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



**Title: Employee\_Hierarchy\_SSIS\_DD**

**SSIS Detail Design Document**

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

|  |  |  |
| --- | --- | --- |
| Last Revision | Last Review | Description |
| 10/22/2018 |  | TFS 12438 - Accommodate middle name long values |

Prepared by: Susmitha Palacherla Date: 12/04/2013

Department, Location: Health Solutions Division

Approved by: Date:

**Change History Log**

| **Date** | **Change Description** | **Author** |
| --- | --- | --- |
| 12/04/2013 | Initial revision | Susmitha Palacherla |
| 06/30/2014 | Updated per scr 12982 to move the updates for migrated logs above the employee id to lanid process. | Susmitha Palacherla |
| 10/27/2014 | Updated per SCR 13624 to add step for Inactivating expired warnings logs. | Susmitha Palacherla |
| 01/19/2015 | Updated to import status and lanid from eWFM emp Info file to use in inactivation of ecls for employees in EA and Inactive status per scr 14072. | Susmitha Palacherla |
| 01/28/2015 | Additional update during SCR 14072. Adding a variable for eWFM info file directory to be able to use a test file from a test staging directory. | Susmitha Palacherla |
| 4/15/2016 | Added steps for integrating a supplementary file for HR employees per TFS 2332 | Susmitha Palacherla |
| 5/16/2016 | Added step to populate AT User Table per TFS 1709 | Susmitha Palacherla |
| 5/18/2017 | TFS 6623 – Check for data in files  2.1.1.1 –Variables – Added additional variables  2.1.1.2 – connection Manager – Added new SMTP connection  2.1.1.3 - Package content and workflow steps- Updated workflow diagram showing additional data checks  Updated Details section to reorder steps and add new steps | Susmitha Palacherla |
| 5/19/2017 | TFS 6623 – Check for data in files – Updates from V&V feedback. Replaced references to Vangent with GDIT. | Susmitha Palacherla |
| 8/14/2017 | TFS 7105 – Upgrade to SQL Server 2012 | Susmitha Palacherla |
| 9/25/2017 | TFS 8228 - Revise logic to flag re-used Employee Ids and clean up data. | Susmitha Palacherla |
| 11/15/2017 | TFS 8974 – Add 2 new fields from PeopleSoft and use Encrypted staging directories/files on database server share | Susmitha Palacherla |
| 11/17/2017 | TFS 8974 – Add 2 new fields from PeopleSoft and use Encrypted staging directories/files on database server share – Fixed type per V&V feedback | Susmitha Palacherla |
| 10/22/2018 | TFS 12438 - Accommodate middle name long values. Added new screenshot for Data Conversion task. | Susmitha Palacherla |

Contents

[1. Overview 4](#_Toc482950185)

[1.1 Project Description 4](#_Toc482950186)

[1.2 Document Scope 4](#_Toc482950187)

[1.3 Source Files 4](#_Toc482950188)

[1.4 Module List 4](#_Toc482950189)

[1.5 Software and Hardware Interfaces 4](#_Toc482950190)

[1.5.1 Software 4](#_Toc482950191)

[1.5.2 Hardware 4](#_Toc482950192)

[1.6 Users 4](#_Toc482950193)

[2. Details 5](#_Toc482950194)

[3.1 Source Files 5](#_Toc482950197)

[3.1.1 PS\_Employee\_Information\_mmddyyyy.csv 5](#_Toc482950198)

[3.1.2 Employee\_Information\_WithProgram.csv 5](#_Toc482950199)

[3.1.3 HR\_Employee\_Information.csv 5](#_Toc482950200)

[3.2 Module Details 6](#_Toc482950201)

[3.2.1 SQL agent job 6](#_Toc482950202)

[3.2.2 SSIS Package 6](#_Toc482950203)

[3.2.3 Tables 27](#_Toc482950209)

[3.2.4 Procedures 27](#_Toc482950210)

# Overview

## Project Description

The eCoaching Log (eCL) is an internal GDIT process designed for the entry and storage of the coaching notes that supervisors, managers, Advanced Resolution Center (ARC) CSRs and quality leads make regarding CSR performance. These notes are gathered through escalations, floor walking, training, quality monitors. It also provides a repository of coaching notes to generate reports for management.

