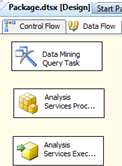
Outline:

* SSIS Control Flow Elements for Data Mining
* SSIS Data Flow Elements for Data Mining

**SSIS Control Flow Elements for Data Mining**

|  |  |
| --- | --- |
|  | * [Analysis Services Execute DDL Task](http://msdn.microsoft.com/en-us/library/ms139988.aspx) * [Analysis Services Processing Task](http://msdn.microsoft.com/en-us/library/ms141779.aspx) * [Data Mining Query Task](http://msdn.microsoft.com/en-us/library/ms141728.aspx) |

These three elements are all used in the authors’ project called ***DMQueryTaskDemo***:



In this case, the project does three distinct things. The **Data Mining Query Task** executes the following Query:

SELECT FLATTENED

t.[CustomerKey],

[TM\_Decision\_Tree].[Bike Buyer],

(PredictProbability([TM\_Decision\_Tree].[Bike Buyer])) as [Probability]

From

[TM\_Decision\_Tree]

PREDICTION JOIN

OPENQUERY([Adventure Works DW2008],

'SELECT

[CustomerKey],

[MaritalStatus],

[Gender],

[YearlyIncome],

[TotalChildren],

[NumberChildrenAtHome],

[EnglishEducation],

[EnglishOccupation],

[HouseOwnerFlag],

[NumberCarsOwned],

[DateFirstPurchase],

[CommuteDistance],

[Region],

[Age],

[BikeBuyer]

FROM

[dbo].[vTargetMail]

') AS t

ON

[TM\_Decision\_Tree].[Marital Status] = t.[MaritalStatus] AND

[TM\_Decision\_Tree].[Gender] = t.[Gender] AND

[TM\_Decision\_Tree].[Yearly Income] = t.[YearlyIncome] AND

[TM\_Decision\_Tree].[Total Children] = t.[TotalChildren] AND

[TM\_Decision\_Tree].[Number Children At Home] = t.[NumberChildrenAtHome] AND

[TM\_Decision\_Tree].[English Education] = t.[EnglishEducation] AND

[TM\_Decision\_Tree].[English Occupation] = t.[EnglishOccupation] AND

[TM\_Decision\_Tree].[House Owner Flag] = t.[HouseOwnerFlag] AND

[TM\_Decision\_Tree].[Number Cars Owned] = t.[NumberCarsOwned] AND

[TM\_Decision\_Tree].[Date First Purchase] = t.[DateFirstPurchase] AND

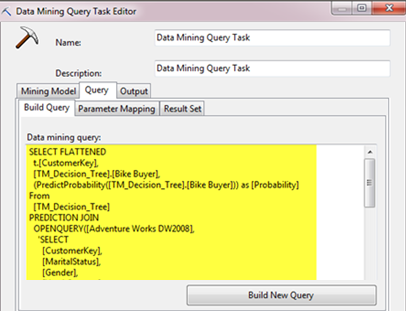
[TM\_Decision\_Tree].[Commute Distance] = t.[CommuteDistance] AND

[TM\_Decision\_Tree].[Region] = t.[Region] AND

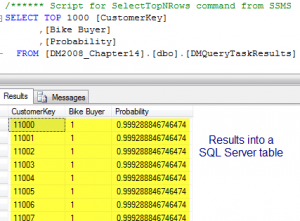
[TM\_Decision\_Tree].[Age] = t.[Age] AND

[TM\_Decision\_Tree].[Bike Buyer] = t.[BikeBuyer]

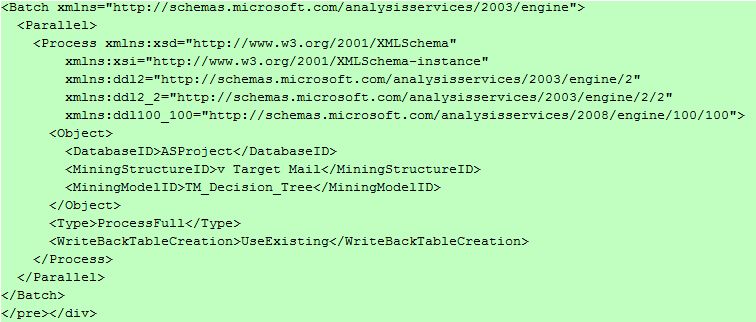
The DMX Query can be typed into Integration Services:



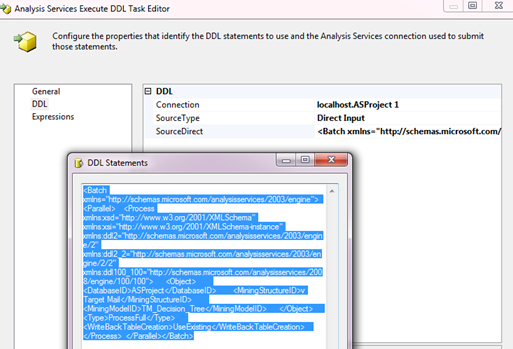
The results go into a SQL Server table.

[](http://www.marktab.net/datamining/2010/08/28/data-mining-with-sql-server-integration-services-ssis/dm2008_1408/)

The Analysis Services Processing Task processes the *AS\_Project* Analysis Services database. And the last element in this project, the **Analysis Services Execute Task** executes the following **DDL code**:

[](http://www.marktab.net/datamining/2010/08/28/data-mining-with-sql-server-integration-services-ssis/dm2008_1417/)

The DDL can be typed into Integration Services:

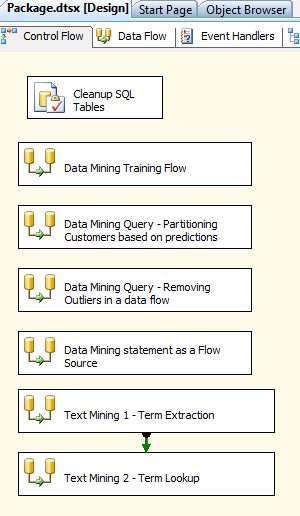


**SSIS Data Flow Elements for Data Mining**

The following graphic shows the data flow elements for data mining. Data flow allows individual record-by-record manipulation logic. Yes, manipulation can be logical.

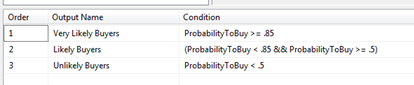
|  |  |
| --- | --- |
|  | * [Data Mining Model Training Destination](http://msdn.microsoft.com/en-us/library/ms139761.aspx) * [Data Mining Query Transformation](http://msdn.microsoft.com/en-us/library/ms140358.aspx) * [Term Extraction Transformation](http://msdn.microsoft.com/en-us/library/ms141809.aspx) * [Term Lookup Transformation](http://msdn.microsoft.com/en-us/library/ms137850.aspx) |

These three elements are all used in the authors’ project called ***Data Flow Samples***:

[](http://www.marktab.net/datamining/2010/08/28/data-mining-with-sql-server-integration-services-ssis/dm2008_1410/)

In the project, the Data Mining Model Training Destination provides a way to build models.  Through percentage random sampling, the Destination provides a way to create a data mining model in this project.  Not all models have to be created from a single database source, and other sources could have been used.  Also, not all models need to apply random sampling, but this model did.  In this case, I would have preferred that the nonselected observations (named *Sampling Unselected Output*) would have gone to a destination where they could be used in subsequent model testing.

|  |  |
| --- | --- |
| Data Flow | The Data Mining Query Transformation allows for observation-level processing. |
|  |  |

The conditional split transformation contains logic to put the results into one of three possible destinations (all are different SQL Server tables) based on the probabilities calculated by the data mining model.  The following screenshot shows how the conditional split is programmed:

The last two transformations are the term extraction and term lookup.  Together they comprise what Microsoft markets (and I agree) is text mining, and sorry it only comes in English.  The Term Extraction Transformation can take a text field and divide it into *terms*, defined as phrases and individual words.

