**CCO\_eCoaching\_V2\_DB\_Change\_Implementations**

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

|  |  |  |
| --- | --- | --- |
| Last Revision | Last Review | Description |
| 07/10/2014 |  | Initial Draft |

Revision Chart

| Version | Primary Author(s) | Description of Version | Date Added |
| --- | --- | --- | --- |
| 1.0 | Susmitha Palacherla | SCR 12978 – Create SP [EC].[sp\_Check\_AppRole] | 06/26/2014 |
| 2.0 | Susmitha Palacherla | SCR 12897 – Additional Sources [EC].[DIM\_Source] Table | 06/26/2014 |
| 3.0 | Susmitha Palacherla | SCR 12963 – Fix Loaded count value in Quality file list table | 07/03/2014 |
| 4.0 | Susmitha Palacherla | SCR 13071 – Create Indexes and Stats for performance tuning | 07/07/2014 |
| 5.0 | Susmitha Palacherla | SCR 12982 – Update Employee Hierarchy Load process | 07/10/2014 |
| 6.0 | Susmitha Palacherla | SCR 12930 – Add VerintFormName to Review SP  SCR 13054 – Import VerintFormName  SCR 13138 – Restrict Quality eCLs loaded from UI  SCR 13213 – Map Coaching Reason ID for Outliers to 9 | 07/22/14 |

Contents

[1.0 Implementation SCR 12978 – Create SP [EC].[sp\_Check\_AppRole] 1](#_Toc393959223)

[2.0 Implementation SCR 12897 – Additional Sources in [EC].[DIM\_Source] Table 2](#_Toc393959224)

[3.0 Implementation SCR 12963 – Fix Loaded count value in Quality file list table 2](#_Toc393959225)

[4.0 Implementation SCR 13071 – Create Indexes and Stats on 6](#_Toc393959226)

[5.0 Implementation SCR 12982 – Update Employee Hierarchy Load process 8](#_Toc393959227)

[6.0 Implementation of SCRs 12930, 13054 and 13138, 13213 9](#_Toc393959228)

# Implementation SCR 12978 – Create SP [EC].[sp\_Check\_AppRole]

Date: 06/26/2014

Implementer: Jourdain Augustin

Description: Installs Stored procedure [EC].[sp\_Check\_AppRole]

Target Environment: eCoaching Database on VDENSSDBP07\SCORP01 Instance

Installation procedure: Run below sql in SQL Environment noted above.

--<Begin Implementation 1>

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Check\_AppRole'

)

DROP PROCEDURE [EC].[sp\_Check\_AppRole]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

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

-- Author: Susmitha Palacherla

-- Create Date: 09/21/2012

-- Description: This procedure returns whether the User is an admin or not.

-- Last Update: 09/18/2013

-- Last Modified By: Susmitha Palacherla

-- Updated per SCR 10617 to return 'N' for all Inactive users.

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

CREATE PROCEDURE [EC].[sp\_Check\_AppRole]

(

@nvcLANID Nvarchar(30)

)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max)

SET @nvcSQL = 'SELECT [ISADMIN]=

CASE WHEN End\_Date = ''99991231'' THEN [ISADMIN]

ELSE ''N''

END

FROM [EC].[Historical\_Dashboard\_ACL]

WHERE [User\_LanID] = '''+@nvcLANID+''''

EXEC (@nvcSQL)

END

GO

--</End Implementation 1>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Implementation SCR 12897 – Additional Sources in [EC].[DIM\_Source] Table

Date: 06/26/2014

Implementer: Jourdain Augustin

Description: Adds/Updates Sources in [EC].[DIM\_Source] Table

Target Environment: eCoaching Database on VDENSSDBP07\SCORP01 Instance

Installation procedure: Run below sql in SQL Environment noted above.

--<Begin Implementation 2>

Update [EC].[DIM\_Source]

set [SubCoachingSource]='CMS Reported Item'

where [SourceID] in (106, 206)

GO

INSERT INTO [EC].[DIM\_Source]

([SourceID]

,[CoachingSource]

,[SubCoachingSource])

VALUES

(118, 'Direct', 'Internal CCO Reporting'),

(218, 'Indirect','Internal CCO Reporting')

GO

--</End Implementation 2>

# Implementation SCR 12963 – Fix Loaded count value in Quality file list table

Date: 07/07/2014

Implementer: Lex Mishra

Description: Correctly populates loaded count value in Quality file list table

Target Environment: eCoaching Database on VDENSSDBP07\SCORP01 Instance

Installation procedure: Run below sql in SQL Environment noted above.

Summary

1. Stage updated IQS\_Coaching.dtsx Package
2. Install Updated stored procedure [EC].[sp\_InsertInto\_Coaching\_Log\_Quality]

Details

1. Stage updated IQS\_Coaching.dtsx Package

Download ssis package IQS\_Coaching.dtsx (VM version 1.1) from \cms\eCoaching\_V2\ETL\

In version manager and copy to [\\vdenssdbp07\scorecard-ssis\Coaching](file:///\\vdenssdbp07\scorecard-ssis\Coaching)

Overwriting existing package

1. Install Updated stored procedure [EC].[sp\_InsertInto\_Coaching\_Log\_Quality]by running below code in ssms in a window pointing to eCoaching database on vdenssdbp07\scorp01.

--<Begin Step 2>

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_InsertInto\_Coaching\_Log\_Quality'

)

DROP PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log\_Quality]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

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

-- Author: Susmitha Palacherla

-- Create Date: 02/23/2014

-- Description: This procedure inserts the Quality scorecards into the Coaching\_Log table.

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

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

-- Modified Date: Per SCR 12963. Added Output parameter @Count to get the @@ROWCOUNT on insert for

-- populating in File List table.

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

CREATE PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log\_Quality]

@Count INT OUTPUT

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

DECLARE @maxnumID INT

-- Fetches the maximum CoachingID before the insert.

