. Always try to place as many conditions as possible in the WHERE clause and as few as possible in the HAVING clause. The main reason is that indexes are not used for conditions specified in the HAVING clause.
- Make the SELECT clause as small as possible. Avoid the unnecessary columns because it can affect the processing speed in a negative way.
- Avoid DISTINCT when it is not required or even redundant. If the SELECT clause contains a primary key then the DISTINCT clause will be unnecessary.
- Prefer OUTER JOINS to UNION operators
- Avoid Data Type Conversions. Converting data types adversely affects the processing speed.
- The largest table last. If possible specify the largest table in the FROM clause last.
- Avoid the ANY and ALL operators

In summary try to avoid OR, UNION, NOT, HAVING, DISTINCT, ANY, ALL and unnecessary columns.

Checking for Duplicates and Other Checks

PaymentLink

The combination of **InvoiceID + PaymentId** should be unique.

InvoiceHeader

InvoiceID should be unique

Payments

PaymentId should be unique

VendorMaster

VendorId should be unique.

- You can **SUM** or **MAX** Invoicetaxcode, MWSKZ, SGTX, Invoicecomment, InvoiceGrossamount/netamount/ Doc/Loc, PurchaseOrder
- Duplicates on AUGBL, AUGBT can be left on as they will not be taken to the InvoiceHeader.
- After using max on AmountDoc and AmountLoc, you will need to check the addition calculation. Sometimes it could be wrong. Do it before and after using max.

- Make sure No Invoice process date is NULL
- Depending on project, there might be no need to join Reguh and Regup, since the info needed might be from only one table.
- No need to max BUZEI (line number). You just need to take it out, so that it is not shown.

SQL Common Errors

A. Make sure you are using the correct JOIN e.g. using JOIN instead of LEFT OUTER JOIN

B. **SUM** (GrossAmountDoc) GrossAmountDoc

MAX (InvoiceComment) InvoiceComment

On SUM or MAX you need to put the name of what you are summing at the end and you also need to remove it from the GROUP BY list.

C. Do not use **SELECT *** in production code! One exception here is when using the **EXISTS** predicate. The select list in the sub-query for the **EXISTS** predicate is ignored since only the existence of rows is important. **Note:** If a view is create using the **SCHEMABINDING** option, then the base tables cannot be modified in a way that will affect the view definition.

D. **NULL** is not matched with any value (even with **NULL**).

E. Predicate evaluation Order: FROM, WHERE, GROUP BY, HAVING, SELECT

F. Data type mismatch in predicates

G. Functions on indexed columns in predicates.

WHERE LEFT(customer_name, 1) = 'L';

WHERE DATEPART(YEAR, sale_date) = 2009 AND DATEPART(MONTH, sale_date) = 1;

The problem arises from the fact that the index columns are being passed to a function, which the query engine must then evaluate for every single row in the table. In cases such as these, the **WHERE** clause predicate is deemed "non-SARGable" and the best that the query optimizer can do is perform a full index or table scan.

To make sure the indexes get used, we need to avoid the use of functions on the indexed columns. In our two examples, it is a relatively simple task to rewrite the queries to use SARG-able predicates. The first requested can be expressed with this logically equivalent query:

```
WHERE customer_name LIKE 'L%';
```

```
WHERE sale_date >= '20090101' AND sale_date < '20090201';
```

In Summary here is the list of common mistakes:

1. NULLs and the NOT IN predicate
2. Functions on indexed columns in predicates
3. Incorrect sub-query column
4. Data type mismatch in predicates
5. Predicate evaluation order
6. Outer joins and placement of predicates
7. Sub-queries that return more than one value
8. Use of SELECT *
9. Scalar user-defined functions
10. Overuse of cursors

```
SELECT a.IdNumber, p.LastName FROM addressTable a, peopleTable p;
```

Without the use of a join in the WHERE clause, each row in the first row is matched up with each row in the second. Your query can become a run-away query.

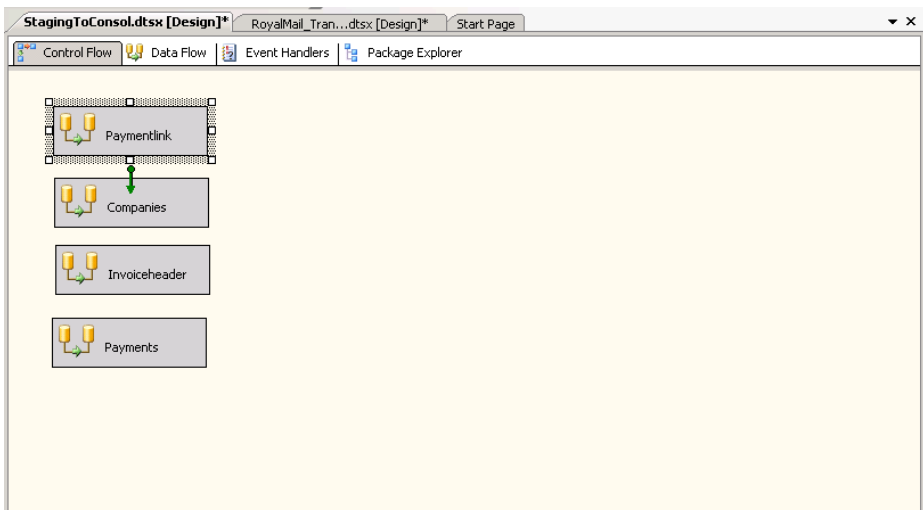
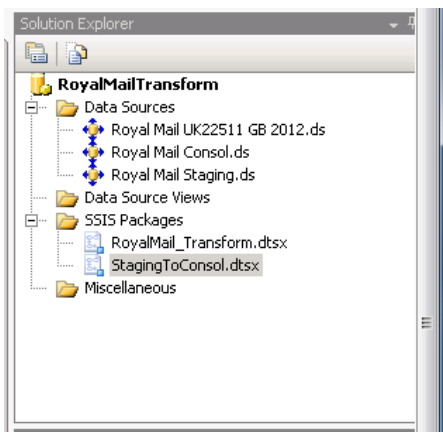
Consol

Ignore all the keys except CompanyKey. Allow CompanyKey to be moved from staging to consol. All other keys are populated automatically during the process.

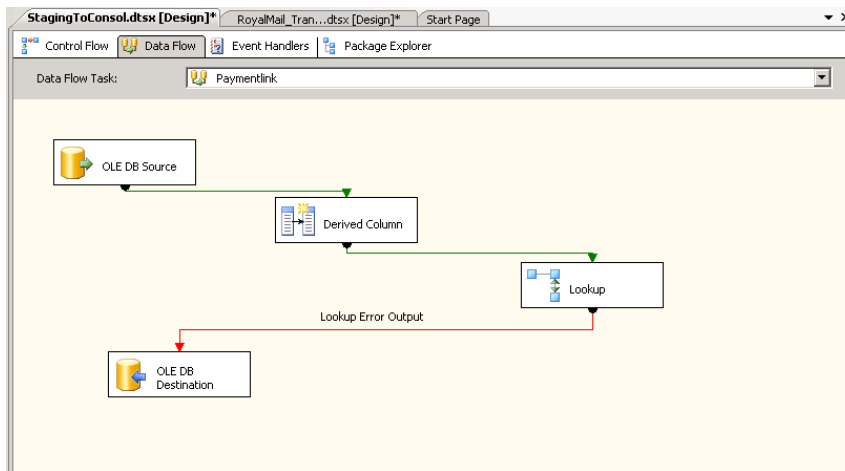
- InvoiceID is the primary key for the InvoiceHeader, along with CompanyRef and LedgerID.
- LedgerID should be the same on PaymentLink and payments
- CompanyRef should be in the tables; otherwise you will have errors when running the process package.

Moving Data from Staging to Consol

The Consol database is the final stage in the process, where the final version of the fact table (InvoiceHeader) and dimensions (Payments, VendorMaster, and Companies) are loaded. This is the database which is linked to the APTrax tool and used by the auditors and clients. In the consol database, the 'Process Package' is used to generate various keys and create various reports.



PaymentLink

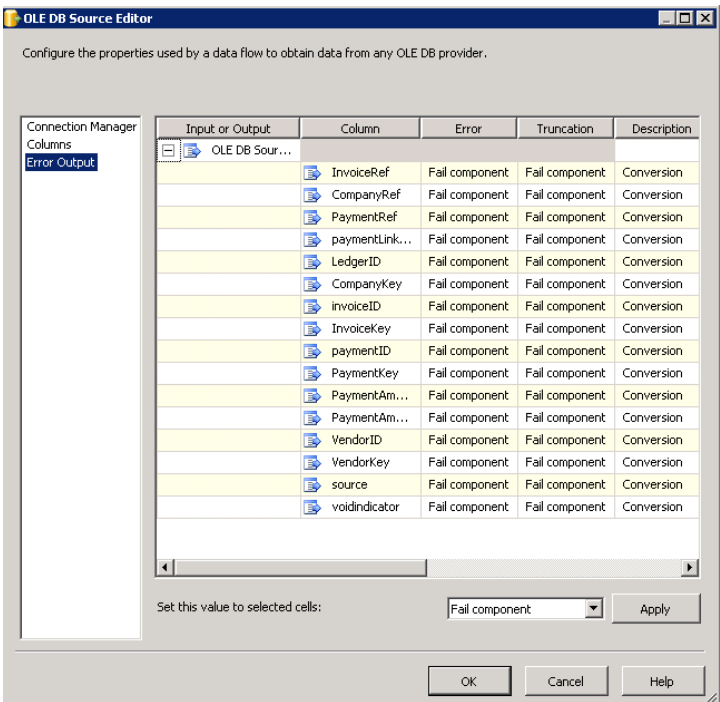
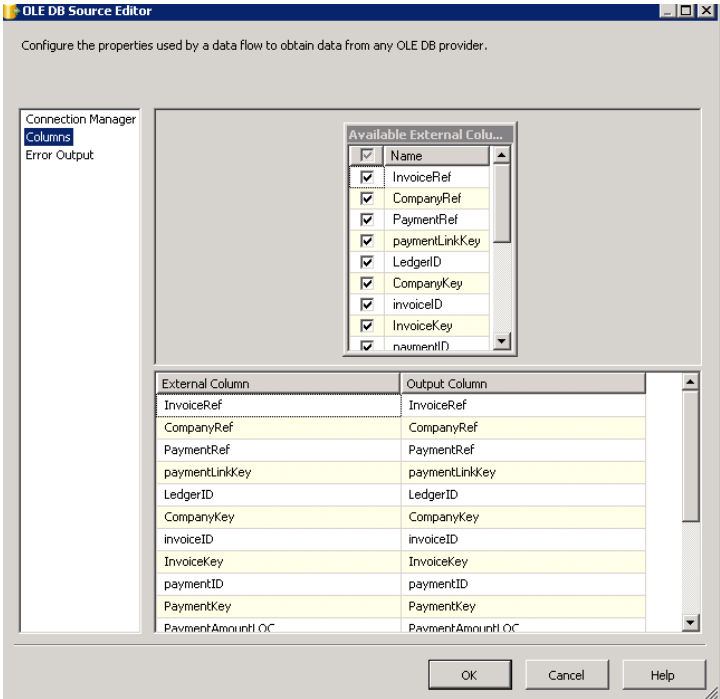


OLE DB Source

The 'OLE DB Source Editor' dialog box is shown, used for configuring the properties of an OLE DB source. The configuration is as follows:

- Connection Manager**: Royal Mail Staging (with a 'New...' button).
- Data access mode**: Table or view.
- Name of the table or the view**: [dbo].[PaymentLink].

Buttons at the bottom include 'Preview...', 'OK', 'Cancel', and 'Help'.



Derived Column

Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Variables
Columns

Mathematical Functions
String Functions
Date/Time Functions
NULL Functions
Type Casts
Operators

Description:

Derived Column Name	Derived Column	Expression	Data Type	Length
InsertDate	<add as new column>	GETDATE()	database timestamp [DT_DBTIMESTAMP]	

Configure Error Output... OK Cancel Help

Lookup

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
Columns
Advanced
Error Output

Cache mode

☒ Full cache
☐ Partial cache
☐ No cache

Connection type

☐ Cache connection manager
☒ OLE DB connection manager

Specify how to handle rows with no matching entries

Redirect rows to error output

OK Cancel Help

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
Connection
 Columns
 Advanced
 Error Output

Specify a data source to use. You can select a table in a data source view, a table in a database connection, or the results of an SQL query.

OLE DB connection manager:
 Royal Mail Consol New...

☐ Use a table or a view:
 New...

☒ Use results of an SQL query:
 select invoiceid,paymentid from paymentlink
 Build Query...
 Browse...
 Parse Query

Preview...

OK Cancel Help

Lookup Transformation Editor

This transform enables the performance of simple equi-joins between the input and a reference data set.

General
 Connection
Columns
 Advanced
 Error Output

Available Input Columns

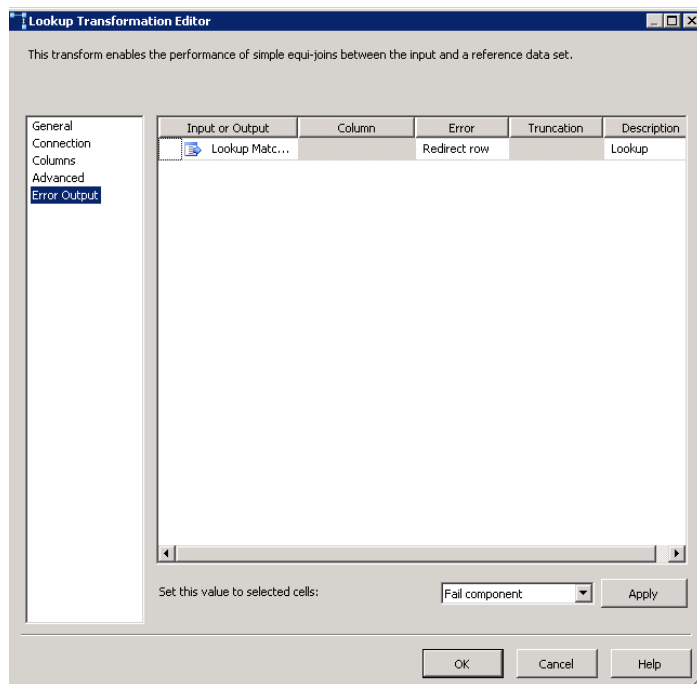
Name
InvoiceRef
CompanyRef
PaymentRef
paymentLinkKey
LedgerID
CompanyKey
invoiceID
InvoiceKey
paymentID

Available Lookup Columns

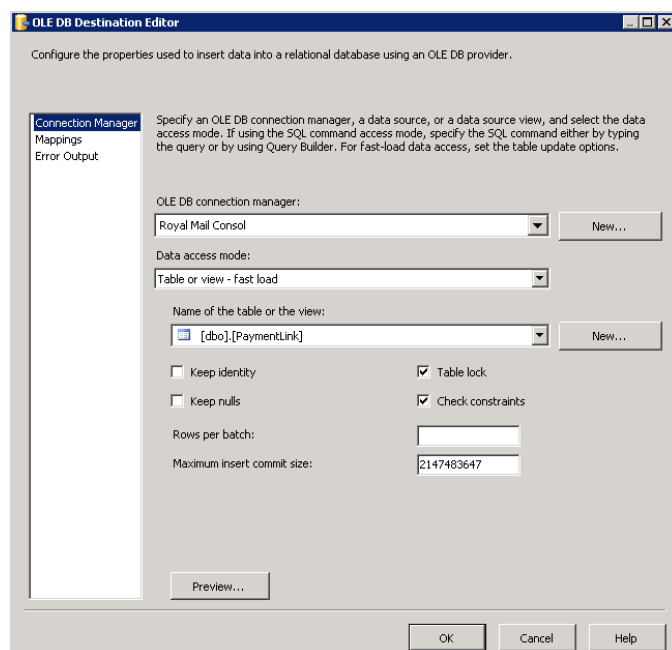
Name
invoiceid
paymentid

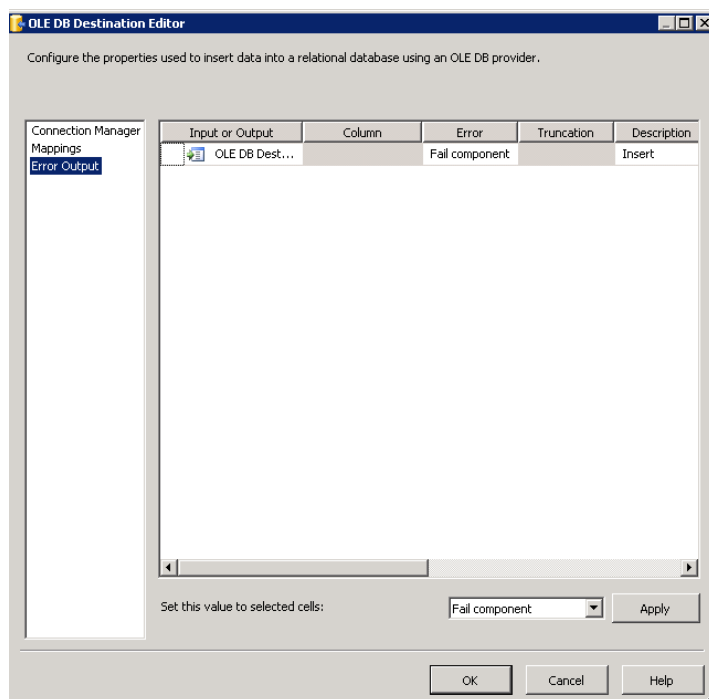
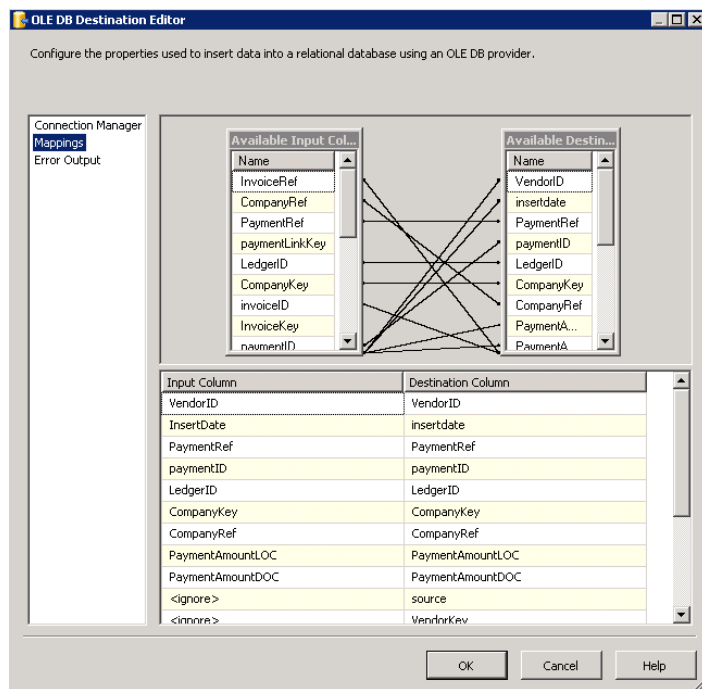
Lookup Column Lookup Operation Output Alias

