\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**Title: eCoaching\_Quality\_Load**

**SSIS Detail Design Document**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Last Modified | Description |
| 9/18/2017 | TFS 7541 – Support ATA Forms for quality Module in IQS Feed |

Prepared by: Susmitha Palacherla Date: 05/01/2014

Department, Location: Health Solutions Division

Approved by: Date:

**Change History Log**

| **Date** | **Version** | **Change Description** | **Author** |
| --- | --- | --- | --- |
| 05/01/2014 | 1.0 | Initial revision | Susmitha Palacherla |
| 07/01/2014 | 2.0 | Updated per SCR 12963 to fix Loaded Count value in File List table | Susmitha Palacherla |
| 07/20/2014 | 3.0 | Updated per SCR 13054 to import Verint Form Name | Susmitha Palacherla |
| 09/08/2014 | 4.0 | Updated to add a variable for File Prefix to support loads into dev and test using a different file name | Susmitha Palacherla |
| 10/31/2016 | 5.0 | Updated per TFS 3757 Include Yes/No value to coaching monitor question | Susmitha Palacherla |
| 03/01/2-17 | 6.0 | Updated per TFS 5653 – to Create new NPN coaching logs from existing quality submitted scorecard logs | Susmitha Palacherla |
| 8/14/2017 | 7.0 | TFS 7105 – Upgrade to SQL Server 2012 | Susmitha Palacherla |
| 9/18/2017 | 8.0 | TFS 7541 – Incorporate ATA Forms in feed for Quality Module | Susmitha Palacherla |

Contents

[1. Overview 4](#_Toc493688661)

[1.1 Project Description 4](#_Toc493688662)

[1.2 Document Scope 4](#_Toc493688663)

[1.3 Source File 4](#_Toc493688664)

[1.4 Module List 4](#_Toc493688665)

[1.5 Software and Hardware Interfaces 4](#_Toc493688666)

[1.5.1 Software 4](#_Toc493688667)

[1.5.2 Hardware 4](#_Toc493688668)

[1.6 Users 5](#_Toc493688669)

[2. Details 5](#_Toc493688670)

[2.1 Source Files 5](#_Toc493688673)

[2.1.1 eCL\_IQS\_Scorecard\_yyyymmdd.csv.zip.encrypt 5](#_Toc493688674)

[2.2 Module Details 5](#_Toc493688675)

[2.2.1 SQL agent job 5](#_Toc493688676)

[2.2.2 SSIS Package 6](#_Toc493688677)

[2.2.3 Tables 44](#_Toc493688678)

[2.2.4 Procedures 44](#_Toc493688679)

[2.2.5 Functions 45](#_Toc493688680)

# Overview

## Project Description

Quality is an important metric that contributes to a CCO CSR’s overall Performance. Quality monitoring can be performed internally by Vangent/GDIT staff or external evaluators using an internal application like Lime Survey or Verint Quality Monitoring tools. Irrespective of the source system in which the Quality evaluations are created, the records are fed into the eCoaching database with an overall classification of opportunity or Reinforcement. Once entered into the eCoaching database Reinforcements are simply acknowledged by CSRs and their Supervisors while Opportunities trigger a Coaching and a Coaching workflow is initiated.

## Document Scope

This document describes the load of the Quality feed sourced from all of the source systems in use at the time. Quality evaluations from all sources are consolidated into the IQS system from which a daily feed is generated for loading into the Coaching database.

## Source File

A daily file is generated from IQS and staged in an encrypted form on a file share.

Files have a naming convention eCL\_IQS\_Scorecard\_yyyymmdd.csv.zip.encrypt (prod) or Test\_eCL\_IQS\_Scorecard\_yyyymmdd.csv.zip.encrypt (Test)

and staged at [\\vrivscors01\BCC Scorecards\Coaching\Apps\Encryption\Encrypt\_out\](file:///\\vrivscors01\BCC%20Scorecards\Coaching\Apps\Encryption\Encrypt_out\)

## Module List

* SQL agent job
* SSIS Package
* Tables
* Procedures

## Software and Hardware Interfaces

### Software

* SQL Server 2012 SP3 Suite
* Encryption/Decryption utility

### Hardware

* F3420-ECLDBD01 – Dev DB Server
* F3420-ECLDBT01 – Test DB Server
* F3420-ECLDBP01 – Prod Server
* [\\F3420-ECLDBD01\data\](file:///\\F3420-ECLDBD01\data\) - Dev File staging share
* [\\F3420-ECLDBT01\data\](file:///\\F3420-ECLDBT01\data\) - Test File staging share
* [\\VRIVSCORS01\BCC Scorecards\](file:///\\VRIVSCORS01\BCC%20Scorecards\) - Prod File staging share

## Users

* CCO CSRs, Supervisors, Managers, Quality and support staff.

# Details



## Source Files

### eCL\_IQS\_Scorecard\_yyyymmdd.csv.zip.encrypt

* Description: A daily file containing all active evaluations from the previous day is staged on file staging server. This file is loaded into the eCoaching database and depending on whether the evaluation is an opportunity or Reinforcement a Coaching is initiated or simply acknowledged. The source file is in an encrypted form and decrypted during the load process.
* Source system: IQS
* Staging location: [\\vrivscors01\BCC Scorecards\Coaching\Apps\Encryption\Encrypt\_out\](file:///\\vrivscors01\BCC%20Scorecards\Coaching\Apps\Encryption\Encrypt_out\)
* File name eCL\_IQS\_Scorecard\_yyyymmdd.csv.zip.encrypt
* Frequency: Daily
* File arrival time: ~4:00 AM EST
* Destination Table: Coaching\_Log and Coaching\_Log\_Reason

## Module Details

### SQL agent job

* Production Instance : F3420-ECLDBP01
* Production Job: CoachingQualityLoad
* Production Package: [\\F3420-ECLDBP01\ssis\Coaching\Packages\IQS\_Coaching.dtsx](file:///\\VDENSSDBP07\scorecard-ssis\Coaching\IQS_Coaching.dtsx)
* Production Config File: Prod\_IQS\_Coaching.dtsConfig
* Owner: ecljobowner
* Run As: ECLProxy (ECL Credential using application service account VNGT\SVC-SQLECLP01
* Schedule: Daily 8:00 PM EST Daily
* Destination File Location: N/A

### SSIS Package

#### Variables

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Scope** | **DataType** | **Value** |
| Decrypt\_In | IQS\_Coaching | String | \\vrivscors01\BCC Scorecards\Coaching\Apps\Encryption\Decrypt\_in\ |
| Encrypt\_In | IQS\_Coaching | String | \\vrivscors01\BCC Scorecards\Coaching\Apps\Encryption\Encrypt\_in\ |
| Decrypt\_Out | IQS\_Coaching | String | \\vrivscors01\BCC Scorecards\Coaching\Apps\Encryption\Decrypt\_out\ |
| Encrypt\_Out | IQS\_Coaching | String | \\vrivscors01\BCC Scorecards\Coaching\Apps\Encryption\Encrypt\_out\ |
| FileName | IQS\_Coaching | String | @[User::FilePrefix] +  (DT\_STR,4,1252) DatePart("yyyy",getdate())+  Right("0" + (DT\_STR,4,1252) DatePart("m",getdate()),2) +  Right("0" + (DT\_STR,4,1252) DatePart("d",getdate()),2)+".csv" |
| SystemName | IQS\_Coaching | String | Dev |
| LoadedCount | IQS\_Coaching | Int32 | 0 |
| RejectedCount | IQS\_Coaching | Int32 | 0 |
| StagedCount | IQS\_Coaching | Int32 | 0 |
| FilePrefix | IQS\_Coaching | String | Test\_eCL\_IQS\_Scorecard\_(Dev/test) and eCL\_IQS\_Scorecard\_(Prod) |
| AdhocNPN | IQS\_Coaching | Boolean | False |
| BeginDate | IQS\_Coaching | Int32 | 0 |
| EndDate | IQS\_Coaching | Int32 | 0 |
| DayOfWeek | IQS\_Coaching | Int32 | 2 |
| NPNLoad | IQS\_Coaching | String | "SELECT \* FROM EC.fn\_NPNQualityRecs(" + (DT\_WSTR,20) @[User::BeginDate] + "," + (DT\_WSTR,20) @[User::EndDate] + ")" |

