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



**Title: eCoaching\_Quality\_Now\_Load**

**SSIS Detail Design Document**

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

|  |  |  |
| --- | --- | --- |
| Last Revision | Last Review | Description |
| 03/26/2019 |  | TFS 13332 – Quality Now Feed Load |

Prepared by: Susmitha Palacherla Date: 03/26/2019

Department, Location: Health Solutions Division

Approved by: Date:

**Change History Log**

| **Date** | **Change Description** | **Author** |
| --- | --- | --- |
| 03/26/2019 | TFS 13332 – Initial revision. Quality Now Feed Load | Susmitha Palacherla |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[1. Overview 3](#_Toc4511506)

[1.1 Project Description 3](#_Toc4511507)

[1.2 Document Scope 4](#_Toc4511508)

[1.3 Module List 4](#_Toc4511509)

[1.4 Software and Hardware Interfaces 4](#_Toc4511510)

[1.4.1 Software 4](#_Toc4511511)

[1.4.2 Hardware 4](#_Toc4511512)

[1.5 Users 4](#_Toc4511513)

[2. Details 5](#_Toc4511514)

[2.1 Source Files 5](#_Toc4511517)

[2.2 Destination Tables 5](#_Toc4511518)

[2.3 Procedures 8](#_Toc4511519)

[2.4 SSIS Package 8](#_Toc4511520)

[2.5 Config File(s) 34](#_Toc4511521)

[2.6 SQL agent job 34](#_Toc4511522)

# Overview

## Project Description

The coaching process is a standard process to facilitate consistent communication when a performance opportunity or reinforcement exists. It provides a method to make a determination of whether coaching is needed, when to initiate coaching and verification that the performance issue has been addressed. This is the feed of Quality Now evaluations from the Quality system. Evaluations are grouped into batches and sent over from IQS to eCL when the minimum batch threshold is reached. For each batch of evaluations in the feed file only one eCL is created and defaults to Pending Supervisor Review. Details evaluations are stored in a separate detail table. The eCL follows the regular workflow.

## Document Scope

This document describes the load of the ‘Quality Now feed’ that will be provided from the IQS system which will in turn get the data from the CMT application. The Quality Now evaluations will be grouped by batches of a predetermined number. All logic to identify a valid batch and its status will be handled at either the CMT and or IQS systems and eCL will only receive batches and their status which will determine if an eCL will be created and or inactivated.

## Module List

* Source File
* Tables
* Procedures
* SSIS Package
* SQL agent job

## Software and Hardware Interfaces

### Software

* SQL Server 2012 SP3 Suite

### 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
* [\\F3420-ECLDBP01\data\](file:///\\F3420-ECLDBP01\data\) - Prod File staging share

## Users

* The first set of users for the Quality Nowrecords loaded into the eCL database are the CCO Supervisors and their immediate supervisors. Logs are loaded at pending Acknowledgement status and can be acknowledged by either the Supervisor (owner) or Manager of the Supervisor first.
* CCO Supervisors and Managers and specific HR staff will be able to access the logs from the historical Dashboards.

# Details



## Source Files

eCL\_QN\_Scorecard\_yyyymmdd.csv is the naming convention for the feed

Source system: NA

* Staging location: [\\f3420-ecldbp01\data\Coaching\IQS\Dencrypt\_Out](file:///\\f3420-ecldbp01\data\Coaching\IQS\Dencrypt_Out)
* File name eCL\_QN\_Scorecard\_yyyymmdd.csv
* Frequency: Daily
* File arrival time: 8:00 AM EST
* Destination Tables: Coaching\_Log and Coaching\_Log\_Reason

## Destination Tables

|  | **Table Name** | **Table Description** |
| --- | --- | --- |
| 1. | Quality\_Now\_Coaching\_Stage | Table used for staging the Quality Now records during the insert into the fact tables. |
| 2. | Quality\_Now\_Coaching\_Rejected | Table used for storing the rejected Quality Now records during the insert into fact tables. Logs can be rejected if an active employee with appropriate role is not found or if logs violate business rules like having non unique batch attributes. |
| 3. | Quality\_Now\_FileList | Table used for storing the Quality Now file name, counts and load dates |
| 4. | Coaching\_Log  Coaching\_Log\_Reason  Coaching\_Log\_Quality\_Now\_Evaluations | Fact tables where the created ecl and its related attributes like the Coaching Reason and Evaluation details are stored. |