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

-- Inserts records from the Quality\_Coaching\_Stage table to the Coaching\_Log Table

INSERT INTO [EC].[Coaching\_Log]

([FormName]

,[ProgramName]

,[SourceID]

,[StatusID]

,[SiteID]

,[CSR]

,[CSRID]

,[SubmitterID]

,[EventDate]

,[isAvokeID]

,[isNGDActivityID]

,[isUCID]

,[isVerintID]

,[VerintID]

,[VerintEvalID]

,[Description]

,[SubmittedDate]

,[StartDate]

,[isCSE]

,[isCSRAcknowledged]

)

SELECT DISTINCT

lower(csr.Emp\_LanID) [FormName],

CASE qs.Program

WHEN NULL THEN csr.Emp\_Program

WHEN '' THEN csr.Emp\_Program

ELSE qs.Program END [ProgramName],

211 [SourceID],

[EC].[fn\_strStatusIDFromIQSEvalID](qs.CSE, qs.Oppor\_Rein )[StatusID],

[EC].[fn\_intSiteIDFromEmpID](LTRIM(qs.User\_EMPID))[SiteID],

lower(csr.Emp\_LanID) [CSR],

qs.User\_EMPID [CSRID],

qs.Evaluator\_ID [SubmitterID],

qs.Call\_Date [EventDate],

0 [isAvokeID],

0 [isNGDActivityID],

0 [isUCID],

1 [isVerintID],

qs.Journal\_ID [VerintID],

qs.Eval\_ID [VerintEvalID],

EC.fn\_nvcHtmlEncode(qs.Summary\_CallerIssues)[Description],

GetDate() [SubmittedDate],

qs.Eval\_Date [StartDate],

CASE WHEN qs.CSE = '' THEN 0

ELSE 1 END [isCSE],

0 [isCSRAcknowledged]

FROM [EC].[Quality\_Coaching\_Stage] qs

join EC.Employee\_Hierarchy csr on qs.User\_EMPID = csr.Emp\_ID

left outer join EC.Coaching\_Log cf on qs.[Eval\_ID] = cf.[VerintEvalID]

where cf.[VerintEvalID] is null

OPTION (MAXDOP 1)

SELECT @Count =@@ROWCOUNT

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

UPDATE [EC].[Coaching\_Log]

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

where [FormName] not like 'eCL%'

OPTION (MAXDOP 1)

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

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

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

([CoachingID]

,[CoachingReasonID]

,[SubCoachingReasonID]

,[Value])

SELECT cf.[CoachingID],

10,

42,

qs.[Oppor\_Rein]

FROM [EC].[Quality\_Coaching\_Stage] qs JOIN [EC].[Coaching\_Log] cf

ON qs.[Eval\_ID] = cf.[VerintEvalID] LEFT OUTER JOIN [EC].[Coaching\_Log\_Reason] cr

ON cf.[CoachingID] = cr.[CoachingID]

WHERE cr.[CoachingID] 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\_Coaching\_Log\_Quality

GO

--</End Step 2>

This completes installation for SCR 12963.

# Implementation SCR 13071 – Create Indexes and Stats on

Date: 07/07/2014

Implementer: Lex Mishra

Description: Create Indexes and Stats on Coaching\_Log and employee\_Hierarchy

Target Environment: eCoaching Database on VDENSSDBP07\SCORP01 Instance

Installation procedure: Run below sql statements one at a time in SQL Environment noted above.

--Create Statistics

CREATE STATISTICS [\_dta\_stat\_2105058535\_4\_5\_8\_24] ON [EC].[Coaching\_Log]([SourceID], [StatusID], [CSRID], [SubmittedDate])

CREATE STATISTICS [\_dta\_stat\_69575286\_2\_1] ON [EC].[Employee\_Hierarchy]([Emp\_Name], [Emp\_ID])

CREATE STATISTICS [\_dta\_stat\_69575286\_1\_12\_2] ON [EC].[Employee\_Hierarchy]([Emp\_ID], [Sup\_LanID], [Emp\_Name])

CREATE STATISTICS [\_dta\_stat\_2105058535\_8\_5\_4] ON [EC].[Coaching\_Log]([CSRID], [StatusID], [SourceID])

CREATE STATISTICS [\_dta\_stat\_2105058535\_24\_8\_5\_4] ON [EC].[Coaching\_Log]([SubmittedDate], [CSRID], [StatusID], [SourceID])

CREATE STATISTICS [\_dta\_stat\_2105058535\_24\_4\_8] ON [EC].[Coaching\_Log]([SubmittedDate], [SourceID], [CSRID])

CREATE STATISTICS [\_dta\_stat\_69575286\_16\_1\_12\_2] ON [EC].[Employee\_Hierarchy]([Mgr\_Name], [Emp\_ID], [Sup\_LanID], [Emp\_Name])

CREATE STATISTICS [\_dta\_stat\_69575286\_2\_1\_12] ON [EC].[Employee\_Hierarchy]([Emp\_Name], [Emp\_ID], [Sup\_LanID])

CREATE STATISTICS [\_dta\_stat\_69575286\_1\_12] ON [EC].[Employee\_Hierarchy]([Emp\_ID], [Sup\_LanID])

CREATE STATISTICS [\_dta\_stat\_69575286\_2\_12] ON [EC].[Employee\_Hierarchy]([Emp\_Name], [Sup\_LanID])

CREATE STATISTICS [\_dta\_stat\_2105058535\_4\_2\_1\_5\_6\_8\_9] ON [EC].[Coaching\_Log]([SourceID], [FormName], [CoachingID], [StatusID], [SiteID], [CSRID], [SubmitterID])

CREATE STATISTICS [\_dta\_stat\_2105058535\_2\_8\_9\_1\_5\_6] ON [EC].[Coaching\_Log]([FormName], [CSRID], [SubmitterID], [CoachingID], [StatusID], [SiteID])

CREATE STATISTICS [\_dta\_stat\_2105058535\_8\_9] ON [EC].[Coaching\_Log]([CSRID], [SubmitterID])

CREATE STATISTICS [\_dta\_stat\_2105058535\_6\_2\_1\_5] ON [EC].[Coaching\_Log]([SiteID], [FormName], [CoachingID], [StatusID])