OK Cancel Help



OLE DB Destination

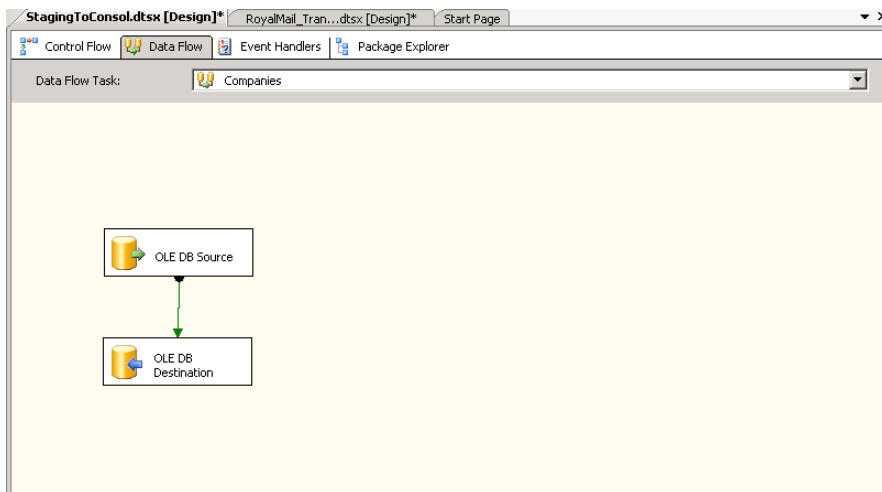




Companies

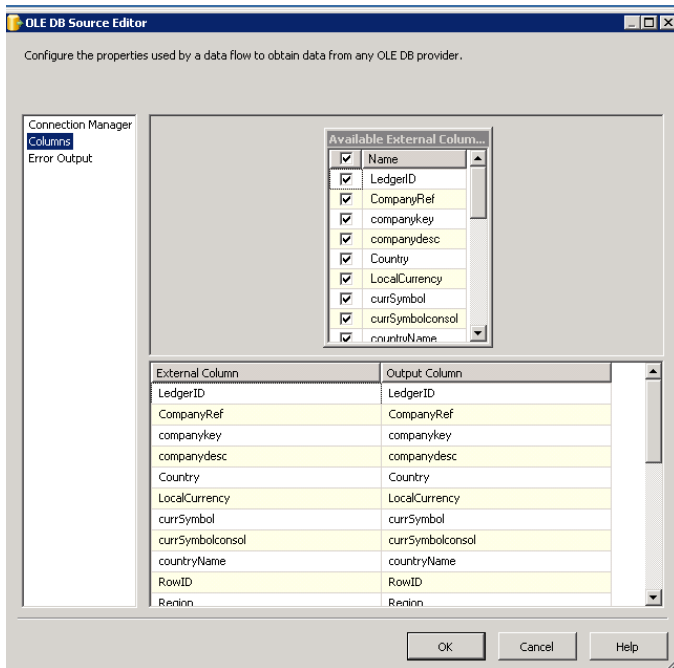
This is used to differentiate the various multiple business units if they exist. Companies table contains information in relation to the business units within the organisation. Details include the name, currency details etc. Company key is the primary key for the companies table. CompanyKey is the numeric equivalent of CompanyRef. CompanyRef is the company code (business division) of the organisation.

Check for the number of companies by looking for different BUKRS (CompanyRef). Company Ref should be the same in companies table, payments, InvoiceHeader and PaymentLink.

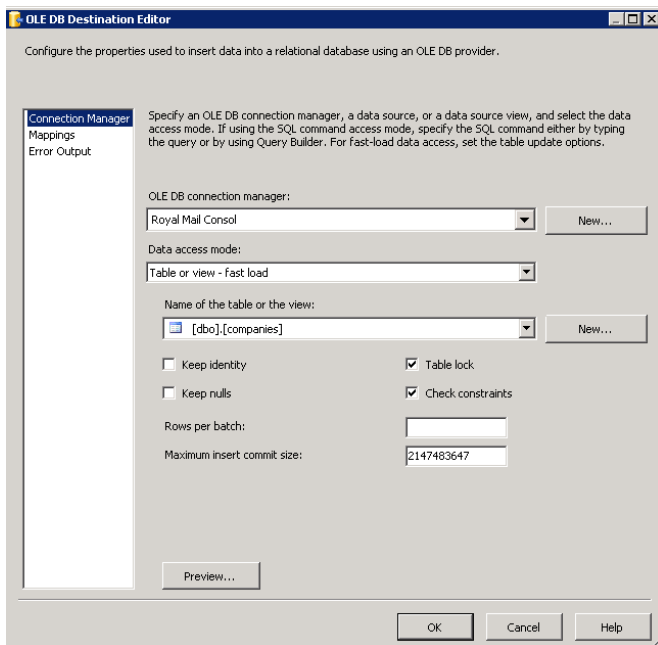


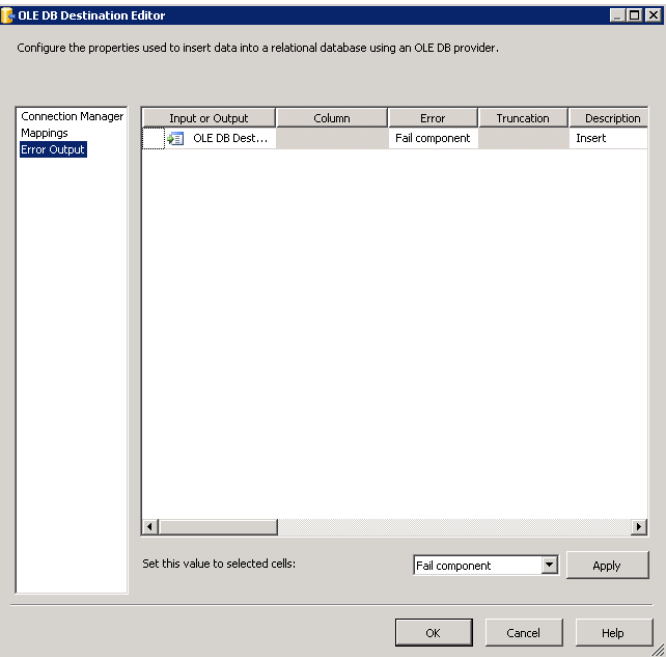
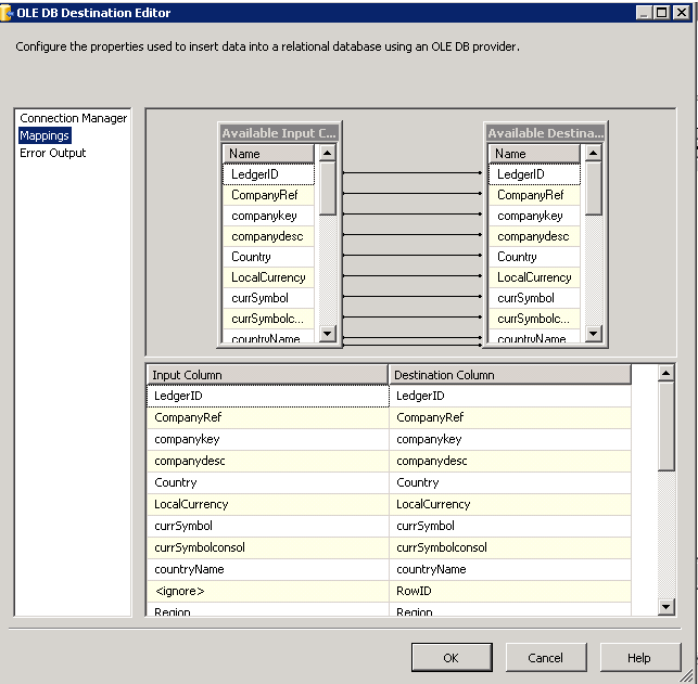
The OLE DB Source Editor dialog box is shown with the 'Connection Manager' tab selected. The 'OLE DB connection manager' dropdown is set to 'Royal Mail Staging'. The 'Data access mode' dropdown is set to 'Table or view'. The 'Name of the table or the view' dropdown is set to '[dbo].[companies]'. The 'Preview...' button is visible at the bottom.

OLE DB Source

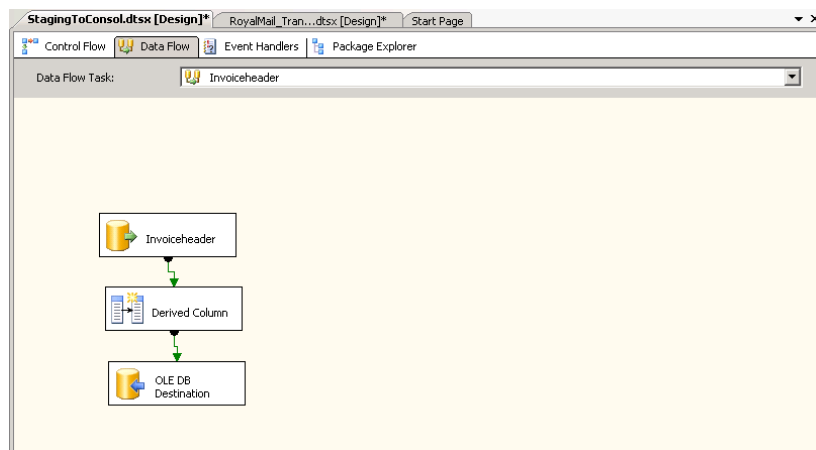


OLE DB Destination





InvoiceHeader

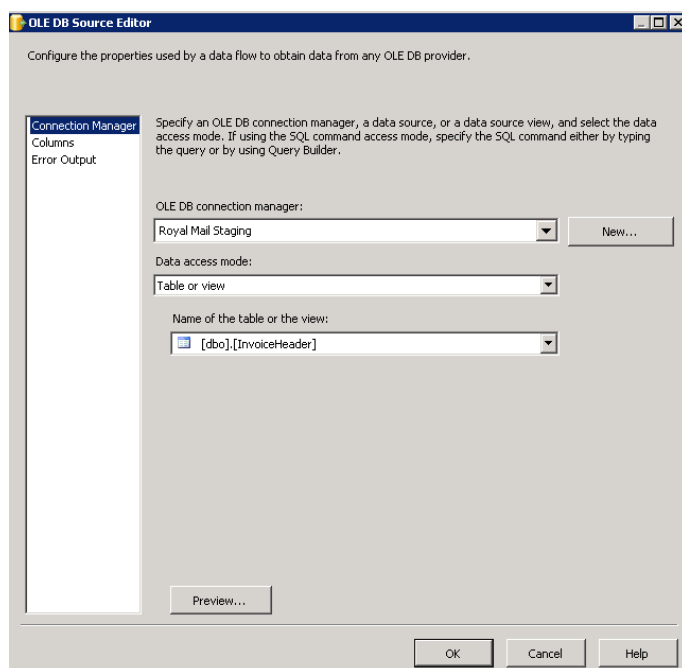


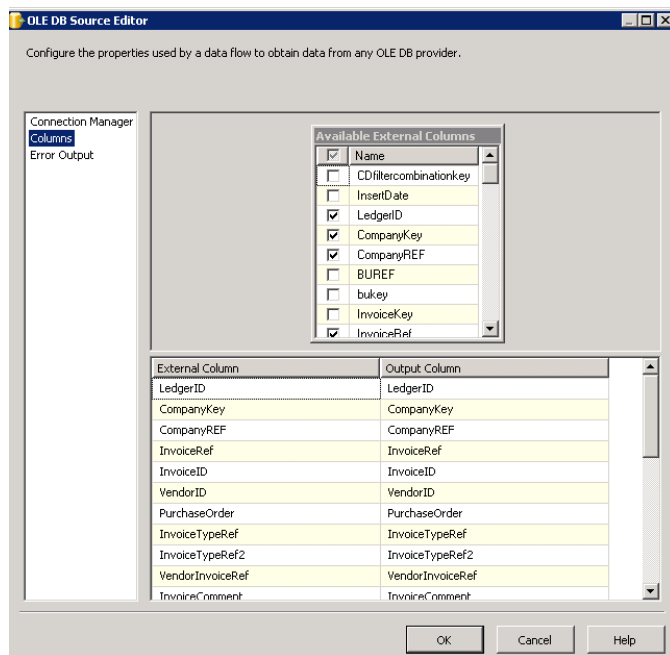
OLE DB Source

Deselect fields that we are not expected to push i.e. those not required by the mapping document.

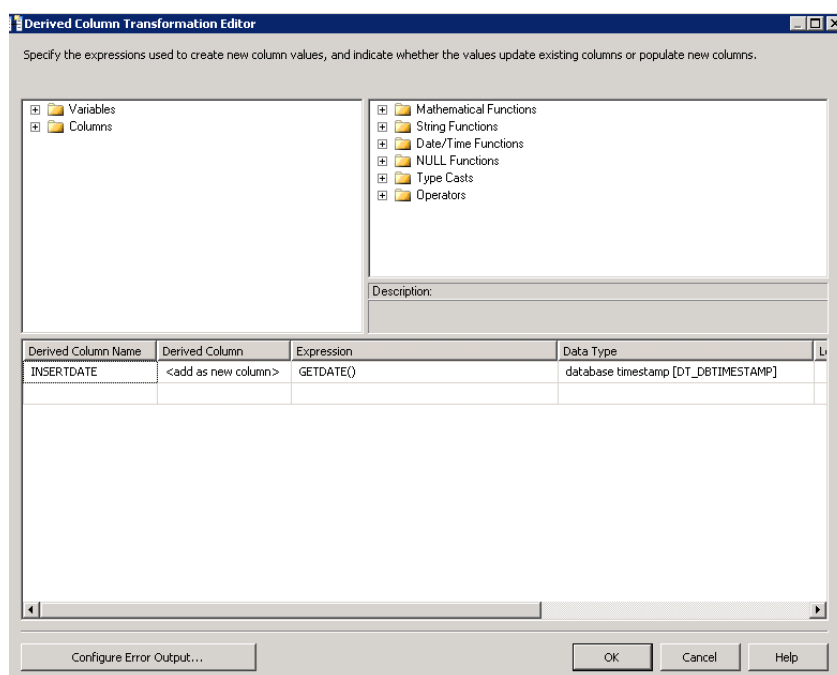
Deselect the Insert date.

Ignore all keys except CompanyKey, as they are populated automatically.





Derived Column



Error: "Microsoft SQL Server Native Client 10.0" Hresult: 0x80004005 Description: "Cannot insert duplicate key row in object 'dbo.InvoiceHeader' with unique index 'ind_InvoiceHeader_invoicekey'.".

Solution: Delete index: 'ind_InvoiceHeader_invoicekey'

OLE DB Destination

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:
Royal Mail Consol New...

Data access mode:
Table or view - fast load

Name of the table or the view:
[dbo].[InvoiceHeader] New...

☐ Keep identity ☒ Table lock
☐ Keep nulls ☒ Check constraints

Rows per batch:
Maximum insert commit size: 2147483647

Preview...

OK Cancel Help

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 Columns:

Name
LedgerID
CompanyKey
CompanyREF
InvoiceRef
InvoiceID
VendorID
PurchaseOrder
InvoiceType...
InvoiceTime

Available Destination Columns:

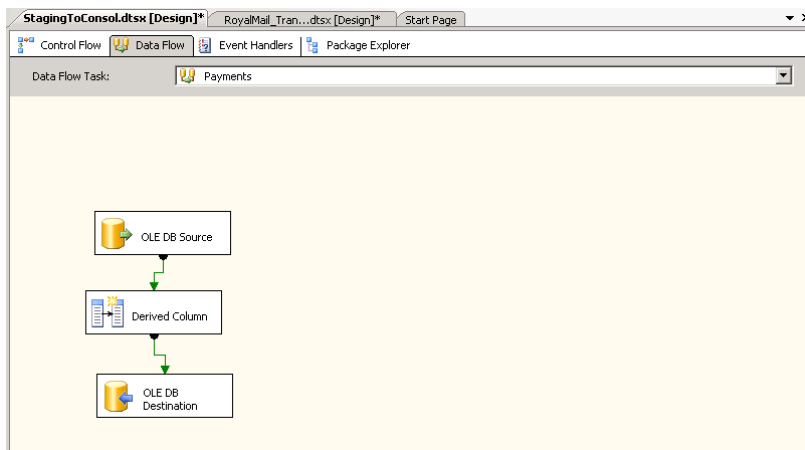
Name
CDfiltercombinationkey
InsertDate
LedgerID
CompanyKey
CompanyREF
BUREF
bukey
InvoiceKey
InvoiceRef

Input Column Destination Column

<ignore>	CDfiltercombinationkey
INSERTDATE	InsertDate
LedgerID	LedgerID
CompanyKey	CompanyKey
CompanyREF	CompanyREF
<ignore>	BUREF
<ignore>	bukey
<ignore>	InvoiceKey
InvoiceRef	InvoiceRef
InvoiceID	InvoiceID
<ignore>	INVOICFID_HASH

OK Cancel Help

Payments



OLE DB Source

The 'OLE DB Source Editor' dialog box is used to configure the properties for an OLE DB source. The 'Connection Manager' tab is selected, showing the following configuration:

- OLE DB connection manager:** Royal Mail Staging (with a 'New...' button)
- Data access mode:** Table or view
- Name of the table or the view:** [dbo].[Payments]

Buttons at the bottom include 'Preview...', 'OK', 'Cancel', and 'Help'.