|  |  |  |  |
| --- | --- | --- | --- |
| Feed Element | Coaching\_Log | Comments | |
| QN Batch ID | QNBatchID |  | |
| QN Batch Status | QNBatchStatus |  | |
| CSR Emp ID | EmpID |  | |
| Evaluation Site ID | SiteID | Looked up from Site | |
| Supervisor Emp ID | SupID |  | |
| Manager Emp ID | MgID |  | |
| Source | SourceID | Looked up from Source | |
| Strengths/Opportunities | QNStrengthsOpportunities |  | |
|  |  |  | |
| For each evaluation record within a batch, the following will be present | | | |
|  | | | |
| Feed Element | Coaching\_Log\_Quality\_Now\_Evaluations | Comments | |
| Evaluation ID | Eval\_ID |  | |
| Evaluation Date | Eval\_Date |  | |
| Evaluator ID | Evaluator\_ID |  | |
| Call Date / Time | Call\_Date |  | |
| Journal File Number | Journal\_ID |  | |
| Active Status | EvalStatus |  | |
| Summary of Caller's Issues | Summary\_CallerIssues |  | |
| Program | Program |  | |
| Evaluation Form | VerintFormName |  | |
| Coaching Monitor | isCoachingMonitor |  | |
| Business\_Process | Business\_Process |  | |
| Business\_Process\_Reason | Business\_Process\_Reason |  | |
| Business\_Process\_Comment | Business\_Process\_Comment |  | |
| Info\_Accuracy | Info\_Accuracy |  | |
| Info\_Accuracy\_Reason | Info\_Accuracy\_Reason |  | |
| Info\_Accuracy\_Comment | Info\_Accuracy\_Comment |  | |
| Privacy\_Disclaimers | Privacy\_Disclaimers |  | |
| Privacy\_Disclaimers\_Reason | Privacy\_Disclaimers\_Reason |  | |
| Privacy\_Disclaimers\_Comment | Privacy\_Disclaimers\_Comment | |  |
| Issue\_Resoluton | Issue\_Resoluton |  | |
| Issue\_Resoluton\_Comment | Issue\_Resoluton\_Comment |  | |
| Call\_Efficiency | Call\_Efficiency |  | |
| Call\_Efficiency\_Comment | Call\_Efficiency\_Comment |  | |
| Active\_Listening | Active\_Listening |  | |
| Active\_Listening\_Comment | Active\_Listening\_Comment |  | |
| Personality\_Flexing | Personality\_Flexing |  | |
| Personality\_Flexing\_Comment | Personality\_Flexing\_Comment |  | |
| Customer\_Temp\_Start | Customer\_Temp\_Start |  | |
| Customer\_Temp\_Start\_Comment | Customer\_Temp\_Start\_Comment |  | |
| Customer\_Temp\_End | Customer\_Temp\_End |  | |
| Customer\_Temp\_End\_Comment | Customer\_Temp\_End\_Comment |  | |

Table structure and details in document: eCoaching\_Database\_DD.docx

## Procedures

|  | **Procedure Name** |
| --- | --- |
| 1. | sp\_InsertInto\_Coaching\_Log\_Quality\_Now |
| 2. | sp\_InsertInto\_Quality\_Now\_Rejected |
| 3. | sp\_Update\_Coaching\_Log\_Quality\_Now |
| 4. | sp\_SelectReviewFrom\_Coaching\_Log\_Quality\_Now |