## Document Scope

This document describes the process for maintaining the Employee Hierarchy table. It describes the source files and the workflow to maintain the user table.

## Source Files

* PS\_Employee\_Information\_mmddyyyy.csv
* Employee\_Information\_WithProgram.csv
* HR\_Employee\_Information.csv

## Module List

* SQL agent job
* SSIS Package
* Tables
* Procedures

## 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

* N/A

# Details



## Source Files

### PS\_Employee\_Information\_mmddyyyy.csv

* Description: This is an Employee file from Peoplesoft.
* Used for capturing Employee attributes like Employee ID, Name, Lan ID, Location, Job Code and Manager(Supervisor’s supervisor) and last hire date and Active Flag.
* Source system: PeopleSoft
* Staging location: [\\f3420-ecldbp01\data\coaching\hrinfo\Encryption\_In\](file:///\\f3420-ecldbp01\data\coaching\hrinfo\Encryption_In\)
* File name: PS\_Employee\_Information\_mmddyyyy.csv
* Encrypted file directory: \\f3420-ecldbp01\data\coaching\hrinfo\Encryption\_Out\
* Encrypted file: PS\_Employee\_Information\_mmddyyyy.csv.zip.encrypt
* Frequency: Arrives daily at ~4:30 AM EST
* Load time: 7 PM EST (5:00 PM Database time)
* Destination Table: Employee\_Hierarchy

### Employee\_Information\_WithProgram.csv

* Description: This is an Employee file from Aspect eWFM. Used to capture the Supervisor for employees having a CSR job code and the program value. The program value should be associated with the call or activity and will be input from the web interface or provided in the Coaching requests from Quality system and Outliers feeds. The program value based on the agents 1MULTI value from eWFM will only be sued when a program cannot be identified for the call activity for a specific coaching log.
* Source system: Aspect eWFM
* Staging location: [\\f3420-ecldbp01\data\coaching\hrinfo\Encryption\_In\](file:///\\f3420-ecldbp01\data\coaching\hrinfo\Encryption_In\)
* File name: Employee\_information\_WithProgram.csv
* Encrypted file directory: \\f3420-ecldbp01\data\coaching\hrinfo\Encryption\_Out\
* Encrypted file: Employee\_Information\_WithProgram.csv.zip.encrypt
* Frequency: Arrived daily at ~6:00 AM EST
* Load time: 7 PM EST (5:00 PM Database time)
* Destination Table: EmpID\_To\_SupID\_Stage

### HR\_Employee\_Information.csv

* Description: This is a manually created file sourced from HCSD HR team.
* Used for capturing Employee attributes like Employee ID, Name, Lan ID, Location, Job Code and Manager(Supervisor’s supervisor) and last hire date and Active Flag for HCSD HR staff.
* Source system: NA
* Staging location: [\\f3420-ecldbp01\data\coaching\hrinfo\Encryption\_In\](file:///\\f3420-ecldbp01\data\coaching\hrinfo\Encryption_In\)
* File name: HR\_Employee\_Information.csv
* Encrypted file directory: \\f3420-ecldbp01\data\coaching\hrinfo\Encryption\_Out\
* Encrypted file: HR\_Employee\_Information.csv.zip.encrypt
* Frequency: Ad-hoc (when there are changes to HR Employee records)
* Load time: 7 PM EST (5:00 PM Database time)
* Destination Table: Employee\_Hierarchy

## Module Details

### SQL agent job

* Production Instance : F3420-ECLDBP01
* Production Job: CoachingHierarchyLoad
* Production Package: [\\F3420-ECLDBP01\ssis\Coaching\Packages\Employee\_Hierarchy.dtsx](file:///\\F3420-ECLDBP01\ssis\Coaching\Packages\Employee_Hierarchy.dtsx)
* Production Config File: Prod\_Employee\_Hierarchy.dtsConfig
* Owner: ecljobowner
* Run As: ECLProxy (ECL Credential using application service account VNGT\SVC-SQLECLP01
* Schedule: Daily 7:00 PM EST
* Destination File Location: N/A