OLE DB Source Editor

Configure the properties used by a data flow to obtain data from any OLE DB provider.

Connection Manager
Columns
Error Output

Available External Columns

☒ Name
☐ paymentKey
☒ PaymentRef
☒ paymentID
☒ LedgerID
☒ CompanyKey
☒ CompanyRef
☒ VendorID
☒ PaymentCurrencyCode
☒ PaymentAmount LOC

External Column	Output Column
PaymentRef	PaymentRef
paymentID	paymentID
LedgerID	LedgerID
CompanyKey	CompanyKey
CompanyRef	CompanyRef
VendorID	VendorID
PaymentCurrencyCode	PaymentCurrencyCode
PaymentAmountLOC	PaymentAmountLOC
PaymentAmountDOC	PaymentAmountDOC
PaymentMethod	PaymentMethod
PaymentDate	PaymentDate

OKCancelHelp

Derived Column

Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

- Variables
- Columns

- Mathematical Functions
- String Functions
- Date/Time Functions
- NULL Functions
- Type Casts
- Operators

Description:

Derived Column Name	Derived Column	Expression	Data Type	Length	Prec
Insertdate	<add as new column>	GETDATE()	database timestamp [DT_DBTIMESTAMP]		

Configure Error Output... OK Cancel Help

OLE DB Destination

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

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:
Royal Mail Consol New...

Data access mode:
Table or view - fast load

Name of the table or the view:
[dbo].[Payments] New...

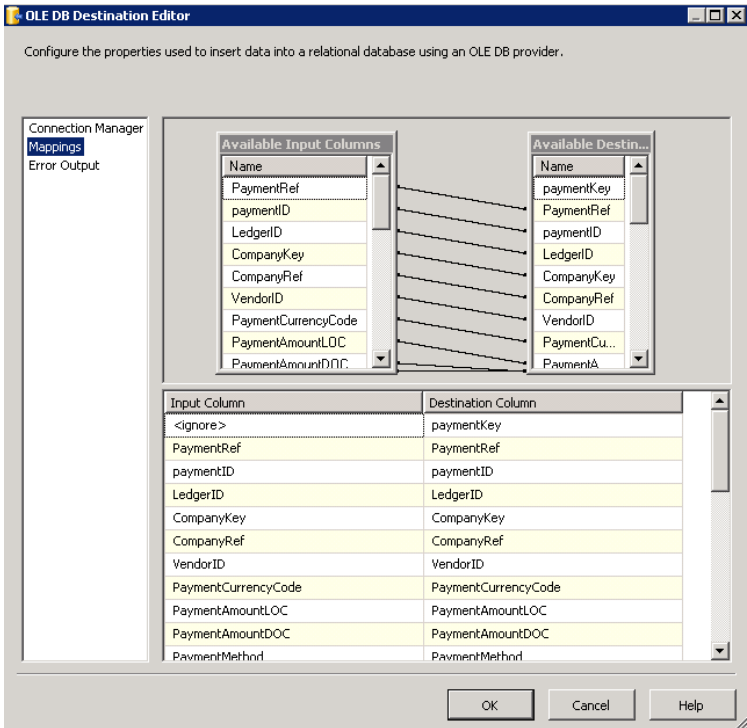
☐ Keep identity ☒ Table lock
☐ Keep nulls ☒ Check constraints

Rows per batch:

Maximum insert commit size:

Preview...

OK Cancel Help



Common Data conversion Errors

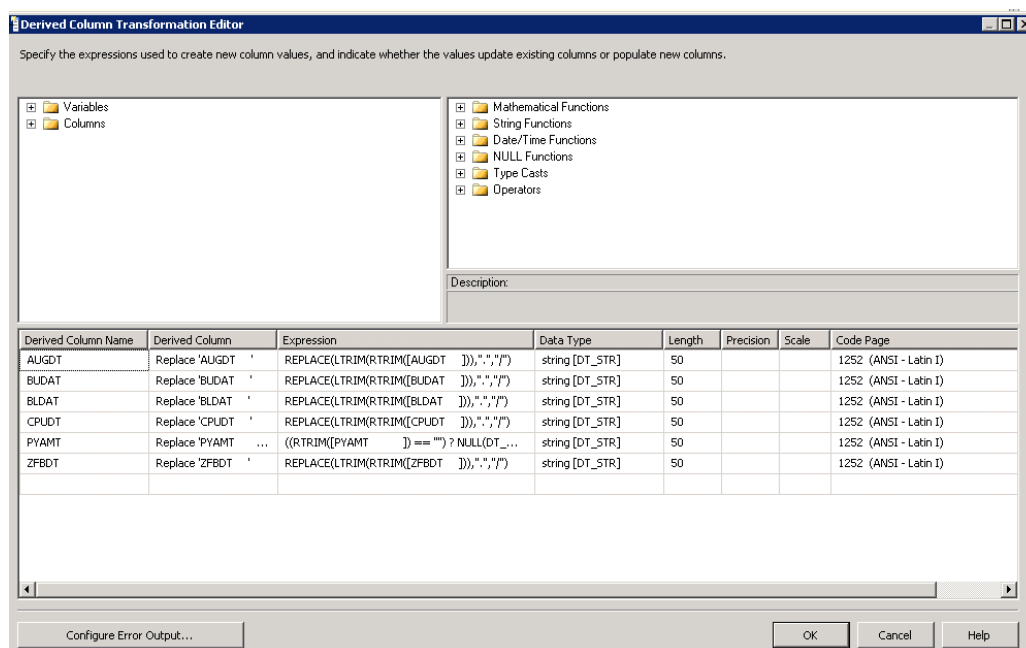
Date in form dd.mm.yyyy Error

You will need to convert the date to dd/mm/yyyy or to dd-mm-yyyy. If you have any other form in that date field, you may get any the error (see below). Sometimes the field might be populated with 00.00.0000 or blanks / spaces, NULLS

```
Error: 0xC02020A1 at DFTC-2011-REGUH, Flat File Source [1]: Data conversion failed. The data conversion for column "LAUFD" returned status value 2 and status text "The value could not be converted because of a potential loss of data."
Error: 0xC0209029 at DFTC-2011-REGUH, Flat File Source [1]: SSIS Error Code DTS_E_INDUCEDTRANSFORMFAILUREONERROR. The "output column "LAUFD" (40)" failed because error code 0xC0209084 occurred, and the error row disposition on "output column "LAUFD" (40)" specifies failure on error. An error occurred on the specified object of the specified component. There may be error messages posted before this with more information about the failure.
```

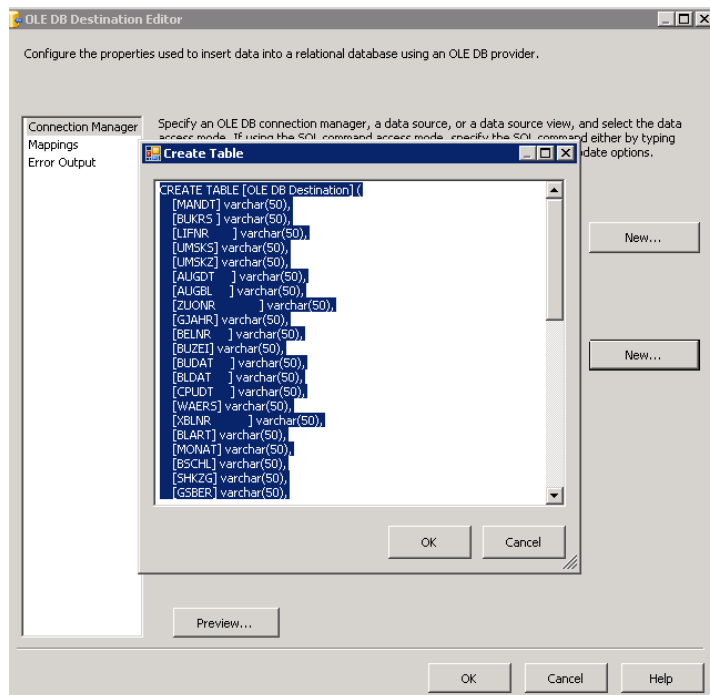
If loading from flat files one suggestion is to convert every data field which needs to be converted on the advanced page of the Flat File Connection Manager except the date field. The date field to be put to string [DT_STR] on this page.

From the flat file source, we then connect it to a derived column where we will write our date fields like this:



The first line basically means for every AUGDT field replace with an equivalent AUGDT, but for every . (dot) replace with a / (forward slash). This is the statement that is converting dd.mm.yyy with an dd/mm/yyyy.

The conversion to a date field will then be done in the OLE DB Destination. On the OLE DB Destination connection manager, click New under Name of table or the view.



On the first line **CREATE TABLE [OLE DB Destination](**, you can change **OLE DB Destination** to the name you want to give the table. Then under the column names you can now change the data type ie under AUGDT change it to **[AUGDT] datetime**.. You can also do it here for real or decimal (15,2) if it was not changed on connection manager.

Sometimes under date fields you will find spaces, where you can then substitute NULLs. In derived columns you can write like this under the expression:

```
REPLACE(LTRIM(RTRIM([AUGDT ])),",","/") == "00/00/0000" ? NULL(DT_WSTR,150) :
REPLACE(LTRIM(RTRIM([AUGDT ])),",","/")
```

The language of building expressions can be a bit disorienting. The key to being proficient in building expressions, is in the understanding of the syntax of this new scripting language which is a combination of C#, Visual Basic and sometimes TSQL. Some examples can be found in the appendix.

From the previous Derived column there is an PYAMT amount field which was causing errors as well. Some of the fields in PYAMT fields were empty. The solution is replace Only empty fields with NULLs.

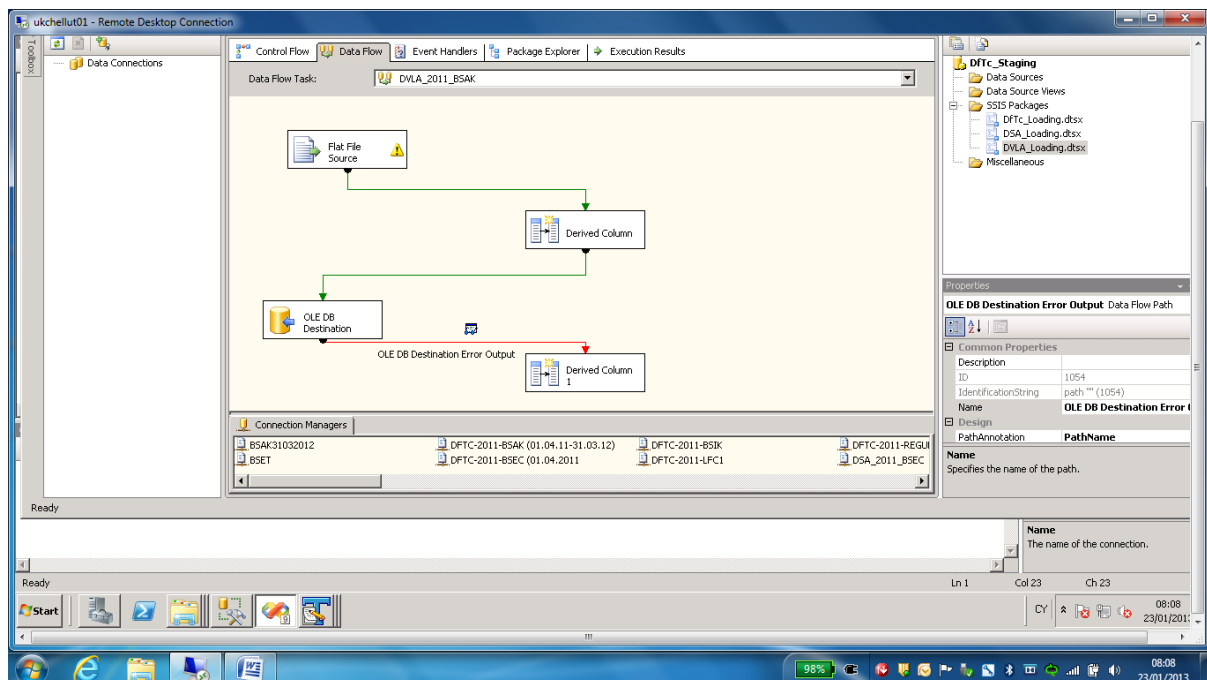
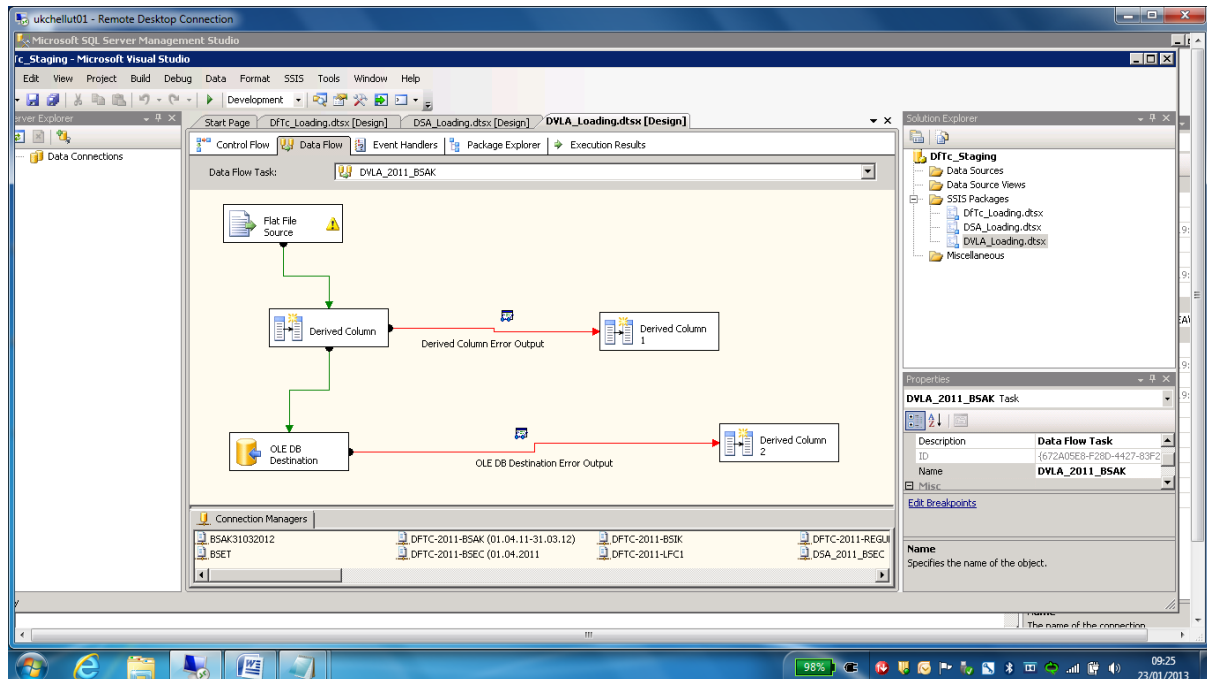
```
((RTRIM([PYAMT ]) == "") ? NULL(DT_WSTR,50) : [PYAMT ])
```

Sometimes it is confusing to find out where the errors are. The error output can be directing to the destination, yet it can be at source. The following diagram with data viewers can be used to find out the errors.

The following code will check for a blank Column

```
LTRIM([State]) == "" || LTRIM([City]) == ""
```

If you have multiple cases, always place conditions that you feel will capture most of the records at the top of the list because the list is read top to bottom and you don't want to evaluate records more times than needed.



The data viewers capture a certain number of rows at a time. You need to keep pressing play > on data viewer until data tasks turn to RED. You have to view the ‘ before’ and ‘after’ and to be sure you have the right data viewer.

To add a Data Viewer, select the connector error, leaving the component that you want to see data for. Right click this connection and select Data Viewers. The Data Flow Path Editor will pop up. Click Add to add the Data Viewer. On the Configure Data viewer screen, select Grid as the type. Click the Grid tab and make sure all the columns you wish to see are in the Displayed Columns list. Close out this window and the Data Path Flow window by clicking OK.

BELNR	BUZEI	BUDAT	BLDAT	CPUDT	WAERS	XBLNR	BLART	MONAT
11208	001	20/02/2012	20/02/2012	20/02/2012	GBP		ZP	11
12354	001	05/03/2012	29/02/2012	06/03/2012	GBP	N-000948183	RE	12
12356	001	05/03/2012	29/02/2012	06/03/2012	GBP	N-000948182	RE	12
12357	001	05/03/2012	29/02/2012	06/03/2012	GBP	N-000948181	RE	12
12358	001	05/03/2012	28/02/2012	06/03/2012	GBP	N-000948167	RE	12
12359	001	05/03/2012	28/02/2012	05/03/2012	GBP	N-000948166	RE	12
12361	001	05/03/2012	28/02/2012	06/03/2012	GBP	N-000948168	RE	12
12362	001	05/03/2012	28/02/2012	06/03/2012	GBP	N-000948163	RE	12
12363	001	05/03/2012	28/02/2012	06/03/2012	GBP	N-000948162	RE	12
12364	001	05/03/2012	28/02/2012	05/03/2012	GBP	N-000948165	RE	12
12366	001	05/03/2012	28/02/2012	05/03/2012	GBP	N-000948164	RE	12
12381	001	05/03/2012	29/02/2012	05/03/2012	GBP	N-000000499	RE	12
12382	001	05/03/2012	29/02/2012	05/03/2012	GBP	N-000000498	RE	12
10862	001	06/03/2012	06/03/2012	06/03/2012	GBP		ZP	12
12367	001	05/03/2012	29/02/2012	21/03/2012	GBP	N-000948208	RE	12
13059	001	21/03/2012	15/03/2012	21/03/2012	GBP	N-000948873	RE	12
14747	001	22/03/2012	22/03/2012	22/03/2012	GBP		ZP	12
12895	001	19/03/2012	15/03/2012	19/03/2012	GBP	N-000948873	RE	12
12897	001	19/03/2012	15/03/2012	20/03/2012	GBP	N-000948845	RE	12
10042	001	20/03/2012	20/03/2012	20/03/2012	GBP		ZP	12

The previous screen helps with the debugging.