Procedure details in document: eCoaching\_Database\_DD.docx

## SSIS Package

* + - 1. **Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Scope** | **DataType** | **Value** |
| CCEmail | Quality\_Now\_Coaching | String | susmithacpalacherla@maximus.com |
| Decrypte\_Out | Quality\_Now\_Coaching | String | \\f3420-ecldbp01\data\coaching\IQS\Decrypt\_out\ |
| Decrypt\_In | Quality\_Now\_Coaching | String | \\f3420-ecldbp01\data\coaching\IQS\Decrypt\_in\ |
| Encrypte\_Out | Quality\_Now\_Coaching | String | \\f3420-ecldbp01\data\coaching\IQS\Encrypt\_out\ |
| Encrypt\_In | Quality\_Now\_Coaching | String | \\f3420-ecldbp01\data\coaching\IQS\Encrypt\_in\ |
| EncryptedFile  (Expression) | Quality\_Now\_Coaching | String | @[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" |
| EncryptedFileexists | Quality\_Now\_Coaching | Boolean |  |
| FileName  (Expression) | Quality\_Now\_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" |
| FilePrefix | Quality\_Now\_Coaching | String | eCL\_QN\_Scorecard\_ |
| FromEmail | Quality\_Now\_Coaching | String | eCoachingDev@maximus.com |
| MailSubject  (Expression) | Quality\_Now\_Coaching | String | "IQS QN Load failed in " + @[User::SystemName] |
| StagedCount | Quality\_Now\_Coaching | Int32 | 0 |
| RejectedCount | Quality\_Now\_Coaching | Int32 | 0 |
| LoadedCount | Quality\_Now\_Coaching | Int32 | 0 |
| SystemName | Quality\_Now\_Coaching | String | Prod |
| ToEmail | Quality\_Now\_Coaching | String | susmithacpalacherla@maximus.com |

* + - 1. **Connection Manager Entries**
         1. Destinationdb
* Dev – F3420-ECLDBD01 DB – eCoachingdev
* Test - F3420-ECLDBT01 DB – eCoachingtest
* Prod - F3420-ECLDBP01 – eCoaching
  + - * 1. QN\_File

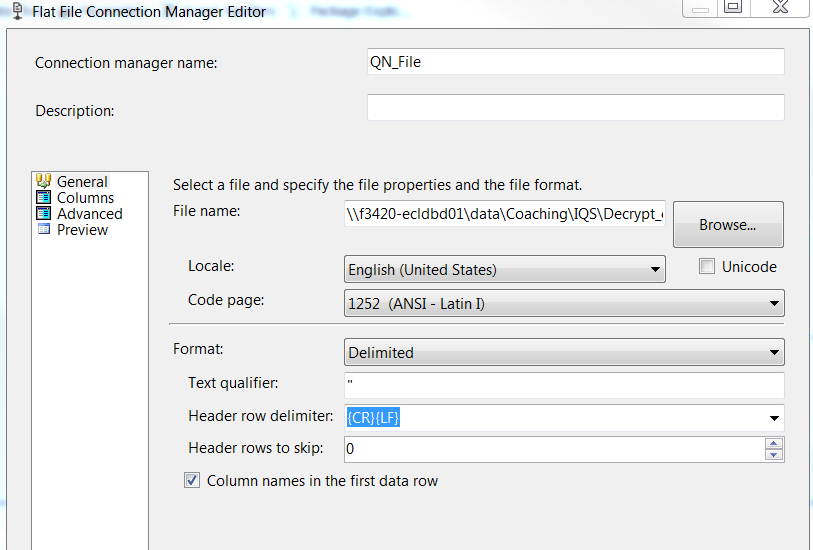
Connection String Expression:

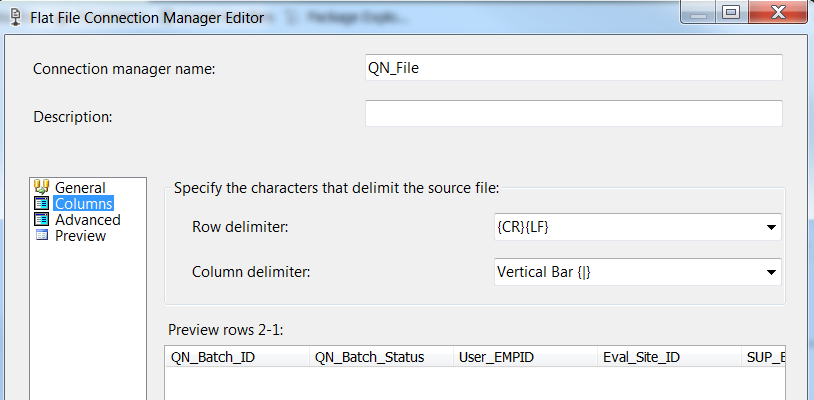
@[User::Decrypt\_Out] +@[User::FilePrefix] +