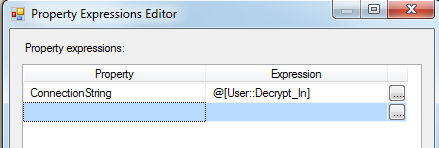
#### Connection Manager Entries

ConnectionStrings

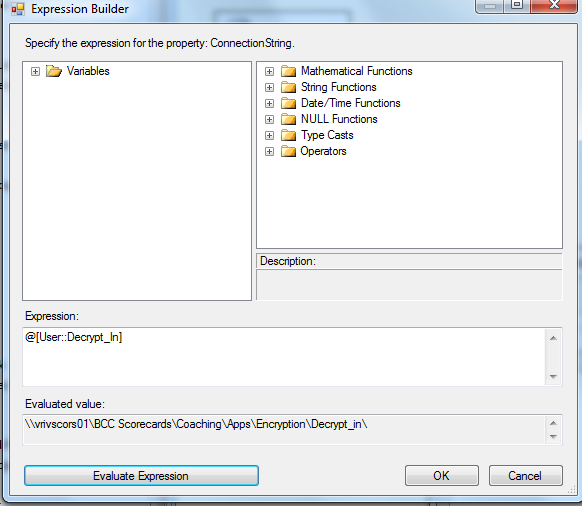
1. Decrypt\_In
2. Encrypt\_out
3. IQS\_File
4. Destinationdb
5. Decrypt\_In

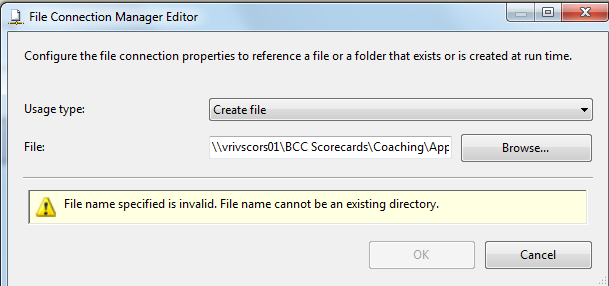
ConnectionString: \\vrivscors01\BCC Scorecards\Coaching\Apps\Encryption\Decrypt\_in\

Expression

: 

Expression Evaluates to:

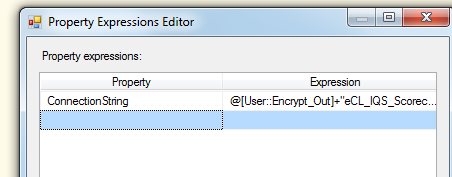


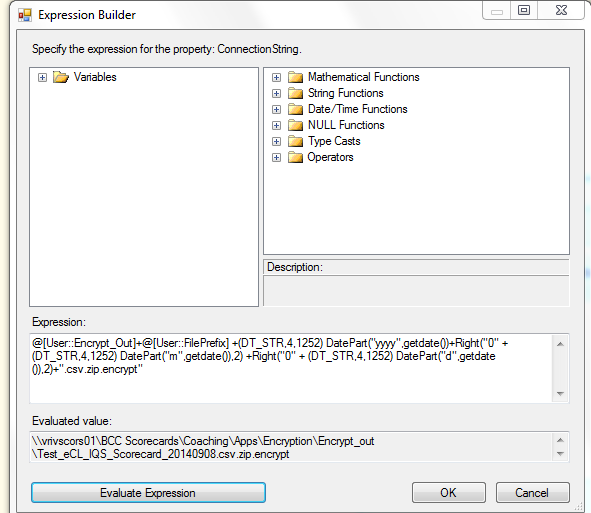


1. Encrypt\_out

ConnectionString: [\\vrivscors01\BCC Scorecards\Coaching\Apps\Encryption\Encrypt\_out\eCL\_IQS\_Scorecard\_20140315.csv.zip.encrypt](file:///\\vrivscors01\BCC%20Scorecards\Coaching\Apps\Encryption\Encrypt_out\eCL_IQS_Scorecard_20140315.csv.zip.encrypt)

Expression

\

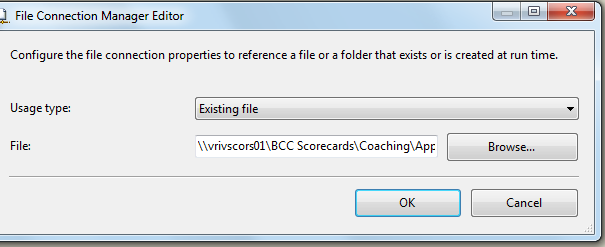


@[User::Encrypt\_Out]+@[User::FilePrefix] +

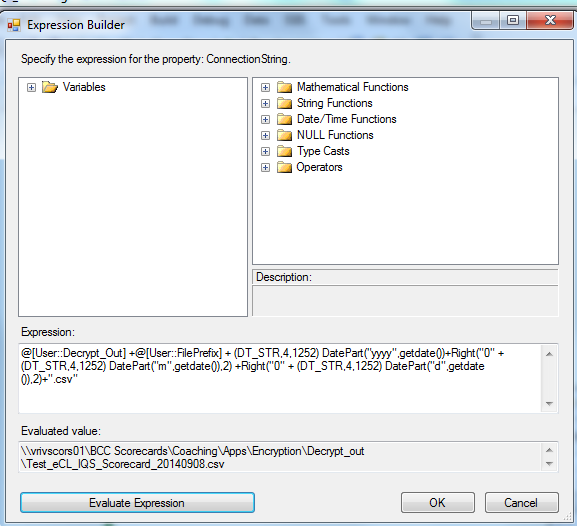
(DT\_STR,4,1252) DatePart("yyyy",getdate())+

Right("0" + (DT\_STR,4,1252) DatePart("m",getdate()),2) +

Right("0" + (DT\_STR,4,1252) DatePart("d",getdate()),2)+".csv.zip.encrypt"



1. IQS\_File

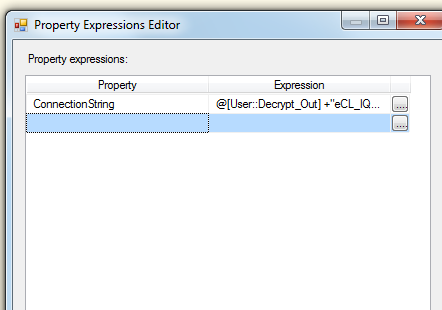


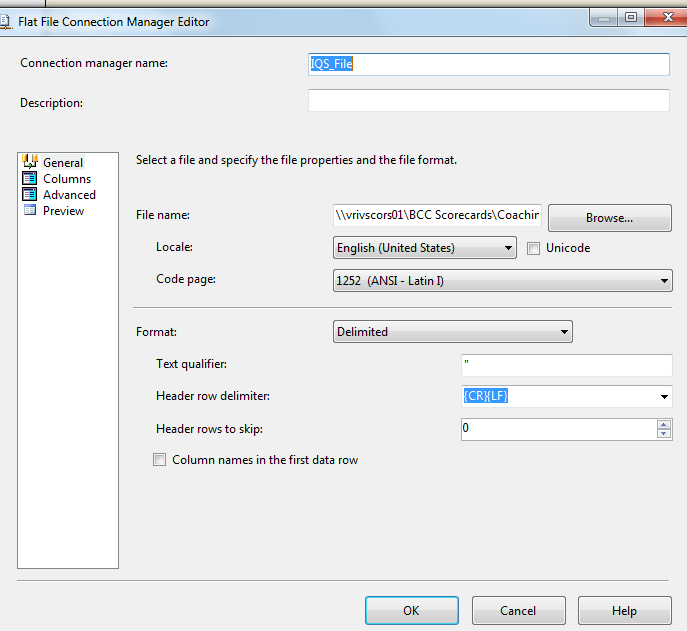
@[User::Decrypt\_Out] +@[User::FilePrefix] +

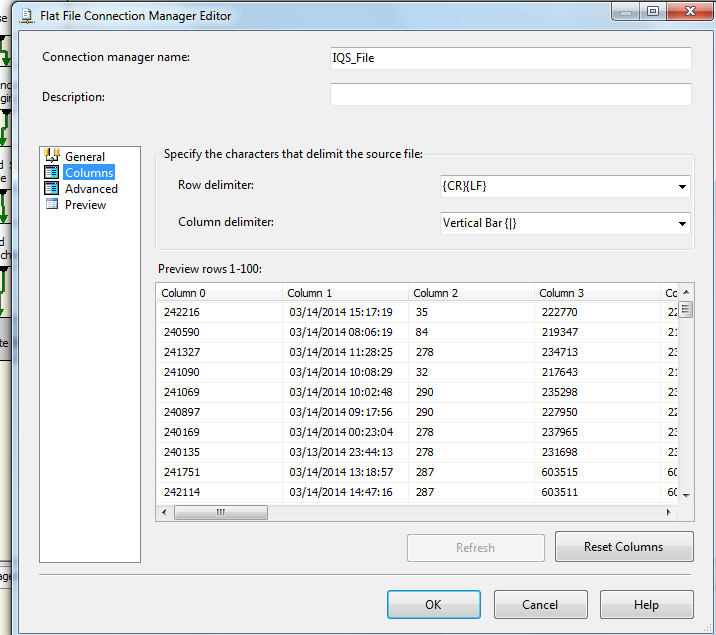
(DT\_STR,4,1252) DatePart("yyyy",getdate())+