Text Qualifiers' Errors

If you have the most common text qualifier of double-quotes around your data, a row may look like the following, where there are only three columns, even though the commas may indicate five:

“Knight, Brian”, 123, “ Bedford, BEDS”

OR another case where the text in one column might be treated as belonging to several columns, just because of an extra | in between the text.

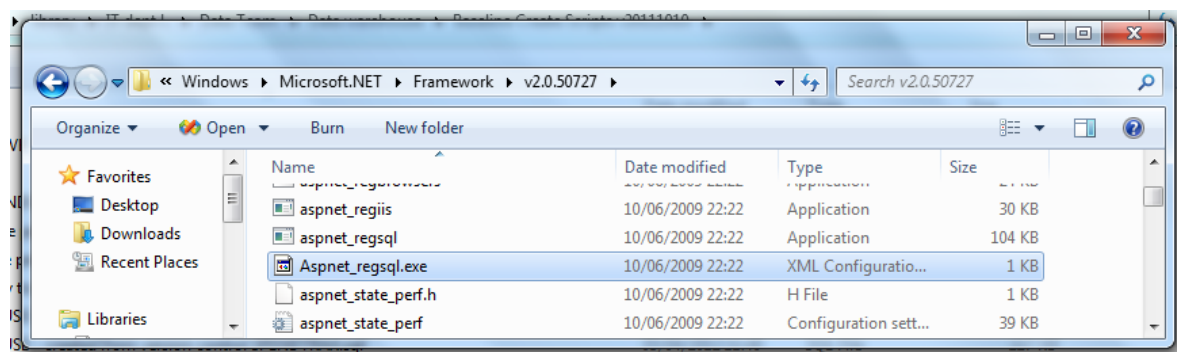
Knight | Brian | 123 |Living in | Bedford |BEDS

Creating Consol and ASP Database for APTrax

1. The location of the scripts for creating the Consol and ASPnet tables is
<\\192.168.3.16\\library\\IT dept L\\Data Team\\Data warehouse\\Baseline Create Scripts v20111010>
2. \\ukcfd01srfs01
3. Create a database with the company in the format *[Company Name][Consol]*
4. Open 1.Tables USE - created from version control INTERNAL APTRAX file and run the script within the Consol db (Run twice)

Name	Date modified	Type	Size
INTERSERVE	09/01/2012 11:34	File folder	
Legacy	08/06/2012 11:05	File folder	
NEW SPENDCATS PREVIOUS VERSIONS	27/01/2012 10:42	File folder	
SPROC file per object	09/01/2012 13:10	File folder	
TABLE file per object DNU - now in version control	16/01/2012 10:54	File folder	
(odi query to supplement the new spend cat proc).sql	12/09/2011 16:08	SQL File	5 KB
1.Tables USE - created from version control INTERNAL APTRAX.sql	13/12/2012 17:33	SQL File	167 KB
1.Tables USE - created from version control SPENDTRAX.sql	05/04/2012 11:46	SQL File	227 KB
2 - Create aspnetdb tables.sql	09/01/2012 12:37	SQL File	3 KB
3 - Create Views ADDITIONAL SCRIPTS INTERNAL APTRAX.sql	11/06/2012 11:39	SQL File	8 KB
3 - Create Views.sql	09/01/2012 12:41	SQL File	9 KB
4 - Create Functions v3 - created from version control INTERNAL APTRAX.sql	23/07/2012 10:39	SQL File	10 KB
4 - Create Functions.sql	07/10/2011 11:58	SQL File	35 KB
5 - Create Sprocs - created from version control INTERNAL APTRAX.sql	12/11/2012 16:18	SQL File	1,707 KB
5 - Create Sprocs v3.sql	09/01/2012 13:10	SQL File	1,137 KB
6 - Populate Config Tables ADDITIONAL SCRIPTS ATL_DUP 1 set up defaults.sql	03/12/2012 13:02	SQL File	11 KB
6 - Populate Config Tables ADDITIONAL SCRIPTS ATL_DUP 2 TEST vs Run xref.sql	06/06/2012 16:33	SQL File	8 KB
6 - Populate Config Tables ADDITIONAL SCRIPTS ATL_vat 1 set up defaults.sql	07/06/2012 14:14	SQL File	6 KB
6 - Populate Config Tables.sql	11/06/2012 13:09	SQL File	21 KB
7 - Create DocumentExplorer Tables.sql	26/01/2010 19:32	SQL File	5 KB
8 - CREATE SCRIPT FOR NEW UPDATE SPENDCATS v4.sql	20/12/2012 09:06	SQL File	20 KB
9 - Triggers.sql	11/06/2012 11:13	SQL File	15 KB

5. Run C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regsql.exe



From remote desktop run script to add tables to the aspnetdb*client* database

```
-- Run C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regsql.exe to
create the security database
```

-- THEN SET THE CONNECTION TO THE NEW ASPNET DB YOU JUST CREATED

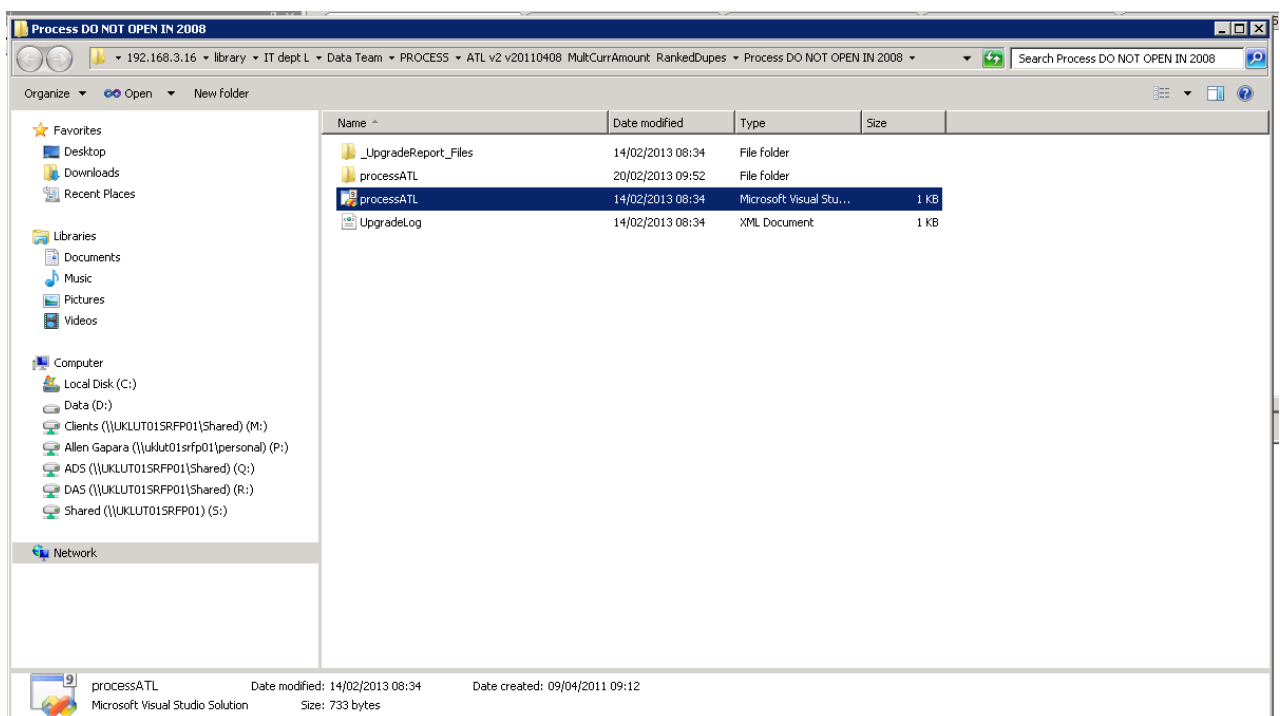
6. In the ASP.NET SQL server setup wizard, enter the destination server and enter a database name for the ASP database in the format *[aspnetdb_][Company Name]*
7. Open [2 - Create aspnetdb tables] file and run the script in the Aspnet db created above.
8. Open [3 - Create Views ADDITIONAL SCRIPTS INTERNAL APTRAX] file and run the script twice in the Consol db
9. Open [3 - Create Views] file and point the *set @aspnetdb=* to the aspnet database created in step 5 then run the script from the Consol database
10. Open [4 - Create Functions v3 - created from version control INTERNAL APTRAX] file and run the script from the consol database (If there is an error about Parsing GO, then Just replace every instance of */****** with *--* in the script). Run twice and ignore any Error about Invalid Object name 'dbo.Category UNSPSC V12'
11. Open [4 - Create Functions] file and run the script in the consol database
12. Open [5 - Create Sprocs - created from version control INTERNAL APTRAX] file and run the script in the consol database twice (Ignore any errors after running twice)
13. Open [5 - Create Sprocs v3] and run the script from the consol database
14. Open [6 - Populate Config Tables ADDITIONAL SCRIPTS ATL_DUP 1 set up defaults] and run in the consol database
15. Open [6 - Populate Config Tables ADDITIONAL SCRIPTS ATL_DUP 2 TEST vs Run xref] and run in the consol database
16. Open [6 - Populate Config Tables ADDITIONAL SCRIPTS ATL_vat 1 set up defaults] and run in the consol database
17. Open [6 - Populate Config Tables] and run the script in the consol database
18. Open [9 - Triggers] and run the script in the consol db
19. Move the data from Staging to Consol using SSIS
 - Deselect All keys except companykey
 - Deselect Buref, Insertdate, RowID
 - Make sure there are no spaces *LTRIM(RTRIM(a.mandt))*
 - Check to see whether the table & Data have moved.
 - Check where columns / destination are pointing
 - Check the Lookup paths

20. Update system_config table with the appropriate reporting currency

System Config table

ConfigItem	Value
DateAnalysisField	invoiceprocessdate
DateFunction	yearquartermonth
ConsolCurrency	USD
ResourcePath	D:\home\Default\SpendGuardianDev\App_GlobalResources
ResourceFile	ResourcesAll
CategoryLevels	4
IncludeCodesInCategoryDescriptions	No
MaxDetailDrillRows	2000
consolcurrencysymbol	\$
NULL	NULL

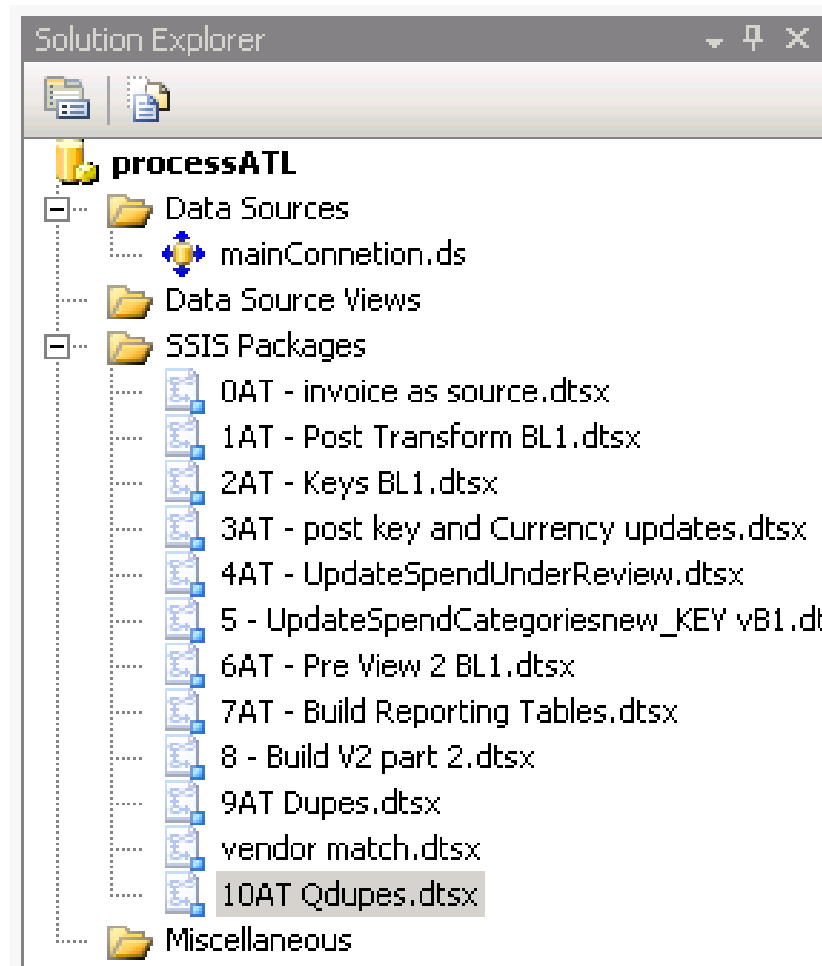
21. Go to <\\192.168.3.16\library\IT dept L\Data Team\PROCESS\ATL v2 v20110408 MultCurrAmount RankedDupes\Process DO NOT OPEN IN 2008> to copy the process package.



Do not open in 2008 is just a name, it is not an instruction. Copy the above URL and then click start on the remote desktop, and then click RUN. Paste the URL in the RUN window.

Click OK and then double-click the above and it will open an upgrade package. Just run the upgrade package and ignore any errors.

A warning about vendor match.dtsx may come up. "The file has been modified outside of the source editor. Do you want to reload it? ". Just click **NO to all**.



22. Set the mainConnection data source to point to the right server as well as Consol Database. On main connection under Data sources double-click it, to open it and then select Native ... and then Select Server under connection string.
23. In the Process package, execute steps 1-4, 6-7 to populate various calculated fields and summary reports

Alternatives:

If the data received is line level, populate the Invoice table and execute step 0AT in the process package. This would roll up the invoices and push them into the InvoiceHeader. DO NOT run step 0AT if the InvoiceHeader is populated. This would truncate the InvoiceHeader table.

Error Handling:

While running any of the baseline scripts, there could be error messages when executing them the first time. This is because few of the tables/sprocs could be dependent on other tables/sprocs getting generated within the same script. Execute the scripts till they are successfully executed.

After Process Step 1

```
SELECT * FROM InvoiceHeader WHERE datekey is NULL
```

Error on Step 1: CREATE INDEX failed because the following SET options have incorrect setting: 'ANSI_PADDING' Verify that SET options are correct for use with indexed views..
SOLUTION: Check the code, and change OFF to ON.

64 bit error. SOLUTION

In checking for nulls, if you find not nulls then you know you have a problem. In process package, Step 1AT, if there are any errors, the tables to concentrate are InvoiceHeader, Dim_date, Dim_date_days, and Companies. After step 1AT – Please check that there is no null DateKey in the InvoiceHeader

After Process step 2

```
SELECT * FROM InvoiceHeader WHERE Invoicekey is NULL
```

```
SELECT * FROM InvoiceHeader WHERE Vendorkey is NULL
```

```
SELECT * FROM PaymentLink WHERE Invoicekey is NULL
```

```
SELECT * FROM PaymentLink WHERE Paymentkey is NULL
```

```
SELECT * FROM PaymentLink WHERE Vendorkey is NULL
```

Errors after Process step 2:

After Step 2AT, check if there are any null InvoiceKeys, VendorKeys in the InvoiceHeader. If there are null VendorKeys in the InvoiceHeader, then check the VendorMaster to see if the VendorID exists. If not, add the VendorID in the VendorMaster and populate the VendorName as '[VendorID] Blank at Source'. Execute Step 2AT again.

LedgerID should be the same on PaymentLink and payments.
CompanyRef should be in the tables (This means you have to import it as well)
When transforming, use identical ways to select columns between payments and PaymentLink, otherwise the systems thinks there are different columns. Select LTRIM(RTRIM(a.mandt)) is different to Select a.mandt
Pay careful attention to trailing spaces, case sensitivity and errors on lookup.

Error: Invalid column name 'voidindicator' SOLUTION: Disable "temp asda delete". This is only for the ASDA project.

If you have NULL Invoicekeys in PaymentLink, it means there is no record of that invoice in the InvoiceHeader.

If you have NULL PaymentKeys in PaymentLink, it means there is no record of that payment in Payments.

If you have NULL VendorKeys in PaymentLink, it means there is no record of that vendor in the Vendor table.

After Process step 3

```
SELECT * FROM InvoiceHeader WHERE InvoiceGrossAmount is NULL
SELECT * FROM InvoiceHeader WHERE InvoiceNetAmount is NULL
SELECT * FROM InvoiceHeader_CalculatedColumns
```

The last query will show some items in the table.

After Process step 4

```
SELECT * FROM VendorMaster WHERE SpendUnderReview is NOT NULL
SELECT DISTINCT VendorID FROM InvoiceHeader
```

Pick one of the vendors and compare spending under review amount with the Gross amount for that vendor from InvoiceHeader.

```
select * from Vendormaster where spendunderreview is not null
```

```
select [InvoiceGrossAmount] from InvoiceHeader where [VendorID] =
'0004015342'
select [InvoiceGrossAmount] from InvoiceHeader where [VendorID] =
'0004015344'
```

```
select * from InvoiceHeader where [VendorID] = '0004015344'
```

Skip Process STEP 5

After Process step 6

```
SELECT * FROM VendorNameDimension
```

After Process step 7

```
SELECT * FROM summaryatl_all
```

Error: If the website already exists there is no need to run “ add in dummyspendcategorycodetable”. Disable this box first and then run &AT

24. In the Process package, execute step 10 to create the Dupe reports

25. Execute the VendorMatch package as well.

In SQL Server, go the Consol database and check if the reports 1-9 have been populated and the scores have been calculated for all the reports

Skip Process Step8 and Step 9

After Process step 10

```
select COUNT (*) from report1
select COUNT (*) from report2
select COUNT (*) from report3
select COUNT (*) from report4
select COUNT (*) from report5
select COUNT (*) from report6
select COUNT (*) from report7
select COUNT (*) from report8
select COUNT (*) from report9
```

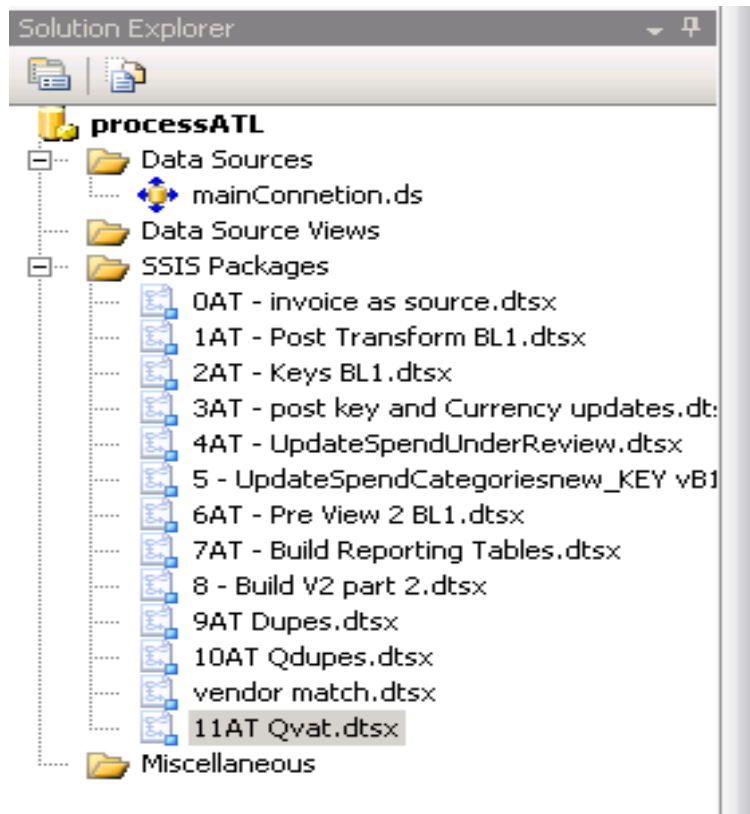
In Step 10AT, if there is an error at the start, then executing ‘Execute SQL Task 2’ individually and try executing the entire step.