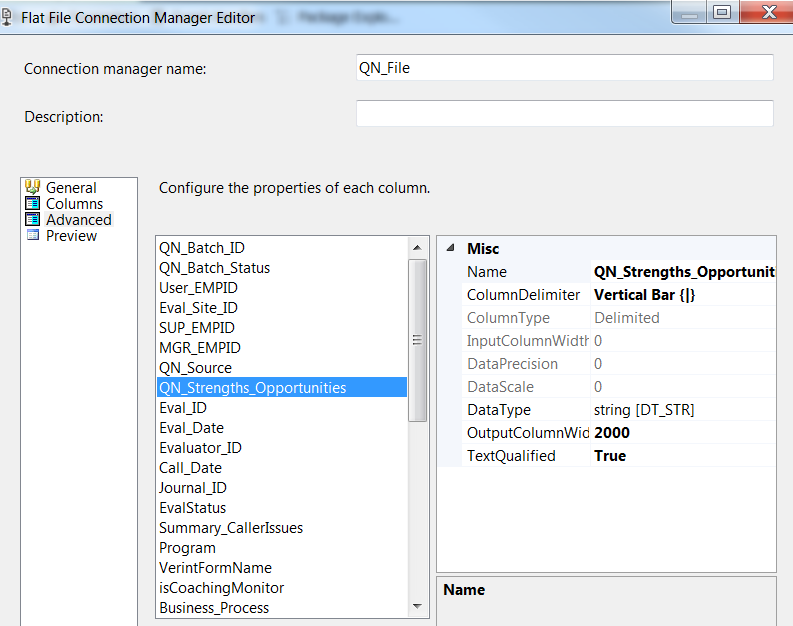
(DT\_STR,4,1252) DatePart("yyyy",getdate())+

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

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







* + - * 1. Decrypt\_In

Expression: @[User::Decrypt\_In]

* + - * 1. Encrypt\_Out

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

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

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

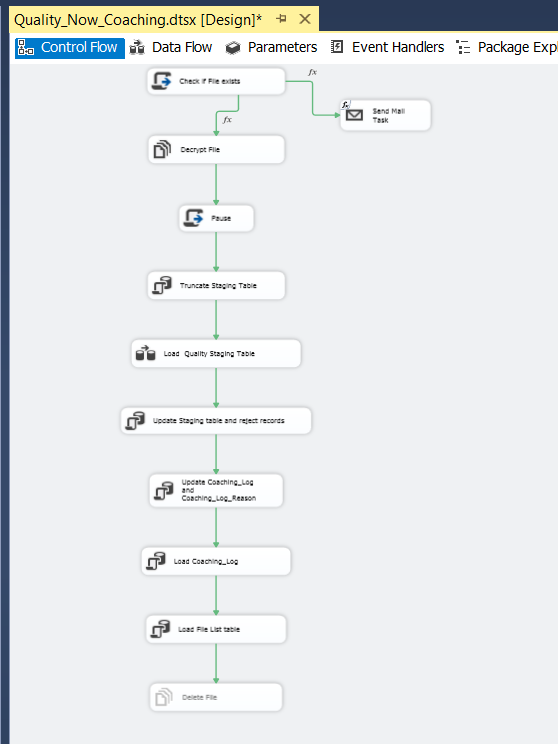
Right("0" + (DT\_STR,4,1252) DatePart("d",getdate()),2)+".csv.zip.encrypt"

* + - * 1. SMTP Connection Manager

smtpout.gdit.com



* + - 1. **Package Content(Control Flow)**



**Logical Workflow**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step #** | **Task** | **On Success** | **On Failure** |
| **1.** | **Check If File Exists** | If file exists  Go to Next step | If File Not exists  Execute Send Mail Task  Stop execution. |
| **2.** | **Decrypt file** | Go to next step. | Stop execution. |
| **3.** | **Pause 3 seconds** | Go to next step. | Stop execution. |
| **4.** | **Truncate Staging table** | Go to next step. | Stop execution. |
| **5.** | **Load Staging table** | Go to next step. | Stop execution. |
| **6.** | **Update Staging table and reject logs** | Go to next step. | Stop execution. |
| **7.** | **Update existing logs** | Go to next step. | Stop execution. |
| **8.** | **Insert Logs** | Go to next step. | Stop execution. |
| **9.** | **Load FileList table** | Go to next step. | Stop execution. |
| **10.** | **Delete decrypted file** | Stop execution. | Stop execution. |
|  |  |  |  |