CREATE STATISTICS [\_dta\_stat\_2105058535\_1\_5\_6\_4\_8\_9] ON [EC].[Coaching\_Log]([CoachingID], [StatusID], [SiteID], [SourceID], [CSRID], [SubmitterID])

CREATE STATISTICS [\_dta\_stat\_2105058535\_2\_5] ON [EC].[Coaching\_Log]([FormName], [StatusID])

CREATE STATISTICS [\_dta\_stat\_2105058535\_2\_1\_5] ON [EC].[Coaching\_Log]([FormName], [CoachingID], [StatusID])

CREATE STATISTICS [\_dta\_stat\_2105058535\_5\_8\_24] ON [EC].[Coaching\_Log]([StatusID], [CSRID], [SubmittedDate])

CREATE STATISTICS [\_dta\_stat\_69575286\_1\_4] ON [EC].[Employee\_Hierarchy]([Emp\_ID], [Emp\_LanID])

--Create Indexes

CREATE NONCLUSTERED INDEX [\_dta\_index\_Coaching\_Log\_28\_2105058535\_\_K8\_K4\_K5\_K24\_2] ON [EC].[Coaching\_Log]

(

[CSRID] ASC,

[SourceID] ASC,

[StatusID] ASC,

[SubmittedDate] ASC

)

INCLUDE ( [FormName]) WITH (SORT\_IN\_TEMPDB = OFF, IGNORE\_DUP\_KEY = OFF, DROP\_EXISTING = OFF, ONLINE = OFF) ON [PRIMARY]

GO

CREATE NONCLUSTERED INDEX [\_dta\_index\_Employee\_Hierarchy\_28\_69575286\_\_K4\_K1\_2\_10\_16] ON [EC].[Employee\_Hierarchy]

(

[Sup\_LanID] ASC,

[Emp\_LanID] ASC,

[Emp\_ID] ASC

)

INCLUDE ( [Emp\_Name],

[Sup\_Name],

[Mgr\_Name]) WITH (SORT\_IN\_TEMPDB = OFF, IGNORE\_DUP\_KEY = OFF, DROP\_EXISTING = OFF, ONLINE = OFF) ON [PRIMARY]

GO

This completes installation for SCR 13071..

# Implementation SCR 12982 – Update Employee Hierarchy Load process

Date: 07/10/2014

Implementer: Lex Mishra

Description: Stages an updated Employee\_Hierarchy.dtsx package that runs the updates

For migrated users prior to attempting to run the updates to employee ID To Lan ID process.

Target Environment: eCoaching Database on VDENSSDBP07\SCORP01 Instance

Summary

1. Stage updated Employee\_Hierarchy.dtsx Package
2. Install Updated stored procedure [EC].[sp\_Update\_Migrated\_User\_Logs]

Details

1. Stage updated Employee\_Hierarchy.dtsx Package

Download ssis package Employee\_Hierarchy.dtsx (VM version 1.1) from \cms\eCoaching\_V2\ETL\

In version manager and copy to [\\vdenssdbp07\scorecard-ssis\Coaching](file:///\\vdenssdbp07\scorecard-ssis\Coaching)

Overwriting existing package

1. Install Updated stored procedure [EC].[sp\_Update\_Migrated\_User\_Logs] by running below code in ssms in a window pointing to eCoaching database on vdenssdbp07\scorp01.

--<Begin Step 2>

IF EXISTS (

SELECT \*

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE SPECIFIC\_SCHEMA = N'EC'

AND SPECIFIC\_NAME = N'sp\_Update\_Migrated\_User\_Logs'

)

DROP PROCEDURE [EC].[sp\_Update\_Migrated\_User\_Logs]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

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

-- Author: Susmitha Palacherla

-- Create date: 04/05/2014

-- Description: Updates historical Coaching logs for Migrated users

-- Last Modified By: Susmitha Palacherela

-- Last Modified Date: 06/30/2014

-- Updated per SCR 12982 to remove update to Employee ID To Lan ID table

-- until correct logic can be identified.

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

CREATE PROCEDURE [EC].[sp\_Update\_Migrated\_User\_Logs]

AS

BEGIN

-- Update CSR value in Coaching logs for migrated users

BEGIN

UPDATE [EC].[Coaching\_Log]

SET [CSR] = H.[Emp\_LanID]

FROM [EC].[Coaching\_Log]F JOIN [EC].[Employee\_Hierarchy]H

ON F.[CSRID] = H.[Emp\_ID]

WHERE F.[CSR] <> H.[Emp\_LanID]

OPTION (MAXDOP 1)

END

END -- [EC].[sp\_Update\_Migrated\_User\_Logs]

GO

--</End Step 2>

# Implementation of SCRs 12930, 13054 and 13138, 13213

Date: 07/22/2014

Implementer:

Description:

SCR 12930 – Add VerintFormName to Review SP

SCR 13054 – Import VerintFormName

SCR 13138 – Restrict Quality eCLs loaded from UI

SCR 13213 – Map Coaching Reason ID for Outliers to 9

Summary

1. Alter required tables to Add new column VerintFormName - 13054
2. Update necessary procedures to load and display new column – 13054/12930
3. Update procedure to restrict load of Quality logs submitted from UI - 13138
4. Stage updated Load Package – 13054
5. Update procedures for Outliers Load - 13213

Details

1. Alter required tables to Add new column VerintFormName

Run the Following commands in an SSMS window pointing to eCoaching database on

VDENSSDBP07\SCORP01

ALTER table [EC].[Quality\_Coaching\_Stage]

ADD VerintFormName nvarchar(50)

ALTER table [EC].[Quality\_Coaching\_Fact]

ADD VerintFormName nvarchar(50)

ALTER table [EC].[Quality\_Coaching\_Rejected]

ADD VerintFormName nvarchar(50)

ALTER table [EC].[Coaching\_Log]

ADD VerintFormName nvarchar(50)

1. Update necessary procedures to load and display new column

\*This step has 3 Procedures to run 2a, 2b and 2c.

Note the <Begin Step> and </End Step> Tags for each procedure.