26. To run Qvat go to <\\192.168.3.16\library\IT dept L\Data Team\PROCESS\ATL v2 v20120111 MCurrAmt RDup QVat> to copy the Qvat process package . Double-clickProcess package ATL when it comes up first.

27. Set the MainConnection data source to point to the right server as well as Consol Database and execute just step 11

If you need to run Step 11 again for some reason you will get an error. Depending on the data modified in the consol, you may need to rerun some of the 1-11 steps again. The solution to an error after running step 11 again is to delete an Ind3 index. Go to consol database and look for **dbo.ATL_VAT_Testscores**, Indexes, **Ind3(Non-unique, Non-clustered)**

To upload the reports in the site, go to [\\192.168.3.16\library\IT dept L\Data Team\Report Baselining\Audit Trax](http://192.168.3.16/library/IT dept L/Data Team/Report Baselining/Audit Trax)



28. (If the report packages have been run before, there is no need to run them again)

The report packages to be run first time are:

- ATL v2 dupes v20111013
- ATL v2 v20110424
- Usage
- VendorATL v2 v20110424
- ATL QVAT v20110509

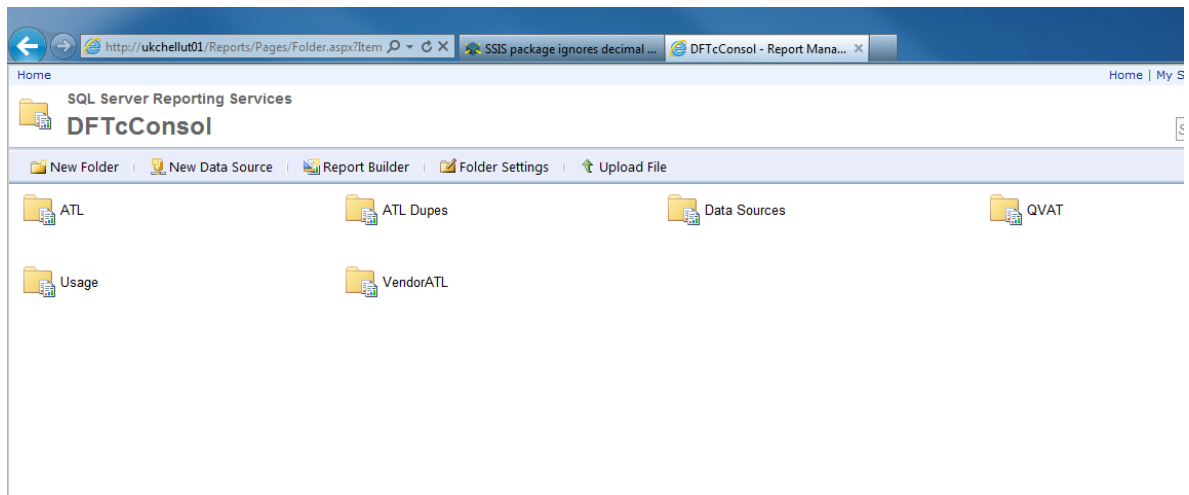
29. For each report package, change the main source as the source database

30. For each report package, go to Projects -> ATL/ATL Dupe properties and make the following changes

- TargetDataSourceFolder – [Source DB]/Data Sources
- TargetReportFolder – [Source DB]/ATL(ATL Dupes]
- TargetServerURL – http:// [Server Name]/reportserver

31. For each report package, select all the reports within the folder and deploy them

32. Go to [http://\[Server name\]/reports](http://[Server name]/reports) to check whether the reports have been deployed



33. The folder to be copied is [within the server] -> **C:\inetpub\wwwroot** Change the folder name to **[APTrax][ConsolName]** (This would be the site name)

WebSite

34. Open Microsoft Visual Web Developer 2010 Express as any administrator. Click File-> Open Website. Navigate to the destination of the copied folder and click Open
35. In the Solution Explorer window on the right of the panel, navigate to Web.Config and open the file.
36. In line 13, change the value field in `<add key="ReportFolder" value="<!--Enter the consol name-->"/>` from `<!--Enter the consol name-->` to `"[ConsolName]"`
37. In line 18, change the Initial catalog field in `<add name="mainConnectionString" connectionString="Data Source=localhost;Initial Catalog=<!--enter the consol name-->"` from `<!--enter the consol name-->` to `[ConsolName]`
38. In line 19, change the Initial catalog field in `<add name="aspnetdbConnectionString" connectionString="Data Source=localhost;Initial Catalog=<!--enter the aspnet database name-->"` from `<!--enter the aspnet database name-->` to `[aspnetdb name]` and click Save
39. Open Internet Information Services (IIS) manager in administrator mode.
40. In the Connections window on the left hand panel, navigate to ServerName-> sites->default web site-> created website
41. Right click on the website and click convert to application
42. Click select in the add application window and change the Application pool to 'ASP.NET v4.0'. Click Ok to convert to application
43. Go to Microsoft Visual Web Developer 2010 Express. Click on the right most icon in the top of Solution Explorer window. It should say ASP.NET configuration

44. In the opened webpage, click on Security-> Users-> Create user
45. Enter a username in the format *[last name][first letter of the first name]* and enter the other details
46. The roles to be (ticked) selected are: (Tick all for one user)
 - a. **Administrators**
 - b. **CCMUser**
 - c. **SuperAdmin**
 - d. **SupplierUser**
 - e. **Users**

And click Create User

47. Click on Manage Users->edit user and enter the name of the person in the description field and click Save
48. Go to the Consol Database in SQL server management studio (**Consol, Programmability, Stored Procedures, System Stored Procedures, RHC dbo.updateusercompanies**) and Execute the Stored Procedure 'UpdateUserCompanies' Every time you update users, you have to execute this stored procedure.
49. Go to Internet Explorer and enter [http://\[ServerName\]/\[APTrax\]/\[ConsolName\]](http://[ServerName]/[APTrax]/[ConsolName]) (the folder name of the website) and check if the site opens

When moving the database from development server to the production server, truncate the workflow tables, they are

- CCMCommFiles
- CCMfiles
- CCMItems
- CCMItemsAmend
- CCMItemsBankDetails
- CCMItemsComm
- CCMItemsNotes
- CCMPotentialItems
- CCMPotentialItemsBackup
- CCMSelection
- CCMStatements
- CCMTemplates
- CCMTemplatesAmend
- CCMVendorComm
- CCMVendorNotes

Going Live

1) Backup

- Before going to the Live site, backup the database first. This is normally done in backups in ukchellut01\D\Backups. For sharing, backup in dropzone.
- Backup both the security and consol database. Backing up a compressed file, makes it easier to transfer the files.
- Chellut02 is the live site. We work on 01 and 03 and then deploy on 02 when going live.
- After compression move the compresses file to 02 and then delete the uncompressed file.
- In ukchellut02 Run AS: Restore database: Do it, for only the consol and not the security databases aspnetdb.
- If it is an on-going project, you will need to keep the workflow tables. Create a new temporary database in unchellut02 then;

Copying Method1

Open SQL Server management studio

Right Click on Database name > Tasks > Export Data.

Provide login credentials and Select Database

In next screen select database where table is to be copied

Copy data from one or more tables or views

check tables you want to copy

finish the wizard

Copying Method2

You can write this query to copy tables.

```
SELECT * INTO DestinationDB.dbo.tableName FROM SourceDB.dbo.SourceTable
```

This command only copies table schema and data

After copying the tables, you can then backup the consol and then restore the consol from ukchellut01. You can then update the workflows by the following: Set Identity if you have an identity error otherwise omit it.

```
SET IDENTITY_INSERT dbo.CCMCommFiles ON
INSERT INTO CCMCommFiles ([Folder] , [FilePath] , [UploadDate] , [ItemRef]
, [ItemID] , [FileContents] , [FileType] , [RowID] , [UploadedBy])
SELECT [Folder]
      , [FilePath]
      , [UploadDate]
      , [ItemRef]
      , [ItemID]
      , [FileContents]
      , [FileType]
      , [RowID]
      , [UploadedBy] FROM [TempExperian].[dbo].[CCMCommFiles]

SET IDENTITY_INSERT dbo.CCMCommFiles OFF
```

After that, you just type updateusercompanies and run it.

2) Website

For a new site, you do the following: (for updating sites you do not need to do these steps)

Copy the website from 01, C:\inetpub\wwwroot\APTraxHO , to 02 , C:\inetpub\wwwroot\ APTraxHO.
(you can copy the latest site and change the name and then paste it into 02) If you change the name, you will have to change the paths in web developer studio as an administrator.

Then you need to deploy reports on the new sites like what was done in 01.

To upload the reports in the site, go to [\\192.168.3.16\library\IT dept L\Data Team\Report Baselining\Audit Trax](http://192.168.3.16/library/IT dept L\Data Team\Report Baselining\Audit Trax)

(If the report packages have been run before, there is no need to run them again)

The report packages to be run first time are:

- a. ATL v2 dupes v2011013
- b. ATL v2 v20110424
- c. Usage
- d. VendorATL v2 v20110424
- e. ATL QVAT v20110509

For each report package, change the main source as the source database

For each report package, go to Projects -> ATL/ATL Dupe properties and make the following changes

- f. TargetDataSourceFolder – [Source DB]/Data Sources
- g. TargetReportFolder – [Source DB]/ATL(ATL Dupes]
- h. TargetServerURL – http:// [Server Name]/reportserver

For each report package, select all the reports within the folder and deploy them

Go to http://[Server name]/reports to check whether the reports have been deployed

Avoiding Misleading Reports

- Make sure DOC amounts and LOC amounts are populated.
- VendorInvoiceRef should have valid characters. VendorInvoiceRef is derived from SupplierInvoiceNumber and VendorInvoiceRef3 is derived from SupplierInvoiceNumber. VendorInvoiceRef3 is populated automatically by taking VendorInvoiceRef and removing letters, symbols, commas and special characters to remain with letters only.

Uses of Process Package reports

0AT - invoice as source.dtsx

1AT - Post Transform BL1.dtsx

2AT - Keys BL1.dtsx
3AT - post key and Currency updates.dtsx
4AT - UpdateSpendUnderReview.dtsx
5 - UpdateSpendCategoriesnew_KEY vB1.dtsx
6AT - Pre View 2 BL1.dtsx
7AT - Build Reporting Tables.dtsx
8 - Build V2 part 2.dtsx
9AT Dupes.dtsx
vendor match.dtsx
10AT Qdupes.dtsx
11AT Qvat.dtsx

Running Statistics for loaded Files

Purpose

- The statistics are needed by the mapping team
- They are PRGX way of data profiling. They provide a summary of the data, looking at NULLs, dates, date range, number of lines / files etc.

The stats are run by three scripts

USP_PRGDS
USP_PRGDSREPORT
USP_PRGLINKREPORT

Procedure

The three scripts are normally kept as .txt files on desktop. You will have to copy the text files to a new query window in SQL Server Management Studio.

- After copying the script to the new query window, remove **USE [UDF]**, and make sure it is pointing to the right database. You can select the right database, on the available databases window, top right hand side.
- **Execute** the script.
- Do the same for the second and third scripts.
- In Object Explorer, expand your database, expand Programmability, and expand **Stored Procedures**. (you should be able to see the 3 procedures)
- Click the 1st procedure dbo.USP_PRGDS and then right-hand click it. Select **Execute** the **Stored Procedure**.
- When a window opens write the name of the database under the **Value** column, on the first row. (You can copy the name of the db and then paste it.)
- Click **OK**
- Do the same for the 3rd script dbo.USP_PRGLINKREPORT. The third one is normally has errors when executed. Please just ignore the errors.
- Do the same for the 2nd script dbo.USP_PRGDSREPORT
- Select **Results to File** and then Click **EXECUTE**
- A **Save Results** window opens. You can then save to remote desktop as a **.txt** file. You should write the name as **NAME.txt**, but select **All files** on the drop-down window underneath the name. You can then copy to from the remote desktop to your normally desktop.
- You can now email these results to the mapping team.

Backup Database using SQL Server Management Studio

1. All ukchellut01 backups are to be done in **ukchellut01\D\Backups**.
2. Expand Databases and then select user database
3. Right-click the database, point to Tasks, and then click Backup. The Backup Database dialog box appears.
4. In the Database list box, verify the database name. You can optionally select a different database from the list.
5. In the Backup type list box, select **FULL**
6. For backup component click Database
7. Either accept, the default backup set name suggested in the Name text box, or enter a different name for the backup set.
8. Optionally, in the Description text box, enter a description of the backup set.
9. Choose the type of backup destination by clicking **Disk**. If the destination does not agree with what you want, click Remove. To view the contents of a backup destination, select it and click Contents.
10. Click Add to specify a destination of the backup. E.g.
D\BACKUPS\FULLBACKUPS\ExperianConsol\ExperianConsol_backup_2013_01_30
11. You can optionally add an extension to the file names, **.full.bak** for full backups or **.diff.bak** for differential backups.
12. Click OK
13. After getting the message “the backup was successful”, go to the destination folder.
Right-click your backup file and compress it as .rar
14. When it completes successfully, you can delete the uncompressed file.
15. Do the same for the security database **aspnetdb_Experian**

Restore Database

To restore a full database backup

After you connect to the appropriate instance of the Microsoft SQL Server Database Engine, in Object Explorer, click the server name to expand the server tree.

Expand Databases. Depending on the database, select a user database.

Right-click the database, point to Tasks, point to Restore, and then click Database, which opens the Restore Database dialog box.

On the General page, use the Source section to specify the source and location of the backup sets to restore. Select one of the following options:

Database: Select the database to restore from the drop-down list. The list contains only databases that have been backed up according to the msdb backup history.

If the backup is taken from a different server, the destination server will not have the backup history information for the specified database. In this case, select Device to manually specify the file or device to restore.

Device

Click the browse (...) button to open the Select backup devices dialog box. In the Backup media type box, select one of the listed device types. To select one or more devices for the Backup media box, click Add.

After you add the devices you want to the Backup media list box, click OK to return to the General page.

In the Source: Device: Database list box, select the name of the database which should be restored.

Note: This list is only available when Device is selected. Only databases that have backups on the selected device will be available.

Backup location

View, add, or remove media for the restore operation.

Add

Adds the location of a backup device to the Backup location list. Depending on the type of media you select in the Backup media field, clicking Add opens one of the following dialog boxes.

Media type	Dialog box	Description
File	Locate	In this dialog box, you can select a local file from the tree or specify a

	Backup File	remote file using its fully qualified universal naming convention (UNC) name.
--	-------------	---

If the list is full, the Add button is unavailable.

Remove: Removes one or more selected files, tapes, or logical backup devices.

Contents: Displays the media contents of a selected file, tape, or logical backup device.

In the Destination section, the Database box is automatically populated with the name of the database to be restored. To change the name of the database, enter the new name in the Database box.

In the Restore to box, leave the default as To the last backup taken or click on Timeline to access the Backup Timeline dialog box to manually select a point in time to stop the recovery action.

In the Backup sets to restore grid, select the backups to restore. This grid displays the backups available for the specified location. By default, a recovery plan is suggested. To override the suggested recovery plan, you can change the selections in the grid. Backups that depend on the restoration of an earlier backup are automatically deselected when the earlier backup is deselected.

Optionally, click Files in the Select a page pane to access the Files dialog box. From here, you can restore the database to a new location by specifying a new restore destination for each file in the Restore the database files as grid.

To view or select the advanced options, on the Options page, in the Restore options panel, you can select any of the following options, if appropriate for your situation:

WITH options (not required):

Overwrite the existing database (**WITH REPLACE**)

Preserve the replication settings (**WITH KEEP_REPLICATION**)

Restrict access to the restored database (**WITH RESTRICTED_USER**)

Select an option for the Recovery state box. This box determines the state of the database after the restore operation.

RESTORE WITH RECOVERY is the default behaviour which leaves the database ready for use by rolling back the uncommitted transactions. Additional transaction logs cannot be restored. Select this option if you are restoring all of the necessary backups now.

RESTORE WITH NORECOVERY which leaves the database non-operational, and does not roll back the uncommitted transactions. Additional transaction logs can be restored. The database cannot be used until it is recovered.

RESTORE WITH STANDBY which leaves the database in read-only mode. It undoes uncommitted transactions, but saves the undo actions in a standby file so that recovery effects can be reverted.

Take tail-log backup before restore will be selected if it is necessary for the point in time that you have selected. You do not need to modify this setting, but you can choose to backup the tail of the log even if it is not required.

Restore operations may fail if there are active connections to the database. Check the Close existing connections option to ensure that all active connections between Management Studio and the database are closed. This check box sets the database to single user mode before performing the restore operations, and sets the database to multi-user mode when complete.

