IF OBJECT\_ID('nr\_EvidencesAndRecognizedApplicationsFromProduct', 'P') IS NOT NULL

DROP PROCEDURE dbo.nr\_EvidencesAndRecognizedApplicationsFromProduct

GO

CREATE PROCEDURE dbo.nr\_EvidencesAndRecognizedApplicationsFromProduct (@Productname nvarchar(500)) as

--V5: Nicolas Rousseau: stored procedure used by the "Recognized Applications Transparency Report (NR)" view

IF ISNULL (@Productname, '') = ''

BEGIN

SELECT

Max(ComplianceComputerID) - Max(ComplianceComputerID) + 1 As 'Exit Code',

'Please enter a Product Name to allow the report to run' as Message,

@Productname as Input

FROM ComplianceComputer

ORDER BY [Exit Code] asc

END

IF ISNULL (@Productname, '') <> ''

BEGIN

IF OBJECT\_ID ('tempdb..#ApplicationsAndEvidences') IS NOT NULL

DROP TABLE #ApplicationsAndEvidences

Create Table #ApplicationsAndEvidences

(

ComputerName nvarchar (500) COLLATE DATABASE\_DEFAULT,

EvidenceName nvarchar (500) COLLATE DATABASE\_DEFAULT,

Version nvarchar (500) COLLATE DATABASE\_DEFAULT,

Company nvarchar (500) COLLATE DATABASE\_DEFAULT,

Description nvarchar (500) COLLATE DATABASE\_DEFAULT,

FilePath nvarchar (1000) COLLATE DATABASE\_DEFAULT,

MatchedARLEvidence nvarchar (3000) COLLATE DATABASE\_DEFAULT,

RecognizedApplication nvarchar (3000) COLLATE DATABASE\_DEFAULT,

ProductName nvarchar (3000),

ApplicationVersion nvarchar (3000),

ApplicationEdition nvarchar (3000),

SuiteMemberApplication nvarchar (3000) COLLATE DATABASE\_DEFAULT,

Classification nvarchar (500) COLLATE DATABASE\_DEFAULT,

OperatingSystem nvarchar (500) COLLATE DATABASE\_DEFAULT,

LocationPath nvarchar (1000) COLLATE DATABASE\_DEFAULT,

CostCenterPath nvarchar (1000) COLLATE DATABASE\_DEFAULT,

CorporateUnitPath nvarchar (1000) COLLATE DATABASE\_DEFAULT,

FileSize int,

EvidenceType nvarchar (1000) COLLATE DATABASE\_DEFAULT,

EvidenceSubType nvarchar (1000) COLLATE DATABASE\_DEFAULT,

ConnectionNameDisplayName nvarchar (1000) COLLATE DATABASE\_DEFAULT,

EvidenceInventoryDate date,

AssignedLicense nvarchar (1000) COLLATE DATABASE\_DEFAULT

)

IF OBJECT\_ID ('tempdb..#InstalledSoftware') IS NOT NULL

DROP TABLE #InstalledSoftware

Create Table #InstalledSoftware

(

ComplianceComputerID int,

SoftwareTitleID int,

SuiteSoftwareTitleID int,

SoftwareLicenseID int

)

IF OBJECT\_ID ('tempdb..#SoftwareTitleFileEvidence') IS NOT NULL

DROP TABLE #SoftwareTitleFileEvidence

Create Table #SoftwareTitleFileEvidence

(

SoftwareTitleID int,

FileEvidenceID bigint

)

IF OBJECT\_ID ('tempdb..#ImportedFileEvidenceMapping') IS NOT NULL

DROP TABLE #ImportedFileEvidenceMapping

Create Table #ImportedFileEvidenceMapping

(

ExternalFileID bigint,

ComplianceConnectionID int,

FileEvidenceID int,

SoftwareTitleID int,

)

IF OBJECT\_ID ('tempdb..#ImportedInstalledFileEvidence') IS NOT NULL

DROP TABLE #ImportedInstalledFileEvidence

Create Table #ImportedInstalledFileEvidence

(

ExternalFileID bigint,

ComplianceConnectionID int,

ExternalID int,

InventoryDate date,

SoftwareTitleID int,

SoftwareLicenseID int,

SuiteSoftwareTitleID int,

ExternalFilePathID bigint,

MatchedARLEvidence nvarchar (3000) COLLATE DATABASE\_DEFAULT

)

IF OBJECT\_ID ('tempdb..#SoftwareTitleInstallerEvidence') IS NOT NULL

DROP TABLE #SoftwareTitleInstallerEvidence

Create Table #SoftwareTitleInstallerEvidence

(

SoftwareTitleID int,

InstallerEvidenceID bigint

)

IF OBJECT\_ID ('tempdb..#ImportedInstallerEvidenceMapping') IS NOT NULL

DROP TABLE #ImportedInstallerEvidenceMapping

Create Table #ImportedInstallerEvidenceMapping

(

ExternalInstallerEvidenceID bigint,

ComplianceConnectionID int,

InstallerEvidenceID int,

SoftwareTitleID int,

)

IF OBJECT\_ID ('tempdb..#ImportedInstalledInstallerEvidence') IS NOT NULL

DROP TABLE #ImportedInstalledInstallerEvidence

Create Table #ImportedInstalledInstallerEvidence

(

ExternalInstallerEvidenceID bigint,

ComplianceConnectionID int,

ExternalID int,

InventoryDate date,

SoftwareTitleID int,

SoftwareLicenseID int,

SuiteSoftwareTitleID int,

EvidenceType nvarchar (500) COLLATE DATABASE\_DEFAULT,

MatchedARLEvidence nvarchar (3000) COLLATE DATABASE\_DEFAULT

)

