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



**Title: eCoaching\_Warning\_History\_Load**

**SSIS Detail Design Document**

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

|  |  |  |
| --- | --- | --- |
| Last Revision | Last Review | Description |
| 12/8/2014 |  | Initial revision |

Prepared by: Susmitha Palacherla Date: 12/8/2014

Department, Location: Health Solutions Division

Approved by: Date:

**Change History Log**

| **Date** | **Version** | **Change Description** | **Author** |
| --- | --- | --- | --- |
| 12/8/2014 | 1.0 | Initial revision. Load process setup per SCR 13623. | Susmitha Palacherla |
| 12/8/2014 | 2.0 | Updates from system testing for SCR 13623 | Susmitha Palacherla |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1. Overview 4](#_Toc404161672)

[1.1 Project Description 4](#_Toc404161673)

[1.2 Document Scope 4](#_Toc404161674)

[1.3 Source Files 4](#_Toc404161675)

[1.4 Module List 5](#_Toc404161676)

[1.5 Software and Hardware Interfaces 6](#_Toc404161677)

[1.5.1 Software 6](#_Toc404161678)

[1.5.2 Hardware 6](#_Toc404161679)

[1.6 Users 6](#_Toc404161680)

[2. Details 6](#_Toc404161681)

[2.1 Source Files 6](#_Toc404161684)

[2.2 Module Details 7](#_Toc404161685)

[2.2.1 SQL agent job 7](#_Toc404161686)

[2.2.2 SSIS Package 7](#_Toc404161687)

[2.2.3 Tables 33](#_Toc404161688)

[2.2.4 Procedures 33](#_Toc404161689)

# Overview

## Project Description

The eCoaching Log (eCL) is an internal Vangent process designed for the entry and storage of the coaching notes that supervisors, managers, Advanced Resolution Center (ARC) CSRs and quality leads make regarding Employee performance. This document describes the process for loading the historical Verbal and Written progressive disciplinary warnings the employee or supervisor receives from their supervisor or manager. A file will be loaded for each of the CCO sites.

## Document Scope

This document describes the load of the historical warnings reports provided by each site into the eCoaching database. The logs are loaded into the system as ‘completed’ and similar to the new Warnings logs submitted from the user interface there will be no notifications sent out for the records.

## Source Files

A file with historical warning logs will be provided by each site.

Files are staged at [\\VRIVSCORS01\BCC Scorecards\Coaching\WH\](file:///\\VRIVSCORS01\BCC%20Scorecards\Coaching\WH\)

Files will have a naming convention of SiteXwarnings.csv where SiteX is the site name.

## Module List

* SQL agent job
* SSIS Package
* Tables
* Procedures

## Software and Hardware Interfaces

### Software

* SQL Server 2008 R2 SP1 Suite

### Hardware

* VRIVFSSDBT02\SCORD01,1437 – Dev DB Instance
* VRIVFSSDBT02\SCORT01,1438 – Test DB Instance
* VDENSSDBP07\SCORP01 – Prod DB Instance
* [\\VRIVSCORS01\BCC Scorecards\](file:///\\VRIVSCORS01\BCC%20Scorecards\) - File staging share

# Details



## Source Files

This is a one time load where each site will provide a file containing all the historical warnings records . Files will sue naming convention siteXwarnings.csv where SiteX represents the sitename.

* Description: Files containing historical warnings for CSRs and supervisors
* Source system: CCO Sites
* Staging location: [\\vrivscors01\BCC Scorecards\Coaching\WH\](file:///\\vrivscors01\BCC%20Scorecards\Coaching\WH\)
* File name SiteXwarnings.csv
* Frequency: One Time Load
* File arrival time: NA
* Destination Table: warning\_Log and warning\_Log\_Reason

## Module Details

### SQL agent job

* Production Instance : VDENSSDBP07\SCORP01
* Production Job: CoachingWHLoad
* Production Package: [\\VDENSSDBP07\scorecard-ssis\Coaching\Warning\_History.dtsx](file:///\\VDENSSDBP07\scorecard-ssis\Coaching\Warning_History.dtsx)
* Production Config File: Prod\_warning\_History.dtsConfig
* Owner: jobs\_admin
* Run As: BccScrdSQLAgent
* Schedule: Once on demand
* Destination File Location: N/A

### SSIS Package

* + - 1. **Variables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Scope** | **DataType** | **Value** | **Evaluate As expression** |
| FileNum | Warning\_History | Int32 | 0 | False |
| FilePath | Warning\_History | String | \\vrivscors01\BCC Scorecards\Coaching\WH | False |
| varDBName | Warning\_History | String | Dev | False |
| varFileName | Warning\_History | String |  | False |
| FailMessage | Warning\_History | String | WH Load failed in Dev | True |
| StagedCount | Warning\_History | Int32 | 0 | False |
| RejectedCount | Warning\_History | Int32 | 0 | False |
| LoadedCount | Warning\_History | Int32 | 0 | False |
| CC | Warning\_History | String | Susmitha.palacherla@gdit.com | True |
| From | Warning\_History | String | Susmitha.palacherla@gdit.com | True |
| To | Warning\_History | String | Susmitha.palacherla@gdit.com | True |
| Subject | Warning\_History | String | WH Load failed in Dev | True |
|  |  |  |  |  |
|  |  |  |  |  |

* + - 1. **Connection Manager Entries**
         1. Destinationdb
* Dev - VRIVFSSDBT02\SCORD01,1437) DB – eCoachingdev
* Prod - VDENSSDBP07\SCORP01 DB – eCoaching

