**CCO eCoaching DB Unit Test Document**

* **Warnings in eCL**

July 23, 2015

Change History Log

| Date | Revision | Change Description | Author |
| --- | --- | --- | --- |
| 10/03/2014 | 1.0 | SCR – 3479 – Incorporate CSR warnings in eCL | Susmitha Palacherla |
| 10/22/2014 | 1.1 | SCR - 13479 – Requirement changes – Additional testing | Susmitha Palacherla |
| 10/28/2014 | 2.0 | SCR – 13624 – Inactivate expired warnings | Susmitha Palacherla |
| 07/23/2015 | 3.0 | TFS 363/402 – Fix issue with Duplicate Form names for insert from UI | Susmitha Palacherla |
|  |  |  |  |

Contents

[1.0 SCR 13479 – Incorporate CSR warnings in eCL 4](#_Toc425431270)

[2.0 SCR 13624 – Inactivate Expired warnings 14](#_Toc425431271)

[3.0 TFS 363/402 Fix issue with duplicate form names for insert from UI 16](#_Toc425431272)

## SCR 13479 – Incorporate CSR warnings in eCL

| Item | Description |
| --- | --- |
| Change Type | New Functionality |
| Change Description | Operations is looking to track Warnings as a new Reason in eCL. |
| Test Environment | eCoaching\_Dev database on vrivfssdbt02\scord01,1437 |
| Tables Updated | Multiple |
| Code doc | Multiple |
| Notes |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test #** | **Action** | **Expected Results** | **Result Pass/Fail** | **Comments** |
|  | Test new Function fn\_strDirectUserHierarchy  SELECT EC].[fn\_strDirectUserHierarchy]  ('susmitha.palacherla','timothy.queen',Getdate()) | Yes | Pass |  |
|  | Test new Function fn\_strDirectUserHierarchy  SELECT [eCoachingDev].[EC].[fn\_strDirectUserHierarchy]  ('susmitha.palacherla','jourdain.augustin', GetDate()) | No | Pass |  |
| 3. | Test new Function fn\_strDirectUserHierarchy  SELECT [EC].[fn\_strDirectUserHierarchy]  ('susmitha.palacherla','timothy.queen','2002-01-01') | No | Pass |  |
| 4. | Test updated procedure sp\_Select\_CoachingReasons\_By\_Module  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_Select\_CoachingReasons\_By\_Module]  @strModulein = N'CSR',  @strSourcein = N'direct',  @isSplReason = 1,-- 0 for non WARN and CSE  @splReasonPrty = 1, -- use 1 for WARN; 2 for CSE  @strCSRin = N'jourdain.augustin', --  @strSubmitterin = N'timothy.queen'  SELECT 'Return Value' = @return\_value  GO | CoachingReasonID CoachingReason  28 Verbal Warning  29 Written Warning  30 Final Written Warning | Pass |  |
| 5. | Test updated procedure sp\_Select\_CoachingReasons\_By\_Module  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_Select\_CoachingReasons\_By\_Module]  @strModulein = N'CSR',  @strSourcein = N'indirect',  @isSplReason = 1,-- 0 for non WARN and CSE  @splReasonPrty = 1, -- use 1 for WARN; 2 for CSE  @strCSRin = N'jourdain.augustin', --  @strSubmitterin = N'timothy.queen'  SELECT 'Return Value' = @return\_value  GO | CoachingReasonID CoachingReason | Pass |  |
| 6. | Test updated procedure sp\_Select\_CoachingReasons\_By\_Module  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_Select\_CoachingReasons\_By\_Module]  @strModulein = N'CSR',  @strSourcein = N'indirect',  @isSplReason = 1,-- 0 for non WARN and CSE  @splReasonPrty = 2, -- use 1 for WARN; 2 for CSE  @strCSRin = N'jourdain.augustin', --  @strSubmitterin = N'timothy.queen'  SELECT 'Return Value' = @return\_value  GO | CoachingReasonID CoachingReason  6 Customer Service Escalation | Pass |  |
| 7. | Test updated procedure sp\_Select\_CoachingReasons\_By\_Module  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_Select\_CoachingReasons\_By\_Module]  @strModulein = N'CSR',  @strSourcein = N'direct',  @isSplReason = 1,-- 0 for non WARN and CSE  @splReasonPrty = 2, -- use 1 for WARN; 2 for CSE  @strCSRin = N'jourdain.augustin', --  @strSubmitterin = N'timothy.queen'  SELECT 'Return Value' = @return\_value  GO | CoachingReasonID CoachingReason  6 Customer Service Escalation | Pass |  |
| 8. | Test updated procedure sp\_Select\_CoachingReasons\_By\_Module  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_Select\_CoachingReasons\_By\_Module]  @strModulein = N'CSR',  @strSourcein = N'direct',  @isSplReason = 0,-- 0 for non WARN and CSE  @splReasonPrty = 2, -- use 1 for WARN; 2 for CSE  @strCSRin = N'jourdain.augustin', --  @strSubmitterin = N'timothy.queen'  SELECT 'Return Value' = @return\_value  GO | CoachingReasonID CoachingReason  1 AHT  2 ARC Issue  3 Attendance  4 CCO Processes and Procedures  5 Current Coaching Initiative  7 Feedback  8 HR Guideline Issues  9 OMR / Exceptions  10 Quality  11 Recognition  12 Schedule Adherence  13 Secure Floor Violations | Pass |  |
| 9. | Test updated procedure sp\_Select\_CoachingReasons\_By\_Module  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_Select\_CoachingReasons\_By\_Module]  @strModulein = N'CSR',  @strSourcein = N'indirect',  @isSplReason = 0,-- 0 for non WARN and CSE  @splReasonPrty = 2, -- use 1 for WARN; 2 for CSE  @strCSRin = N'jourdain.augustin', --  @strSubmitterin = N'timothy.queen'  SELECT 'Return Value' = @return\_value | CoachingReasonID CoachingReason  1 AHT  2 ARC Issue  3 Attendance  4 CCO Processes and Procedures  5 Current Coaching Initiative  7 Feedback  8 HR Guideline Issues  9 OMR / Exceptions  10 Quality  11 Recognition  12 Schedule Adherence  13 Secure Floor Violations | Pass |  |