Select **Prompt** before restoring each backup if you wish to be prompted between each restore operation. This is not usually necessary unless the database is large and you wish to monitor the status of the restore operation.

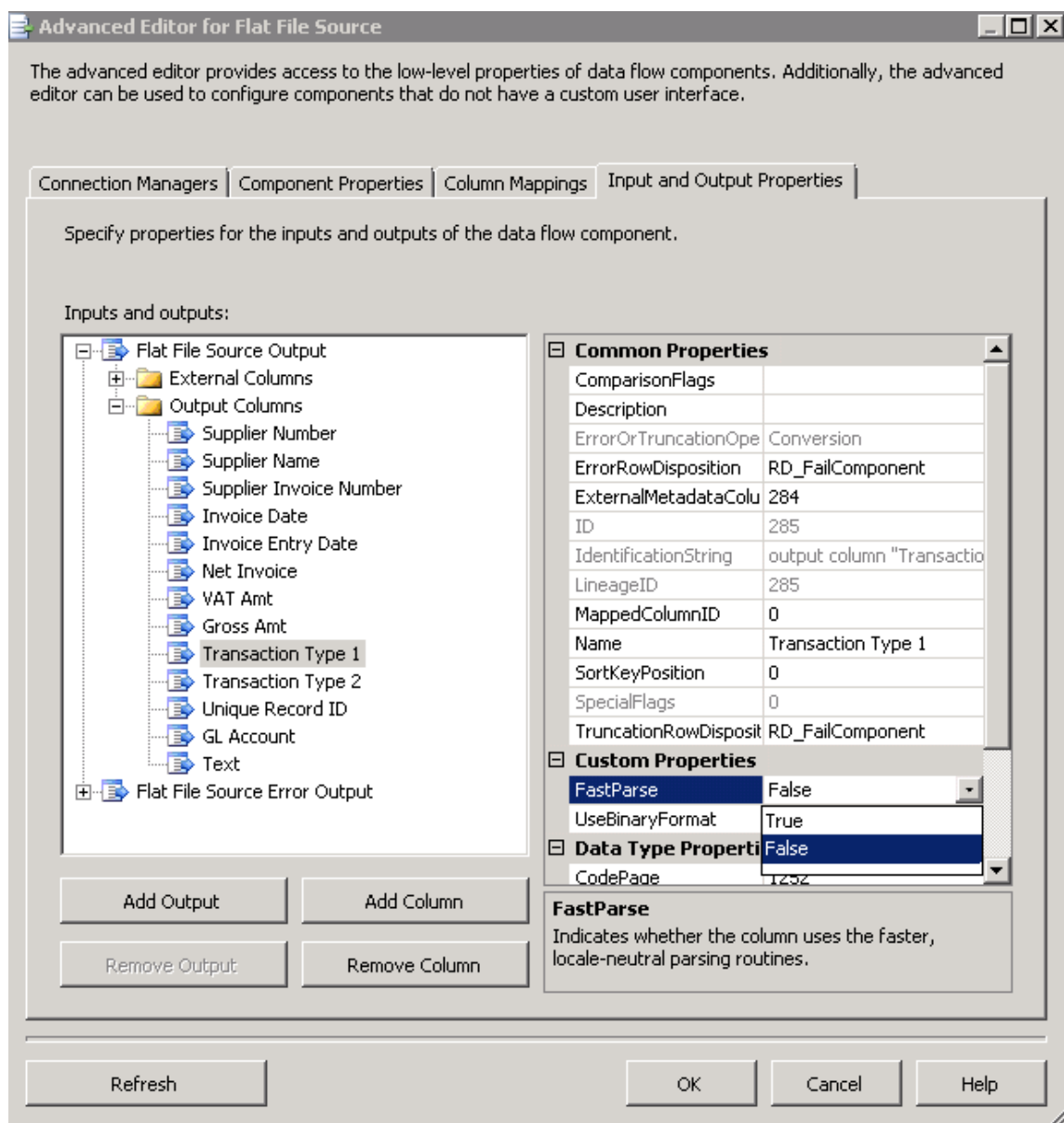
Click **OK**.

Best Practices

Extracting Data from Sources

Using Fast Parse can drastically improve performance of your package when you are using a Flat File source. By default, SSIS validates any numeric or date columns but with Fast Parse set to **True** this step will be bypassed. To enable Fast Parse, follow these steps:

- Right-click the Flat File Source or Data Conversion transformation, and click Show Advanced Editor.
- In the Advanced Editor dialog box, click the column for which you want to enable Fast Parse
- In the Inputs and Outputs pane, click the column for which you want to enable Fast Parse
- In the Properties window, expand the Custom properties node, and set the Fast Parse property to True
- Click OK



Appendices

The following consol tables will help us know what the final table is like. We can easily determine which are primary keys, names, column- width and data types.

Companies Table

The following table helps in matching our columns to the final companies table, which should not be changed.

```
CREATE TABLE [dbo].[companies](
    [LedgerID] [int] NOT NULL,
    [CompanyRef] [varchar](20) NOT NULL,
    [companykey] [int] NOT NULL,
    [companydesc] [varchar](255) NOT NULL,
    [Country] [varchar](50) NULL,
    [LocalCurrency] [varchar](50) NULL,
    [currSymbol] [varchar](2) NULL,
    [currSymbolconsol] [varchar](2) NULL,
    [countryName] [nvarchar](100) NULL,
    [RowID] [int] IDENTITY(1,1) NOT NULL,
    [Region] [nvarchar](100) NULL,
    [joinsOn] [char](1) NULL,
    [joinsOnVendor] [varchar](1) NULL,
    [Group] [nvarchar](50) NULL,
    [Platform] [nvarchar](50) NULL,
    [Cluster] [nvarchar](50) NULL,
    [VendorGroup] [int] NULL,
    [MatlGroup] [int] NULL,
    [PhaseGroup] [int] NULL,
    [CustGroup] [int] NULL,
    [TaxGroup] [int] NULL,
    [EMGroup] [int] NULL,
    [phase] [int] NULL,
    [AuditNumber] [nvarchar](50) NULL,
```

InvoiceHeader Table

The following table helps in matching our columns to the final table, which should not be changed.

```
CREATE TABLE [dbo].[InvoiceHeader] (
    [CDfiltercombinationkey] [int] NULL,
    [InsertDate] [smalldatetime] NOT NULL,
    [LedgerID] [int] NULL,
    [CompanyKey] [int] NULL,
    [CompanyREF] [varchar] (20) NULL,
    [BUREF] [varchar] (50) NULL,
    [bukey] [int] NULL,
    [InvoiceKey] [int] NULL,
    [InvoiceRef] [varchar] (100) NULL,
    [InvoiceID] [varchar] (100) NULL,
    [INVOICEID_HASH] [varbinary] (8000) NULL,
    [VendorID] [nvarchar] (50) NULL,
    [VendorKey] [int] NULL,
    [PurchaseOrder] [varchar] (50) NULL,
    [purchaseOrderKey] [int] NULL,
    [InvoiceTypeRef] [varchar] (50) NULL,
    [InvoiceTypeRef2] [varchar] (50) NULL,
    [VendorInvoiceRef] [nvarchar] (50) NULL,
    [VendorInvoiceRef3] [nvarchar] (50) NULL,
    [InvoiceComment] [nvarchar] (255) NULL,
    [InvoiceDate] [smalldatetime] NULL,
    [InvoiceDatekey] [int] NULL,
    [InvoiceProcessDate] [smalldatetime] NULL,
    [invoiceProcessDateKey] [int] NULL,
    [GLDate] [smalldatetime] NULL,
    [InvoiceCurrencyCode] [varchar] (50) NULL,
    [InvoiceGrossAmount] [decimal] (15, 2) NULL,
    [InvoiceGrossAmountDoc] [decimal] (15, 2) NULL,
    [invoiceGrossAmountLoc] [decimal] (15, 2) NULL,
    [InvoiceNetAmount] [decimal] (15, 2) NULL,
    [InvoiceNetAmountLOC] [decimal] (15, 2) NULL,
    [InvoiceNetAmountDOC] [decimal] (15, 2) NULL,
    [InvoiceTaxAmount] [decimal] (15, 2) NULL,
    [InvoiceTaxAmountLoc] [decimal] (15, 2) NULL,
    [InvoiceTaxAmountDoc] [decimal] (15, 2) NULL,
    [InvoiceDiscountAmountLOC] [decimal] (15, 2) NULL,
    [InvoiceDiscountAmountDOC] [decimal] (15, 2) NULL,
    [InvoiceFreightAmountLOC] [decimal] (15, 2) NULL,
    [InvoiceFreightAmountDOC] [decimal] (15, 2) NULL,
    [InvoiceTaxCode] [varchar] (50) NULL,
    [TaxRate] [float] NULL,
    [PaymentRef] [varchar] (50) NULL,
    [PaymentDate] [smalldatetime] NULL,
    [PaymentDatekey] [int] NULL,
    [paymentTermsRef] [varchar] (50) NULL,
    [paymentTermsKey] [int] NULL,
    [VendorSite] [varchar] (50) NULL,
    [EmployeeProcessor] [nvarchar] (50) NULL,
    [bucket] [varchar] (50) NULL,
    [Bucketkey] [int] NULL,
    [invoiceTypeKEY] [smallint] NULL,
    [DueDate] [smalldatetime] NULL,
    [DiscountTermsRef] [varchar] (50) NULL,
    [DiscountTermsKey] [int] NULL,
    [lines] [int] NULL,
    [datekey] [int] NULL,
```

Payments Table

The following table helps in matching our columns to the final table, which should not be changed.

```
CREATE TABLE [dbo].[Payments] (
    [paymentKey] [int] IDENTITY(1,1) NOT NULL,
    [PaymentRef] [varchar] (503) NULL,
    [paymentID] [varchar] (503) NULL,
    [LedgerID] [bigint] NOT NULL,
    [CompanyKey] [int] NULL,
    [CompanyRef] [varchar] (10) NOT NULL,
    [VendorID] [varchar] (50) NULL,
    [PaymentCurrencyCode] [varchar] (100) NULL,
    [PaymentAmountLOC] [money] NULL,
    [PaymentAmountDOC] [money] NULL,
    [PaymentMethod] [varchar] (100) NULL,
    [PaymentDate] [smalldatetime] NULL,
    [voidIndicator] [varchar] (100) NULL,
    [VoidReasonCode] [varchar] (100) NULL,
    [CheckNumber] [varchar] (100) NULL,
    [CheckPrintDate] [smalldatetime] NULL,
    [CheckVoidedDate] [smalldatetime] NULL,
    [source] [varchar] (4000) NULL,
    [chequeAmount] [money] NULL,
    [BankAccount] [varchar] (100) NULL,
    [SortCode] [varchar] (100) NULL,
    [PaymentTermsRef] [varchar] (255) NULL,
    [PaymentTermsKey] [int] NULL,
    [VendorKey] [int] NULL,
    [insertdate] [datetime] NOT NULL,
    [BatchNumber] [varchar] (50) NULL,
    [ReversalFlag] [bit] NULL,
    [ClearedDate] [smalldatetime] NULL,
```

PaymentLink Table

The following PaymentLink table helps in seeing the columns we want to match.

```
CREATE TABLE [dbo].[PaymentLink] (
    [PaymentLinkKey] [int] IDENTITY(1,1) NOT NULL,
    [LedgerID] [bigint] NOT NULL,
    [CompanyKey] [int] NULL,
    [CompanyRef] [varchar](10) NOT NULL,
    [InvoiceRef] [varchar](100) NULL,
    [invoiceID] [varchar](100) NULL,
    [InvoiceKey] [int] NULL,
    [PaymentRef] [varchar](503) NULL,
    [paymentID] [varchar](503) NULL,
    [payItem] [varchar](50) NULL,
    [PaymentKey] [int] NULL,
    [PaymentAmountLOC] [money] NULL,
    [PaymentAmountDOC] [money] NULL,
    [VendorID] [nvarchar](50) NOT NULL,
    [VendorKey] [int] NULL,
    [source] [varchar](4000) NULL,
    [insertdate] [datetime] NOT NULL,
```


VendorMaster Table

The following vendormaster table helps in seeing the columns we want to match.

```
CREATE TABLE [dbo].[VendorMaster] (
    [vendorkey] [int] IDENTITY(1,1) NOT NULL,
    [RowID] AS ([vendorkey]),
    [ledgerid] [int] NOT NULL,
    [companyREF] [varchar](20) NOT NULL,
    [companykey] [int] NULL,
    [VendorID] [nvarchar](50) NOT NULL,
    [VendorNumber] [varchar](12) NULL,
    [VendorName] [nvarchar](100) NOT NULL,
    [origVendorName] [nvarchar](100) NOT NULL,
    [VendorAddr1] [nvarchar](255) NULL,
    [VendorAddr2] [nvarchar](255) NULL,
    [VendorAddr3] [nvarchar](255) NULL,
    [VendorAddr4] [nvarchar](255) NULL,
    [VendorAddr5] [nvarchar](255) NULL,
    [VendorPostcode] [nvarchar](15) NULL,
    [VendorCountry] [nvarchar](30) NULL,
    [VendorState] [nvarchar](50) NULL,
    [vendorContact] [varchar](50) NULL,
    [vendorTelephone] [varchar](50) NULL,
    [vendorFax] [varchar](50) NULL,
    [VendorGroupCompany] [varchar](255) NULL,
    [vendorType] [varchar](50) NULL,
    [ContactName] [nvarchar](255) NULL,
    [ContactTelNo] [varchar](255) NULL,
    [BuyingContact] [varchar](255) NULL,
    [vatNumber] [varchar](20) NULL,
    [bankaccount] [varchar](50) NULL,
    [bankname] [nvarchar](255) NULL,
    [sortcode] [nvarchar](50) NULL,
    [spendUnderReview] [money] NULL,
    [internalExternal] [varchar](50) NULL,
    [SICCode] [varchar](10) NULL,
    [companyreg] [varchar](50) NULL,
    [DUNS] [varchar](50) NULL,
    [Turnover] [money] NULL,
    [Employees] [varchar](50) NULL,
    [proximity] [int] NULL,
    [BME] [decimal](18, 2) NULL,
    [CorpAffCategory] [varchar](max) NULL,
    [spendcategorycode] [varchar](20) NULL,
    [spendcategoryname] [varchar](255) NULL,
    [sckey] [int] NULL,
    [MappingOverride] [bit] NULL,
    [NAICS] [varchar](20) NULL,
    [USSIC] [varchar](20) NULL,
    [vendornamekey] [int] NULL,
    [vendorGroupCompanykey] [int] NULL,
    [vendororignamekey] [int] NULL,
    [EMAILADDRESS] [varchar](100) NULL,
    [_keyout] [int] NULL,
    [_keyoutspend] [money] NULL,
    [paymentTermsRef] [varchar](50) NULL,
    [paymentTermsKey] [int] NULL,
    [insertdate] [smalldatetime] NULL,
    [exclude] [bit] NULL,
    [lastupdated] [smalldatetime] NULL,
```

[firstactivitydate] [smalldatetime] NULL,
[lastactivitydate] [smalldatetime] NULL,
[lastupdatedby] [nvarchar](255) NULL,
[CatValLock] [int] NULL,
[CatValLockDate] [smalldatetime] NULL,
[CatValLockBy] [nvarchar](50) NULL,
[CatValLockHistory] [nvarchar](255) NULL,
[oneTimeIndicator] [bit] NOT NULL,
[createdBy] [varchar](50) NULL,
[CreatedOn] [smalldatetime] NULL,
[LastActivityHistory] [varchar](2) NULL,
[Inactive] [bit] NULL,
[vendorCurrencyCode] [varchar](5) NULL,
[MappingOverride] [bit] NULL,
[statementexclude] [bit] NULL,
[OneTimeFlag] [varchar](50) NULL,
[newCategoryCode] [varchar](10) NULL,
[EmailValidated] [bit] NULL,

Working with Common Expressions

Expression Operator	Description
	Logical OR operation
&&	Logical AND operation
==	Comparison of two expressions to determine if equivalent
!=	Comparison of two expression to determine inequality
?:	Conditional operator
POWER()	Raise numeric to a power
LOWER()	Convert to a lowercase
GETDATE()	Returns current date
DATEPART()	Parses date part from a date
ISNULL	Tests an expression for NULL

The Expression column is a blank slate upon which you write the code that controls value. You can hard code a value, such as a space, by enclosing a space inside double-quotes: " " Because SSIS is so strict about data typing, if you evaluate or return a Null in your expression, you have to pull a Null from the Null functions list – for example if assessing if a Date value is null, you would need to use the Null function NULL (DT_DATE). Similarly Type Casts convert data to a specific type, so for example if you wanted to put a string source type value into a float column, you would have to convert it using the Type Cast (DT_R8) [Sting Column].

You can conditionally replace the MiddleName value. In this case, we'll replace Null middle names with a space. If not Null, we'll allow the existing value to continue in the data flow pipeline.

[Test Condition] ? [True Operation] : [False Operation]

To replace Null MiddleName values with a space, use the following expression:

IsNull(MiddleName) ? " " : MiddleName

Replace blanks with NULL

TRIM(ColumnNane)=="" ? (DT_STR, 4, 1252)NULL(DT_STR, 4, 1252) : ColumnName

Replace NULL with another value

ISNULL(ColumnNane) ? "New Value" : ColumnName

Remove a given character from a string (eg remove dashes from DateofBirth)

REPLACE(DateofBirth, "-", " ")

If the column is blank or null, it will be set to unknown. To make a logical AND condition, use && instead of the || operator.

ISNULL(ColumnNane) || TRIM(ColumnNane) == "" ? "Unknown " : ColumnName

Return the first three characters from a postcode

SUBSTRING(Postcode, 1,3)

Add A Column

Add the following expression:

FirstName + (IsNull(MiddleName) ? " " : MiddleName) + LastName

The <<length>> placeholder means the summed lengths of the columns and the <<code_page>> placeholder with 1252 for en-us. The FullName expression should now read:
 (DT_STR, 150, 1252)(FirstName + (IsNull(MiddleName) ? " " : MiddleName) + LastName)
 Note the Data Type column is now a String:

It is not possible to convert the data type of a column when you are merely replacing the value. You can, however, accomplish the same goal by creating a new column in the Data Flow.