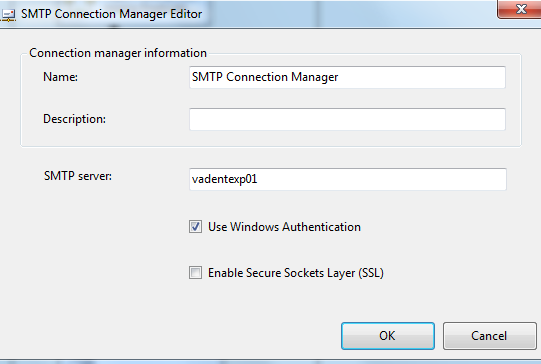


* + - * 1. WH\_File
* \\vrivscors01\bcc scorecards\Coaching\WH\eCL\_WH\_feed\_XXXXAyyyymmdd.csv

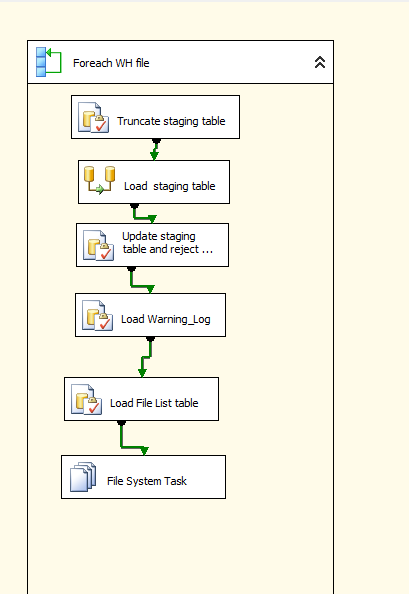




* + - * 1. SMTP Connection Manager



* + - 1. **Package Content**



**Workflow**

**Summary of Steps**

1. For Each Loop – For each WH file in Staging directory
2. SQL task – Truncate WH Staging table
3. DFT – Stage WH Staging table
4. SQL task Update WH Staging table
5. SQL task – Load Warning Log
6. SQL task – Load File list table
7. File system task – Delete File

**Details**

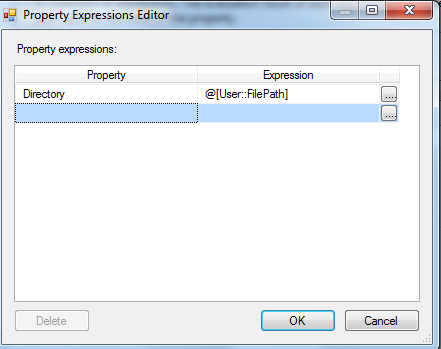
1. Step1: For Each Loop – For each WH file in Staging directory

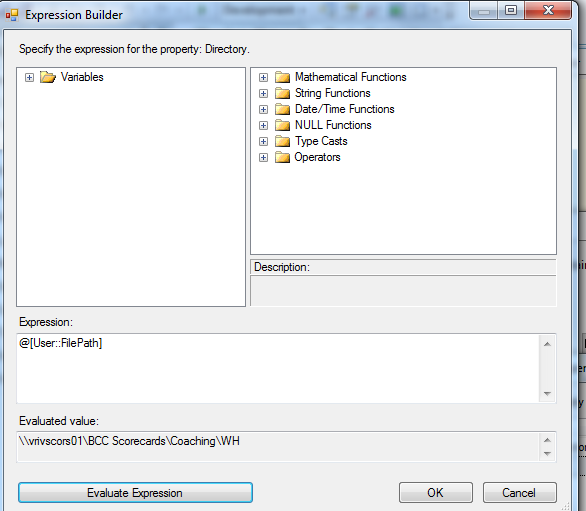
Container: For each Loop

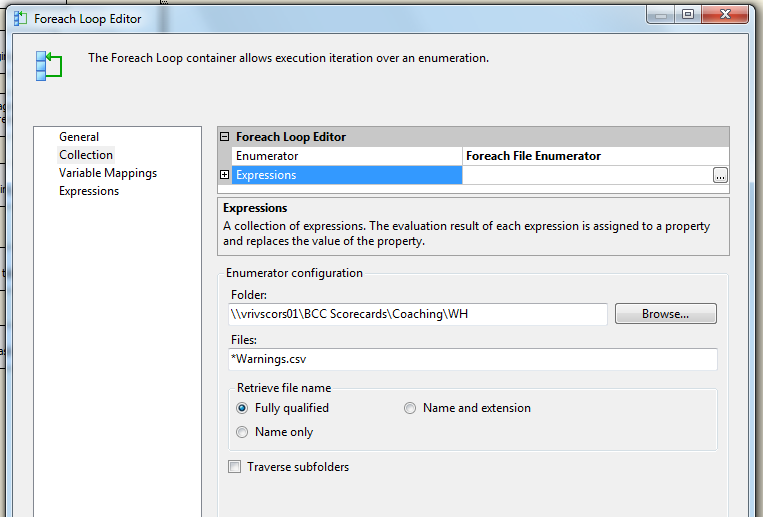
Folder: \\vrivscors01\BCC Scorecards\Coaching\WH