--2a. <Begin step 2a>

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

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

-- Author: Susmitha Palacherla

-- Create Date: 02/23/2014

-- Description: This procedure inserts the Quality scorecards into the Coaching\_Log table.

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

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

-- Modified Date: 07/18/2014

-- Description: Updated per SCR 13054 to add additional column VerintFormName.

-- Updated per SCR 13138 to insert new records based on Journal ID and Submitter ID

-- If eCL already submitted for a Journal ID from web interface should not be inserted.

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

ALTER PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log\_Quality]

@Count INT OUTPUT

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

DECLARE @maxnumID INT

-- Fetches the maximum CoachingID before the insert.

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

-- Inserts records from the Quality\_Coaching\_Stage table to the Coaching\_Log Table

INSERT INTO [EC].[Coaching\_Log]

([FormName]

,[ProgramName]

,[SourceID]

,[StatusID]

,[SiteID]

,[CSR]

,[CSRID]

,[SubmitterID]

,[EventDate]

,[isAvokeID]

,[isNGDActivityID]

,[isUCID]

,[isVerintID]

,[VerintID]

,[VerintEvalID]

,[Description]

,[SubmittedDate]

,[StartDate]

,[isCSE]

,[isCSRAcknowledged]

,[VerintFormName]

)

SELECT DISTINCT

lower(csr.Emp\_LanID) [FormName],

CASE qs.Program

WHEN NULL THEN csr.Emp\_Program

WHEN '' THEN csr.Emp\_Program

ELSE qs.Program END [ProgramName],

211 [SourceID],

[EC].[fn\_strStatusIDFromIQSEvalID](qs.CSE, qs.Oppor\_Rein )[StatusID],

[EC].[fn\_intSiteIDFromEmpID](LTRIM(qs.User\_EMPID))[SiteID],

lower(csr.Emp\_LanID) [CSR],

qs.User\_EMPID [CSRID],

qs.Evaluator\_ID [SubmitterID],

qs.Call\_Date [EventDate],

0 [isAvokeID],

0 [isNGDActivityID],

0 [isUCID],

1 [isVerintID],

qs.Journal\_ID [VerintID],

qs.Eval\_ID [VerintEvalID],

EC.fn\_nvcHtmlEncode(qs.Summary\_CallerIssues)[Description],

GetDate() [SubmittedDate],

qs.Eval\_Date [StartDate],

CASE WHEN qs.CSE = '' THEN 0

ELSE 1 END [isCSE],

0 [isCSRAcknowledged],

qs.VerintFormname [verintFormName]

FROM [EC].[Quality\_Coaching\_Stage] qs

join EC.Employee\_Hierarchy csr on qs.User\_EMPID = csr.Emp\_ID

left outer join EC.Coaching\_Log cf on qs.[Journal\_ID] = cf.[VerintID]

AND qs.Evaluator\_ID = cf.[SubmitterID]

where cf.[VerintID] is null and cf.[SubmitterID] is NULL

OPTION (MAXDOP 1)

SELECT @Count =@@ROWCOUNT

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

UPDATE [EC].[Coaching\_Log]

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

where [FormName] not like 'eCL%'

OPTION (MAXDOP 1)

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

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

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

([CoachingID]

,[CoachingReasonID]

,[SubCoachingReasonID]

,[Value])

SELECT cf.[CoachingID],

10,

42,

qs.[Oppor\_Rein]

FROM [EC].[Quality\_Coaching\_Stage] qs JOIN [EC].[Coaching\_Log] cf

ON qs.[Eval\_ID] = cf.[VerintEvalID]

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

ON cf.[CoachingID] = cr.[CoachingID]

WHERE cr.[CoachingID] 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\_Coaching\_Log\_Quality

GO

--</End step 2a>

--2b. <Begin step 2b>

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

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

-- Author: Susmitha Palacherla

-- Create Date: 05/14/2014

-- Description: This procedure updates the existing records in the Quality Fact table

-- and inserts new records.

-- Modified Date: 07/18/2014

-- Description: Updated per SCR 13054 to add additional column VerintFormName.

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

ALTER PROCEDURE [EC].[sp\_Update\_Quality\_Fact]

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

-- Update existing records in quality Fact Table

UPDATE [EC].[Quality\_Coaching\_Fact]

SET [Summary\_CallerIssues] = S.[Summary\_CallerIssues],

[Date\_Inserted]=S.[Date\_Inserted],

[Oppor\_Rein]= S.[Oppor\_Rein]

FROM [EC].[Quality\_Coaching\_Stage]S INNER JOIN [EC].[Quality\_Coaching\_Fact]F

ON S.[Eval\_ID] = F.[Eval\_ID]

AND S.[Journal\_ID] = F.[Journal\_ID]

WHERE F.[Eval\_ID] is NOT NULL

OPTION (MAXDOP 1)

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

-- Append new records to Quality Fact Table

INSERT INTO [EC].[Quality\_Coaching\_Fact]

([Eval\_ID]

,[Eval\_Date]

,[Eval\_Site\_ID]

,[User\_ID]

,[User\_EMPID]

,[User\_LANID]

,[SUP\_ID]

,[SUP\_EMPID]

,[MGR\_ID]

,[MGR\_EMPID]

,[Journal\_ID]

,[Call\_Date]

,[Summary\_CallerIssues]

,[Coaching\_Goal\_Discussion]

,[CSE]

,[Evaluator\_ID]

,[Program]

,[Source]

,[Oppor\_Rein]

,[Date\_Inserted]

,[VerintFormName])

SELECT

S.[Eval\_ID]

,S.[Eval\_Date]

,S.[Eval\_Site\_ID]

,S.[User\_ID]

,S.[User\_EMPID]

,S.[User\_LANID]

,S.[SUP\_ID]

,S.[SUP\_EMPID]

,S.[MGR\_ID]

,S.[MGR\_EMPID]

,S.[Journal\_ID]

,S.[Call\_Date]

,S.[Summary\_CallerIssues]

,S.[Coaching\_Goal\_Discussion]

,S.[CSE]

,S.[Evaluator\_ID]

,S.[Program]

,S.[Source]

,S.[Oppor\_Rein]

,S.[Date\_Inserted]

,S.[VerintFormName]

FROM

[EC].[Quality\_Coaching\_Stage] S LEFT OUTER JOIN

[EC].[Quality\_Coaching\_Fact] F ON

S.[Eval\_ID] = F.[Eval\_ID]

WHERE(F.[Eval\_ID] 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\_Update\_Quality\_Fact

GO

--</End step 2b>

--2c. <Begin step 2c>

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

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

-- Author: Susmitha Palacherla

-- Create Date: 11/16/11

-- Description: This procedure selects an e-Coaching record from the Coaching\_Log table for review.