### SSIS Package

* + - 1. **Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **DataType** | **Value** | **Evaluate as Expression** |
| Decrypt\_In | String | \\f3420-ecldbp01\data\coaching\HRInfo\Decrypt\_In\  \*set from config file | False |
| Decrypt\_Out | String | \\f3420-ecldbp01\data\coaching\HRInfo\Decrypt\_Out\  \*set from config file | False |
| Encrypt\_Out | String | \\f3420-ecldbp01\data\coaching\HRInfo\Encrypt\_Out\  \*set from config file | False |
| CCEmail | String | [Susmitha.palacherla@gdit.com](mailto:Susmitha.palacherla@gdit.com)  \*set from config file | False |
| FromEmail | String | [Susmitha.palacherla@gdit.com](mailto:Susmitha.palacherla@gdit.com)  \*set from config file | False |
| ToEmail | String | [Susmitha.palacherla@gdit.com](mailto:Susmitha.palacherla@gdit.com)  \*set from config file | False |
| FileBackup | String | \\f3420-ecldbp01\data\coaching\HRInfo\Backups\ \*set from config file | False |
| PSFile | String | PS\_Employee\_Information\_mmddyyyy.csv | True - "PS\_Employee\_Information\_" +  Right("0" + (DT\_STR,4,1252) DatePart("m",getdate()),2) +  Right("0" + (DT\_STR,4,1252) DatePart("d",getdate()),2) +  (DT\_STR,4,1252) DatePart("yyyy",getdate()) + ".csv" |
| PSFile | String | \\f3420-ecldbp01\data\coaching\HRInfo\Decrypt\_Out\PS\_Employee\_Information\_mmddyyyy.csv | @[User::Decrypt\_Out] + @[User::PSFile] |
| PSFileExists | Boolean | True/False  \*Checked from Script task | False |
| PSRecordCount | Int32 | 0  \*Checked from Row Count task | False |
| HRRecordCount | Int32 | 0  \*Checked from Row Count task | False |
| WFMRecordCount | Int32 | 0  \*Checked from Row Count task | False |
|  |  |  |  |

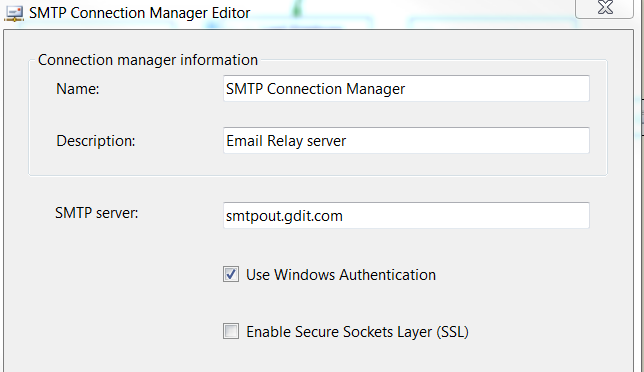
* + - 1. **Connection Manager Entries**
         1. Destinationdb
* Dev – F3420-ECLDBD01 DB – eCoachingdev
* Test - F3420-ECLDBT01 DB – eCoachingtest
* Prod - F3420-ECLDBP01 – eCoaching
  + - * 1. EmpFile
* \\f3420-ecldbp01\data\coaching\HRInfo\Decrypt\_Out \Employee\_Information\_WithProgram.csv
* Expression: @[User::WFMFileDir]+"Employee\_Information\_WithProgram.csv"
  + - * 1. PSFile
* [\\f3420-ecldbp01\data\coaching\HRInfo\Decrypt\_Out \PS\_Employee\_Information\_mmddyyyy.csv](file:///\\f3420-ecldbp01\data\coaching\HRInfo\Decrypt_Out%20\PS_Employee_Information_mmddyyyy.csv%20)
* Expression: @[User:: Decrypt\_Out] + "PS\_Employee\_Information\_" +

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

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

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

* + - * 1. HRFile
* [\\f3420-ecldbp01\data\coaching\HRInfo\Decrypt\_Out \HR\_Employee\_Information.csv](file:///\\f3420-ecldbp01\data\coaching\HRInfo\Decrypt_Out%20\HR_Employee_Information.csv)
* Expression: @[User::FileDir] + "HR\_Employee\_Information.csv"
  + - * 1. SMTP Connection Manager