Files: \*Warnings.csv











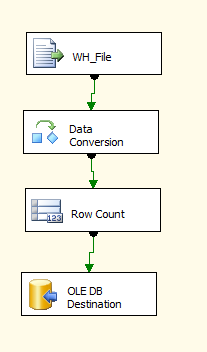
1. SQL task – Truncate WH Staging table



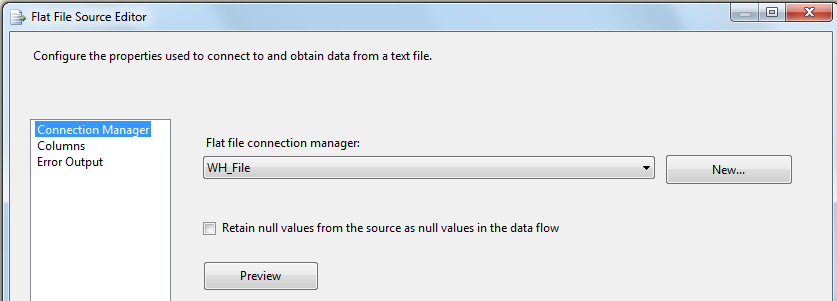
Executes Truncate Table [EC].[Warning\_History\_Stage]

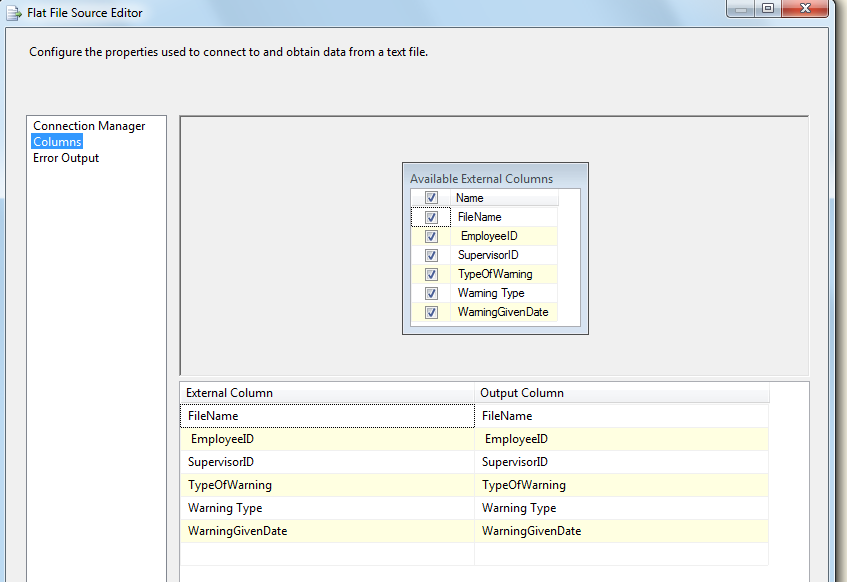
1. DFT: Load WH staging table



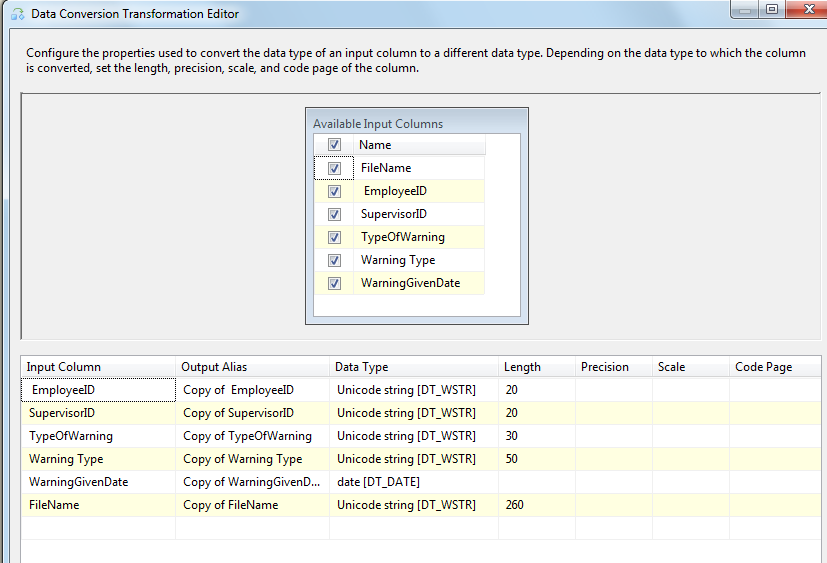


1. Flat File source

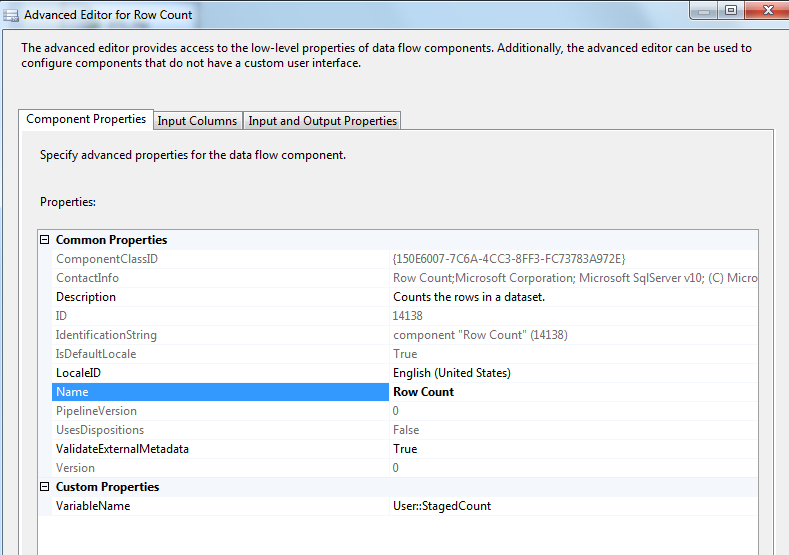




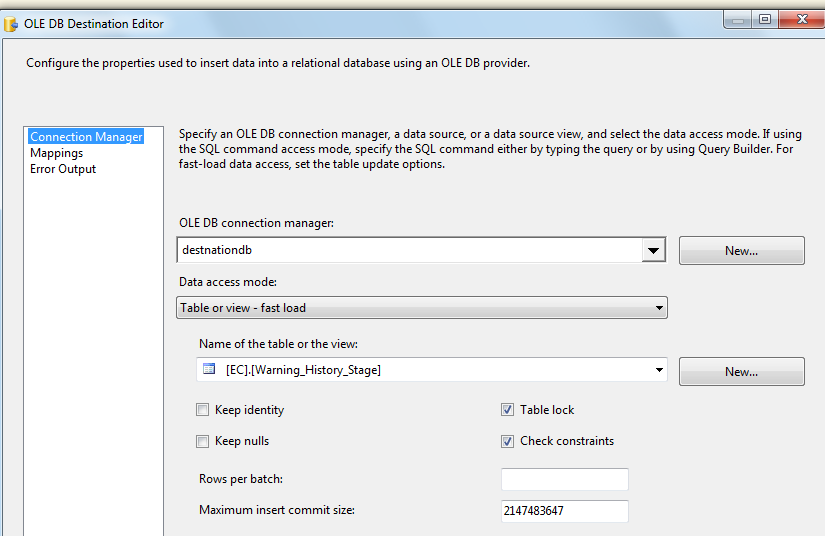