-- Last Modified by: Susmitha Palacherla

-- Last Update: 07/18/2014

-- Updated per SCR 12930 to add isUCID and VerintFormName to the Return

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

ALTER PROCEDURE [EC].[sp\_SelectReviewFrom\_Coaching\_Log] @strFormIDin nvarchar(50)

AS

BEGIN

DECLARE

@nvcSQL nvarchar(max)

SET @nvcSQL = 'SELECT cl.CoachingID numID,

cl.FormName strFormID,

sc.CoachingSource strFormType,

s.Status strFormStatus,

cl.EventDate EventDate,

cl.CoachingDate CoachingDate,

sh.Emp\_LanID strSubmitter,

sh.Emp\_Name strSubmitterName,

sh.Emp\_Email strSubmitterEmail,

cl.CSR strCSR,

eh.Emp\_Name strCSRName,

eh.Emp\_Email strCSREmail,

st.City strCSRSite,

eh.Sup\_LanID strCSRSup,

eh.Sup\_Name strCSRSupName,

eh.Sup\_Email strCSRSupEmail,

eh.Mgr\_LanID strCSRMgr,

eh.Mgr\_Name strCSRMgrName,

eh.Mgr\_Email strCSRMgrEmail,

sc.SubCoachingSource strSource,

CL.isUCID isUCID,

CL.UCID strUCID,

CL.isVerintID isVerintMonitor,

CL.VerintID strVerintID,

CL.VerintFormName VerintFormName,

CL.isAvokeID isBehaviorAnalyticsMonitor,

CL.AvokeID strBehaviorAnalyticsID,

CL.isNGDActivityID isNGDActivityID,

CL.NGDActivityID strNGDActivityID,

CC.AHT,

CC.ARC "ARC Issue",

CC.Attendance,

CC.BCCPPI "BCC Process Procedure Issues",

CC.CCI "Current Coaching Initiative",

CC.CSE "Customer Service Escalation",

CC.Feedback,

CC.HR "HR Guideline Issues",

CC.OMR "OMR / Exceptions",

CC.Quality,

CC.Recognition,

CC.Schedule "Schedule Adherence",

CC.SecFloor "Secure Floor Violations",

CC.Other,

CL.Description txtDescription,

CL.CoachingNotes txtCoachingNotes,

CL.isVerified,

CL.SubmittedDate,

CL.StartDate,

CL.SupReviewedAutoDate,

CL.isCSE,

CL.MgrReviewManualDate,

CL.MgrReviewAutoDate,

CL.MgrNotes txtMgrNotes,

CL.isCSRAcknowledged,

CL.isCoachingRequired,

CL.CSRReviewAutoDate,

CL.CSRComments txtCSRComments

FROM [EC].[Employee\_Hierarchy] eh,

[EC].[Employee\_Hierarchy] sh,

[EC].[DIM\_Status] s,

[EC].[DIM\_Source] sc,

[EC].[DIM\_Site] st,

(SELECT ccl.FormName,

MAX(CASE WHEN [cr].[CoachingReason] = ''AHT'' THEN [clr].[Value] ELSE NULL END) AHT,

MAX(CASE WHEN [cr].[CoachingReason] = ''ARC Issue'' THEN [clr].[Value] ELSE NULL END) ARC,

MAX(CASE WHEN [cr].[CoachingReason] = ''Attendance'' THEN [clr].[Value] ELSE NULL END) Attendance,

MAX(CASE WHEN [cr].[CoachingReason] = ''BCC Process Procedure Issues'' THEN [clr].[Value] ELSE NULL END) BCCPPI,

MAX(CASE WHEN [cr].[CoachingReason] = ''Current Coaching Initiative'' THEN [clr].[Value] ELSE NULL END) CCI,

MAX(CASE WHEN [cr].[CoachingReason] = ''Customer Service Escalation'' THEN [clr].[Value] ELSE NULL END) CSE,

MAX(CASE WHEN [cr].[CoachingReason] = ''Feedback'' THEN [clr].[Value] ELSE NULL END) Feedback,

MAX(CASE WHEN [cr].[CoachingReason] = ''HR Guideline Issues'' THEN [clr].[Value] ELSE NULL END) HR,

MAX(CASE WHEN [cr].[CoachingReason] = ''OMR / Exceptions'' THEN [clr].[Value] ELSE NULL END) OMR,

MAX(CASE WHEN [cr].[CoachingReason] = ''Quality'' THEN [clr].[Value] ELSE NULL END) Quality,

MAX(CASE WHEN [cr].[CoachingReason] = ''Recognition'' THEN [clr].[Value] ELSE NULL END) Recognition,

MAX(CASE WHEN [cr].[CoachingReason] = ''Schedule Adherence'' THEN [clr].[Value] ELSE NULL END) Schedule,

MAX(CASE WHEN [cr].[CoachingReason] = ''Secure Floor Violations'' THEN [clr].[Value] ELSE NULL END) SecFloor,

MAX(CASE WHEN [cr].[CoachingReason] = ''Other'' THEN [clr].[Value] ELSE NULL END) Other

FROM [EC].[Coaching\_Log\_Reason] clr,

[EC].[DIM\_Coaching\_Reason] cr,

[EC].[Coaching\_Log] ccl WITH(NOLOCK)

WHERE [ccl].[FormName] = '''+@strFormIDin+'''

AND [clr].[CoachingReasonID] = [cr].[CoachingReasonID]

AND [ccl].[CoachingID] = [clr].[CoachingID]