IF OBJECT\_ID ('tempdb..#SoftwareTitleWMIEvidence') IS NOT NULL

DROP TABLE #SoftwareTitleWMIEvidence

Create Table #SoftwareTitleWMIEvidence

(

SoftwareTitleID int,

WMIEvidenceID bigint

)

IF OBJECT\_ID ('tempdb..#ImportedWMIEvidenceMapping') IS NOT NULL

DROP TABLE #ImportedWMIEvidenceMapping

Create Table #ImportedWMIEvidenceMapping

(

ExternalWMIEvidenceID bigint,

ComplianceConnectionID int,

WMIEvidenceID int,

SoftwareTitleID int,

)

IF OBJECT\_ID ('tempdb..#ImportedInstalledWMIEvidence') IS NOT NULL

DROP TABLE #ImportedInstalledWMIEvidence

Create Table #ImportedInstalledWMIEvidence

(

ExternalWMIEvidenceID bigint,

ComplianceConnectionID int,

ExternalID int,

InventoryDate date,

SoftwareTitleID int,

SoftwareLicenseID int,

SuiteSoftwareTitleID int,

MatchedARLEvidence nvarchar (3000) COLLATE DATABASE\_DEFAULT

)

INSERT INTO #InstalledSoftware

SELECT

isd.ComplianceComputerID,

isd.SoftwareTitleID,

NULL,

isd.SoftwareLicenseID

FROM InstalledSoftwareData isd

LEFT JOIN SoftwareTitle st on st.SoftwareTitleID = isd.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

WHERE stp.ProductName = @Productname

INSERT INTO #SoftwareTitleFileEvidence

SELECT

stfe.SoftwareTitleID,

stfe.FileEvidenceID

FROM SoftwareTitleFileEvidence stfe WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = stfe.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN FileEvidence fe WITH (NOLOCK) on stfe.FileEvidenceID = fe.FileEvidenceID

WHERE stfe.EvidenceExistenceRuleID IN (1,4) -- Mandatory and at least one

AND fe.Ignored = 0

AND stp.ProductName = @Productname

INSERT INTO #ImportedFileEvidenceMapping

SELECT

ifem.ExternalFileID,

ifem.ComplianceConnectionID,

ifem.FileEvidenceID,

stfe.SoftwareTitleID

FROM ImportedFileEvidenceMapping ifem WITH (NOLOCK)

INNER JOIN #SoftwareTitleFileEvidence stfe ON stfe.FileEvidenceID = ifem.FileEvidenceID

--WHERE ifem.FileEvidenceID IN (SELECT FileEvidenceID FROM #SoftwareTitleFileEvidence)

--delete from #ImportedInstalledFileEvidence

CREATE INDEX index1 ON #ImportedFileEvidenceMapping (ExternalFileID)

CREATE INDEX index2 ON #ImportedFileEvidenceMapping (ComplianceConnectionID)

INSERT INTO #ImportedInstalledFileEvidence

SELECT

iife.ExternalFileID,

iife.ComplianceConnectionID,

iife.ExternalID,

ic.InventoryDate,

ifem.SoftwareTitleID,

isd.SoftwareLicenseID,

isd.SuiteSoftwareTitleID,

iife.ExternalFilePathID,

ISNULL (fe.FileName, '') + ' ' + ISNULL (fe.FileVersion, '') + ISNULL (' (' + fe.Company + ')', '')

FROM ImportedInstalledFileEvidence iife WITH (NOLOCK)

LEFT JOIN ImportedComputer ic WITH (NOLOCK) on ic.ExternalID = iife.ExternalID AND ic.ComplianceConnectionID = iife.ComplianceConnectionID

INNER JOIN #ImportedFileEvidenceMapping ifem on ifem.ExternalFileID = iife.ExternalFileID AND ifem.ComplianceConnectionID = iife.ComplianceConnectionID

INNER JOIN #InstalledSoftware isd on isd.ComplianceComputerID = ic.ComplianceComputerID

AND isd.SoftwareTitleID = ifem.SoftwareTitleID -- enhances performance

INNER JOIN FileEvidence fe on fe.FileEvidenceID = ifem.FileEvidenceID

-- Installer evidence now

INSERT INTO #SoftwareTitleInstallerEvidence

SELECT

stfe.SoftwareTitleID,

stfe.InstallerEvidenceID

FROM SoftwareTitleInstallerEvidence stfe WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = stfe.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN InstallerEvidence fe WITH (NOLOCK) on stfe.InstallerEvidenceID = fe.InstallerEvidenceID

WHERE fe.Ignored = 0

AND stp.ProductName = @Productname

INSERT INTO #ImportedInstallerEvidenceMapping

SELECT

ifem.ExternalInstallerID,

ifem.ComplianceConnectionID,

ifem.InstallerEvidenceID,

stfe.SoftwareTitleID

FROM ImportedInstallerEvidenceMapping ifem WITH (NOLOCK)

INNER JOIN #SoftwareTitleInstallerEvidence stfe ON stfe.InstallerEvidenceID = ifem.InstallerEvidenceID