1. Data Conversion

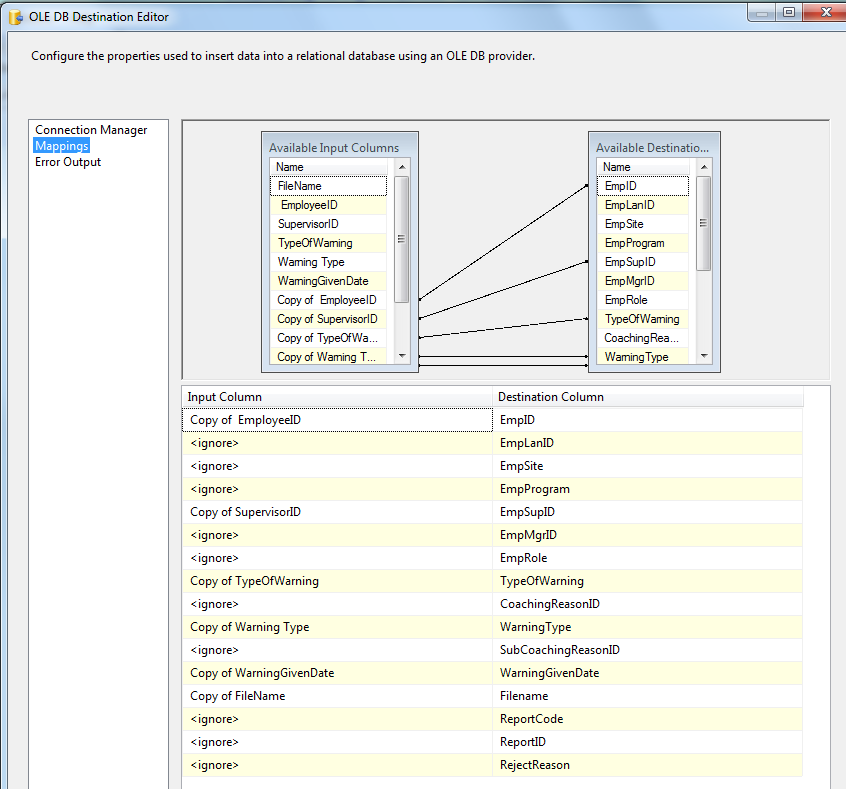


1. Row count

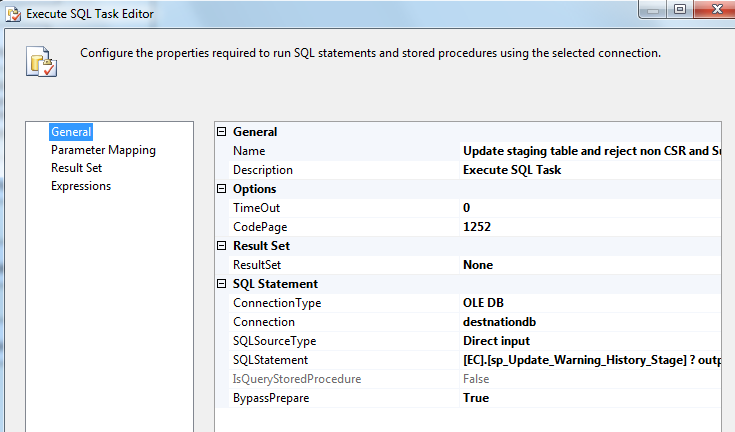


1. OLEDB Destination



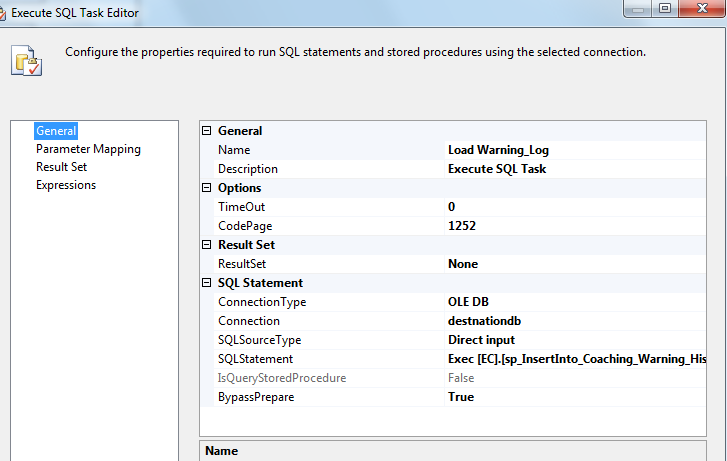


1. SQL task: Update WH Staging table

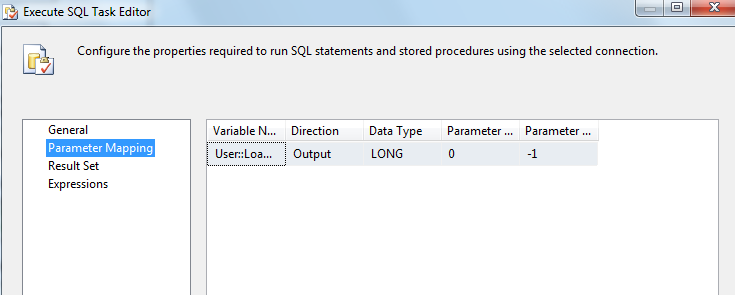


Exec [EC].[sp\_Update\_Warning\_History\_Stage] ? output

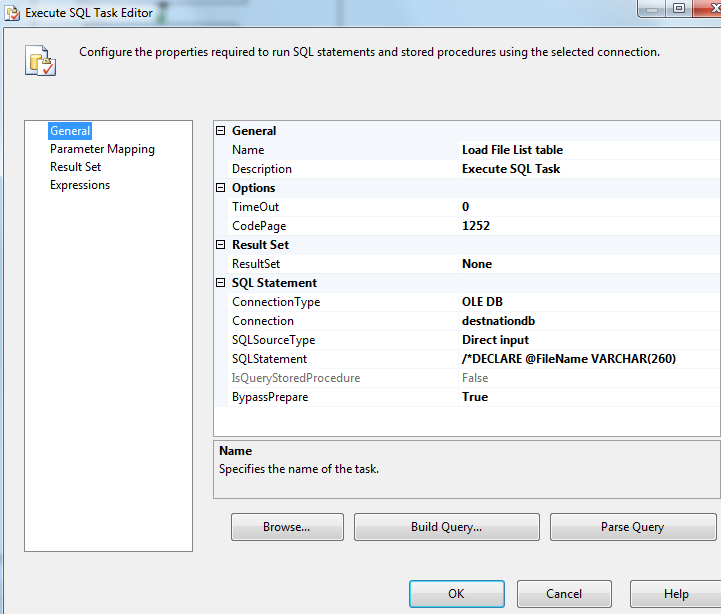
1. SQL task: Load Warning\_Log Table