| 10. | Test updated procedure sp\_Select\_SubCoachingReasons\_By\_Reason  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_Select\_SubCoachingReasons\_By\_Reason]  @strReasonin = N'Verbal Warning',  @strModulein = N'CSR',  @strSourcein = N'Direct',  @nvcEmpLanIDin = N'susmitha.palacherla'  SELECT 'Return Value' = @return\_value  GO | SubCoachingReasonID SubCoachingReason  92 Attendance  93 Call Avoidance  94 Conduct  95 Dress Code  48 ETS  96 GDIT Policy  98 Performance  97 Quality  77 Secure Floor Policy  1 Security and Privacy Incident  62 Other: Specify reason under coaching details. | Pass | Retested after adding ETS as a Sub Coaching Source. Failed Initially.  Had to modify sp sp\_Select\_SubCoachingReasons\_By\_Reason to display ETS as a Sub coaching Reason irrespective of Job Code for Warnings related Coaching Reasons.  11 Sub Coaching Reasons returned including ETS as expected after update. |
| 11. | Test updated procedure sp\_Select\_SubCoachingReasons\_By\_Reason  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_Select\_SubCoachingReasons\_By\_Reason]  @strReasonin = N'Written Warning',  @strModulein = N'CSR',  @strSourcein = N'Direct',  @nvcEmpLanIDin = N'susmitha.palacherla'  SELECT 'Return Value' = @return\_value  GO | SubCoachingReasonID SubCoachingReason  92 Attendance  93 Call Avoidance  94 Conduct  95 Dress Code  48 ETS  96 GDIT Policy  98 Performance  97 Quality  77 Secure Floor Policy  1 Security and Privacy Incident  62 Other: Specify reason under coaching details. | Pass | See Comments for # 10 above. |
| 12. | Test updated procedure sp\_Select\_SubCoachingReasons\_By\_Reason  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_Select\_SubCoachingReasons\_By\_Reason]  @strReasonin = N'Final Written Warning',  @strModulein = N'CSR',  @strSourcein = N'Direct',  @nvcEmpLanIDin = N'susmitha.palacherla'  SELECT 'Return Value' = @return\_value  GO | SubCoachingReasonID SubCoachingReason  92 Attendance  93 Call Avoidance  94 Conduct  95 Dress Code  48 ETS  96 GDIT Policy  98 Performance  97 Quality  77 Secure Floor Policy  1 Security and Privacy Incident  62 Other: Specify reason under coaching details. | Pass | See Comments for # 10 above. |
| 13. | Test updated procedure sp\_Select\_SubCoachingReasons\_By\_Reason  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_Select\_SubCoachingReasons\_By\_Reason]  @strReasonin = N'Final Written Warning',  @strModulein = N'CSR',  @strSourcein = N'InDirect',  @nvcEmpLanIDin = N'susmitha.palacherla'  SELECT 'Return Value' = @return\_value  GO | SubCoachingReasonID SubCoachingReason | Pass |  |
| 14. | Test Insert from backend  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_InsertInto\_Warning\_Log]  @nvcFormName = N'jourdain.augustin-1',  @nvcProgramName = N'NA',  @nvcEmpLanID = N'jourdain.augustin',  @SiteID = NULL,  @nvcSubmitter = N'timothy.queen',  @dtmEventDate = N'2014-10-10',  @intCoachReasonID1 = 28,  @nvcSubCoachReasonID1 = N'92',  @nvcDescription = N'This is a test to insert a warning record.',  @nvcCoachingNotes = N'This is a test to insert a warning record.',  @dtmSubmittedDate = N'2014-10-10',  @ModuleID = 1  SELECT 'Return Value' = @return\_value  GO | 1 record inserted to warning\_log table with identity 1  And 2 records inserted to warning\_log\_reason table | Pass | Obsolete as of 10/22.  Replaced with test # 20 |
| 15. | exec ec.sp\_openkeys  SELECT [WarningID]  ,[FormName]  ,[ProgramName]  ,[SourceID]  ,[StatusID]  ,[SiteID]  ,[EmpLanID]  ,[EmpID]  ,[SubmitterID]  ,[SupID]  ,[MgrID]  ,[WarningGivenDate]  ,[Description]  ,[EC].fn\_Decrypt([Description])  ,[CoachingNotes]  ,[EC].fn\_Decrypt([CoachingNotes])  ,[SubmittedDate]  ,[ModuleID]  ,[Active]  FROM [eCoachingDev].[EC].[Warning\_Log]  GO  CLOSE SYMMETRIC KEY WarnDescKey; | Expecting to see decrypted values for Decsription and CoachingNotes as submitted. | Pass | Obsolete as of 10/22 |
| 16. | USE [eCoachingDev]  GO  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_SelectReviewFrom\_Warning\_Log]  @strFormIDin = N'Endalkachew.Negussie-1'  SELECT 'Return Value' = @return\_value  GO | Returns the warning log | Pass |  |
| 17. | USE [eCoachingDev]  GO  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_SelectReviewFrom\_Warning\_Log\_Reasons]  @strFormIDin =N'Endalkachew.Negussie-1'  SELECT 'Return Value' = @return\_value  GO | Returns the warnings logs for the above record in test 16. | Pass |  |
| 18. | USE [eCoachingDev]  GO  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_SelectFrom\_Warning\_Log\_SUPCSRCompleted]  @strCSRSUPin = N'susmitha.palacherla',  @strSDatein = N'2014-10-01',  @strEDatein = N'2014-10-14',  @bitActive = 1  SELECT 'Return Value' = @return\_value  GO | Returns 2 records where sup is Susmitha palacherla | Pass |  |
| 19. | USE [eCoachingDev]  GO  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_SelectFrom\_Warning\_Log\_MGRCSRCompleted]  @strCSRMGRin = N'Andy.Ma',  @strSDatein = N'2014-10-01',  @strEDatein = N'2014-10-13',  @bitActive = N'1'  SELECT 'Return Value' = @return\_value  GO | Returns 2 records where mgr is Andy ma | Pass |  |
| 20. | Test Insert from backend  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_InsertInto\_Warning\_Log]  @nvcFormName = N'jourdain.augustin-9',  @nvcProgramName = N'NA',  @nvcEmpLanID = N'jourdain.augustin',  @SiteID = NULL,  @nvcSubmitter = N'timothy.queen',  @dtmEventDate = N'2014-10-22',  @intCoachReasonID1 = 29,  @nvcSubCoachReasonID1 = N'92',  @dtmSubmittedDate = N'2014-10-22',  @ModuleID = 1  SELECT 'Return Value' = @return\_value  GO | 1 record inserted to warning\_log table with identity 1  And 1 records inserted to warning\_log\_reason table | Pass |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## SCR 13624 – Inactivate Expired warnings