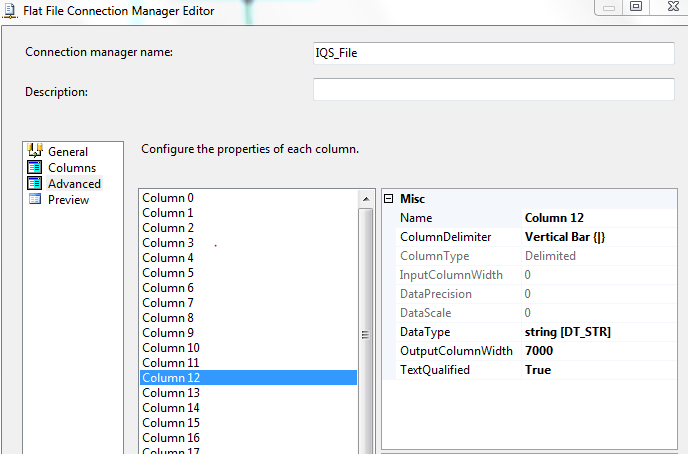
Right("0" + (DT\_STR,4,1252) DatePart("m",getdate()),2) +

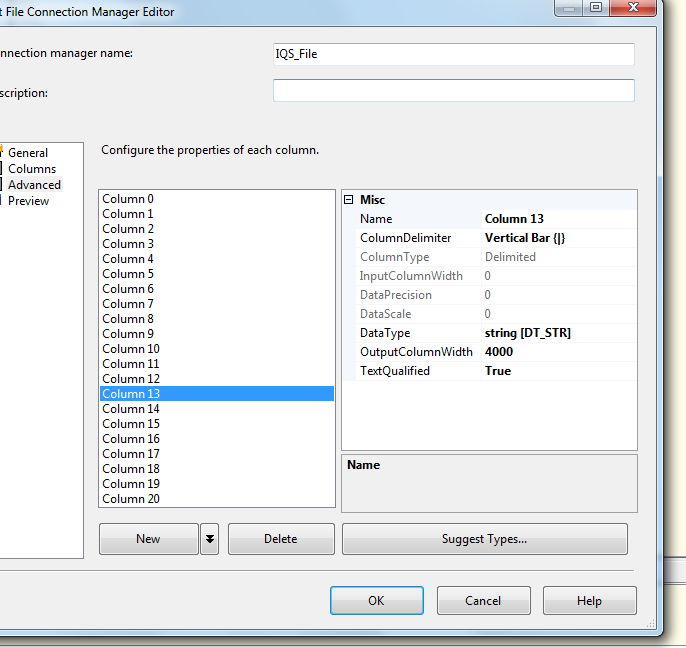
Right("0" + (DT\_STR,4,1252) DatePart("d",getdate()),2)+".csv"







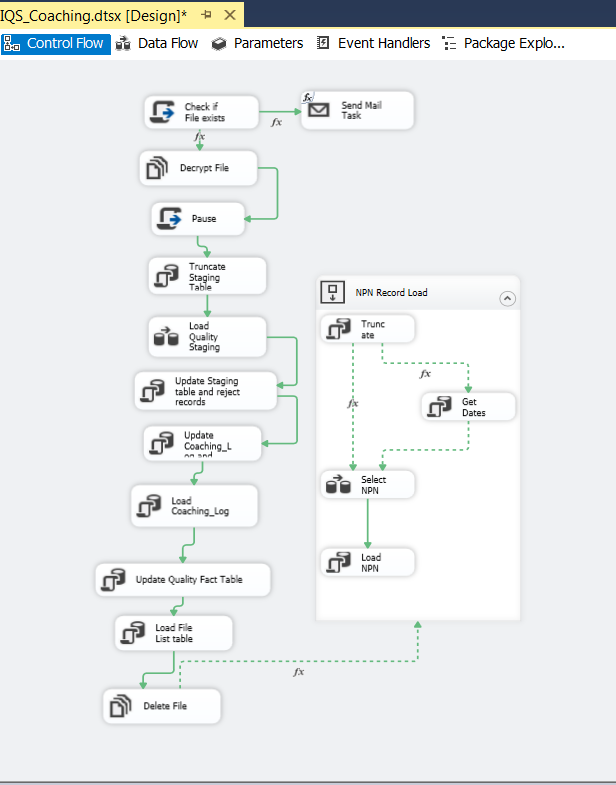




1. destinationdb

* Dev – F3420-ECLDBD01 DB – eCoachingdev
* Test - F3420-ECLDBT01 DB – eCoachingtest
* Prod - F3420-ECLDBP01 – eCoaching

#### Package Content



**Workflow**

**IQS File Load**

1. Check if file Exists

2. If File Exists Decrypt File in Encrypt\_out and place with that days date in decrypt\_out folder

Else Send Notification for Missing File and End Process

3. Pause for 90 seconds.

4. Truncate staging table

5. Load staging table

6. Update staging table and reject records

7. Update coaching log and coaching log reason table tables

8. Load coaching log and coaching log reason table tables

9. Update Quality Fact table

10. Load Quality File List table

11. Delete file

**NPN Load**

1. Truncate staging table

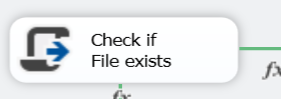
2. Get Dates for Previous Week

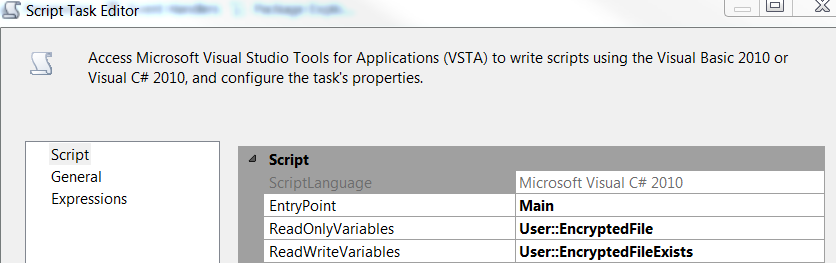
3. Select NPN logs into staging table

4. Load NPN logs into coaching log table

**Details: IQS File Load**

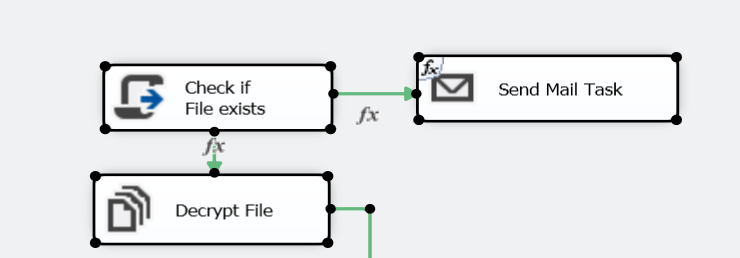
**Step1: Script task:** Check if file Exists





|  |
| --- |
| #region Namespaces  using System;  using System.Data;  using Microsoft.SqlServer.Dts.Runtime;  using System.Windows.Forms;  using System.IO;  #endregion  namespace ST\_c8c2463bd5c3454d9df6e94c8922f66f  {  /// <summary>  /// ScriptMain is the entry point class of the script. Do not change the name, attributes,  /// or parent of this class.  /// </summary>  [Microsoft.SqlServer.Dts.Tasks.ScriptTask.SSISScriptTaskEntryPointAttribute]  public partial class ScriptMain : Microsoft.SqlServer.Dts.Tasks.ScriptTask.VSTARTScriptObjectModelBase  {    /// <summary>  /// This method is called when this script task executes in the control flow.  /// Before returning from this method, set the value of Dts.TaskResult to indicate success or failure.  /// To open Help, press F1.  /// </summary>  public void Main()  {  // TODO: Add your code here  string file = Dts.Variables["User::EncryptedFile"].Value.ToString();  Dts.Variables["User::EncryptedFileExists"].Value = File.Exists(file);  Dts.TaskResult = (int)ScriptResults.Success;  }  #region ScriptResults declaration  /// <summary>  /// This enum provides a convenient shorthand within the scope of this class for setting the  /// result of the script.  ///  /// This code was generated automatically.  /// </summary>  enum ScriptResults  {  Success = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Success,  Failure = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Failure  };  #endregion  }  } |

**Precedence Constraint from Step1**

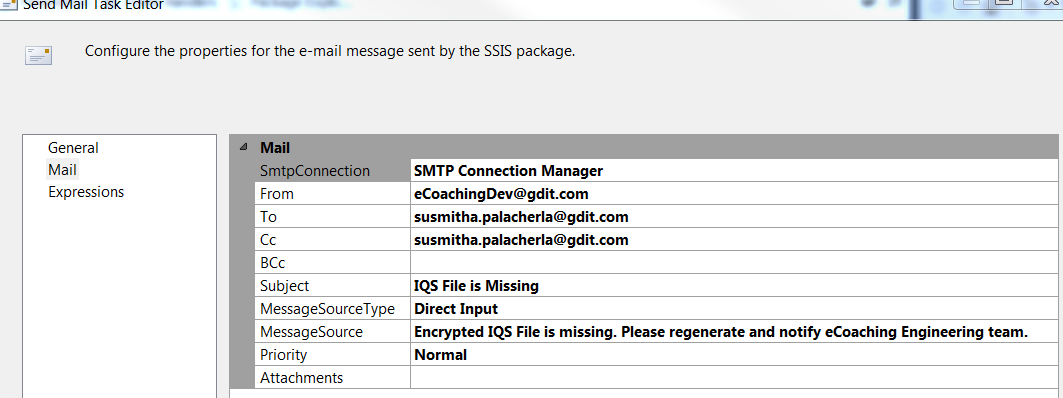


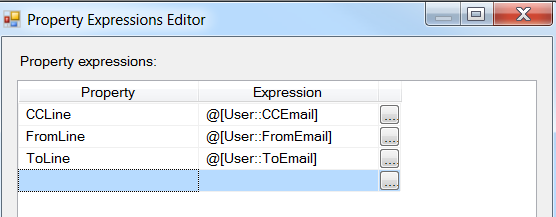
|  |  |
| --- | --- |
| Evaluation Operation | Expression and Constraint |
| Value | Success |
| Expression | @[User::EncryptedFileExists]== False |
| Multiple Constraints | Logical AND. |