Exec [EC].[sp\_InsertInto\_Warning\_History] ? output



1. SQL Task: Load File List Table



INSERT INTO [EC].[Warning\_History\_FileList]

([File\_Name]

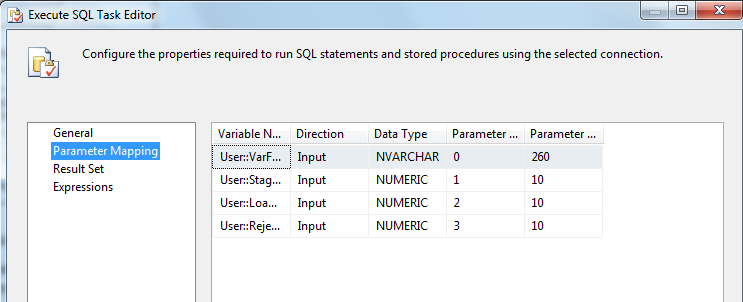
,[File\_LoadDate]

,[Count\_Staged]

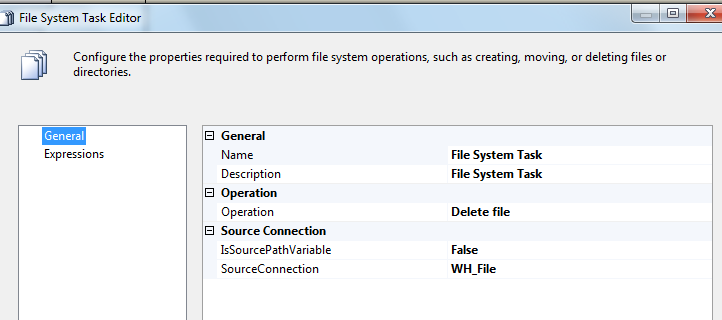
,[Count\_Loaded]

,[Count\_Rejected])

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



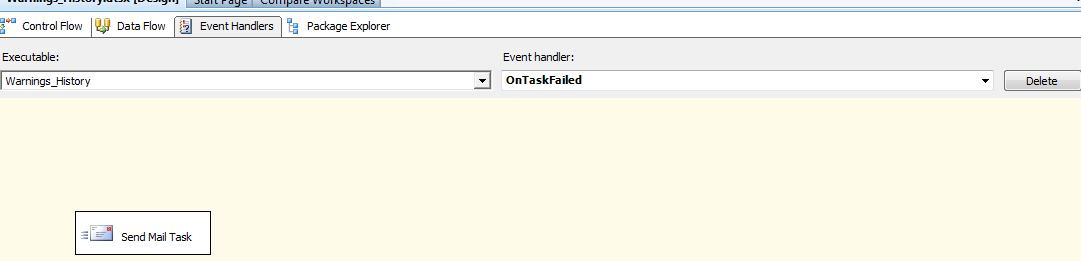
1. File System Task: Delete Loaded Files