Other options: You can also use a Data Conversion Transformation to change the data type of a column in the data flow pipeline, but the Data Conversion also adds a column to the data flow pipeline. You can also cast the value in T-SQL from the source (in the OLE DB Source Adapter).

It is highly recommend, performing an explicit cast in any Derived Column expression that uses a string variable, particularly if the value is subject to change.

Example: To write an expression in derived column with the following conditions :

If(ColA ISNULL then ColB)

If(ColB ISNULL then ColC)

If(ColC ISNULL then "Other")

`ISNULL(ColA)?(ISNULL(ColB)?(ISNULL(ColC) ? "Other" : ColC):ColB):ColA` OR
`IsNull(ColA)? (IsNull(ColB)? ColC: ColB) : ColA`

Boolean Condition (If) ? True Expression Result (Then) : False Expression Result (Else)

Simple If...Then Examples, With an Equal Condition:

`Field == "TrueCondition" ? "True" : "False"`

With a Type Cast:

`(DT_STR,50,1252)(Field == "TrueCondition" ? "True" : "False")`

With an OR Condition:

`Field == "TrueCondition1" || Field == "TrueCondition2" ? "True" : "False"`

With an AND Condition:

`Field1 == "TrueCondition1" && Field2 == "TrueCondition2" ? "True" : "False"`

Checking for a Null:

`ISNULL([Amount]) ? 0 : [Amount]`

Nested with an Equal Condition:

`Field == "TrueCondition1" ? "True" : (Field == "TrueCondition2" ? "True" : (Field == "TrueCondition3" ? "True" : (Field == "TrueCondition4" ? "True" : "False")))`

Nested with a Type Cast:

`(DT_STR,3,1252)Field == "TrueCondition1" ? "True" : (Field == "TrueCondition2" ? "True" : (Field == "TrueCondition3" ? "True" : "False"))`

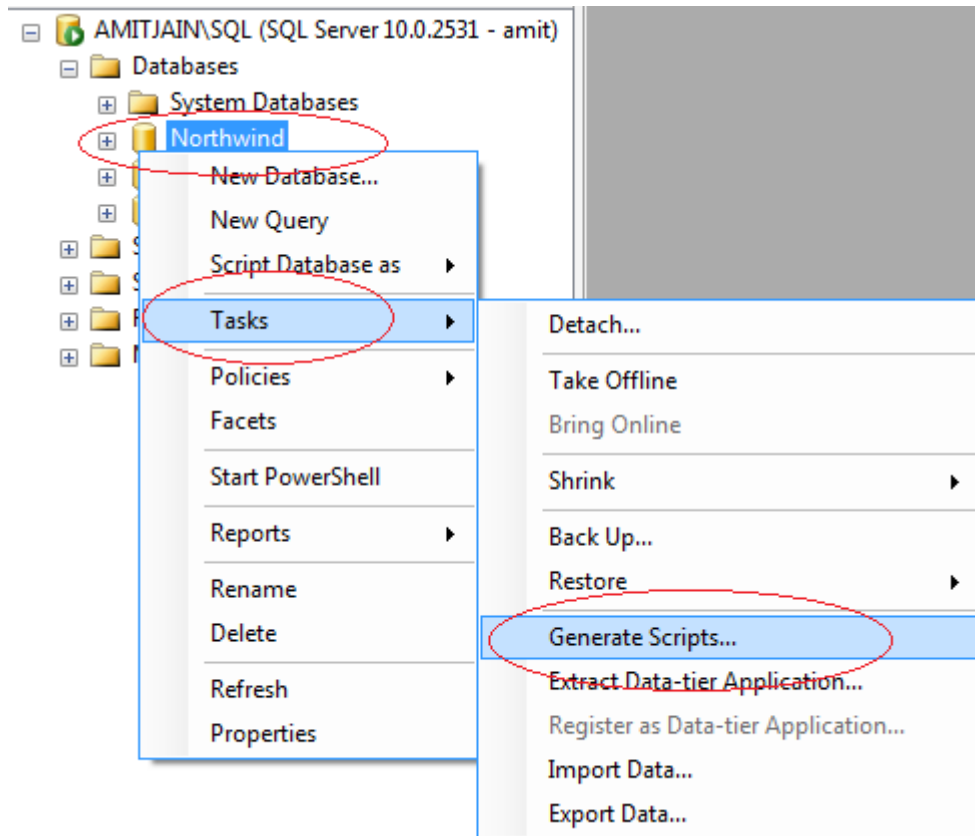
Nested Check for a Null in Numerator and Denominator:

`ISNULL([Income]) ? [Expense] * -1 : ISNULL([Expense]) ? [Income] : [Income] – [Expense])`

Database Copy

If you want to copy all objects, indexes, triggers, constraints etc then do it as mentioned below.

Right click on DB name > Tasks > Generate Scripts





Select the database objects to script.

- ☒ Script entire database and all database objects
- ☐ Select specific database objects

☐ Tables

☐ Views

☐ Stored Procedures

Specify how scripts should be saved or published.

Output Type

☒ Save scripts to a specific location

☐ Publish to Web service

☒ Save to file

Files to generate:

☒ Single file

☐ Single file per object

File name: ...

☒ Overwrite existing file

Advanced

Click on Advance button and select “Schema And Data” From Type of Data to Script section

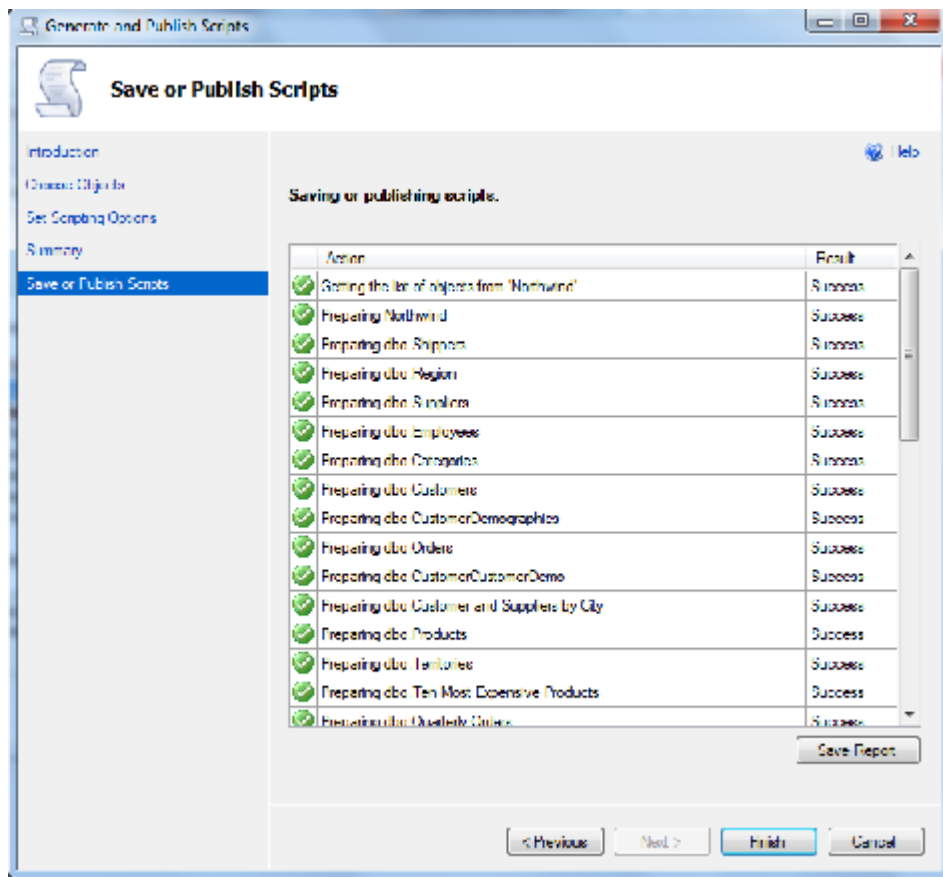
Advanced Scripting Options

Options

Script Logins	False
Script Object-Level Permissions	False
Script Statistics	Do not script statistics
Script USE DATABASE	True
Types of data to script	Schema and data
<input checked="" type="checkbox"/> Table/View Options	Data only
Script Change Tracking	Schema and data
Script Check Constraints	Schema only
Script Data Compression Options	False
Script Foreign Keys	True
Script Full-Text Indexes	False
Script Indexes	True
Script Primary Keys	True
Script Triggers	False

Types of data to script
Generates script that contains schema only or schema and data.

OK Cancel



Quick Reference of SQL Commands

```
SELECT * INTO New_Table_Name FROM
```

```
SELECT * INTO New_Table_Name FROM WHERE
```

```
UPDATE BSAK_ALL  
SET DMBTR = DMBTR *-1, WRBTR = WRBTR *-1  
WHERE SHKZG = 'S'
```

```
SELECT Vendor_Code FROM Vendors  
WHERE Char_Length(Ltrim(Rtrim(Vendor_code))) = 4
```

```
CASE WHEN (a.WMST = 0) THEN (a.[WRBTR] – b.FWSTE) WHEN a.WMST IS NULL THEN  
a.[WRBTR] – b.[WRBTR] ELSE a.[WRBTR] – a.WMWST END InvoiceNetAmountDoc
```

```
CASE WHEN (SHKZG = 'S') THEN FWBAS WHEN (SHKZG = 'H') THEN FWBAS *-1 END FWBAS
```

The only areas that you can use SUM or MAX are the following Invoice Comment, Invoice Gross Amount / Net Amount, DOC/LOC, Purchase Order, Invoice Tax Code. Take out the line number BUZEI in grouping / MAX / SUM

```
CREATE VIEW View_Name AS
```

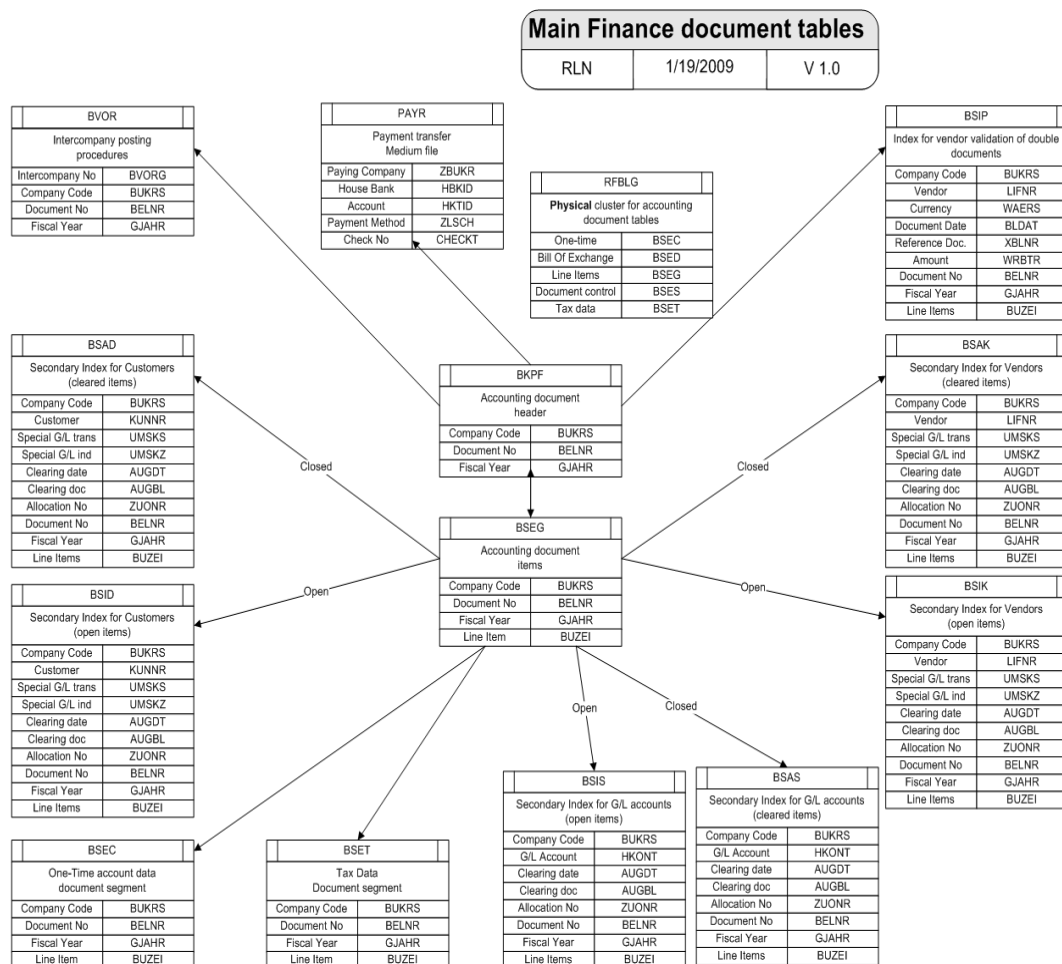
```
SELECT InvoiceRef, ISNULL(UnitsOnOrder,0.00) InvoiceNetAmount,  
SELECT ProductName,UnitPrice*(UnitsInStock+ISNULL(UnitsOnOrder,0))  
If "UnitsOnOrder" is NULL , ISNULL() returns a zero if the value is NULL:
```

Why your Nvarchar field got stored with scientific notation?

When you convert real/float values to Varchar/Nvarchar, it will convert as the regular decimal number when the digits are less than 7 otherwise it will use the scientific notation. So, here we are first parsing the string value into real and then parsing into money (money never uses the scientific notation). From money we then convert into Varchar.

```
SELECT REPLACE(CAST(CAST(CAST(SupplierInvoiceNumber as real) as money) as nvarchar(255)),'.00','')
```

SAP Tables



To view the SAP tables and SAP fields you should check out the following website.

www.dendrites.com/sapfields/sapfields_search_form.asp

Website testing (<http://Servername/APTraxNameof Site>)

Session filters

The "session filters" is an area of the tool in which the user can select how much data to include in the reports around the site.

	Instructions	Expected result
STEP 1	Log into the site	Homepage appears
STEP 2	Check the site summary in the bottom left hand corner	Should see Volume of spend, number of vendors, number of companies and the first and last transaction date
STEP 3	Click on the "APTrax" icon	APTrax homepage appears with reports
STEP 5	Click on the Session Filters icon (globe in the top right hand corner)	The Session Filters page is displayed
STEP 6	Ensure all dates are correctly populated	Dates should relate to time period we have for data
STEP 7	Ensure all companies are populated	Internal companies should all be populated based on the data we get
STEP 8	Un-tick some dates and companies then click update	Should take you back to the home page
STEP 9	Click back on session filters	Dates and companies that you chose should still be un-ticked

Vendor

The vendor section of APTrax is designed to store vendor information as well as vendor reports

	Instruction	Expected result
STEP 1	Click on Vendor at the top left of the tool	Vendor detail page comes up with Vendor Name, Address 1, and Address 2 search boxes as well as the filter and clear box
STEP 2	Type in a vendor in the Vendor Name box e.g. IBM followed by the wildcard *and press return	All instances of the vendor should populate onto the screen
STEP 3	Check that headings: Options, Company Ref, Vendor ID, Vendor Name, Address 1, Address 2, Post/ZIP code, Country, and Spend Under review are populated	You should see data under each of those headings. Please note not every vendor will have spend under review populated
STEP 4	Choose a vendor with Spend Under Review and click select under the options heading	The vendor details should populate for that vendor
STEP 5	Click on Transactions on the headings across the top of the page	All transactions for that vendor should now populate on the screen. Ensure that under the headings of Company, Invoice Number, Vendor Invoice Ref, Vendor ID, Original Vendor name, Invoice Gross, Invoice Gross (doc) Invoice Net (Doc), Invoice Tax (doc) Invoice currency, Invoice Type Ref, Invoice Date, Payment Ref, Payment Date and ABS Invoice Gross Amount Invoice Comment, Purchase Order
STEP 6	Where there are arrows on the headings across the page click twice to make sure the sort functionality works correctly	Alpha characters should filter in Alphabetical order and Numerical numbers should filter by high and low numbers
STEP 7	Currency check	Ensure that the correct currency is populated into the tool
STEP 8	Invoice comment and PO	Check to see whether we should we get Invoice comment Description or PO number
STEP 9	Click on payment Ref Hyperlinks	Remittance details show now show on screen

STEP 10	Click on reports under resources on the left hand side	A list of tasks should now populate
STEP 11	Click on Inactive Vendors	Report should populate
STEP 12	Click on Red view details	All invoice details for that vendor should now show
STEP 13	Click on Invoice Statistics by Vendor	Supplier search box should appear along with the view report button
STEP 14	Type name of vendor into supplier search box e.g. IBM followed by return	Select supplier from supplier drop down box and click view report
STEP 15	Review Graph	Check Graphs and Tables to ensure data is populated
STEP 16	Click on KPI Dashboard by Vendor	Supplier search box should appear along with the view report button
STEP 17	Type name of vendor into supplier search box e.g. IBM followed by return	Select supplier from supplier drop down box and click view report
STEP 18	Review Graphs	Ensure that dates correspond to what has been selected in the global parameters. Click on parameters to change dates to ensure they are filtering correctly. Ensure graphs are populated.
STEP 19	Click on supplier address with spend	Ensure all columns are populated with data. If there is missing data ensure it is because we do not get it in the data provided to us
STEP 20	Click on export button	Ensure report exports to whichever format you selected

AP Stats

The AP Stats reports provide key statistics for the invoice processing and settlement functions. Use the reports for regular monitoring, measure actual performance against targets and identify performance gaps.