IF @[User::EncryptedFileExists]== False

Then

**Step2: Send mail Task**



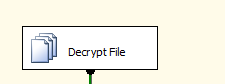


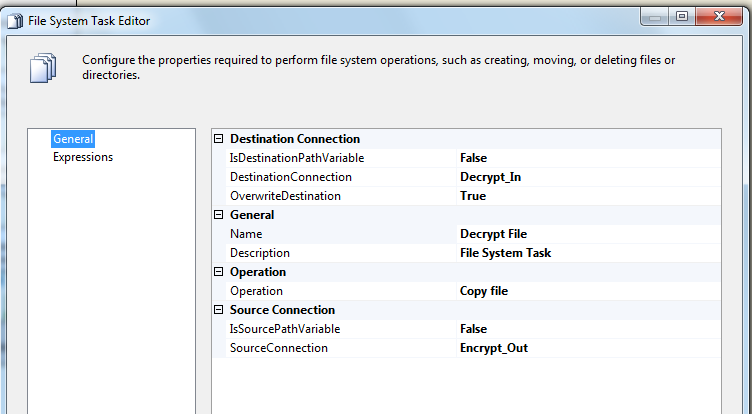
|  |  |
| --- | --- |
| Evaluation Operation | Expression and Constraint |
| Value | Success |
| Expression | @[User::EncryptedFileExists]== True |
| Multiple Constraints | Logical AND. |

IF @[User::EncryptedFileExists]== True

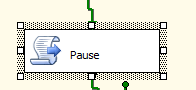
Then

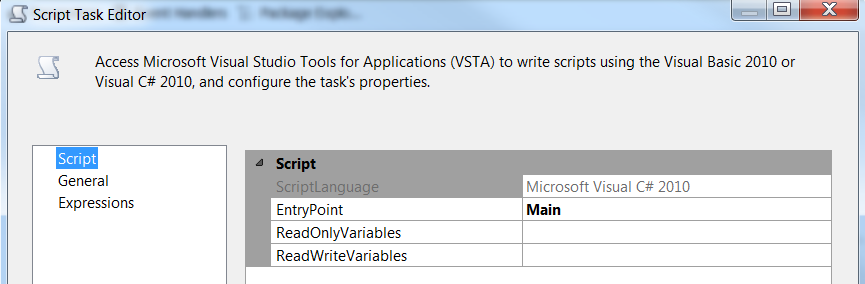
**Step2: File System task: Decrypt file**





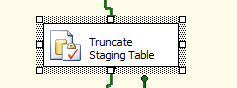
**Step3**: **Script Task: Pause** (to pause 90 seconds after the file decrypt. This ensures that the file decrypts completely before moving onto the next steps of loading the staging table.

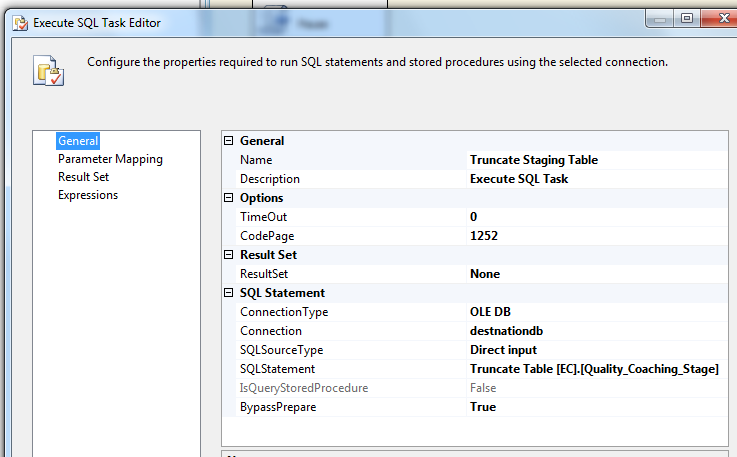




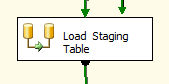
|  |
| --- |
| #region Namespaces  using System;  using System.Data;  using Microsoft.SqlServer.Dts.Runtime;  using System.Windows.Forms;  #endregion  namespace ST\_193e5478e5864ef598e1c7f28cb71fb9  {  /// <summary>  /// ScriptMain is the entry point class of the script. Do not change the name, attributes,  /// or parent of this class.  /// </summary>  [Microsoft.SqlServer.Dts.Tasks.ScriptTask.SSISScriptTaskEntryPointAttribute]  public partial class ScriptMain : Microsoft.SqlServer.Dts.Tasks.ScriptTask.VSTARTScriptObjectModelBase  {  /// <summary>  /// This method is called when this script task executes in the control flow.  /// Before returning from this method, set the value of Dts.TaskResult to indicate success or failure.  /// To open Help, press F1.  /// </summary>  public void Main()  {  // TODO: Add your code here  // Sleep for 90 seconds (\*1000)  System.Threading.Thread.Sleep(90000);  Dts.TaskResult = (int)ScriptResults.Success;    }  #region ScriptResults declaration  /// <summary>  /// This enum provides a convenient shorthand within the scope of this class for setting the  /// result of the script.  ///  /// This code was generated automatically.  /// </summary>  enum ScriptResults  {  Success = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Success,  Failure = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Failure  };  #endregion  }  } |

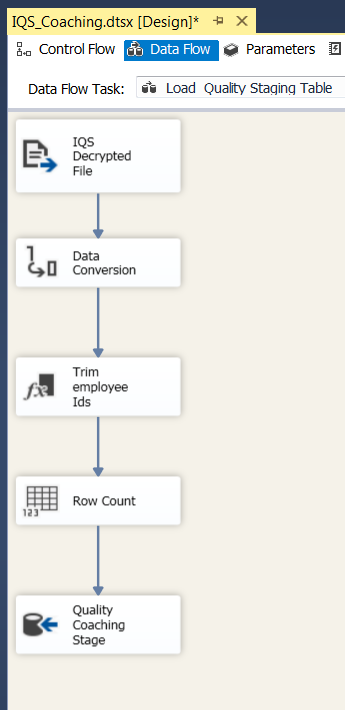
**Step 4: SQL task -** Truncate staging table (Quality\_Coaching\_Stage)





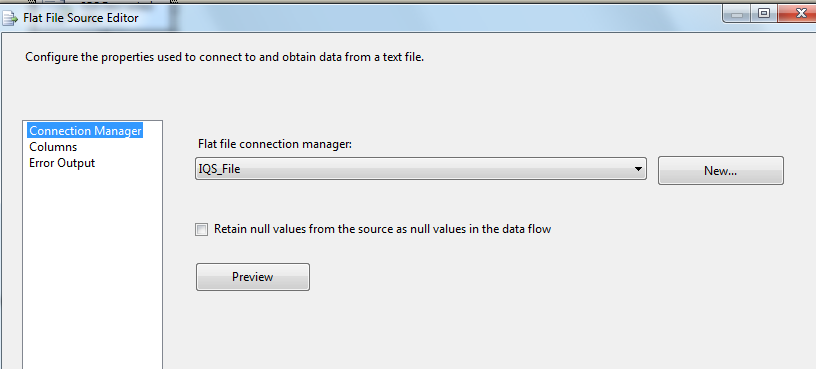
**Step 5: DFT –** Load Quality\_Coaching\_Stage table