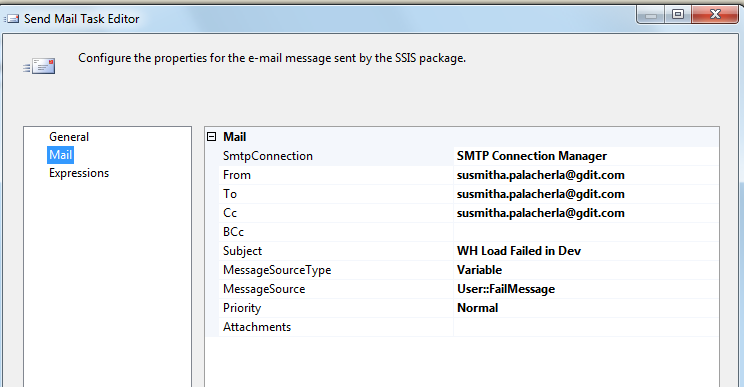
* + - 1. **Event Handlers**

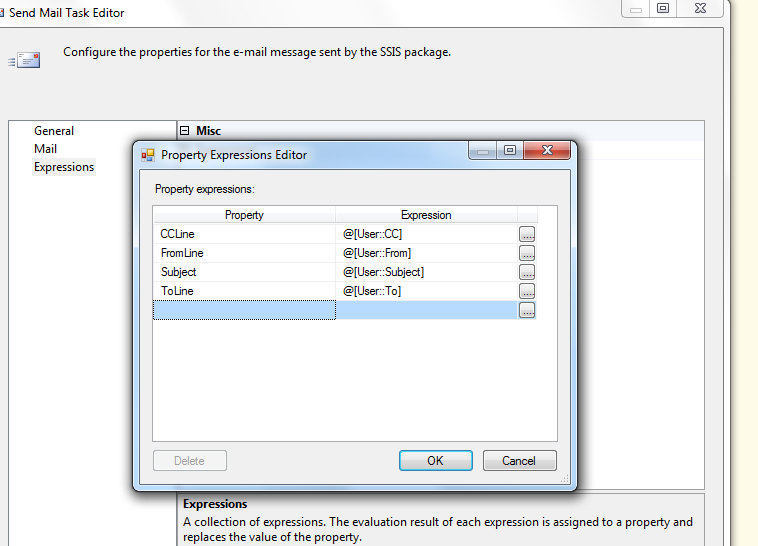
Executable: Warning\_History

EventHandler: OntaskFailed









### Tables

|  | **Table Name** | **Table Description** |
| --- | --- | --- |
| 1. | Warning\_History\_Stage | Table used for staging the WH records during the insert into Warning\_Log table. |
| 2. | Warning\_History\_Rejected | Table used for storing the rejected WH records during the insert into warning\_Log table. |
| 3. | Warning\_History\_FileList | Table used for storing the WH file names and Load dates. |

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

### Procedures

|  | **Procedure Name** | **Usage** |
| --- | --- | --- |
| 1. | sp\_Update\_Warning\_History\_Stage | Feed Load: Warning History |
| 2. | sp\_InsertInto\_Warning\_History\_Rejected | Feed Load: Warning History |
| 3. | sp\_InsertInto\_Coaching\_Warning\_History | Feed Load: Warning History |
| 4. |  |  |
| 6. |  |  |

### Code

#### Create Staging table

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [EC].[Warning\_History\_Stage](

[EmpID] [nvarchar](20) NULL,

[EmpLanID] [nvarchar](30) NULL,

[EmpSite] [nvarchar](20) NULL,

[EmpProgram] [nvarchar](30) NULL,

[EmpSupID] [nvarchar](20) NULL,

[EmpMgrID] [nvarchar](20) NULL,

[EmpRole] [nvarchar](3) NULL,

[TypeOfWarning] [nvarchar](30) NULL,

[CoachingReasonID] [int] NULL,

[WarningType] [nvarchar](50) NULL,

[SubCoachingReasonID] [int] NULL,

[WarningGivenDate] [datetime] NULL,

[Filename] [nvarchar](260) NULL,

[ReportCode] [nvarchar](40) NULL,

[ReportID] [int] NULL,

[RejectReason] [nvarchar](200) NULL

) ON [PRIMARY]

#### GOCreate Rejected table

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [EC].[Warning\_History\_Rejected](

[EmpID] [nvarchar](20) NULL,

[EmpSupID] [nvarchar](20) NULL,

[EmpRole] [nvarchar](3) NULL,

[TypeOfWarning] [nvarchar](30) NULL,

[WarningType] [nvarchar](50) NULL,

[WarningGivenDate] [datetime] NULL,

[Filename] [nvarchar](260) NULL,

[Reject\_Reason] [nvarchar](200) NULL,

[Reject\_Date] [datetime] NULL

) ON [PRIMARY]

GO

#### Create File List table

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [EC].[Warning\_History\_FileList](

[File\_SeqNum] [int] IDENTITY(1,1) NOT NULL,

[File\_Name] [nvarchar](260) NULL,

[File\_LoadDate] [datetime] NULL,

[Count\_Staged] [int] NULL,

[Count\_Loaded] [int] NULL,

[Count\_Rejected] [int] NULL

) ON [PRIMARY]

GO

#### Create Update Staging table procedure

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 12/4/2014

-- Description: Performs the following actions.

-- Removes Alpha characters from first 2 positions of Emp\_ID

-- Populate Employee and Hierarchy attributes from Employee Table

-- Inserts into Rejected table and Deletes rejected records.