--WHERE ifem.FileEvidenceID IN (SELECT FileEvidenceID FROM #SoftwareTitleFileEvidence)

--delete from #ImportedInstalledFileEvidence

CREATE INDEX index1 ON #ImportedInstallerEvidenceMapping (ExternalInstallerEvidenceID)

CREATE INDEX index2 ON #ImportedInstallerEvidenceMapping (ComplianceConnectionID)

INSERT INTO #ImportedInstalledInstallerEvidence

SELECT

iife.ExternalInstallerEvidenceID,

iife.ComplianceConnectionID,

iife.ExternalComputerID,

ic.InventoryDate,

ifem.SoftwareTitleID,

isd.SoftwareLicenseID,

isd.SuiteSoftwareTitleID,

ife.Evidence,

ISNULL (fe.DisplayName, '') + ' ' + ISNULL (fe.Version, '') + ISNULL (' (' + fe.Publisher + ')', '')

FROM ImportedInstalledInstallerEvidence iife WITH (NOLOCK)

LEFT JOIN ImportedInstallerEvidence ife on ife.ExternalInstallerID = iife.ExternalInstallerEvidenceID

AND ife.ComplianceConnectionID = iife.ComplianceConnectionID

LEFT JOIN ImportedComputer ic WITH (NOLOCK) on ic.ExternalID = iife.ExternalComputerID AND ic.ComplianceConnectionID = iife.ComplianceConnectionID

INNER JOIN #ImportedInstallerEvidenceMapping ifem on ifem.ExternalInstallerEvidenceID = iife.ExternalInstallerEvidenceID AND ifem.ComplianceConnectionID = iife.ComplianceConnectionID

LEFT JOIN InstallerEvidence fe on fe.InstallerEvidenceID = ifem.InstallerEvidenceID

INNER JOIN #InstalledSoftware isd on isd.ComplianceComputerID = ic.ComplianceComputerID

AND isd.SoftwareTitleID = ifem.SoftwareTitleID -- enhances performance

-- WMI now

INSERT INTO #SoftwareTitleWMIEvidence

SELECT

stfe.SoftwareTitleID,

stfe.WMIEvidenceID

FROM SoftwareTitleWMIEvidence stfe WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = stfe.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN WMIEvidence fe WITH (NOLOCK) on stfe.WMIEvidenceID = fe.WMIEvidenceID

WHERE fe.Ignored = 0

AND stp.ProductName = @Productname

INSERT INTO #ImportedWMIEvidenceMapping

SELECT

ifem.ExternalEvidenceID,

ifem.ComplianceConnectionID,

ifem.EvidenceRuleID,

stfe.SoftwareTitleID

FROM ImportedWMIEvidenceRuleMapping ifem WITH (NOLOCK)

INNER JOIN #SoftwareTitleWMIEvidence stfe ON stfe.WMIEvidenceID = ifem.EvidenceRuleID

CREATE INDEX index1 ON #ImportedWMIEvidenceMapping (ExternalWMIEvidenceID)

CREATE INDEX index2 ON #ImportedWMIEvidenceMapping (ComplianceConnectionID)

INSERT INTO #ImportedInstalledWMIEvidence

SELECT DISTINCT

iife.ExternalEvidenceID,

iife.ComplianceConnectionID,

iife.ExternalComputerID,

ic.InventoryDate,

ifem.SoftwareTitleID,

isd.SoftwareLicenseID,

isd.SuiteSoftwareTitleID,

ISNULL (fe.ClassName, '') + ' ' + ISNULL (fe.PropertyName, '') + ISNULL (fe.PropertyValue, '')

FROM ImportedInstalledWMIEvidence iife WITH (NOLOCK)

LEFT JOIN ImportedWMIEvidence ife on ife.ExternalEvidenceID = iife.ExternalEvidenceID

AND ife.ComplianceConnectionID = iife.ComplianceConnectionID

LEFT JOIN ImportedComputer ic WITH (NOLOCK) on ic.ExternalID = iife.ExternalComputerID AND ic.ComplianceConnectionID = iife.ComplianceConnectionID

INNER JOIN #ImportedWMIEvidenceMapping ifem on ifem.ExternalWMIEvidenceID = iife.ExternalEvidenceID AND ifem.ComplianceConnectionID = iife.ComplianceConnectionID

LEFT JOIN WMIEvidence fe on fe.WMIEvidenceID = ifem.WMIEvidenceID

INNER JOIN #InstalledSoftware isd on isd.ComplianceComputerID = ic.ComplianceComputerID

AND isd.SoftwareTitleID = ifem.SoftwareTitleID -- enhances performance

-- Files first

INSERT INTO #ApplicationsAndEvidences

SELECT

cc.ComputerName as 'Computer Name',

ife.FileName as EvidenceName,

ife.FileVersion as version,

ife.Company,

ife.Description,

ifep.ExternalFilePath as 'File Path',

isd.MatchedARLEvidence,

st.FullName,

stp.ProductName,

stv.VersionName,

ste.EditionName,

NULL, --Members are imported later

stc.DefaultValue,

cc.OperatingSystem as 'Operating System',

l.Path as 'Location Path',

c.Path as 'Cost Center Path',

cu.Path,

ife.FileSize as FileSize,

'File' as EvidenceType,

'N/A' as EvidenceSubType,

ccx.ConnectionNameDisplayName,

isd.InventoryDate,

sl.Name

FROM #ImportedInstalledFileEvidence isd WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = isd.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN SoftwareTitleVersion stv WITH (NOLOCK) on stv.SoftwareTitleVersionID = st.SoftwareTitleVersionID