*You can use the Term Extraction transformation to discover the content of a data set. For example, text that contains e-mail messages may provide useful feedback about products, so that you could use the Term Extraction transformation to extract the topics of discussion in the messages, as a way of analyzing the feedback.*

*The Term Extraction transformation can extract nouns only, noun phrases only, or both nouns and noun phases. A noun is a single noun; a noun phrases is at least two words, of which one is a noun and the other is a noun or an adjective. For example, if the transformation uses the nouns-only option, it extracts terms like bicycle and landscape; if the transformation uses the noun phrase option, it extracts terms like new blue bicycle, bicycle helmet, and boxed bicycles.*

*Articles and pronouns are not extracted. For example, the Term Extraction transformation extracts the term bicycle from the text the bicycle, my bicycle, and that bicycle.*

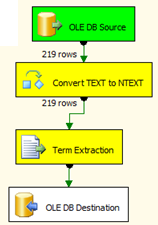
*The Term Extraction transformation normalizes words so that the capitalized and noncapitalized versions of words are not treated as different terms. For example, in the text You see many bicycles in Seattle and Bicycles are blue, bicycles and Bicycles are recognized as the same term and the transformation keeps only bicycle. Proper nouns and words that are not listed in the internal dictionary are not normalized.*

*The Term Extraction transformation also stems nouns to extract only the singular form of a noun. For example, the transformation extracts man from men, mouse from mice, andbicycle from bicycles. The transformation uses its dictionary to stem nouns. Gerunds are treated as nouns if they are in the dictionary.*

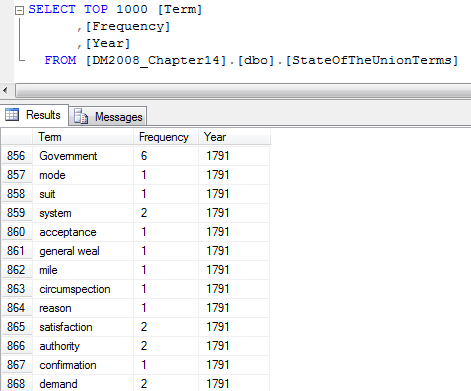
*Retrieved from*[*http://msdn.microsoft.com/en-us/library/ms141809.aspx*](http://msdn.microsoft.com/en-us/library/ms141809.aspx)

I decided to include the above details because some people might want a quick overview of what exactly this text mining capability includes.  The authors’ project focuses on analyzing State of the Union addresses (the annual address that the American President gives to the public from the House of Representatives).  The project uses Term Extraction to detect terms in these speeches (each of which completely fills one particular string field in a database), builds a dictionary with these terms (saved as a SQL Server table), and then determines the frequency of terms by address (putting those frequencies into another SQL Server table).

Manipulators like me will find Integration Services entertaining because you get to see the progress of the data flow while it runs:



The final destination in this project is a SQL Server table:

[](http://www.marktab.net/datamining/2010/08/28/data-mining-with-sql-server-integration-services-ssis/dm2008_1416/)

In the book, the authors provide how they used Association Rules to subsequently model this information. People who can program or develop or script (in this discussion, words I am using interchangably) can extend the abilities of text mining to do more than just make a dictionary (term extraction) or analyze against the dictionary (term lookup).  Those preprogrammed Transformations provide complex functionality.

Here are some extended programming challenges to extend the State of the Union address:

* Add a categorization field to determine when during the speech a term was used.  Use this additional information to help select political party.
* Break the speeches into sentences, and use word order as the basis for Sequence Analysis.  Because you will have individual sentences, you can then analyze by President.
* As an advanced twist to the second idea, use only what Integration Services considers*terms*, and do the same sequence analysis again based on sequences of terms.  How can the terms be used to determine what President spoke?

**Conclusion**

Text mining becomes possible starting with Integration Services’ Text Extraction Transformation and Text Lookup Transformation.