* + - * 1. **Script task: Check If file exists Files**

|  |
| --- |
|  |
|  |
|  |
|  |
| #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  {  #region Help: Using Integration Services variables and parameters in a script  /\* To use a variable in this script, first ensure that the variable has been added to  \* either the list contained in the ReadOnlyVariables property or the list contained in  \* the ReadWriteVariables property of this script task, according to whether or not your  \* code needs to write to the variable. To add the variable, save this script, close this instance of  \* Visual Studio, and update the ReadOnlyVariables and  \* ReadWriteVariables properties in the Script Transformation Editor window.  \* To use a parameter in this script, follow the same steps. Parameters are always read-only.  \*  \* Example of reading from a variable:  \* DateTime startTime = (DateTime) Dts.Variables["System::StartTime"].Value;  \*  \* Example of writing to a variable:  \* Dts.Variables["User::myStringVariable"].Value = "new value";  \*  \* Example of reading from a package parameter:  \* int batchId = (int) Dts.Variables["$Package::batchId"].Value;  \*  \* Example of reading from a project parameter:  \* int batchId = (int) Dts.Variables["$Project::batchId"].Value;  \*  \* Example of reading from a sensitive project parameter:  \* int batchId = (int) Dts.Variables["$Project::batchId"].GetSensitiveValue();  \* \*/  #endregion  #region Help: Firing Integration Services events from a script  /\* This script task can fire events for logging purposes.  \*  \* Example of firing an error event:  \* Dts.Events.FireError(18, "Process Values", "Bad value", "", 0);  \*  \* Example of firing an information event:  \* Dts.Events.FireInformation(3, "Process Values", "Processing has started", "", 0, ref fireAgain)  \*  \* Example of firing a warning event:  \* Dts.Events.FireWarning(14, "Process Values", "No values received for input", "", 0);  \* \*/  #endregion  #region Help: Using Integration Services connection managers in a script  /\* Some types of connection managers can be used in this script task. See the topic  \* "Working with Connection Managers Programatically" for details.  \*  \* Example of using an ADO.Net connection manager:  \* object rawConnection = Dts.Connections["Sales DB"].AcquireConnection(Dts.Transaction);  \* SqlConnection myADONETConnection = (SqlConnection)rawConnection;  \* //Use the connection in some code here, then release the connection  \* Dts.Connections["Sales DB"].ReleaseConnection(rawConnection);  \*  \* Example of using a File connection manager  \* object rawConnection = Dts.Connections["Prices.zip"].AcquireConnection(Dts.Transaction);  \* string filePath = (string)rawConnection;  \* //Use the connection in some code here, then release the connection  \* Dts.Connections["Prices.zip"].ReleaseConnection(rawConnection);  \* \*/  #endregion  /// <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  }  } |
|  |
| **If User::EncryptedFileexists ==0**  **Then**  **Execute Send Mail Task** |
|  |
|  |
|  |
|  |
|  |
|  |

* + - * 1. **Decrypt Encrypted File**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

