Table of Contents

[1. Old Connector 4](#_Toc11336878)

[2. Neuer Connector 5](#_Toc11336879)

[2.1. 400\_rayventory\_devices 5](#_Toc11336880)

[2.2. 400\_rayventory\_devices\_chassis 6](#_Toc11336881)

[2.3. 499\_rayventory\_devices\_relations 6](#_Toc11336882)

[2.4. 500\_rayventory\_inv\_raw\_arp 6](#_Toc11336883)

[2.5. 500\_rayventory\_inv\_raw\_file 6](#_Toc11336884)

[2.6. 500\_rayventory\_inv\_raw\_generic 7](#_Toc11336885)

[2.7. 500\_rayventory\_inv\_raw\_msi 7](#_Toc11336886)

[3. SQL Statements 7](#_Toc11336887)

[3.1. FQDN 7](#_Toc11336888)

[3.2. Powerstate 9](#_Toc11336889)

[3.3. ISO Tag 10](#_Toc11336890)

[3.4. ISO Tag and all Files in one 11](#_Toc11336891)

[4. Liste of Functions 12](#_Toc11336892)

[4.1. Base 12](#_Toc11336893)

[4.1.1. CIMBoolean(sValue) 12](#_Toc11336894)

[4.1.2. CIMUniqueidentifier(sValue) 12](#_Toc11336895)

[4.1.3. f\_SQLDateTime(TimeStamp) 13](#_Toc11336896)

[4.1.4. f\_StringSplit(TimeStamp) 14](#_Toc11336897)

[4.1.5. f\_GetInventoryDate(iType) 14](#_Toc11336898)

[4.1.6. f\_GetLastInventoryDate(iType) 15](#_Toc11336899)

[4.2. Device 15](#_Toc11336900)

[4.2.1. f\_GetDeviceFQDN(@bOnlyComputer) 15](#_Toc11336901)

[4.2.2. f\_GetDeviceChassis(bAll) 16](#_Toc11336902)

[4.2.3. f\_GetDeviceDiskDriveBase() 17](#_Toc11336903)

[4.2.4. f\_GetDeviceDiskDrive() 17](#_Toc11336904)

[4.2.5. f\_GetDeviceLogicalDisk() 18](#_Toc11336905)

[4.2.6. f\_GetDeviceMemoryModul() 18](#_Toc11336906)

[4.2.7. f\_GetDeviceNetworkInterface() 19](#_Toc11336907)

[4.2.8. f\_GetDeviceNetworkInfo() 19](#_Toc11336908)

[4.2.9. f\_GetDeviceProcessors() 20](#_Toc11336909)

[4.2.10. f\_GetDeviceSerialNumber() 20](#_Toc11336910)

[4.2.11. f\_GetDeviceVideoController() 21](#_Toc11336912)

[4.2.12. f\_GetDeviceModel() 21](#_Toc11336915)

[4.2.13. f\_GetDeviceInfo() 22](#_Toc11336917)

[4.2.14. f\_GetDeviceOSLanguage() 22](#_Toc11336918)

[4.2.15. f\_GetDeviceOSLastLoginUser() 23](#_Toc11336919)

[4.2.16. f\_GetDeviceOSInfo() 24](#_Toc11336920)

[4.2.17. Deprecated Functions 25](#_Toc11336921)

[4.3. Virtualization 27](#_Toc11336922)

[4.3.1. VMware 27](#_Toc11336923)

[4.3.2. Hyper-V 30](#_Toc11336924)

[4.3.3. Sun/Oracle Solaris Zone 32](#_Toc11336925)

[4.4. Software 34](#_Toc11336926)

[4.4.1. f\_SoftwareValuesByPropertyName(@sProperty) 34](#_Toc11336927)

[4.4.2. f\_GetSoftwareFileUsage () 34](#_Toc11337001)

[4.4.3. f\_GetSoftwareTagFiles() 35](#_Toc11337002)

[4.4.4. f\_GetSoftwareFiles(iType) 36](#_Toc11337003)

[4.4.5. f\_GetSoftwareMicrosoftSQL() 37](#_Toc11337004)

[4.4.6. f\_GetSoftwareMicrosoftExchange() 38](#_Toc11337005)

[4.4.7. f\_GetSoftwareMicrosoftSharePoint() 38](#_Toc11337006)

[4.5. Oracle 39](#_Toc11337007)

[4.5.1. f\_OracleRAC 39](#_Toc11337008)

[4.5.2. f\_OracleNodes 39](#_Toc11337009)

[4.5.3. f\_OracleInstances(@InstanceID) 40](#_Toc11337010)

[4.5.4. f\_OracleDFUS (@DBObjectID) 41](#_Toc11337011)

[4.5.5. f\_OracleOptions (@InstanceID, @ClassName) 42](#_Toc11337012)

[4.5.6. f\_OracleUserCounts (@IncludeSystemUser) 43](#_Toc11337013)

[4.5.7. f\_OracleAccessCheck (@All) 43](#_Toc11337014)

[4.5.8. f\_OracleFailures () 44](#_Toc11337015)

[4.5.9. v\_OracleReport 44](#_Toc11337016)