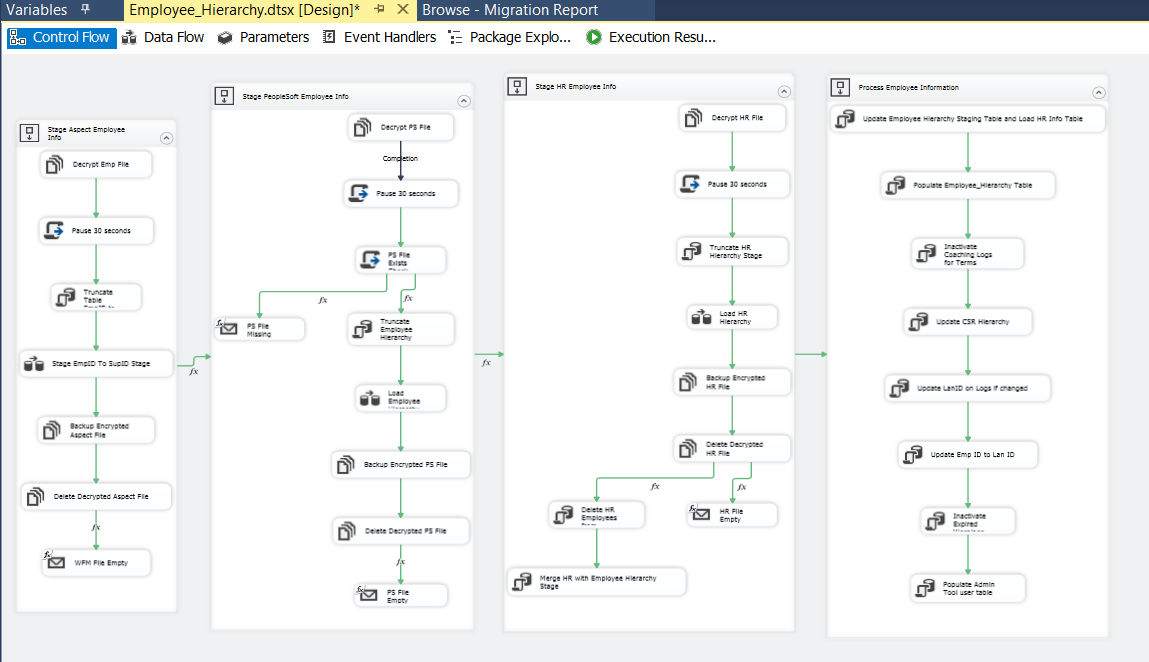


* + - * 1. Decrypt\_In
* [\\f3420-ecldbp01\data\coaching\HRInfo\Decrypt\_In\](file:///\\f3420-ecldbp01\data\coaching\HRInfo\Decrypt_In\)
* Expression: @[User::Decrypt\_In]
  + - * 1. EmpFile\_Encrypt\_Out
* [\\f3420-ecldbp01\data\coaching\HRInfo\Encrypt\_Out\Employee\_Information\_WithProgram.csv.zip.encrypt](file:///\\f3420-ecldbp01\data\coaching\HRInfo\Encrypt_Out\Employee_Information_WithProgram.csv.zip.encrypt)
* Expression: @[User::Encrypt\_Out] + "Employee\_Information\_WithProgram.csv.zip.encrypt"
  + - * 1. PSFile\_Encrypt\_Out
* [\\f3420-ecldbp01\data\coaching\HRInfo\Encrypt\_Out\PS\_Employee\_Information\_mmddyyyy.csv.zip.encrypt](file:///\\f3420-ecldbp01\data\coaching\HRInfo\Encrypt_Out\PS_Employee_Information_mmddyyyy.csv.zip.encrypt)
* Expression: @[User::Encrypt\_Out]+"PS\_Employee\_Information\_" +