| Item | Description |
| --- | --- |
| Change Type | New Functionality |
| Change Description | Inactivate all Warning logs 13 weeks after Warning Given Date |
| Test Environment | eCoaching\_Dev database on vrivfssdbt02\scord01,1437 |
| Tables Updated | EC.Warning\_Log |
| Code doc | CCO\_eCoaching\_Warning\_Log\_Create.sql |
| Notes | Using 91 days for 13 weeks. The Inactivation is done using a stored procedure that is called as a step in the Employee Hierarchy load process that is a daily process. For the unit testing the stored procedure will be executed directly in ssms. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test #** | **Action** | **Expected Results** | **Result Pass/Fail** | **Comments** |
|  | Set up a test record that has a warninggivendate older than 91 days.  update ec.warning\_log  set WarningGivenDate = '2014-06-22 00:00:00.000'  where formname = 'eCL-Jourdain.Augustin-739834'  Then Exceute sp [EC].[sp\_InactivateExpiredWarningLogs]  USE [eCoachingDev]  GO  DECLARE @return\_value int  EXEC @return\_value = [EC].[sp\_InactivateExpiredWarningLogs]  SELECT 'Return Value' = @return\_value  GO | When querying the warning\_log table, of the 4 logs present in the table , 1 log should have Active = 0 and the other 3 Active = 1 | Pass |  |
|  |  |  |  |  |