LEFT JOIN SoftwareTitleEdition ste WITH (NOLOCK) on ste.SoftwareTitleEditionID = st.SoftwareTitleEditionID

LEFT JOIN SoftwareTitleClassificationI18N stc WITH (NOLOCK) on stc.SoftwareTitleClassificationID = st.SoftwareTitleClassificationID

LEFT JOIN ImportedFileEvidence ife on ife.ExternalFileID = isd.ExternalFileID AND ife.ComplianceConnectionID = isd.ComplianceConnectionID

LEFT JOIN ImportedComputer icc WITH (NOLOCK) on icc.ExternalID = isd.ExternalID AND icc.ComplianceConnectionID = isd.ComplianceConnectionID

INNER JOIN ComplianceComputerCurrentUserScoped cc WITH (NOLOCK) on cc.ComplianceComputerID = icc.ComplianceComputerID -- apply access restriction

LEFT JOIN SoftwareLicense sl on sl.SoftwareLicenseID = isd.SoftwareLicenseID

LEFT JOIN Location l WITH (NOLOCK) on cc.LocationID = l.GroupEXID

LEFT JOIN CostCenter c WITH (NOLOCK) on c.GroupexID = cc.CostCenterID

LEFT JOIN CorporateUnit cu WITH (NOLOCK) on cu.GroupexID = cc.BusinessUnitID

LEFT JOIN ImportedInstalledFileEvidencePath ifep WITH (NOLOCK) on ifep.ComplianceConnectionID = isd.ComplianceConnectionID

AND ifep.ExternalFilePathID = isd.ExternalFilePathID

-- LEFT OUTER JOIN ARLStagingFileEvidence arlsfe on arlsfe.FileEvidenceID = fe.FileEvidenceID

LEFT JOIN ComplianceConnection ccx WITH (NOLOCK) on isd.ComplianceConnectionID = ccx.ComplianceConnectionID

WHERE cc.ComplianceComputerStatusID = 1

-- Installer now

INSERT INTO #ApplicationsAndEvidences

SELECT

cc.ComputerName as 'Computer Name',

ife.DisplayName as EvidenceName,

ife.Version as version,

ife.Publisher,

NULL,

'N/A' as 'File Path',

isd.MatchedARLEvidence,

st.FullName,

stp.ProductName,

stv.VersionName,

ste.EditionName,

NULL, --Members are imported later

stc.DefaultValue,

cc.OperatingSystem as 'Operating System',

l.Path as 'Location Path',

c.Path as 'Cost Center Path',

cu.Path,

NULL as FileSize,

'Installer' as EvidenceType,

isd.EvidenceType as EvidenceSubType,

ccx.ConnectionNameDisplayName,

isd.InventoryDate,

sl.Name

FROM #ImportedInstalledInstallerEvidence isd WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = isd.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN SoftwareTitleVersion stv WITH (NOLOCK) on stv.SoftwareTitleVersionID = st.SoftwareTitleVersionID

LEFT JOIN SoftwareTitleEdition ste WITH (NOLOCK) on ste.SoftwareTitleEditionID = st.SoftwareTitleEditionID

LEFT JOIN SoftwareTitleClassificationI18N stc WITH (NOLOCK) on stc.SoftwareTitleClassificationID = st.SoftwareTitleClassificationID

LEFT JOIN ImportedInstallerEvidence ife on ife.ExternalInstallerID = isd.ExternalInstallerEvidenceID AND ife.ComplianceConnectionID = isd.ComplianceConnectionID

LEFT JOIN ImportedComputer icc WITH (NOLOCK) on icc.ExternalID = isd.ExternalID AND icc.ComplianceConnectionID = isd.ComplianceConnectionID

INNER JOIN ComplianceComputerCurrentUserScoped cc WITH (NOLOCK) on cc.ComplianceComputerID = icc.ComplianceComputerID -- apply access restriction

LEFT JOIN SoftwareLicense sl on sl.SoftwareLicenseID = isd.SoftwareLicenseID

LEFT JOIN Location l WITH (NOLOCK) on cc.LocationID = l.GroupEXID

LEFT JOIN CostCenter c WITH (NOLOCK) on c.GroupexID = cc.CostCenterID

LEFT JOIN CorporateUnit cu WITH (NOLOCK) on cu.GroupexID = cc.BusinessUnitID

LEFT JOIN ComplianceConnection ccx WITH (NOLOCK) on isd.ComplianceConnectionID = ccx.ComplianceConnectionID

WHERE cc.ComplianceComputerStatusID = 1

-- WMI Now

INSERT INTO #ApplicationsAndEvidences

SELECT

cc.ComputerName as 'Computer Name',

ife.ClassName + ' ' + ife.PropertyName + ' ' + ife.PropertyValue as EvidenceName,

'N/A' as version,

'N/A' as Publisher,

NULL,

'N/A' as 'File Path',

isd.MatchedARLEvidence,

st.FullName,

stp.ProductName,

stv.VersionName,

ste.EditionName,

NULL, --Members are imported later

stc.DefaultValue,

cc.OperatingSystem as 'Operating System',

l.Path as 'Location Path',

c.Path as 'Cost Center Path',

cu.Path,

NULL as FileSize,

'WMI' as EvidenceType,

NULL as EvidenceSubType,

ccx.ConnectionNameDisplayName,

isd.InventoryDate,

sl.Name

FROM #ImportedInstalledWMIEvidence isd WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = isd.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN SoftwareTitleVersion stv WITH (NOLOCK) on stv.SoftwareTitleVersionID = st.SoftwareTitleVersionID