	Instruction	Expected result
STEP 1	Click on 'AP Manager'	Resources and Reports are populated on the left hand side of the screen
STEP 2	Click on 'AP Stats'	9 Reports are displayed. These are: Company Summary, Invoice Statistics, Invoice Statistics Vendor, KPI Dashboard, KPI Dashboard Vendor, Ledger Investigation, Purchase Order Spend, Vendor Summary, Transaction Increases
NOTE	Where there are arrows on the headings across the page click twice to make sure the sort functionality works correctly	Alpha characters are filtered in Alphabetical order and Numerical numbers are filtered by high and low numbers
STEP 3	Click on 'Company Summary'	The Internal companies and their relevant spend, invoices, transaction count, average spend and average lines per invoice(optional) are populated
STEP 4	Click on 'Invoice Statistics'	Ensure all the spend, invoice, transaction values and relevant graphs are populated according to the criteria. When the 'Show vendors' link is clicked, it should show all the vendors for a particular criteria
STEP 5	Click on 'Invoice Statistics Vendor'	Supplier search box is displayed
STEP 5.1	Type name of vendor into supplier search e.g. IBM followed by return	The Supplier drop down shows all the relevant vendors according to the search criteria
STEP 5.2	Select supplier from supplier drop down box and click view report.	The report is displayed with all columns and graph populated for that particular vendor
STEP 6	Click on KPI dashboard	The report is displayed with the all the graphs populated

STEP 7	Click on KPI dashboard by vendor and type name of vendor into supplier search e.g. IBM followed by return	The Supplier drop down should show all the relevant vendors according to the search criteria
STEP 7.1	Select supplier from supplier drop down box and click 'view report'	The report is displayed with all the graphs populated
STEP 8	Click on 'Ledger Investigator'	Supplier search box is displayed
STEP 8.1	Type name of vendor into supplier search e.g. IBM followed by return	The Supplier drop down shows all the relevant vendors according to the search criteria
STEP 8.2	Select supplier from supplier drop down box and click 'view report'	The report is displayed with all columns populated
STEP 8.3	Click on 'Payment Ref'	The report is displayed with all columns populated
STEP 9	Click on 'Purchase order spend'	If there are PO details in the data, then the report is displayed with all columns populated
STEP 9.1	Click on 'View detail'	All the relevant columns for that particular vendor are populated
STEP 10	Click on 'Vendor Summary'	All the columns should be populated. If there are PO details in the data, then the % of Value on PO and PO spend values are populated
STEP 11	Click on 'Transaction Increases'	The report is displayed with all columns populated
STEP 11.1	Click on 'View detail' for a vendor which has Spend	The report is displayed with all columns and graph populated

Cash Management

The Cash Management reports support the internal control environment. The reports provide details of exceptions and statistical anomalies contained in the transaction data. Use the reports to enhance and supplement operational internal controls.

	Instruction	Expected result
STEP 1	Click on 'AP Manager'	Resources and Reports are populated on the left hand side of the screen
STEP 2	Click on 'Cash Management'	8 Reports are displayed. These are: Grief days, Grief Credits, Invoice to Pay days, Invoice to Process days, Process to pay days, Invoice to process days by vendor, Invoice to pay days by vendor, Process to pay days by vendor
NOTE	Where there are arrows on the headings across the page click twice to make sure the sort functionality works correctly	Alpha characters are filtered in Alphabetical order and Numerical numbers are filtered by high and low numbers
STEP 3	Click on 'Grief days'	The overall average invoice to pay days should have a value. All the columns are populated
STEP 4	Click on 'Grief credits'	The overall average ratio of credits to invoices should have a value. All the columns are populated
STEP 5	Click on 'Invoice to Pay days'	The report is displayed with all columns and graph populated
STEP 5.1	Click on 'Show vendors'	The report is displayed with all columns populated
STEP 5.2	Click on 'Show detail'	The report is displayed with all columns populated
STEP 6	Click on 'Invoice to Process days'	The report is displayed with all columns and graph populated
STEP 6.1	Click on 'Show vendors'	The report is displayed with all columns populated
STEP 6.2	Click on 'Show detail'	The report is displayed with all columns populated

STEP 7	Click on 'Process to Pay days'	The report is displayed with all columns and graph populated
STEP 7.1	Click on 'Show vendors'	The report is displayed with all columns populated
STEP 7.2	Click on 'Show detail'	The report is displayed with all columns populated
STEP 8	Click on 'Invoice to Pay days by vendor'	Supplier search box is displayed
STEP 8.1	Type name of vendor into supplier search e.g. IBM followed by return	The Supplier drop down shows all the relevant vendors according to the search criteria
STEP 8.2	Select supplier from supplier drop down box and click 'view report'	The report is displayed with all columns and graph populated
STEP 9	Click on 'Invoice to Process days by vendor'	Supplier search box is displayed
STEP 9.1	Type name of vendor into supplier search e.g. IBM followed by return	The Supplier drop down shows all the relevant vendors according to the search criteria
STEP 9.2	Select supplier from supplier drop down box and click 'view report'	The report is displayed with all columns and graph populated
STEP 10	Click on 'Process to pay days by vendor'	Supplier search box is displayed
STEP 10.1	Type name of vendor into supplier search e.g IBM followed by return	The Supplier drop down shows all the relevant vendors according to the search criteria
STEP 10.2	Select supplier from supplier drop down box and click 'view report'	The report is displayed with all columns and graph populated

Compliance

The Compliance reports identify unusual transactions that may be indicative of Control Issues. Use these reports to support internal control, identify potential fraudulent activity and support compliance.

	Instruction	Expected result
STEP 1	Click on 'AP Manager'	Resources and Reports are populated on the left hand side of the screen
STEP 2	Click on 'Compliance'	8 Reports are displayed. These are: Benford Law - Invoice Amount, Benford Law - Invoice Number, Round Value Payments, Authority Limits, Small Invoices, Invoice Spikes, More than average credits by transactions, Single Payments
NOTE	Where there are arrows on the headings across the page click twice to make sure the sort functionality works correctly	Alpha characters are filtered in Alphabetical order and Numerical numbers are filtered by high and low numbers
STEP 3	Click on 'Benford Law - Invoice Amount'	The report is displayed with the graph populated
STEP 4	Click on 'Benford Law - Invoice Reference'	The report is displayed with the graph populated
STEP 5	Click on 'Round value Payments'	The report is displayed with all columns populated
STEP 5.1	Click on 'Show Vendors'	The report is displayed with all columns populated
STEP 5.2	Click on 'Show Detail' where applicable	All the invoice fields for that particular vendor are populated
STEP 6	Click on 'Authority Limits'	The OverOrTolerance drop down box, Authority Limit and Tolerance text boxes are displayed with default values. The columns are populated according to the default search criterion
STEP 6.1	Change the 'OverOrTolerance' value to 'Invoice gross value within tolerance of limit'	The report is displayed with all columns populated accordingly

STEP 6.2	Click on 'view detail'	The report is displayed with all columns populated accordingly
STEP 6.3	Change the 'Tolerance' and 'Authority limit' values	The report is displayed with all columns populated accordingly
STEP 7	Click on 'Small Invoices'	The Invoice Value text box is displayed with the default value. All the columns are populated according to the default search criteria
STEP 7.1	Change the 'Invoice Value' field and click 'view report'	The report is displayed with all columns populated accordingly
STEP 7.2	Click on 'View Vendors'	The report is displayed with all columns populated
STEP 7.3	Click on 'view detail'	The report is displayed with all columns populated
STEP 8	Click on 'Invoice Spikes'	The report is displayed with all columns populated
STEP 8.1	Click on 'View Detail'	The report is displayed with all columns and graph populated
STEP 9	Click on 'More than average credit by transactions'	The Average credit/Invoice percentage text box is displayed with the default value. All the columns are populated according to the default search criteria
STEP 9.1	Change the 'Average credit/Invoice percentage' value and click 'view report'	The report is displayed with all columns populated accordingly

STEP 9	Click on 'Single Payments'	The Top vendor's text box is displayed with the default value. All the columns are populated according to the default search criteria
STEP 9	Change the 'Top vendors' value and click 'view report'	The report is displayed with all columns populated accordingly

Tax

The Tax reports identify discrepancies and/or non-standard Sale Tax transactions. Use these reports to support internal control, identify any under-recovery of Sales Tax opportunities and support compliance

	Instruction	Expected result
STEP 1	Click on 'AP Manager'	Resources and Reports are populated on the left hand side of the screen
STEP 2	Click on 'Tax'	3 Reports are displayed. These are QVAT Main potentials, QVAT vendor summary, Invalid UK VAT numbers
NOTE	Where there are arrows on the headings across the page click twice to make sure the sort functionality works correctly	Alpha characters are filtered in Alphabetical order and Numerical numbers are filtered by high and low numbers
STEP 3	Click on 'QVAT main potentials'	The scores drop down box is displayed with the default value. The fields are populated according to the default search criteria
STEP 3.1	Using the drop down, change the 'Scores' values and click 'view report'	The report is displayed with all columns populated accordingly
STEP 3.2	Click the 'score' link	The report is displayed with all columns populated and test description for the scores are displayed (Default score is 20)
STEP 4	Click on 'QVAT vendor summary'	The report is displayed with all columns populated
STEP 4.1	Click on 'Vendor ID'	The report is displayed with all the scored invoices for that particular vendor
STEP 4.1	Click on 'Score'	The report is displayed with all columns populated and test description for the scores are displayed (Default score is 20)
STEP 4.1	Click on 'Invalid UK VAT numbers'	The report is displayed with all columns populated

Master Data

The Master Data reports detail anomalies, variances and data quality issues (such as missing or incorrect data) within the supplier master file. Use these reports to monitor key data supporting 'procure to pay' operations, identify control and process issues and ensure master data is appropriately maintained.

	Instruction	Expected result
STEP 1	Click on 'AP Manager'	Resources and Reports are populated on the left hand side of the screen
STEP 2	Click on 'Master data'	6 Reports are displayed. These are Duplicate vendors, Duplicate vendors (Post/Zip Code), Inactive vendors, Missing addresses, One time vendor accounts, Vendors with PO box addresses
NOTE	Where there are arrows on the headings across the page click twice to make sure the sort functionality works correctly	Alpha characters are filtered in Alphabetical order and Numerical numbers are filtered by high and low numbers
STEP 3	Click on 'Duplicate vendors'	The report is displayed with all columns populated
STEP 4	Click on 'Duplicate vendors (Post/Zip code)'	The report is displayed with all columns populated
STEP 5	Click on 'Inactive vendors'	The Inactive days text field is displayed with the default value. All the columns are populated according to the default search criteria
STEP 5.1	Change the 'Inactive days' value	The report is displayed with all columns populated accordingly
STEP 5.2	Click on 'View detail'	The report is displayed with all columns populated
STEP 6	Click on 'Missing Addresses'	The report is displayed with all columns populated
STEP 7	Click on 'One time vendor Accounts'	All the columns are populated or the message 'No Vendors with names like "One time" ' displayed

STEP 8

Click on 'Vendors with PO box addresses'

The report is displayed with all columns populated

Duplicates

We utilise advanced matching techniques to identify possible duplicate payments. Each report is based upon specific matching criteria; some of which will be more relevant than others to your organisation. Each report provides a list of possible duplicates and any identified contra (adjusting) entries. Use the reports to identify spend recovery opportunities and highlight internal control issues.

	Instruction	Expected result
STEP 1	Click on Duplicates under resources	Reports 1-9 should appear under reports
STEP 2	Click on report 1	All data fields should be populated. Credit transactions should show in red.
NOTE	Review Dupes	Check that there are no dupes groups containing one single line transaction. If there is, query it
STEP 3	Click the export button	Select a format to export to from the drop down list e.g. Excel
STEP 4	Check download	Dupe reports should now be in Excel format
STEP 5	Repeat above for all 9 reports	As above

Identify Dupes

PRGX APTrax includes a workflow for finding and managing potential duplicate payments. This workflow enables users to track the progress of a claim. This test evidences to users that a potential duplicate payment can be investigated and flagged as a "claim" for the next stage in the workflow.

	Instruction	Expected result
STEP 1	Click "Workflow" Select the report "Identify Dupes"	There should be a number of search, filter and drop down boxes
STEP 2	Where the "select dupe run" dropdown box is select report 1 Payment to wrong vendor	All fields should now be populated with correct data
STEP 3	Dupe Groups	Ensure Dupe groups are separated by shading
STEP 4	Credits	Ensure credits are coloured in red
STEP 5	Investigate	Select a dupe group to investigate by clicking on the Investigate circle at the left of the dupes and click save located in the bottom right hand corner
STEP 6	Drop	Select a dupe group to drop by clicking on the drop circle at the left of the dupes- ensure it then defaults to the comment field to the right of the dupe. There should be only one comment field per dupe group. Write a reason why we are dropping and then click on save in the bottom right hand corner
STEP 7	Click on show dropped box	Ensure that the item you dropped is now in the show dropped items page along with the comment you wrote. Check formatting of columns for any errors
STEP 8	Reinstate item	Click on reinstate and the item should now have moved from this screen
STEP 9	Un-tick show dropped items	Item that you reinstated should now be found on the screen

Investigate Dupes

PRGX APTrax includes a workflow for finding and managing potential duplicate payments. This workflow enables users to track the progress of a claim. This test evidences to users that a potential duplicate payment (or "Claim") can be moved along in the workflow.

	Instruction	Expected result
STEP 1	"Work Flow" Click on Investigate Dupes under the task list	All items that have been selected to be investigated should now appear
STEP 2	All fields should be correctly populated	You should see the relevant data fields populated under the headings across the top of the page and the user name and status of each dupe should now be showing on the far right of the screen
STEP 3	Click on the item ref of the dupe	The dupe group you selected should now populate. Ensure there is a column name of "dup Transaction" which has a tick box for each of the legs of the dupe. Ensure that credits if there are any show up in red text.
STEP 4	Click on Add File	Click on Add file and upload a document. Once complete click on the Files tab beneath the dupe screen to ensure file you added is populated there and can be downloaded
STEP 5	User Notes	Type in some notes and click on add notes ensuring that the notes you added are populated below
STEP 6	Communication	Ensure you can add communication files by clicking on the browse button and uploading a document
STEP 7	Files	Click on the files tab and ensure that any attachments that were added are showing here

STEP 8	Amendments	Click on the Amendments tab and ensure that the changes to the status of the dupe are populated along with the username of the person who changed it as well as the time and date it got changed
Step 9	Dup Transaction tick box	Click on one of the dup transaction tick boxes. This identifies which payment is the error payment
Step 10	Select Reason	You must then click on the select reason dropdown box and choose one of the reason codes
Step 11	Ready to send	Click the ready to send button on the right hand side of the screen. The screen should then take you back to the original Investigate dupes page and the item you looked at would have moved from this screen.

Statement Management

PRGX APTrax includes a workflow for finding and managing potential duplicate payments. This workflow enables users to track the progress of a claim. This test evidences to users that they are able to add "claims" to the workflow by uploading statements into APTrax and creating manual claim items.

	Instruction	Expected result
STEP 6	Select the Task "Statement Management" on the left hand side of the screen	You should see a vendor box with (select) in it as well as a view statements button
STEP 7	Click on the select field to find a supplier	Supplier pop out box appears
STEP 8	Select a supplier from the pop up box by typing in the supplier name followed by the wildcard option *	Select the supplier of your choice by clicking on the green supplier id. The pop up box disappears and the supplier name appears in the supplier box
STEP 9	Click "View Statements" and then upload a "Statement" by using the "Browse" button and click "Upload"	Statement will be shown below the "View Statements" text
STEP 10	Complete all the fields in the "Claim Item" section and click "Create Claim Item"	Fields are reset and the claim is shown at the bottom of the page
STEP 11	Hover over the "Recovery Auditing" module and select "Claims Management" from the "Workflow" submenu	Claims Management page is displayed
STEP 12	Click on the name of the supplier you added a claim to.	User is taken to the "Ready to Send" section and your claim you just made will now be found here.

Claims Management

	Instruction	Expected result
Step 1	Click on "Claims Management" on the left hand side of the screen	Claim Summary page appears
Step 2	Click on a green vendor name under the vendor name heading	The claim item/s for that vendor selected at Investigate dupes and Statement Management stage should appear
Step 3	Click on the Item Ref	A pop out screen should appear showing the item and its relating dupe leg and any notes etc added to the item. You should also now be at the "Ready to send" tab located along the top of the screen.
Step 4	Close the Pop out box and click on the send to vendor tick box. Select which template you wish to use using the drop down box in the bottom right hand corner and then click on send	A popup box should come up asking "Are you sure you want to send these items to vendor". Click ok. That item should now have disappeared from the screen and moved to the next step of the workflow which is send to vendor
Step 5	Click on the send to vendor tab at the top of the screen	You should see the item that you was just looking at in the previous screen.
Step 6	In the dropdown box located on the right hand side select vendor confirmed and click update	The item you will now have moved from this screen onto the next stage of the workflow process which is the Vendor Confirmed tab
Step 7	Click on the vendor confirmed tab and select one of the options from the dropdown box. Type a comment into the comment field and click ready to bill	The Item will now have moved on to the next stage of the process which is Ready to Bill
Step 8	Click on the ready to bill tab and select the "Bill" box located on the right hand side. Then click on Bill	A popup box asking "Are you sure you wish to send the items to CMS" should appear. Select OK. An excel spreadsheet should now appear and the item has now moved into the Billed tab

Workflow Reports

	Instruction	Expected result
Step 1	Click on WorkFlow Reports under the resources section	All the reports in this section should now populate underneath
Step 2	Click on the first report which is Investigation Dash	The report should populate showing the user id and the progress of the recoveries. Two pie charts showing Investigations and Recoveries should be present
Step 3	Click on Recovery Pipeline report	This report should show the different Statuses within APTrax and the number of items as well as the value
Step 4	Click on one of the item numbers	You should now see all transactions at that stage of the process
Step 5	Click on the item reference for one of the transactions	You should now see the history of that transaction showing what it related to, any notes added as well as attachments
Step 6	Click on one of the numbers under the "Item" column	All dupes currently at that status for that particular user will populate on the screen
Step 7	Click on the Recovery Pipeline by company report	Ensure that you can see the different internal companies across the top (Companies will show in the global parameters)
Step 8	Click on a number under the item column	Ensure it drills through to the items at that stage of the process as well as drill down to the detail as shown in the previous report
Step 9	Click on Recovery Pipeline By User report	Ensure the User ID's are populated and you can drill down into the items by clicking on the reference numbers as shown in the reports above
Step 10	Click on the Workflow By Reason Code report	Check that the reason codes are populated for the transactions that have been given a reason code at the "Investigate Dupes" stage of the process.

Step 11	Click on a number under the item column	Ensure that drill downs work as in the previous reports. Check that the values reflect the stages in the workflow
Step 12	Click on Workflow By Vendor report	Ensure the values reflect what is shown in the claims management screen and ensure that the drill downs work as they should
Step 13	Click on Items to Investigate report	This report should show all the items that are at the Investigate stage of the process. You should be able to export this report