[5. List of Store Proceduren 45](#_Toc11337017)

[5.1. Base 45](#_Toc11337018)

[5.1.1. sp\_GetVirtualVMwareLinkGuest 45](#_Toc11337019)

[5.1.2. sp\_GetVirtualHyperVLinkGuest @bWithSerial 47](#_Toc11337020)

[5.1.3. sp\_GetSoftwareInstallInformation @iEvidenceGroup, @sEvidenceFilter, @bNoUpdate, @iLastSeenDays 49](#_Toc11337021)

[5.1.4. sp\_GetSoftwareFiles @iType, @noUsageData 51](#_Toc11337022)

[5.2. Oracle Reports 53](#_Toc11337023)

[5.2.1. sp\_OracleDFUS @DBObjectID 53](#_Toc11337024)

[5.2.2. sp\_OracleOption @DBObjectID 54](#_Toc11337025)

[5.2.3. sp\_OracleFailures @ComputerCN 55](#_Toc11337026)

[5.2.4. sp\_OracleRaynetOptionList @DBObjectID 56](#_Toc11337027)

[5.2.5. sp\_OracleScriptOptionList @DBObjectID 57](#_Toc11337028)

[5.2.6. sp\_rp\_OracleOptionList @DBObjectID, @ComputerID, @OnlyProduct 58](#_Toc11337029)

[5.2.7. sp\_rp\_OracleVirtualInfrastructure @DBObjectID 59](#_Toc11337030)

[5.2.8. sp\_rp\_OracleReport @ComputerCN, @InstanceName 60](#_Toc11337031)

[5.3. Aspera Connector 62](#_Toc11337032)

[5.3.1. csp\_aspera\_connector\_device 62](#_Toc11337033)

[5.3.2. csp\_aspera\_connector\_device\_types 64](#_Toc11337034)

[5.3.3. csp\_aspera\_connector\_device\_relation 65](#_Toc11337035)

[5.3.4. csp\_aspera\_connector\_device\_relation\_types 66](#_Toc11337036)

[5.3.5. csp\_aspera\_connector\_software\_arp 67](#_Toc11337037)

[5.3.6. csp\_aspera\_connector\_software\_msi 68](#_Toc11337038)

[5.3.7. csp\_aspera\_connector\_software\_file 69](#_Toc11337039)

[5.3.8. csp\_aspera\_connector\_software\_tag 70](#_Toc11337040)

[5.3.9. csp\_aspera\_connector\_software\_generic\_othersw 71](#_Toc11337041)

[5.3.10. csp\_aspera\_connector\_software\_generic\_os 72](#_Toc11337042)

[5.3.11. csp\_aspera\_connector\_software\_generic\_microsoft 73](#_Toc11337043)

[5.3.12. csp\_aspera\_connector\_software\_generic\_oracle 74](#_Toc11337044)

[6. Configuration 75](#_Toc11337045)

[6.1. Oracle 75](#_Toc11337046)

# Old Connector

|  |  |  |
| --- | --- | --- |
| Connector | Description | ToDo |
| 400\_rayventory\_devices | Import Device Information | Change to Store Procedure, like Export from Bertelsmann |
| 499\_rayventory\_devices\_relations | Import Virtual Device Relation | Add new routines for HyperV like Export Bertelsmann |
| 500\_rayventory\_inv\_raw\_arp | Import ARP Software Information | Change to Store Procedure, like Export from Bertelsmann |
| 500\_rayventory\_inv\_raw\_file | Import File Software Information | Change to Store Procedure, like Export from Bertelsmann |
| 500\_rayventory\_inv\_raw\_generic | Import following:   * Other Software Evidenes then ARP, File, MSI * Operatimg System * Oracle Database * Oracle Feature and Options * Microsoft SQL | Here the following should be added:   * Microsoft SQL Reporting * Microsoft SQL Agent * Microsoft SQL Analyse Service * Microsoft Exchange * Microsoft Share Point   This should be changed to a Store Procedure like Export Bertelsmann |

# Neuer Connector

## 400\_rayventory\_devices

In the New Connector the Information from RayVentory should be in a Store Procedure like csp\_bertelsmann\_export\_device. The Store Procedure from Bertelsmann is the basis for Import to Aspera by Bertelsmann.

Following Points are there other then in the actual connector

|  |  |
| --- | --- |
| What | Should be Implemented |
| If CSName exists, the CSName is used as ComputerCN else ComputerCN from Computer is used.  This is needed, why some times the ComputerCN has the IP Address and not the Hostname | Yes |
| FQDN will created by:   * If CIM\_UnitaryComputerSystem Domain exits, Hostname + ‘.’ + this Value * If Computer.ComputerCN has points, then this is used as FQDN * If Organisation Information exists, Computer.ComputerCN + ‘.’ + Organization Information (With Replaces) * If NetworkDiscovery.DNSFullname exitsts, NetworkDiscovery.DNSFullname is used * If Win32\_NetworkAdapterConfiguration DNSDomain exist, then Computer.ComputerCN + ‘.’ + this value * If no other then Hostname   This is the best way to get the FQDN, depending on information what we get from systems | Yes |
| Double Machine Entries will be only show one time, it should use the latest inventory | Yes |
| If a Machine is Virtual and on two or more esx Server and one of this is powered on and the other are powered off, all Machine will be created.  The Power On Machine get as ID the FQDN or DeviceName  The Power Off Machine get as ID the {FQDN}\_{0…x} or {DeviceName}\_{0…x}  Sample  machinea.temp.test\_0 for the first Power Off Machine  machinea.temp.test\_1 for the second Power Off Machine  Is in Bertelsmann at the moment only for VMWare, HyperV will be good | Yes |
| The most Hardware Information comes from f\_System\_Base()  This should be the Base for the new | Yes |
| All Queries are in a Store Procedure and give at the end the complete list of devices | Yes |