LEFT JOIN SoftwareTitleEdition ste WITH (NOLOCK) on ste.SoftwareTitleEditionID = st.SoftwareTitleEditionID

LEFT JOIN SoftwareTitleClassificationI18N stc WITH (NOLOCK) on stc.SoftwareTitleClassificationID = st.SoftwareTitleClassificationID

LEFT JOIN ImportedWMIEvidence ife on ife.ExternalEvidenceID = isd.ExternalWMIEvidenceID AND ife.ComplianceConnectionID = isd.ComplianceConnectionID

LEFT JOIN ImportedComputer icc WITH (NOLOCK) on icc.ExternalID = isd.ExternalID AND icc.ComplianceConnectionID = isd.ComplianceConnectionID

INNER JOIN ComplianceComputerCurrentUserScoped cc WITH (NOLOCK) on cc.ComplianceComputerID = icc.ComplianceComputerID -- apply access restriction

LEFT JOIN SoftwareLicense sl on sl.SoftwareLicenseID = isd.SoftwareLicenseID

LEFT JOIN Location l WITH (NOLOCK) on cc.LocationID = l.GroupEXID

LEFT JOIN CostCenter c WITH (NOLOCK) on c.GroupexID = cc.CostCenterID

LEFT JOIN CorporateUnit cu WITH (NOLOCK) on cu.GroupexID = cc.BusinessUnitID

LEFT JOIN ComplianceConnection ccx WITH (NOLOCK) on isd.ComplianceConnectionID = ccx.ComplianceConnectionID

WHERE cc.ComplianceComputerStatusID = 1

-- Insert suites

-- Cleanup tables

DELETE FROM #ImportedInstalledFileEvidence

DELETE FROM #ImportedFileEvidenceMapping

DELETE FROM #SoftwareTitleFileEvidence

DELETE FROM #InstalledSoftware

DELETE FROM #ImportedInstalledInstallerEvidence

DELETE FROM #ImportedInstallerEvidenceMapping

DELETE FROM #SoftwareTitleInstallerEvidence

DELETE FROM #ImportedInstalledWMIEvidence

DELETE FROM #ImportedWMIEvidenceMapping

DELETE FROM #SoftwareTitleWMIEvidence

INSERT INTO #InstalledSoftware

SELECT

isd.ComplianceComputerID,

isr.ReplacedSoftwareTitleID,

isd.SoftwareTitleID,

isd.SoftwareLicenseID

FROM InstalledSoftwareReplacement isr

LEFT JOIN InstalledSoftwareData isd on isd.InstalledSoftwareID = isr.InstalledSoftwareID

LEFT JOIN SoftwareTitle SuiteTitle on SuiteTitle.SoftwareTitleID = isd.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct SuitetitleProduct on SuitetitleProduct.SoftwareTitleProductID = SuiteTitle.SoftwareTitleProductID

WHERE SuitetitleProduct.ProductName = @Productname

INSERT INTO #SoftwareTitleFileEvidence

SELECT

stfe.SoftwareTitleID,

stfe.FileEvidenceID

FROM SoftwareTitleFileEvidence stfe WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = stfe.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN FileEvidence fe WITH (NOLOCK) on stfe.FileEvidenceID = fe.FileEvidenceID

WHERE stfe.EvidenceExistenceRuleID IN (1,4) -- Mandatory and at least one

AND fe.Ignored = 0

AND stfe.SoftwareTitleID IN (SELECT SoftwareTitleID FROM #InstalledSoftware)

INSERT INTO #ImportedFileEvidenceMapping

SELECT

ifem.ExternalFileID,

ifem.ComplianceConnectionID,

ifem.FileEvidenceID,

stfe.SoftwareTitleID

FROM ImportedFileEvidenceMapping ifem WITH (NOLOCK)

INNER JOIN #SoftwareTitleFileEvidence stfe ON stfe.FileEvidenceID = ifem.FileEvidenceID

INSERT INTO #ImportedInstalledFileEvidence

SELECT

iife.ExternalFileID,

iife.ComplianceConnectionID,

iife.ExternalID,

ic.InventoryDate,

ifem.SoftwareTitleID,

isd.SoftwareLicenseID,

isd.SuiteSoftwareTitleID,

iife.ExternalFilePathID,

ISNULL (fe.FileName, '') + ' ' + ISNULL (fe.FileVersion, '') + ISNULL (' (' + fe.Company + ')', '')

FROM ImportedInstalledFileEvidence iife WITH (NOLOCK)

LEFT JOIN ImportedComputer ic WITH (NOLOCK) on ic.ExternalID = iife.ExternalID AND ic.ComplianceConnectionID = iife.ComplianceConnectionID

INNER JOIN #ImportedFileEvidenceMapping ifem WITH (NOLOCK) on ifem.ExternalFileID = iife.ExternalFileID AND ifem.ComplianceConnectionID = iife.ComplianceConnectionID