GROUP BY ccl.FormName ) CC,

[EC].[Coaching\_Log] cl WITH(NOLOCK)

where [cl].[CSRID] = [eh].[Emp\_ID]

and [cl].[StatusID] = [s].[StatusID]

and [cl].[SourceID] = [sc].[SourceID]

and [cl].[SiteID] = [st].[SiteID]

and [cl].[SubmitterID] = [sh].[Emp\_ID]

and [cl].[FormName] = [CC].[FormName]

Order By [cl].[FormName]'

EXEC (@nvcSQL)

END --sp\_SelectReviewFrom\_Coaching\_Log

GO

--</End step 2c>

1. Update procedure to restrict load of Quality logs submitted from UI

--<Begin Step 3>

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

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

-- Author: Susmitha Palacherla

-- Create Date: 02/03/2014

-- Description: This procedure inserts the e-Coaching records into the Coaching\_Log table.

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

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

-- Modified Date: 07/18/2014

-- Description: Updated per SCR 13138 to translate 'Opportunity' and 'Re-Inforcement' values for

-- Quality logs submitted from Web Interface to 'Met Goal' and 'Did Not Meet Goal'

-- on insert into the Coaching\_Reason table.

--

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

ALTER PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log]

( @nvcFormName Nvarchar(50),

@nvcCSR Nvarchar(40),

@nvcProgramName Nvarchar(50),

@intSourceID INT,

@intStatusID INT,

@SiteID INT,

@nvcSubmitter Nvarchar(40),

@dtmEventDate datetime,

@dtmCoachingDate datetime,

@bitisAvokeID bit ,

@nvcAvokeID Nvarchar(40)= NULL ,

@bitisNGDActivityID bit,

@nvcNGDActivityID Nvarchar(40)= NULL ,

@bitisUCID bit,

@nvcUCID Nvarchar(40)= NULL,

@bitisVerintID bit,

@nvcVerintID Nvarchar(255)= NULL,

@intCoachReasonID1 INT,

@intSubCoachReasonID1 INT,

@nvcValue1 Nvarchar(30),

@intCoachReasonID2 INT = NULL,

@intSubCoachReasonID2 INT = NULL,

@nvcValue2 Nvarchar(30) = NULL,

@intCoachReasonID3 INT = NULL,

@intSubCoachReasonID3 INT = NULL,

@nvcValue3 Nvarchar(30) = NULL,

@intCoachReasonID4 INT = NULL,

@intSubCoachReasonID4 INT = NULL,

@nvcValue4 Nvarchar(30) = NULL,

@intCoachReasonID5 INT,

@intSubCoachReasonID5 INT,

@nvcValue5 Nvarchar(30),

@intCoachReasonID6 INT,

@intSubCoachReasonID6 INT,

@nvcValue6 Nvarchar(30),

@intCoachReasonID7 INT,

@intSubCoachReasonID7 INT,

@nvcValue7 Nvarchar(30),

@intCoachReasonID8 INT,

@intSubCoachReasonID8 INT,

@nvcValue8 Nvarchar(30),

@intCoachReasonID9 INT,

@intSubCoachReasonID9 INT,

@nvcValue9 Nvarchar(30),

@intCoachReasonID10 INT,

@intSubCoachReasonID10 INT,

@nvcValue10 Nvarchar(30),

@intCoachReasonID11 INT,

@intSubCoachReasonID11 INT,

@nvcValue11 Nvarchar(30),

@intCoachReasonID12 INT,

@intSubCoachReasonID12 INT,

@nvcValue12 Nvarchar(30),

@nvcDescription Nvarchar(3000) ,

@nvcCoachingNotes Nvarchar(3000) ,

@bitisVerified bit ,

@dtmSubmittedDate datetime ,

@dtmStartDate datetime ,

@dtmSupReviewedAutoDate datetime ,

@bitisCSE bit ,

@dtmMgrReviewManualDate datetime ,

@dtmMgrReviewAutoDate datetime ,

@nvcMgrNotes Nvarchar(3000) ,

@bitisCSRAcknowledged bit ,

@dtmCSRReviewAutoDate datetime ,

@nvcCSRComments Nvarchar(3000),

@bitEmailSent bit

)

AS

BEGIN

DECLARE @RetryCounter INT

SET @RetryCounter = 1

RETRY: -- Label RETRY

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

-- Fetch the Employee ID of the current User (@nvcCSR) and Employee ID of the Submitter (@nvcSubmitter).

DECLARE @nvcEmpID Nvarchar(10),

@nvcSubmitterID Nvarchar(10),

@dtmDate datetime

SET @dtmDate = GETDATE()

SET @nvcEmpID = EC.fn\_nvcGetEmpIdFromLanID(@nvcCSR,@dtmDate)

SET @nvcSubmitterID = EC.fn\_nvcGetEmpIdFromLanID(@nvcSubmitter,@dtmDate)

INSERT INTO [EC].[Coaching\_Log]

([FormName]

,[ProgramName]

,[SourceID]

,[StatusID]

,[SiteID]

,[CSR]

,[CSRID]

,[SubmitterID]

,[EventDate]

,[CoachingDate]

,[isAvokeID]

,[AvokeID]

,[isNGDActivityID]

,[NGDActivityID]

,[isUCID]

,[UCID]

,[isVerintID]

,[VerintID]

,[Description]

,[CoachingNotes]

,[isVerified]

,[SubmittedDate]

,[StartDate]

,[SupReviewedAutoDate]

,[isCSE]

,[MgrReviewManualDate]

,[MgrReviewAutoDate]

,[MgrNotes]

,[isCSRAcknowledged]

,[CSRReviewAutoDate]

,[CSRComments]

,[EmailSent])