* + - * 1. **Script Task: Pause**

|  |
| --- |
|  |
|  |
|  |
|  |
| #region Help: Introduction to the script task  /\* The Script Task allows you to perform virtually any operation that can be accomplished in  \* a .Net application within the context of an Integration Services control flow.  \*  \* Expand the other regions which have "Help" prefixes for examples of specific ways to use  \* Integration Services features within this script task. \*/  #endregion  #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  {  #region Help: Using Integration Services variables and parameters in a script  /\* To use a variable in this script, first ensure that the variable has been added to  \* either the list contained in the ReadOnlyVariables property or the list contained in  \* the ReadWriteVariables property of this script task, according to whether or not your  \* code needs to write to the variable. To add the variable, save this script, close this instance of  \* Visual Studio, and update the ReadOnlyVariables and  \* ReadWriteVariables properties in the Script Transformation Editor window.  \* To use a parameter in this script, follow the same steps. Parameters are always read-only.  \*  \* Example of reading from a variable:  \* DateTime startTime = (DateTime) Dts.Variables["System::StartTime"].Value;  \*  \* Example of writing to a variable:  \* Dts.Variables["User::myStringVariable"].Value = "new value";  \*  \* Example of reading from a package parameter:  \* int batchId = (int) Dts.Variables["$Package::batchId"].Value;  \*  \* Example of reading from a project parameter:  \* int batchId = (int) Dts.Variables["$Project::batchId"].Value;  \*  \* Example of reading from a sensitive project parameter:  \* int batchId = (int) Dts.Variables["$Project::batchId"].GetSensitiveValue();  \* \*/  #endregion  #region Help: Firing Integration Services events from a script  /\* This script task can fire events for logging purposes.  \*  \* Example of firing an error event:  \* Dts.Events.FireError(18, "Process Values", "Bad value", "", 0);  \*  \* Example of firing an information event:  \* Dts.Events.FireInformation(3, "Process Values", "Processing has started", "", 0, ref fireAgain)  \*  \* Example of firing a warning event:  \* Dts.Events.FireWarning(14, "Process Values", "No values received for input", "", 0);  \* \*/  #endregion  #region Help: Using Integration Services connection managers in a script  /\* Some types of connection managers can be used in this script task. See the topic  \* "Working with Connection Managers Programatically" for details.  \*  \* Example of using an ADO.Net connection manager:  \* object rawConnection = Dts.Connections["Sales DB"].AcquireConnection(Dts.Transaction);  \* SqlConnection myADONETConnection = (SqlConnection)rawConnection;  \* //Use the connection in some code here, then release the connection  \* Dts.Connections["Sales DB"].ReleaseConnection(rawConnection);  \*  \* Example of using a File connection manager  \* object rawConnection = Dts.Connections["Prices.zip"].AcquireConnection(Dts.Transaction);  \* string filePath = (string)rawConnection;  \* //Use the connection in some code here, then release the connection  \* Dts.Connections["Prices.zip"].ReleaseConnection(rawConnection);  \* \*/  #endregion  /// <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(3000);  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  }  } |

* + - * 1. **SQL Task: Truncate Staging table**

|  |
| --- |
|  |
|  |
|  |
|  |

* + - * 1. **DFT: Load Staging table**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

* + - * 1. **SQLTask: Update Staging table and Reject Records**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

* + - * 1. **SQLTask: Update Coaching\_Log and Evaluation tables**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

* + - * 1. **SQL Task: Update Coaching\_Log and Evaluation tables**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

* + - * 1. **SQL Task: Load FileList Table**

|  |
| --- |
|  |
|  |
| INSERT INTO [EC].[Quality\_Now\_FileList]  ([File\_Name]  ,[File\_LoadDate]  ,[Count\_Staged]  ,[Count\_Rejected]  ,[Count\_Loaded])  VALUES (?,GetDate(), ?, ?,?); |
|  |
|  |
|  |
|  |

* + - * 1. **File System Task: Delete File**

|  |
| --- |
|  |
|  |
|  |
|  |

* + - * 1. **Event Handlers: Executable Quality\_Now\_Coaching: On Error**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
| "Package: "+ (DT\_WSTR, 50) @[System::PackageName] +".  Time: "+ (DT\_WSTR, 50) @[System::StartTime] +".  Task: "+ (DT\_WSTR, 50) @[System::SourceName] +".  Error Description: "+ (DT\_STR, 2000,1252) @[System::ErrorDescription] |
|  |

## Config File(s)

|  |
| --- |
| [..\..\..\Code\ETL\Dev\_Quality\_Now\_Coaching.dtsConfig](../../../Code/ETL/Dev_Quality_Now_Coaching.dtsConfig)  [..\..\..\Code\ETL\Test\_Quality\_Now\_Coaching.dtsConfig](../../../Code/ETL/Test_Quality_Now_Coaching.dtsConfig)  [..\..\..\Code\ETL\Prod\_Quality\_Now\_Coaching.dtsConfig](../../../Code/ETL/Prod_Quality_Now_Coaching.dtsConfig) |

## SQL agent job

* Production Instance : F3420-ECLDBP01
* Production Job: CoachingQualityNowLoad
* Production Package: [\\F3420-ECLDBP01\ssis\Coaching\Packages\Quality Now\_Coaching.dtsx](file:///\\F3420-ECLDBP01\ssis\Coaching\Packages\Quality_Other%20_Coaching.dtsx)
* Production Config File: Prod\_Quality\_Now\_Coaching.dtsConfig
* Owner: ecljobowner
* Run As: ECLProxy (ECL Credential using application service account AD\SVC-f3420-APPECLP01
* Schedule: Daily 11:30 AM EST