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

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

* + - * 1. HRFile\_Encrypt\_Out
* [\\f3420-ecldbp01\data\coaching\HRInfo\Encrypt\_Out\HR\_Employee\_Information.csv.zip.encrypt](file:///\\f3420-ecldbp01\data\coaching\HRInfo\Encrypt_Out\HR_Employee_Information.csv.zip.encrypt)
* Expression: @[User::Encrypt\_Out] + "HR\_Employee\_Information.csv.zip.encrypt"
  + - 1. **Package Content**



**Workflow**

**Summary of Steps**

**Container 1: Stage Aspect Employee Info**

1. File System Task – Decrypt Emp File
2. Script Task – Pause 30 seconds
3. SQL Task – Truncate EmP ID To sup ID Staging Table
4. DFT – Stage WFM emp Info File data into Emp ID To Sup ID Table
5. File System Task – Backup Encrypted Aspect File
6. File System Task – Delete Decrypted Aspect File
   * IF WFM File empty – Send Mail Task. End Process.
   * IF WFM File not empty – Go to container 2

|  |  |
| --- | --- |
| **Container 1** | **Stage Aspect Employee Info** |
|  |  |
|  |  |
|  |  |
| **Step 1** | File Task: Decrypt Emp File |
|  |  |
|  |  |
| Step 2 | Script Task: Pause 30 secs |
|  |  |
| Step 3 | SQL Task : Truncate Table [EC].[EmpID\_To\_SupID\_Stage] |
| Step 4 | DFT: Load Table [EC].[EmpID\_To\_SupID\_Stage]    Data conversion    OLE DB Destination editor |
|  |  |
| Step 5 | Backup Encrypted Aspect File |
|  |  |
| Step 6 | Delete Decrypted Aspect file |
|  |  |
|  | |  |  | | --- | --- | | **Precedence Constraint**  Evaluation Operation: Expression and Constraint  Value: Success  Expression: @[User::WFMRecordCount]>0  Multiple Constraints: Logical AND  Go to Container 2 | **Precedence Constraint**  Evaluation Operation: Expression and Constraint  Value: Success  Expression: @[User::WFMRecordCount]==0  Multiple Constraints: Logical AND  Go to Step 7 | |
|  |  |
| Step 7 | Mail Task: WFM File Empty |
|  |  |

**Container 2: Stage PeopleSoft Employee Info**

1. File System Task – Decrypt PS File
2. Script Task – Pause 30 seconds
3. Script Task – PS File Exists Check
   * IF PS File Missing – Send Mail Task. End process.
   * IF PS File Not Missing – Go to Step 4.
4. SQL Task – Truncate Employee Hierarchy Staging Table
5. DFT – Load Employee Hierarchy Table
6. File System Task – Backup Encrypted PS File
7. File System Task – Delete Decrypted PS File
   * IF PS File empty – Send Mail Task. End Process.
   * IF PS File not empty – Go to container 3