VALUES

(@nvcFormName

,@nvcProgramName

,@intSourceID

,@intStatusID

,@SiteID

,@nvcCSR

,@nvcEmpID

,@nvcSubmitterID

,@dtmEventDate

,@dtmCoachingDate

,@bitisAvokeID

,@nvcAvokeID

,@bitisNGDActivityID

,@nvcNGDActivityID

,@bitisUCID

,@nvcUCID

,@bitisVerintID

,@nvcVerintID

,@nvcDescription

,@nvcCoachingNotes

,@bitisVerified

,@dtmSubmittedDate

,@dtmStartDate

,@dtmSupReviewedAutoDate

,@bitisCSE

,@dtmMgrReviewManualDate

,@dtmMgrReviewAutoDate

,@nvcMgrNotes

,@bitisCSRAcknowledged

,@dtmCSRReviewAutoDate

,@nvcCSRComments

,@bitEmailSent)

SELECT @@IDENTITY AS 'Identity';

DECLARE @I BIGINT = @@IDENTITY

/\*

IF NOT @intCoachReasonID1 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID1,@intSubCoachReasonID1,

CASE WHEN @intCoachReasonID1 = 6 THEN 'Opportunity' ELSE @nvcValue1 END)

END

\*/

IF NOT @intCoachReasonID1 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID1,@intSubCoachReasonID1,

CASE WHEN @intCoachReasonID1 = 6 THEN 'Opportunity'

WHEN (@intCoachReasonID1 = 10 AND @nvcValue1 = 'Opportunity') THEN 'Did Not Meet Goal'

WHEN (@intCoachReasonID1 = 10 AND @nvcValue1 = 'Reinforcement') THEN 'Met Goal'

ELSE @nvcValue1 END)

END

IF NOT @intCoachReasonID2 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID2,@intSubCoachReasonID2,@nvcValue2)

END

IF NOT @intCoachReasonID3 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID3, @intSubCoachReasonID3,@nvcValue3)

END

IF NOT @intCoachReasonID4 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID4, @intSubCoachReasonID4,@nvcValue4)

END

IF NOT @intCoachReasonID5 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID5, @intSubCoachReasonID5,@nvcValue5)

END

IF NOT @intCoachReasonID6 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID6, @intSubCoachReasonID6,@nvcValue6)

END

IF NOT @intCoachReasonID7 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID7, @intSubCoachReasonID7,@nvcValue7)

END

IF NOT @intCoachReasonID8 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID8, @intSubCoachReasonID8,@nvcValue8)

END

IF NOT @intCoachReasonID9 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID9, @intSubCoachReasonID9,@nvcValue9)

END

IF NOT @intCoachReasonID10 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID10, @intSubCoachReasonID10,@nvcValue10)

END

IF NOT @intCoachReasonID11 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID11, @intSubCoachReasonID11,@nvcValue11)

END

IF NOT @intCoachReasonID12 IS NULL

BEGIN

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

([CoachingID],[CoachingReasonID],[SubCoachingReasonID],[Value])

VALUES (@I, @intCoachReasonID12, @intSubCoachReasonID12,@nvcValue12)

END

COMMIT TRANSACTION

END TRY

BEGIN CATCH

--PRINT 'Rollback Transaction'

ROLLBACK TRANSACTION

DECLARE @DoRetry bit; -- Whether to Retry transaction or not

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SET @doRetry = 0;

SELECT @ErrorMessage = ERROR\_MESSAGE(),

@ErrorSeverity = ERROR\_SEVERITY(),

@ErrorState = ERROR\_STATE()

IF ERROR\_NUMBER() = 1205 -- Deadlock Error Number

BEGIN

SET @doRetry = 1; -- Set @doRetry to 1 only for Deadlock

END

IF @DoRetry = 1

BEGIN

SET @RetryCounter = @RetryCounter + 1 -- Increment Retry Counter By one

IF (@RetryCounter > 3) -- Check whether Retry Counter reached to 3

BEGIN

RAISERROR(@ErrorMessage, 18, 1) -- Raise Error Message if

-- still deadlock occurred after three retries

END

ELSE

BEGIN

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

GOTO RETRY -- Go to Label RETRY

END

END

ELSE

BEGIN

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

END CATCH

END -- sp\_InsertInto\_Coaching\_Log

GO

--</End Step 3>

1. Stage updated IQS\_Coaching.dtsx Package

Download ssis package IQS\_Coaching.dtsx (VM version 1.2) from \cms\eCoaching\_V2\ETL\

In version manager and copy to [\\vdenssdbp07\scorecard-ssis\Coaching](file:///\\vdenssdbp07\scorecard-ssis\Coaching)

Overwriting existing package

1. Update procedures for Outliers Load – 13213

--<Begin Step 5a>

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

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

-- Author: Susmitha Palacherla

-- Create date: 03/10/2014

-- Loads records from [EC].[Outlier\_Coaching\_Stage]to [EC].[Coaching\_Log]

-- Last modified by: Susmitha Palacherla

-- Modified per SCR 13213 to map to Coaching Reason ID 9.

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

ALTER PROCEDURE [EC].[sp\_InsertInto\_Coaching\_Log\_Outlier]

AS

BEGIN

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

DECLARE @maxnumID INT,

@dtmDate DATETIME

-- Fetches the maximum CoachingID before the insert.

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

-- Fetches the Date of the Insert

SET @dtmDate = GETDATE()

-- Inserts records from the Outlier\_Coaching\_Stage table to the Coaching\_Log Table

INSERT INTO [EC].[Coaching\_Log]

([FormName]

,[ProgramName]

,[SourceID]

,[StatusID]

,[SiteID]

,[CSR]

,[CSRID]

,[SubmitterID]

,[EventDate]

,[isAvokeID]

,[isNGDActivityID]

,[isUCID]

,[isVerintID]

,[Description]

,[SubmittedDate]

,[StartDate]

,[isCSRAcknowledged]

,[numReportID]

,[strReportCode]