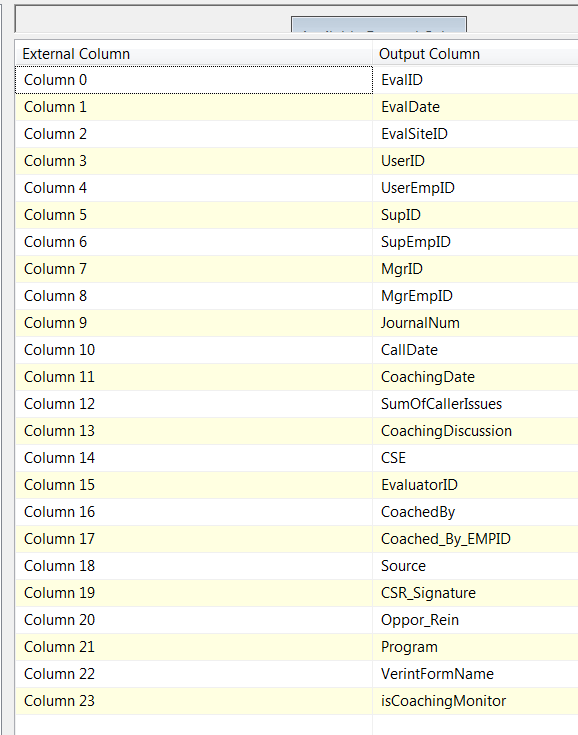




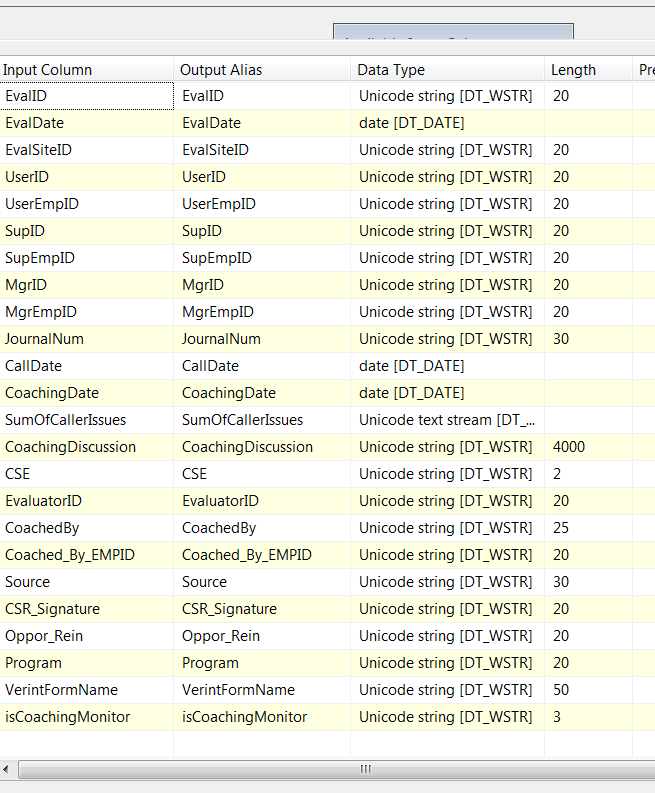
Process flow

1. Source data
2. Data conversion on string values
3. Derived column to trim employee ids and get Program and system date for date inserted
4. Row Count Transformation to get count of records staged
5. OLE DB Destination to write to Staging table
   1. Source data file

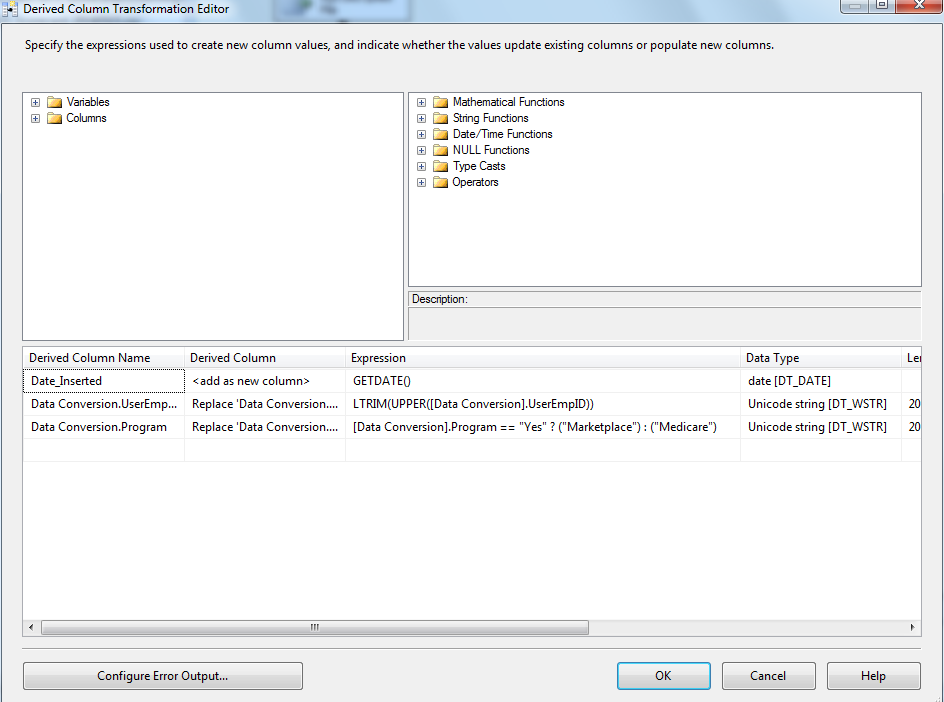




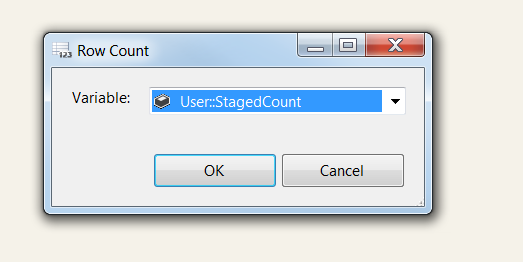
* 1. Data conversion of string values



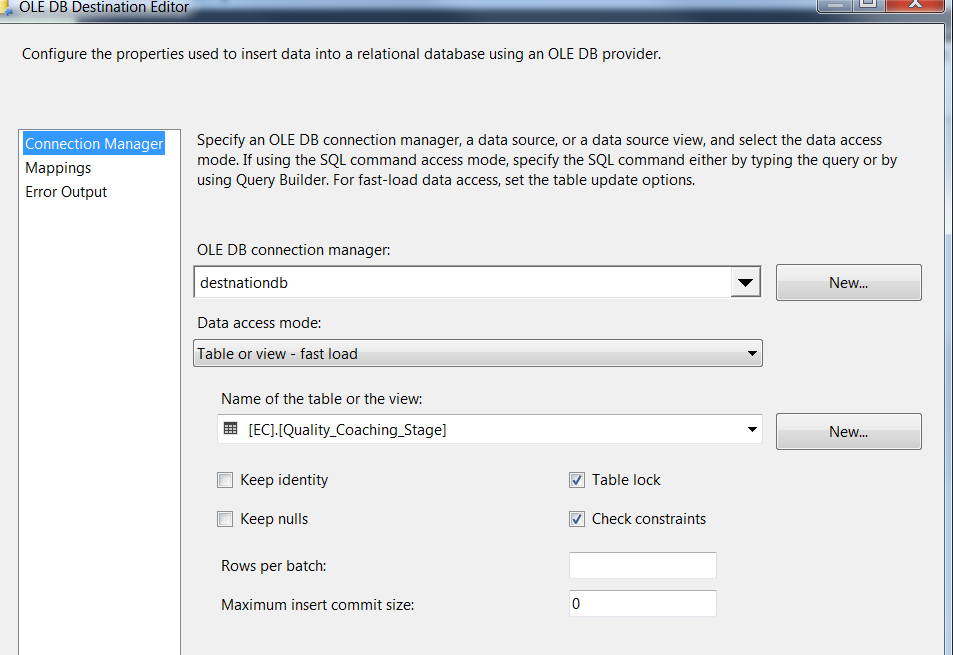
* 1. Derived columns for employee id trim and getdate()