LEFT JOIN FileEvidence fe on fe.FileEvidenceID = ifem.FileEvidenceID

INNER JOIN #InstalledSoftware isd WITH (NOLOCK) on isd.ComplianceComputerID = ic.ComplianceComputerID

AND isd.SoftwareTitleID = ifem.SoftwareTitleID

INSERT INTO #SoftwareTitleInstallerEvidence

SELECT

stfe.SoftwareTitleID,

stfe.InstallerEvidenceID

FROM SoftwareTitleInstallerEvidence stfe WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = stfe.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN InstallerEvidence fe WITH (NOLOCK) on stfe.InstallerEvidenceID = fe.InstallerEvidenceID

WHERE fe.Ignored = 0

AND stfe.SoftwareTitleID IN (SELECT SoftwareTitleID FROM #InstalledSoftware)

INSERT INTO #ImportedInstallerEvidenceMapping

SELECT

ifem.ExternalInstallerID,

ifem.ComplianceConnectionID,

ifem.InstallerEvidenceID,

stfe.SoftwareTitleID

FROM ImportedInstallerEvidenceMapping ifem WITH (NOLOCK)

INNER JOIN #SoftwareTitleInstallerEvidence stfe ON stfe.InstallerEvidenceID = ifem.InstallerEvidenceID

INSERT INTO #ImportedInstalledInstallerEvidence

SELECT

iife.ExternalInstallerEvidenceID,

iife.ComplianceConnectionID,

iife.ExternalComputerID,

ic.InventoryDate,

ifem.SoftwareTitleID,

isd.SoftwareLicenseID,

isd.SuiteSoftwareTitleID,

ife.Evidence,

ISNULL (fe.DisplayName, '') + ' ' + ISNULL (fe.Version, '') + ' (' + fe.Publisher + ')'

FROM ImportedInstalledInstallerEvidence iife WITH (NOLOCK)

LEFT JOIN ImportedInstallerEvidence ife on ife.ExternalInstallerID = iife.ExternalInstallerEvidenceID

AND ife.ComplianceConnectionID = iife.ComplianceConnectionID

LEFT JOIN ImportedComputer ic WITH (NOLOCK) on ic.ExternalID = iife.ExternalComputerID AND ic.ComplianceConnectionID = iife.ComplianceConnectionID

INNER JOIN #ImportedInstallerEvidenceMapping ifem on ifem.ExternalInstallerEvidenceID = iife.ExternalInstallerEvidenceID AND ifem.ComplianceConnectionID = iife.ComplianceConnectionID

LEFT JOIN InstallerEvidence fe on fe.InstallerEvidenceID = ifem.InstallerEvidenceID

INNER JOIN #InstalledSoftware isd on isd.ComplianceComputerID = ic.ComplianceComputerID

AND isd.SoftwareTitleID = ifem.SoftwareTitleID -- enhances performance

--WMI for suites

INSERT INTO #SoftwareTitleWMIEvidence

SELECT

stfe.SoftwareTitleID,

stfe.WMIEvidenceID

FROM SoftwareTitleWMIEvidence stfe WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = stfe.SoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN WMIEvidence fe WITH (NOLOCK) on stfe.WMIEvidenceID = fe.WMIEvidenceID

WHERE fe.Ignored = 0

AND stp.ProductName = @Productname

INSERT INTO #ImportedWMIEvidenceMapping

SELECT

ifem.ExternalEvidenceID,

ifem.ComplianceConnectionID,

ifem.EvidenceRuleID,

stfe.SoftwareTitleID

FROM ImportedWMIEvidenceRuleMapping ifem WITH (NOLOCK)

INNER JOIN #SoftwareTitleWMIEvidence stfe ON stfe.WMIEvidenceID = ifem.EvidenceRuleID

INSERT INTO #ImportedInstalledWMIEvidence

SELECT DISTINCT

iife.ExternalEvidenceID,

iife.ComplianceConnectionID,

iife.ExternalComputerID,

ic.InventoryDate,

ifem.SoftwareTitleID,

isd.SoftwareLicenseID,

isd.SuiteSoftwareTitleID,

ISNULL (fe.ClassName, '') + ' ' + ISNULL (fe.PropertyName, '') + ISNULL ( fe.PropertyValue, '')

FROM ImportedInstalledWMIEvidence iife WITH (NOLOCK)

LEFT JOIN ImportedWMIEvidence ife on ife.ExternalEvidenceID = iife.ExternalEvidenceID

AND ife.ComplianceConnectionID = iife.ComplianceConnectionID

LEFT JOIN ImportedComputer ic WITH (NOLOCK) on ic.ExternalID = iife.ExternalComputerID AND ic.ComplianceConnectionID = iife.ComplianceConnectionID

INNER JOIN #ImportedWMIEvidenceMapping ifem on ifem.ExternalWMIEvidenceID = iife.ExternalEvidenceID AND ifem.ComplianceConnectionID = iife.ComplianceConnectionID

LEFT JOIN WMIEvidence fe on fe.WMIEvidenceID = ifem.WMIEvidenceID

INNER JOIN #InstalledSoftware isd on isd.ComplianceComputerID = ic.ComplianceComputerID

AND isd.SoftwareTitleID = ifem.SoftwareTitleID -- enhances performance

-- Insert the suites items now...