-- =============================================

CREATE PROCEDURE [EC].[sp\_Update\_Warning\_History\_Stage]

@Count INT OUTPUT

AS

BEGIN

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [EmpID]= [EC].[RemoveAlphaCharacters]([EmpID])

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 3 ms

-- Populate Attributes from Employee Table

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [EmpLanID] = EMP.[Emp\_LanID]

,[EmpSite]= EMP.[Emp\_Site]

,[EmpProgram]= EMP.[Emp\_Program]

,[EmpRole]=

CASE WHEN EMP.[Emp\_Job\_Code]in ('WACS01', 'WACS02','WACS03') THEN 'C'

WHEN EMP.[Emp\_Job\_Code] = 'WACS40' THEN 'S'

ELSE 'O' END

,[CoachingReasonID]= CASE [TypeOfWarning]

WHEN 'Verbal Warning' THEN 28

WHEN 'Written warning' THEN 29

WHEN 'Final Written warning' THEN 30 ELSE -1 END

,[SubCoachingReasonID]= CASE [WarningType]

WHEN 'ETS' THEN 48

WHEN 'Attendance' THEN 92

WHEN 'Conduct (including Call Avoidance)' THEN 93

WHEN 'Quality/Performance' THEN 94

WHEN 'Security or Privacy Issue' THEN 95

WHEN 'Other Policy (non-Security/Privacy)' THEN 96 ELSE -1 END

,[ReportCode] = RIGHT(Filename,CHARINDEX('\',REVERSE(Filename))-1)

FROM [EC].[Warning\_History\_Stage] STAGE JOIN [EC].[Employee\_Hierarchy]EMP

ON LTRIM(STAGE.EmpID) = LTRIM(EMP.Emp\_ID)

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 3 ms

-- Populate Sup Emp ID if Missing

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [EmpSupID] = H.SupID

FROM [EC].[Warning\_History\_Stage] STAGE JOIN [EC].[CSR\_Hierarchy]H

ON LTRIM(STAGE.EmpID) = LTRIM(H.EmpID)

AND STAGE.[WarningGivenDate]BETWEEN H.[StartDate] and H.[EndDate]

WHERE ([EmpSupID] IS NULL OR [EmpSupID]= ' ')

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 3 ms

-- Populate Mgr Emp ID if Missing

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [EmpMgrID] = H.MgrID

FROM [EC].[Warning\_History\_Stage] STAGE JOIN [EC].[CSR\_Hierarchy]H

ON LTRIM(STAGE.EmpID) = LTRIM(H.EmpID)

AND STAGE.[WarningGivenDate]BETWEEN H.[StartDate] and H.[EndDate]

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 3 ms

-- Reject records not belonging to CSRs and Supervisors

BEGIN

EXEC [EC].[sp\_InsertInto\_Warning\_History\_Rejected]

END

WAITFOR DELAY '00:00:00.03' -- Wait for 3 ms

-- Delete rejected records

BEGIN

DELETE FROM [EC].[Warning\_History\_Stage]

WHERE [RejectReason]is not NULL

SELECT @Count =@@ROWCOUNT

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 3 ms

-- Assign Record ID

BEGIN

DECLARE @id INT

SET @id = 0

UPDATE [EC].[Warning\_History\_Stage]

SET @id = [ReportID] = @id + 1

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 3 ms

END -- [EC].[sp\_Update\_Warning\_History\_Stage]

GO

#### Create Insert into Rejected table Procedure

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Susmitha Palacherla

-- Create date: 11/19/14

-- Description: Determines rejection Reason for Warning\_History logs.

-- Populates the records with reject reasons to the Reject table.

-- =============================================

CREATE PROCEDURE [EC].[sp\_InsertInto\_Warning\_History\_Rejected]

AS

BEGIN

-- reject records for Employees with missing lan IDs

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [RejectReason]= N'Employee Lan ID is missing.'

WHERE EmpLanID is NULL

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 5 ms

-- reject records that dont belong to CSR or Supervisor

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [RejectReason]= N'Record does not belong to CSR or Superviosr.'

WHERE EmpRole not in ('C', 'S') and [RejectReason] is NULL

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 5 ms

-- reject records for Employee not in Hierarchy table

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [RejectReason]= N'Employee Not found in Hierarchy table.'

WHERE EMPID NOT IN

(SELECT DISTINCT EMP\_ID FROM [EC].[Employee\_Hierarchy])

AND [RejectReason]is NULL

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 5 ms

-- reject records for Employee not in Hierarchy table

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [RejectReason]= N'Employee Not found in Hierarchy table.'

WHERE EMPID NOT IN

(SELECT DISTINCT EMP\_ID FROM [EC].[Employee\_Hierarchy])

AND [RejectReason]is NULL

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 5 ms

-- reject records for Supervisor not in Hierarchy table

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [RejectReason]= N'Supervisor Not found in Hierarchy table.'

WHERE EmpSupID NOT IN

(SELECT DISTINCT EMP\_ID FROM [EC].[Employee\_Hierarchy])

AND [RejectReason]is NULL

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 5 ms

-- reject records for Manager not in Hierarchy table

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [RejectReason]= N'Manager Not found in Hierarchy table.'

WHERE EmpMgrID NOT IN

(SELECT DISTINCT EMP\_ID FROM [EC].[Employee\_Hierarchy])

AND [RejectReason]is NULL

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 5 ms

-- reject records for unexpected Type Of warning values

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [RejectReason]= N'Type of warning value invalid.'

WHERE [TypeOfWarning] NOT IN

('Verbal Warning', 'Written Warning', 'Final Written Warning')

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 5 ms

-- reject records for unexpected WarningType values

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [RejectReason]= N'Warning type value invalid.'