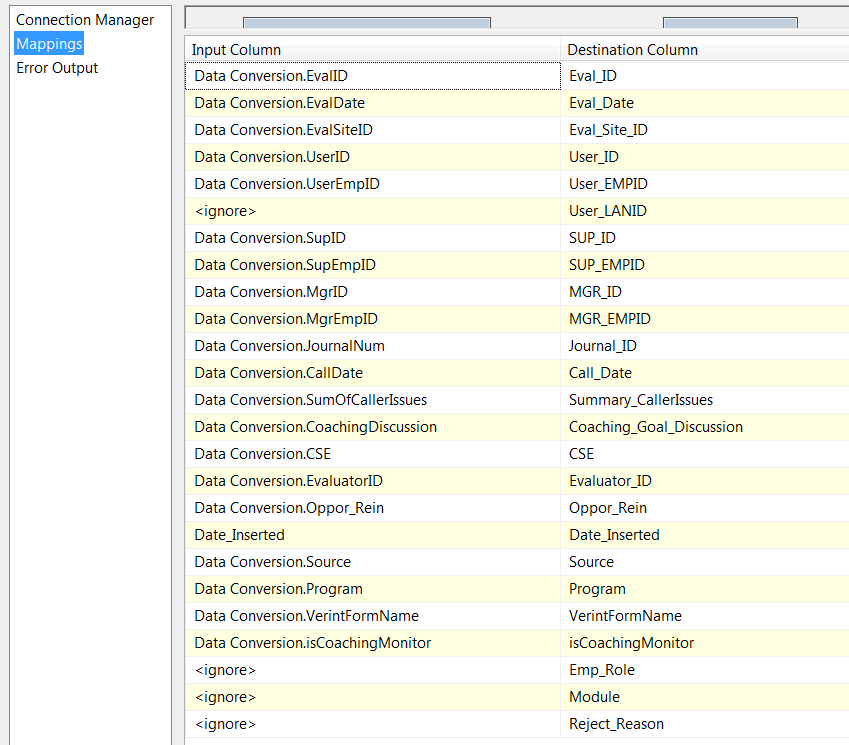


* 1. Row Count



* 1. OLE DB Destination





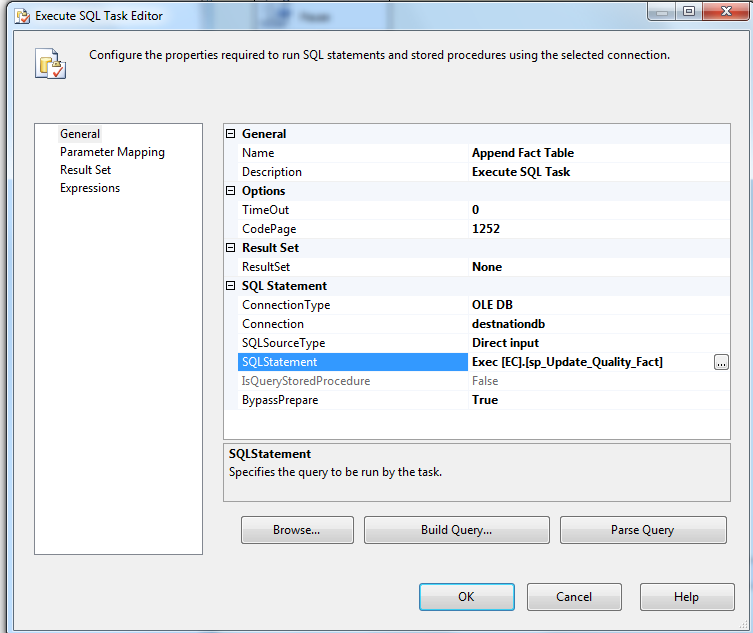
**Step 6: SQL task –** Update Staging table and reject records

**Step 7: Sql task –** Update Coaching\_Log and Coaching\_log\_reason tables

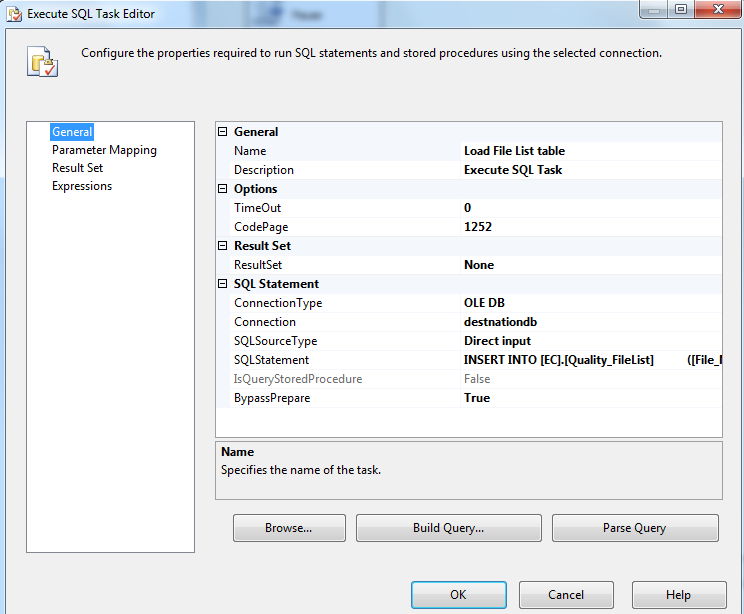
**Step 8: sql task –** Load Coaching\_Log and Coaching\_log\_reason tables

SQL statement – exec [EC].[sp\_InsertInto\_Coaching\_Log\_Quality] ? output

**Step 9: SQL task –** Update Quality Fact Table



**Step 10: SQL task –** Load File List Table



SQL Statement –

INSERT INTO [EC].[Quality\_FileList]

([File\_Name]

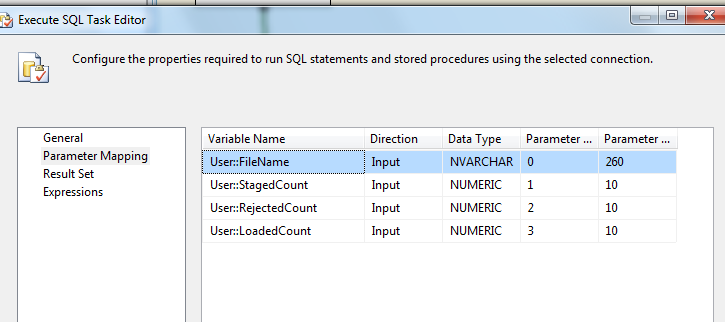
,[File\_LoadDate]

,[Count\_Staged]

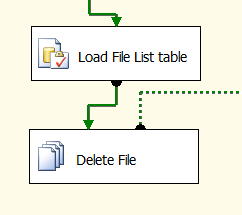
,[Count\_Rejected]

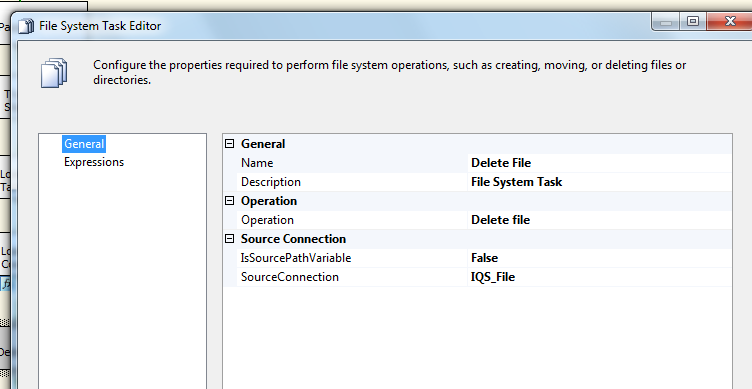
,[Count\_Loaded])

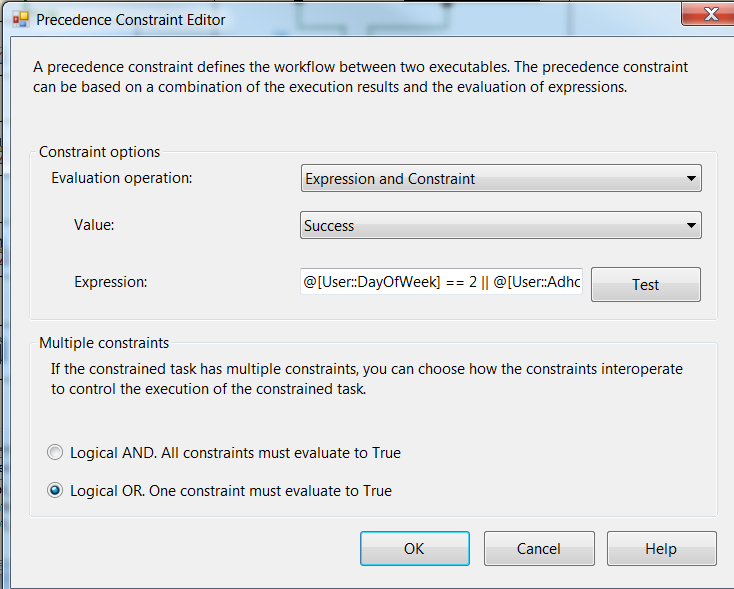
VALUES (?,GetDate(), ?, ?,?)