-- Files First

INSERT INTO #ApplicationsAndEvidences

SELECT

cc.ComputerName as 'Computer Name',

ife.FileName as EvidenceName,

ife.FileVersion as version,

ife.Company,

ife.Description,

ifep.ExternalFilePath as 'File Path',

isd.MatchedARLEvidence,

st.FullName,

stp.ProductName,

stv.VersionName,

ste.EditionName,

Compst.FullName,

stc.DefaultValue,

cc.OperatingSystem as 'Operating System',

l.Path as 'Location Path',

c.Path as 'Cost Center Path',

cu.Path,

ife.FileSize as FileSize,

'File' as EvidenceType,

'N/A' as EvidenceSubType,

ccx.ConnectionNameDisplayName,

isd.InventoryDate,

sl.Name

FROM #ImportedInstalledFileEvidence isd WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = isd.SuiteSoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN SoftwareTitleVersion stv WITH (NOLOCK) on stv.SoftwareTitleVersionID = st.SoftwareTitleVersionID

LEFT JOIN SoftwareTitleEdition ste WITH (NOLOCK) on ste.SoftwareTitleEditionID = st.SoftwareTitleEditionID

LEFT JOIN SoftwareTitleClassificationI18N stc WITH (NOLOCK) on stc.SoftwareTitleClassificationID = st.SoftwareTitleClassificationID

LEFT JOIN ImportedFileEvidence ife on ife.ExternalFileID = isd.ExternalFileID AND ife.ComplianceConnectionID = isd.ComplianceConnectionID

LEFT JOIN ImportedComputer icc WITH (NOLOCK) on icc.ExternalID = isd.ExternalID AND icc.ComplianceConnectionID = isd.ComplianceConnectionID

INNER JOIN ComplianceComputerCurrentUserScoped cc WITH (NOLOCK) on cc.ComplianceComputerID = icc.ComplianceComputerID -- apply access restriction

LEFT JOIN SoftwareLicense sl WITH (NOLOCK) on sl.SoftwareLicenseID = isd.SoftwareLicenseID

LEFT JOIN SoftwareTitle compst WITH (NOLOCK) on compst.SoftwareTitleID = isd.SoftwareTitleID

LEFT JOIN Location l WITH (NOLOCK) on cc.LocationID = l.GroupEXID

LEFT JOIN CostCenter c WITH (NOLOCK) on c.GroupexID = cc.CostCenterID

LEFT JOIN CorporateUnit cu WITH (NOLOCK) on cu.GroupexID = cc.BusinessUnitID

LEFT JOIN ImportedInstalledFileEvidencePath ifep WITH (NOLOCK) on ifep.ComplianceConnectionID = isd.ComplianceConnectionID

AND ifep.ExternalFilePathID = isd.ExternalFilePathID

-- LEFT OUTER JOIN ARLStagingFileEvidence arlsfe on arlsfe.FileEvidenceID = fe.FileEvidenceID

LEFT JOIN ComplianceConnection ccx WITH (NOLOCK) on isd.ComplianceConnectionID = ccx.ComplianceConnectionID

WHERE cc.ComplianceComputerStatusID = 1

-- Installer now

INSERT INTO #ApplicationsAndEvidences

SELECT

cc.ComputerName as 'Computer Name',

ife.DisplayName as EvidenceName,

ife.Version as version,

ife.Publisher,

NULL,

'N/A' as 'File Path',

isd.MatchedARLEvidence,

st.FullName,

stp.ProductName,

stv.VersionName,

ste.EditionName,

compst.FullName,

stc.DefaultValue,

cc.OperatingSystem as 'Operating System',

l.Path as 'Location Path',

c.Path as 'Cost Center Path',

cu.Path,

NULL as FileSize,

'Installer' as EvidenceType,

isd.EvidenceType as EvidenceSubType,

ccx.ConnectionNameDisplayName,

isd.InventoryDate,

sl.Name

FROM #ImportedInstalledInstallerEvidence isd WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = isd.SuiteSoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN SoftwareTitleVersion stv WITH (NOLOCK) on stv.SoftwareTitleVersionID = st.SoftwareTitleVersionID

LEFT JOIN SoftwareTitleEdition ste WITH (NOLOCK) on ste.SoftwareTitleEditionID = st.SoftwareTitleEditionID

LEFT JOIN SoftwareTitleClassificationI18N stc WITH (NOLOCK) on stc.SoftwareTitleClassificationID = st.SoftwareTitleClassificationID

LEFT JOIN SoftwareTitle compst WITH (NOLOCK) on compst.SoftwareTitleID = isd.SoftwareTitleID

LEFT JOIN ImportedInstallerEvidence ife on ife.ExternalInstallerID = isd.ExternalInstallerEvidenceID AND ife.ComplianceConnectionID = isd.ComplianceConnectionID

LEFT JOIN ImportedComputer icc WITH (NOLOCK) on icc.ExternalID = isd.ExternalID AND icc.ComplianceConnectionID = isd.ComplianceConnectionID

INNER JOIN ComplianceComputerCurrentUserScoped cc WITH (NOLOCK) on cc.ComplianceComputerID = icc.ComplianceComputerID -- apply access restriction

LEFT JOIN SoftwareLicense sl on sl.SoftwareLicenseID = isd.SoftwareLicenseID

LEFT JOIN Location l WITH (NOLOCK) on cc.LocationID = l.GroupEXID

LEFT JOIN CostCenter c WITH (NOLOCK) on c.GroupexID = cc.CostCenterID

LEFT JOIN CorporateUnit cu WITH (NOLOCK) on cu.GroupexID = cc.BusinessUnitID

LEFT JOIN ComplianceConnection ccx WITH (NOLOCK) on isd.ComplianceConnectionID = ccx.ComplianceConnectionID