|  |  |
| --- | --- |
| **Container 2** | **Stage PeopleSoft Employee Info** |
|  |  |
|  |  |
|  |  |
| **Step 1** | File Task: Decrypt PSFile |
|  |  |
|  |  |
| Step 2 | Script Task: Pause 30 secs |
|  |  |
| Step 3 | Script Task: PS File Exists Check    public void Main()  {  // TODO: Add your code here  string file = Dts.Variables["User::PSFileName"].Value.ToString();  Dts.Variables["User::PSFileExists"].Value = File.Exists(file);  Dts.TaskResult = (int)ScriptResults.Success;  } |
|  |  |
|  | |  |  | | --- | --- | | **Precedence Constraint**  Evaluation Operation: Expression and Constraint  Value: Success  Expression: @[User::PSFileExists]== True  Multiple Constraints: Logical AND  Go to step 4 | **Precedence Constraint**  Evaluation Operation: Expression and  Constraint  Value: Success  Expression: @[User::PSFileExists]==  False  Multiple Constraints: Logical AND  Go to Step 3a | |
|  |  |
| Step 3a | Mail Task: PS File Missing |
|  |  |
| Step 4 | SQL Task : Truncate Table [EC].[Employee\_Hierarchy\_Stage]  IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[EC].[Employee\_Hierarchy\_Stage]') AND type in (N'U'))  Truncate TABLE [EC].[Employee\_Hierarchy\_Stage] |
|  |  |
|  |  |
| Step 5 | DFT: Load Table [EC].[Employee\_Hierarchy\_Stage]    Data conversion    Derived Columns:    FullPrimaryName: (ISNULL([Copy of last\_name]) ? "" : [Copy of last\_name]) + ", " + (ISNULL([Copy of first\_name]) ? "" : [Copy of first\_name]) + " " + (ISNULL([Copy of middle\_name]) ? "" : [Copy of middle\_name])  FullPreferredName:  (ISNULL([Copy of pref\_name\_last]) ? "" : [Copy of pref\_name\_last]) + ", " + (ISNULL([Copy of pref\_name\_first]) ? "" : [Copy of pref\_name\_first]) + " " + (ISNULL([Copy of pref\_name\_mi]) ? "" : [Copy of pref\_name\_mi])  OLE DB Destination editor |
|  |  |
| Step 6 | Backup Encrypted PS File |
|  |  |
| Step 7 | Delete Decrypted PS file |
|  |  |
|  | |  |  | | --- | --- | | **Precedence Constraint**  Evaluation Operation:  Expression and Constraint  Value: Success  Expression:  @[User::PSRecordCount]>0  Multiple Constraints: Logical AND  Go to Container 3 | **Precedence Constraint**  Evaluation Operation:  Expression and Constraint  Value: Success  Expression:  @[User::PSRecordCount]==0  Multiple Constraints: Logical AND  Go to Step 8 | |
|  |  |
| Step 8 | Mail Task: PS File Empty |
|  |  |

**Container 3: Stage HR Employee Info**

1. File System Task – Decrypt HR File
2. Script Task – Pause 30 seconds
3. SQL Task – Truncate HR Hierarchy Staging Table
4. DFT – Load HR Hierarchy Staging Table
5. File System Task – Backup Encrypted Aspect File
6. File System Task – Delete Decrypted Aspect File
   * IF HR File empty – Send Mail Task. End Process.
   * IF HR File not empty – Go to Step 7
7. SQL Task – Delete HR Employees from Emp Hierarchy Staging Table
8. SQL Task – Merge HR records from HR staging table into Emp Hierarchy Staging Table

|  |  |
| --- | --- |
| **Container 3** | **Stage HR Employee Info** |
|  |  |
|  |  |
|  |  |
| **Step 1** | File Task: Decrypt HR File |
|  |  |
|  |  |
| Step 2 | Script Task: Pause 30 secs |
|  |  |
| Step 3 | SQL Task : Truncate Table [EC].[HR\_Hierarchy\_Stage]  IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[EC].[HR\_Hierarchy\_Stage]') AND type in (N'U'))  Truncate TABLE [EC].[HR\_Hierarchy\_Stage] |
| Step 4 | DFT: Load Table [EC].[HR\_Hierarchy\_Stage]    Data conversion    Derived column:    OLE DB Destination editor |
|  |  |
| Step 5 | Backup Encrypted HR File |
|  |  |
| Step 6 | Delete Decrypted HR file |
|  |  |
|  | |  |  | | --- | --- | | **Precedence Constraint**  Evaluation Operation: Expression and Constraint  Value: Success  Expression: @[User::HRRecordCount]>0  Multiple Constraints: Logical AND  Go to Step 7 | **Precedence Constraint**  Evaluation Operation:  Expression and Constraint  Value: Success  Expression:  @[User::HRRecordCount]==0  Multiple Constraints: Logical AND  Go to Step 6a | |
|  |  |
| Step 6a | **Mail task: HR File Empty** |
|  |  |
| Step 7 | SQL Task: Delete HR Employees from Employee Hierarchy Staging Table  SQL Statement:  DELETE FROM [EC].[Employee\_Hierarchy\_Stage]  WHERE Emp\_JOB\_Code like 'WH%' |
|  |  |
| Step 8 | SQL Task: Merge HR with Employee Hierarchy Stage  SQL Statement:  Exec [EC].[sp\_Merge\_HR\_Employee\_Hierarchy\_Stage] |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Workflow**