**Step 11: SQL task Delete File if Environment is Prod**



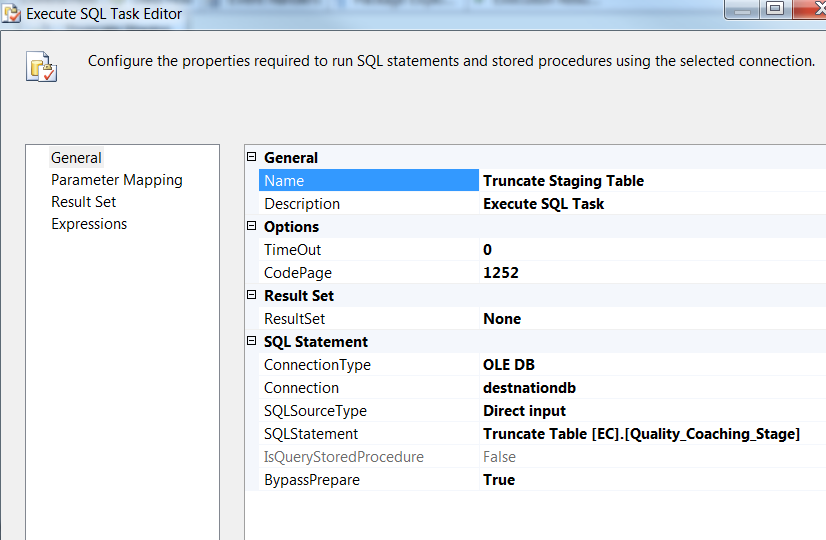




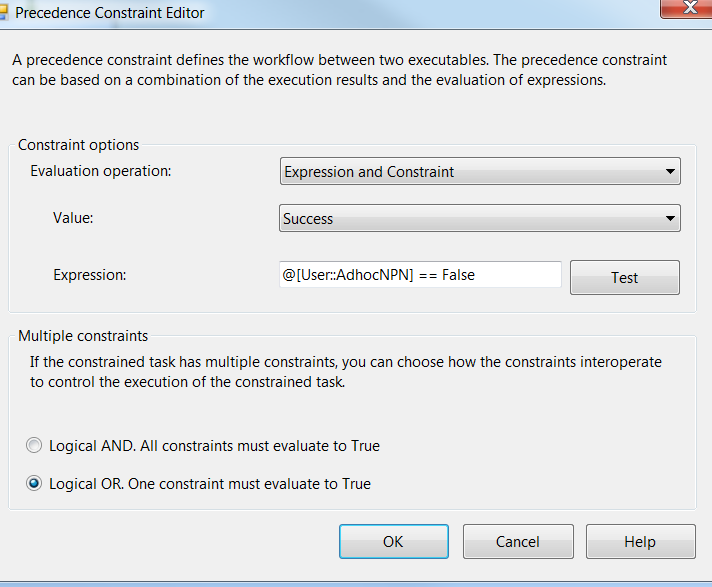
@[User::DayOfWeek] == 2 || @[User::AdhocNPN] == True