## TFS 363/402 Fix issue with duplicate form names for insert from UI

| Item | | Description | | | |
| --- | --- | --- | --- | --- | --- |
| Change Type | | Problem Fix | | | |
| Change Description | | Once in a while, users are not able to access "My Dashboard" and/or "My Submissions" and/or "Historical Dashboard", error page is displayed for the users (A system error has occurred. Please contact the system admin for assistance.".  Under the scene, this is the exception caught in debug mode:  Subquery returned more than 1 value. This is not permitted when the subquery follows =, !=, <, <=, >, >= or when the subquery is used as an expression.  The above exception is thrown when there are more than 1 record with the form name passed to sp\_SelectReviewFrom\_Coaching\_Log\_Reasons/sp\_SelectReviewFrom\_Warning\_Log\_Reasons.  Fix: UI doesn't generate the random digits as part of the form name, instead stored procedure will insert the record, get the inserted record primary key (CoachingID/WarningID), append to the form name, and update the inserted record; | | | |
| Test Environment | | eCoaching\_Dev database on vrivfssdbt02\scord01,1437 | | | |
| Code Modules created/updated | | SP [EC].[sp\_InsertInto\_Warning\_Log] | | | |
| Code doc | | CCO\_eCoaching\_Warning\_Log\_Create.sql | | | |
| Notes | |  | | | |
| TEST# | ACTION | | EXPECTED RESULTS | RESULTS  P/F/I | COMMENTS | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3.1 | USE [eCoachingDev]  GO  DECLARE @return\_value int,  @isDup bit,  @nvcNewFormName nvarchar(50)  EXEC @return\_value = [EC].[sp\_InsertInto\_Warning\_Log]  @nvcFormName = N'Jourdain.Augustin',  @nvcProgramName = N'Marketplace',  @nvcEmpLanID = N'lili.huang',  @SiteID = 19,  @nvcSubmitter = N'Susmitha.Palacherla',  @dtmEventDate = N'2015-07-20 00:00:00.000',  @intCoachReasonID1 = 28,  @nvcSubCoachReasonID1 = N'91',  @dtmSubmittedDate = N'2015-07-23 15:46:00.000',  @ModuleID = 1,  @isDup = @isDup OUTPUT,  @nvcNewFormName = @nvcNewFormName OUTPUT    SELECT @isDup as N'isDup'  SELECT @nvcNewFormName as N'NewFormName'  SELECT 'Return Value' = @return\_value  GO | Record should be inserted with FormName eCL-Jourdain.Augustin-5463 as WarningID for record is based on identity which is 5463. | P |  |
| 3.2 | Re-Insert above record with same values. | Record should not be inserted as it is a duplicate. | P |  |