**Summary of Steps**

**Container 4: Process Employee Information**

1. SQL task – Update Employee Hierarchy Staging table.
2. SQL task – Populate Employee Hierarchy table
3. SQL task – Inactivate Coaching logs for termed users
4. SQL task – Update CSR Hierarchy Table
5. SQL task – Update Coaching Logs for migrated users
6. SQL task – Update Employee ID To Lan ID table
7. SQL task – Inactivate expired warning logs
8. Populate AT User Table

|  |  |
| --- | --- |
| **Container 4** | **Process Employee Information** |
|  |  |
|  |  |
|  |  |
| Step 1 | SQL Task: Update Employee Hierarchy Staging Table and Load HR Info Table  SQL Statement:  EXEC [EC].[sp\_Update\_Employee\_Hierarchy\_Stage] |
|  |  |
| Step 2 | SQL Task: Populate Employee\_Hierarchy Table  SQL Statement:  Exec [EC].[sp\_Populate\_Employee\_Hierarchy] |
|  |  |
| Step 3 | SQL Task: Inactivate Coaching Logs for Terms  SQL Statement:  Exec [EC].[sp\_InactivateCoachingLogsForTerms] |
|  |  |
| Step 4 | SQL Task: Update CSR Hierarchy  SQL Statement:  Exec [EC].[sp\_Update\_CSR\_Hierarchy] |
|  |  |
| Step 5 | SQL Task: Update LanID on Logs if changed  SQL Statement:  Exec [EC].[sp\_Update\_Migrated\_User\_Logs] |
|  |  |
| Step 6 | SQL Task: Update Emp ID to Lan ID  SQL Statement:  Exec [EC].[sp\_Update\_EmployeeID\_To\_LanID] |
|  |  |
| Step 7 | SQL Task: Inactivate Expired Warnings  SQL Statement:  Exec [EC].[sp\_InactivateExpiredWarningLogs] |
|  |  |
| Step 8 | SQL Task: Populate Admin Tool user table  SQL Statement:  Exec [EC].[sp\_AT\_Populate\_User] |
|  |  |
|  |  |

### Tables

|  | **Table Name** | **Table Description** |
| --- | --- | --- |
| 1. | Employee\_Hierarchy | Table for storing Employee Information. |
| 2. | Employee\_Hierarchy\_Stage | Table used for staging the Employee Information from PeopleSoft during the Employee Hierarchy load process. |
| 3. | EmpID\_To\_SupID\_Stage | Table used for staging the Sup Ids from Aspect eWFM during the Employee Hierarchy load process. |
| 4. | EmployeeID\_To\_LanID | Table used for storing Employee ID to Lan ID links over time. |
| 5. | CSR\_Hierarchy | Table used for storing CSR Hierarchy records over time. |
| 6. | HR\_Hierarchy\_Stage | Table used to stage HR Hierarchy records |
| 7. | AT\_User | Table for storing Admin Tool users |
| 8. | Employee\_Ids\_With\_Prefixes | Table used for storing employee Ids that have a re-used numeric employee ID and need the alpha part retained. |

Table creates in TFS Source control at: \cms\eCoaching\_V2\Code\DB\Tables\ CCO\_eCoaching\_EmployeeHierarchy\_Load\_Tables\_Create.sql

### Procedures

|  | **Procedure Name** | **Usage** |
| --- | --- | --- |
| 1. | sp\_Populate\_Employee\_Hierarchy | Load: Employee Data |
| 2. | sp\_Update\_CSR\_Hierarchy | Load: Employee Data |
| 3. | sp\_Update\_Employee\_Hierarchy\_Stage | Load: Employee Data |
| 4. | sp\_Update\_EmployeeID\_To\_LanID | Load: Employee Data |
| 5. | sp\_Update\_Migrated\_User\_Logs | Housekeeping: Migrated user log update |
| 6. | sp\_Merge\_HR\_Employee\_Hierarchy\_Stage | Load: Employee Data |
| 7. | sp\_AT\_Populate\_User | Load: AT user Data |

Stored Procedure creates in TFS Source control at: \cms\eCoaching\_V2\Code\DB\ Stored Procedures\\*.sql