)

select Distinct LOWER(cs.CSR\_LANID) [FormName],

CASE cs.Program

WHEN NULL THEN csr.Emp\_Program

WHEN '' THEN csr.Emp\_Program

ELSE cs.Program END [ProgramName],

212 [SourceID],

[EC].[fn\_strStatusIDFromStatus](cs.Form\_Status)[StatusID],

[EC].[fn\_intGetSiteIDFromLanID](cs.CSR\_LANID,@dtmDate)[SiteID],

LOWER(cs.CSR\_LANID) [CSR],

cs.CSR\_EMPID [CSRID],

[EC].[fn\_nvcGetEmpIdFromLanId](LOWER(cs.Submitter\_LANID),@dtmDate)[SubmitterID],

cs.Event\_Date [EventDate],

0 [isAvokeID],

0 [isNGDActivityID],

0 [isUCID],

0 [isVerintID],

EC.fn\_nvcHtmlEncode(cs.TextDescription) [Description],

cs.Submitted\_Date SubmittedDate,

cs.Start\_Date [StartDate],

0 [isCSRAcknowledged],

cs.Report\_ID [numReportID],

cs.Report\_Code [strReportCode]

from [EC].[Outlier\_Coaching\_Stage] cs join EC.Employee\_Hierarchy csr on cs.CSR\_EMPID = csr.Emp\_ID

left outer join EC.Coaching\_Log cf on cs.Report\_ID = cf.numReportID and cs.Report\_Code = cf.strReportCode

where cf.numReportID is Null and cf.strReportCode is null

-- Updates the strFormID value

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

UPDATE [EC].[Coaching\_Log]

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

where [FormName] not like 'eCL%'

OPTION (MAXDOP 1)

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

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

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

([CoachingID]

,[CoachingReasonID]

,[SubCoachingReasonID]

,[Value])

SELECT cf.[CoachingID],

9,

[EC].[fn\_intSubCoachReasonIDFromRptCode](SUBSTRING(cf.strReportCode,1,3)),

os.[CoachReason\_Current\_Coaching\_Initiatives]

FROM [EC].[Outlier\_Coaching\_Stage] os JOIN [EC].[Coaching\_Log] cf

ON os.[Report\_ID] = cf.[numReportID] AND os.[Report\_Code] = cf.[strReportCode]

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

ON cf.[CoachingID] = cr.[CoachingID]

WHERE cr.[CoachingID] 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

GO

--</End Step 5a>

--<Begin Step 5b>

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

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

-- Author: Susmitha Palacherla

-- Create Date: 11/16/12

-- Description: This procedure allows managers to update the e-Coaching records from the review page for Outlier records.

-- Last modified by: Susmitha Palacherla

-- Modified per SCR 13213 to add Coaching Reason 'OMR / Exceptions' to the filter criteria for the review page.

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

ALTER PROCEDURE [EC].[sp\_Update5Review\_Coaching\_Log]

(

@nvcFormID Nvarchar(50),

@nvcFormStatus Nvarchar(30),

@nvcstrReasonNotCoachable Nvarchar(30),

@dtmMgrReviewAutoDate datetime,

@dtmMgrReviewManualDate datetime,

@bitisCoachingRequired bit,

-- @nvcstrCoachReason\_Current\_Coaching\_Initiatives Nvarchar(30),

@nvcMgrNotes Nvarchar(max),

@nvctxtReasonNotCoachable Nvarchar(max)

)

AS

BEGIN

DECLARE @RetryCounter INT

SET @RetryCounter = 1

RETRY: -- Label RETRY

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRANSACTION

BEGIN TRY

UPDATE EC.Coaching\_Log

SET StatusID = (select StatusID from EC.DIM\_Status where status = @nvcFormStatus),

strReasonNotCoachable = @nvcstrReasonNotCoachable,

isCoachingRequired = @bitisCoachingRequired,

MgrReviewAutoDate = @dtmMgrReviewAutoDate,

MgrReviewManualDate = @dtmMgrReviewManualDate,

MgrNotes = @nvcMgrNotes,

txtReasonNotCoachable = @nvctxtReasonNotCoachable

WHERE FormName = @nvcFormID

OPTION (MAXDOP 1)

UPDATE EC.Coaching\_Log\_Reason

SET Value = (CASE WHEN @bitisCoachingRequired = 'True' then 'Opportunity' ELSE 'Not Coachable' END)

FROM EC.Coaching\_Log cl INNER JOIN EC.Coaching\_Log\_Reason clr

ON cl.CoachingID = clr.CoachingID

INNER JOIN EC.DIM\_Coaching\_Reason cr ON cr.CoachingReasonID = clr.CoachingReasonID

WHERE cl.FormName = @nvcFormID

and cr.CoachingReason in ('OMR / Exceptions', 'Current Coaching Initiative')

OPTION (MAXDOP 1)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

--PRINT 'Rollback Transaction'

ROLLBACK TRANSACTION

DECLARE @DoRetry bit; -- Whether to Retry transaction or not

DECLARE @ErrorMessage NVARCHAR(4000)

DECLARE @ErrorSeverity INT

DECLARE @ErrorState INT

SET @doRetry = 0;

IF ERROR\_NUMBER() = 1205 -- Deadlock Error Number

BEGIN

SET @doRetry = 1; -- Set @doRetry to 1 only for Deadlock

END

IF @DoRetry = 1

BEGIN

SET @RetryCounter = @RetryCounter + 1 -- Increment Retry Counter By one

IF (@RetryCounter > 3) -- Check whether Retry Counter reached to 3

BEGIN

RAISERROR(@ErrorMessage, 18, 1) -- Raise Error Message if

-- still deadlock occurred after three retries

END

ELSE

BEGIN

WAITFOR DELAY '00:00:00.05' -- Wait for 5 ms

GOTO RETRY -- Go to Label RETRY

END

END

ELSE

BEGIN

RAISERROR (@ErrorMessage, -- Message text.

@ErrorSeverity, -- Severity.

@ErrorState -- State.

)

END

END CATCH

END --sp\_Update5Review\_Coaching\_Log

GO

--</End Step 5b>

--<Begin Step 5c>

Update clr

set CoachingReasonID = 9

from [EC].[Coaching\_Log]cl join [EC].[Coaching\_Log\_reason] clr

on cl.coachingid = clr.coachingid

where cl.sourceid = 212 and CoachingReasonID = 5

and cl.submitteddate >= '2014-06-12 00:00:00.000'

--</End Step 5c>

This completes installation for Section 6.0