WHERE cc.ComplianceComputerStatusID = 1

-- WMI

INSERT INTO #ApplicationsAndEvidences

SELECT

cc.ComputerName as 'Computer Name',

ife.ClassName + ' ' + ife.PropertyName + ' ' + ife.PropertyValue as EvidenceName,

'N/A' as version,

'N/A' as Publisher,

NULL,

'N/A' as 'File Path',

isd.MatchedARLEvidence,

st.FullName,

stp.ProductName,

stv.VersionName,

ste.EditionName,

compst.FullName, --Members are imported later

stc.DefaultValue,

cc.OperatingSystem as 'Operating System',

l.Path as 'Location Path',

c.Path as 'Cost Center Path',

cu.Path,

NULL as FileSize,

'WMI' as EvidenceType,

NULL as EvidenceSubType,

ccx.ConnectionNameDisplayName,

isd.InventoryDate,

sl.Name

FROM #ImportedInstalledWMIEvidence isd WITH (NOLOCK)

LEFT JOIN SoftwareTitle st WITH (NOLOCK) on st.SoftwareTitleID = isd.SuiteSoftwareTitleID

LEFT JOIN SoftwareTitleProduct stp WITH (NOLOCK) on stp.SoftwareTitleProductID = st.SoftwareTitleProductID

LEFT JOIN SoftwareTitleVersion stv WITH (NOLOCK) on stv.SoftwareTitleVersionID = st.SoftwareTitleVersionID

LEFT JOIN SoftwareTitleEdition ste WITH (NOLOCK) on ste.SoftwareTitleEditionID = st.SoftwareTitleEditionID

LEFT JOIN SoftwareTitleClassificationI18N stc WITH (NOLOCK) on stc.SoftwareTitleClassificationID = st.SoftwareTitleClassificationID

LEFT JOIN SoftwareTitle compst WITH (NOLOCK) on compst.SoftwareTitleID = isd.SoftwareTitleID

LEFT JOIN ImportedWMIEvidence ife on ife.ExternalEvidenceID = isd.ExternalWMIEvidenceID AND ife.ComplianceConnectionID = isd.ComplianceConnectionID

LEFT JOIN ImportedComputer icc WITH (NOLOCK) on icc.ExternalID = isd.ExternalID AND icc.ComplianceConnectionID = isd.ComplianceConnectionID

INNER JOIN ComplianceComputerCurrentUserScoped cc WITH (NOLOCK) on cc.ComplianceComputerID = icc.ComplianceComputerID -- apply access restriction

LEFT JOIN SoftwareLicense sl on sl.SoftwareLicenseID = isd.SoftwareLicenseID

LEFT JOIN Location l WITH (NOLOCK) on cc.LocationID = l.GroupEXID

LEFT JOIN CostCenter c WITH (NOLOCK) on c.GroupexID = cc.CostCenterID

LEFT JOIN CorporateUnit cu WITH (NOLOCK) on cu.GroupexID = cc.BusinessUnitID

LEFT JOIN ComplianceConnection ccx WITH (NOLOCK) on isd.ComplianceConnectionID = ccx.ComplianceConnectionID

WHERE cc.ComplianceComputerStatusID = 1

SELECT \*, 'To be added' as 'IsEmbedded Reason', @Productname as 'Product Input' FROM #ApplicationsAndEvidences

DROP TABLE #ApplicationsAndEvidences

DROP TABLE #ImportedInstalledFileEvidence

DROP TABLE #ImportedFileEvidenceMapping

DROP TABLE #SoftwareTitleFileEvidence

DROP TABLE #InstalledSoftware

DROP TABLE #ImportedInstalledInstallerEvidence

DROP TABLE #ImportedInstallerEvidenceMapping

DROP TABLE #SoftwareTitleInstallerEvidence

DROP TABLE #ImportedInstalledWMIEvidence

DROP TABLE #ImportedWMIEvidenceMapping

DROP TABLE #SoftwareTitleWMIEvidence

END

GO

DECLARE @ViewName nvarchar(64), @tenant int

SET @ViewName = 'Recognized Applications Transparency Report (NR)' -- Unique Name you want to display for object

SET @tenant = 1 -- Tenant for which this view applies

EXEC ComplianceCustomViewRegister

@TenantID = 1,

@ComplianceSavedSearchSystemID = NULL,

@SearchName = @ViewName,

@SearchNameResourceName = NULL,

@Description = '(v5) PLEASE ENTER A PRODUCT NAME IN THE SEARCH BAR. This report shows the list of installed application on active computers and the file, installer or WMI evidence that allowed to recognize them. Please note that if an application is recognized through two evidence, the installation will have two rows in the report.',

@DescriptionResourceName = NULL,

@SearchGridLayout = NULL,

@SearchXML = NULL,

@SearchSQL = 'EXEC dbo.nr\_EvidencesAndRecognizedApplicationsFromProduct @Productname = @SearchText',

@SearchSQLConnection = 'Live',

@SearchMapping = NULL,

@ComplianceSearchType = 'Custom',

@ComplianceSearchFolderSystemID = -17,--will create the view at the root folder. Pick up the folder ID you wish

@CanDelete = 1,

@CanChangeMasterObject = 0