WHERE [WarningType] NOT IN

('ETS',

'Attendance',

'Conduct (including Call Avoidance)',

'Quality/Performance',

'Security or Privacy Issue',

'Other Policy (non-Security/Privacy)')

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 5 ms

-- reject records for future warning given dates

BEGIN

UPDATE [EC].[Warning\_History\_Stage]

SET [RejectReason]= N'Warning given date is in the future.'

WHERE [WarningGivenDate]> GETDATE()

OPTION (MAXDOP 1)

END

WAITFOR DELAY '00:00:00.03' -- Wait for 5 ms

-- Write rejected records to Rejected table.

BEGIN

INSERT INTO [EC].[Warning\_History\_Rejected]

([EmpID]

,[EmpSupID]

,[EmpRole]

,[TypeOfWarning]

,[WarningType]

,[WarningGivenDate]

,[Filename]

,[Reject\_Reason]

,[Reject\_Date])

SELECT S.[EmpID]

,S.[EmpSupID]

, S.[EmpRole]

, S.[TypeOfWarning]

, S.[WarningType]

, S.[WarningGivenDate]

, S.[Filename]

, S.[RejectReason]

,GETDATE()

FROM [EC].[Warning\_History\_Stage] S

WHERE S.[RejectReason] is not NULL

OPTION (MAXDOP 1)

END

END -- [EC].[sp\_InsertInto\_Warning\_History\_Rejected]

GO

#### Create Insert into warning log table Procedure

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- ====================================================================

-- Author: Susmitha Palacherla

-- Create Date: 12/5/2014

-- Description: This procedure inserts the warnings historical records into the Warning\_Log table.

-- The main attributes of the eCL are written to the Warning\_Log table.

-- The Coaching Reasons are written to the Warning\_Reasons Table.

-- =====================================================================

ALTER PROCEDURE [EC].[sp\_InsertInto\_Warning\_History]

@Count INT OUTPUT

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

DECLARE @maxnumID INT

-- Fetches the maximum WarningID before the insert.

SET @maxnumID = (SELECT IsNUll(MAX([WarningID]), 0) FROM [EC].[Warning\_Log])

-- Inserts records from the Warning\_History\_Stage table to the Warning\_Log Table

INSERT INTO [EC].[Warning\_Log]

([FormName]

,[ProgramName]

,[SourceID]

,[StatusID]

,[SiteID]

,[EmpLanID]

,[EmpID]

,[SubmitterID]

,[SubmittedDate]

,[WarningGivendate]

,[numReportID]

,[strReportCode]

,[ModuleID]

,[SupID]

,[MgrID]

)

SELECT DISTINCT

lower(wh.EmpLanID) [FormName],

wh.EmpProgram [ProgramName],

120 [SourceID],

1 [StatusID],

[EC].[fn\_intSiteIDFromEmpID](LTRIM(wh.EMPID))[SiteID],

lower(wh.EmpLanID) [EmpLanID],

wh.EMPID [EmpID],

ISNULL(NULLIF(wh.EMPSupID, ' '), '999999') [SubmitterID],

GetDate() [SubmittedDate],

wh.WarningGivenDate [WarningGivenDate],

wh.ReportID [numReportID],

wh.ReportCode [strReportCode],

CASE wh.EmpRole when 'C' THEN 1

WHEN 'S' THEN 2 ELSE -1 END [ModuleID],

ISNULL(NULLIF(wh.EmpSupID,' '), '999999') [SupID],

ISNULL(NULLIF(wh.EmpMgrID,' '), '999999') [MgrID]

FROM [EC].[Warning\_History\_Stage] wh

left outer join EC.Warning\_Log cf on wh.empid = cf.empid

and wh.ReportCode = cf.strReportCode and wh.ReportID = cf.numReportID

and wh.WarningGivenDate = cf.WarningGivenDate

where cf.empid is null and cf.WarningGivenDate is NULL and cf.strReportCode is NULL and cf.numReportID is NULL

OPTION (MAXDOP 1)

SELECT @Count =@@ROWCOUNT

WAITFOR DELAY '00:00:00:05' -- Wait for 5 ms

UPDATE [EC].[Warning\_Log]

SET [FormName] = 'eCL-'+[FormName] +'-'+ convert(varchar,WarningID)

where [FormName] not like 'eCL%'

OPTION (MAXDOP 1)

WAITFOR DELAY '00:00:00:05' -- Wait for 5 ms

-- Inserts records into Warning\_Log\_reason table for each record inserted into Warning\_log table.

INSERT INTO [EC].[Warning\_Log\_Reason]

([WarningID]

,[CoachingReasonID]

,[SubCoachingReasonID]

,[Value])

SELECT cf.[WarningID],

wh.[CoachingReasonID],

wh.[SubCoachingReasonID],

'Opportunity'

FROM [EC].[Warning\_History\_Stage] wh INNER JOIN [EC].[Warning\_Log] cf

ON (wh.[ReportCode] = cf.[strReportCode]

and wh.WarningGivenDate = cf.WarningGivenDate and wh.EmpID = cf.EmpID and wh.ReportID = cf.numReportID)

LEFT OUTER JOIN [EC].[Warning\_Log\_Reason] cr

ON cf.[WarningID] = cr.[WarningID]

WHERE cr.[WarningID] IS NULL

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SELECT @ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE()

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

IF ERROR\_NUMBER() IS NULL

RETURN 1

ELSE IF ERROR\_NUMBER() <> 0

RETURN ERROR\_NUMBER()

ELSE

RETURN 1

END CATCH

END -- sp\_InsertInto\_Warning\_History

GO