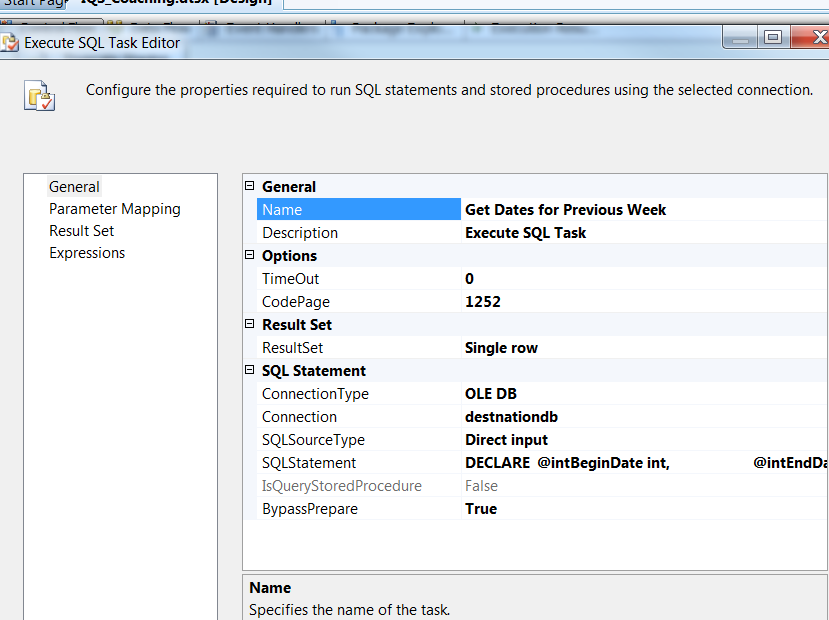
**Details: NPN Load**

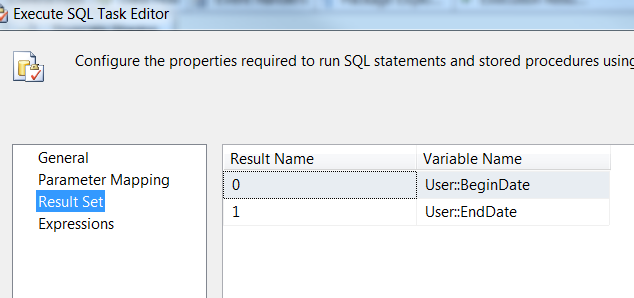
Step 1: Truncate staging table



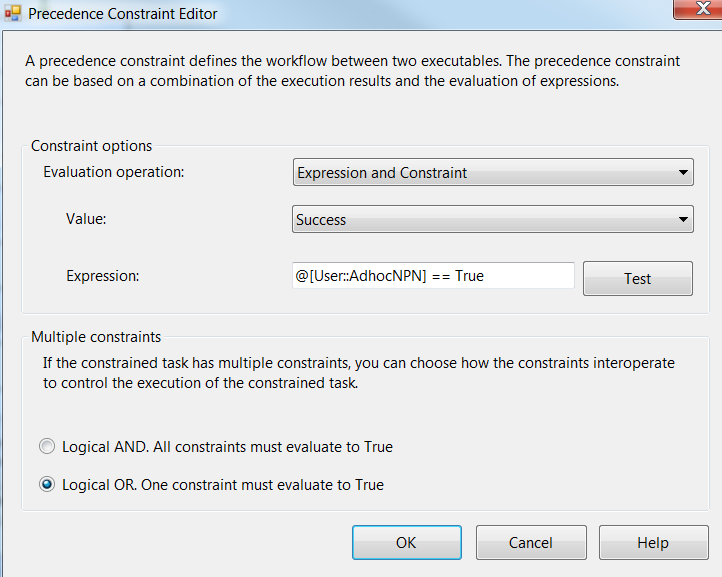
Step 2: Get Dates for Previous Week

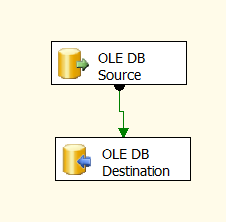


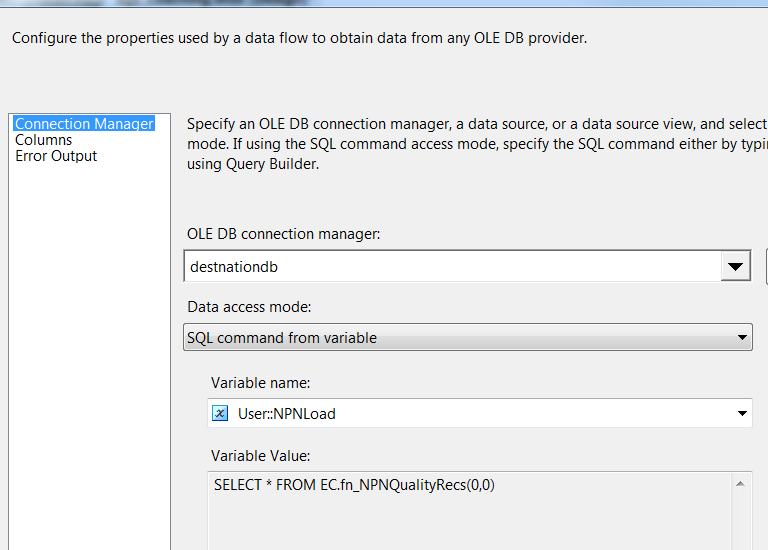


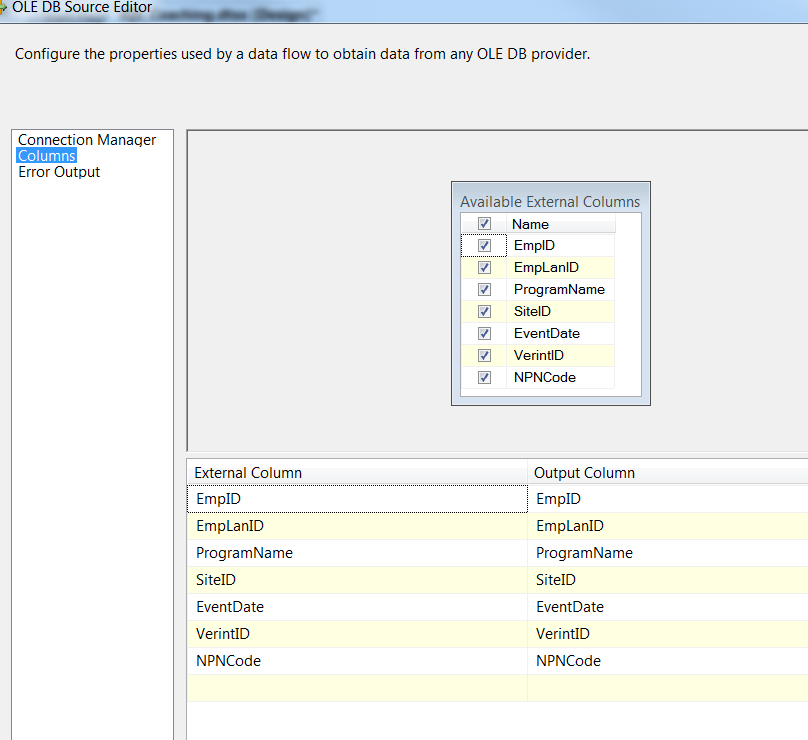


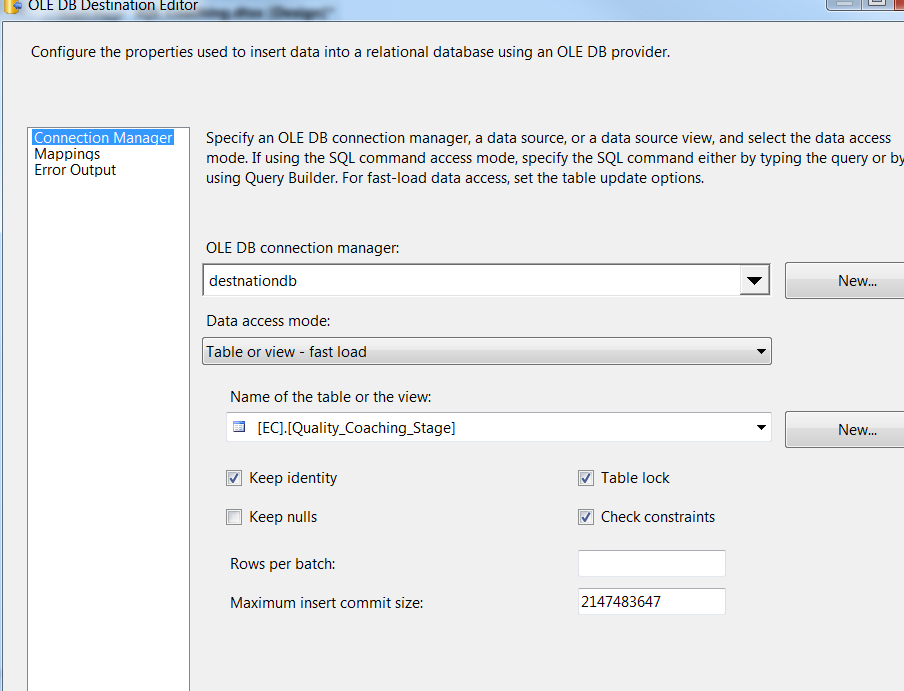
**Step 3: Select NPN logs into staging table**

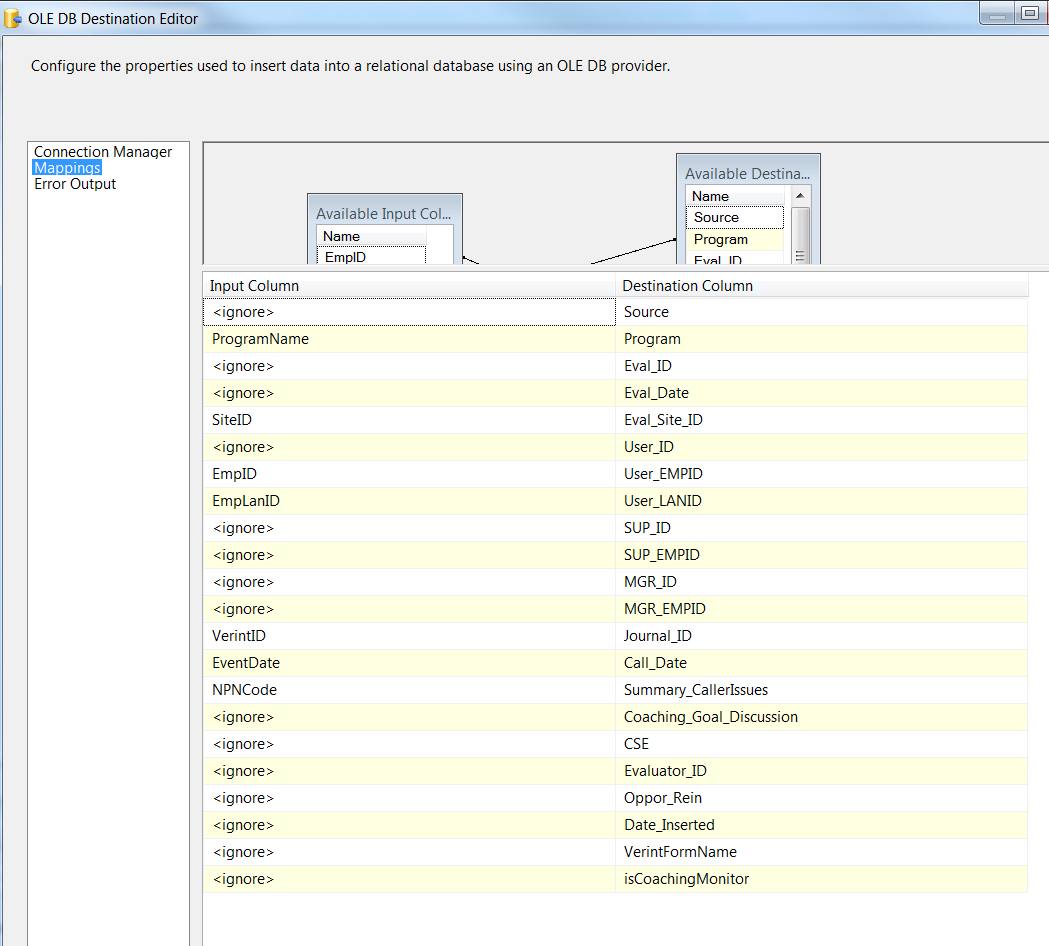




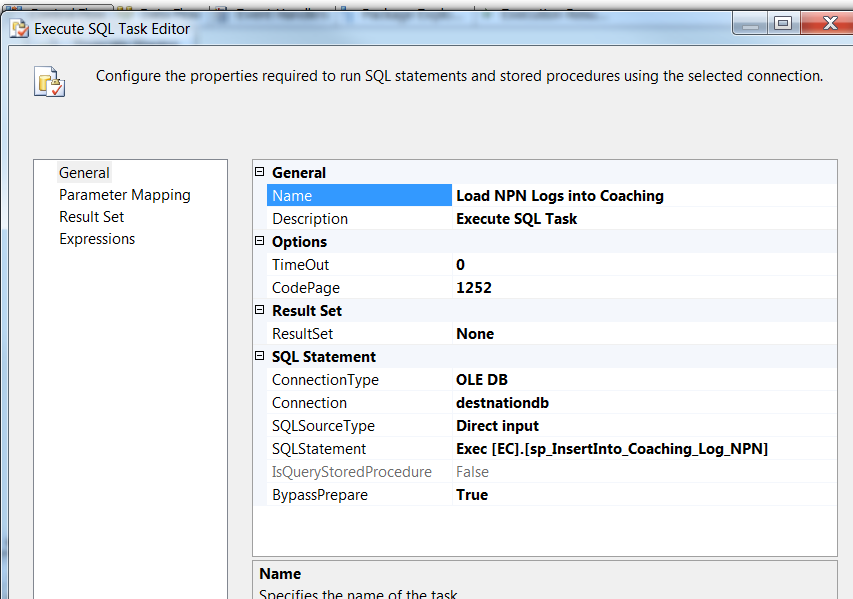








Step 4: Load NPN logs into coaching log table



### Tables

|  | **Table Name** | **Table Description** |
| --- | --- | --- |
| 1. | Quality\_Coaching\_Stage | Table used for staging the Quality evaluations from IQS during the insert into Coaching\_Log table. |
| 2. | Quality\_Coaching\_Rejected | Table used for storing the rejected Quality evaluations from IQS during the insert into Coaching\_Log table. Logs are rejected when a Lan ID cannot be looked up for an incoming Employee ID in the file. |
| 3. | Quality\_Coaching\_Fact | Table used for storing Quality source records loaded into the Coaching Log table. |
| 4. | Quality\_FileList | Table used for storing Quality file names and Load dates. |
| 5. | NPN\_Description | Table used to store Descriptions for VPN codes |

\cms\eCoaching\_V2\Code\DB\Tables

### Procedures

|  | **Procedure Name** | **Usage** |
| --- | --- | --- |
| 1. | sp\_InsertInto\_Coaching\_Log\_Quality | Load: Quality Data |
| 2. | sp\_Update\_Coaching\_Log\_Quality | Load: Quality Data |
| 3. | sp\_Update\_Quality\_Fact | Load: Quality Data |
| 4. | sp\_Get\_Dates\_For\_Previous\_Week.sql | Load: NPN Data |
| 5. | sp\_InsertInto\_Coaching\_Log\_NPN.sql | Load: NPN Data |
| 6. | sp\_InsertInto\_IQS\_Rejected | Load: Quality Data |
|  |  |  |

\cms\eCoaching\_V2\Code\DB\Stored Procedures

### Functions

|  | **Procedure Name** | **Usage** |
| --- | --- | --- |
| 1. | fn\_NPNQualityRecs.sql | Load: NPN Data |
| 2. | fn\_strNPNDescriptionFromCode | Load: NPN Data |
|  |  |  |

\cms\eCoaching\_V2\Code\DB\Functions