At the moment the 400\_rayventory\_devices Connector import more the Devices, out of my opinion should be changed in different parts.

|  |  |  |
| --- | --- | --- |
| Part | At the Moment | New |
| Device Information | 400\_rayventory\_devices | 401\_rayventory\_devices |
| Device Types (Chassis Type) | 400\_rayventory\_devices | 400\_rayventory\_devices\_chassis |

Import Process 400\_rayventory\_operating\_systems\_insert in the connector should be updated for Ubuntu and Debian.

## 400\_rayventory\_devices\_chassis

This exists not at the moment. It is part of 400\_rayventory\_devices, see 400\_Rayventory\_devices.

## 499\_rayventory\_devices\_relations

This create the Relation Table for Virtual Devices. We need to talk with Aspera how the Relation should come. Why by Bertelsmann we need to send this in other way then it is at the moment in the connector. After this we should define this.

One XML per relation (e.g. Hyper-V, VMWare, LPAR,…)

## 500\_rayventory\_inv\_raw\_arp

This works fine, but we should change this to the way how it will created in store procedure csp\_bertelsmann\_export\_software\_arp what we create for Bertelsmann. And use Store Procedure in the Connector

## 500\_rayventory\_inv\_raw\_file

This works fine, but we should change this to the way how it will created in store procedure csp\_bertelsmann\_export\_software\_file what we create for Bertelsmann. And use Store Procedure in the Connector

Following should be added, if Aspera can:

|  |  |
| --- | --- |
| Point | Description |
| MD5 | MD5 Checksum if we have, like SWID Tag Files |
| Content | If we have the Content of a file, like SWID Tag Files |

We can talk to Aspera but Andreas assumes that there is no possibility

Idea: Gathering relevant file information from ISO-files. Discussion with Aspera 🡪 Best-Practice

## 500\_rayventory\_inv\_raw\_generic

This should change in the way how it will created in store procedure csp\_bertelsmann\_export\_software\_generic what we create for Bertelsmann. This store procedure has all the old stuff in it and the new Points

* Microsoft SQL Reporting
* Microsoft SQL Analyse Service
* Microsoft Exchange
* Microsoft Share Point

For the Product:

* Microsoft SQL Agent

Is not in the Procedure of Bertelsmann, but should be in the connector

Discussion with Aspera if SQL agent is relevant/required

## 500\_rayventory\_inv\_raw\_msi

This works fine, but we should change this to the way how it will created in store procedure csp\_bertelsmann\_export\_software\_msi what we create for Bertelsmann. And use Store Procedure in the Connector

# SQL Statements

## FQDN

With this Satement you get the FQDN Information

SELECT

c.ComputerID,

LOWER(COALESCE(opCSName.Value, c.ComputerCN)) AS ComputerCN,

LOWER(CASE

WHEN baseDom.Value IS NOT NULL

AND baseDom.Value != 'WORKGROUP'

THEN COALESCE(opCSName.Value, c.ComputerCN) + '.' + baseDom.Value

WHEN CHARINDEX('.', c.ComputerCN) > 0 THEN c.ComputerCN

WHEN CHARINDEX(',', d.DN) > 1

THEN c.ComputerCN + '.' + REPLACE(REPLACE(d.DN, 'DC=', ''), ',', '.')

WHEN dbo.CIMString(nd.DNSFullName) IS NOT NULL THEN nd.DNSFullName

WHEN dns.DNSSuffix IS NOT NULL THEN c.ComputerCN + '.' + dns.DNSSuffix

ELSE COALESCE(opCSName.Value, c.ComputerCN)

END) FQDN

FROM Computer c WITH (NOLOCK)

LEFT JOIN Organization o WITH (NOLOCK)

ON c.ComputerOUID = o.OrganizationID

LEFT JOIN Domain d WITH (NOLOCK)

ON o.DomainID = d.OrganizationID

LEFT JOIN (SELECT ComputerID, MAX(DNSFullName) AS DNSFullName

FROM NetworkDevice nd WITH (NOLOCK) GROUP BY ComputerID) nd

ON c.ComputerID = nd.ComputerID

LEFT JOIN f\_HardwareProperties('CIM\_UnitaryComputerSystem') baseDom

ON c.ComputerID = baseDom.ComputerID

AND Property = 'Domain'

LEFT JOIN (SELECT

nic.ComputerID,

MAX(dns.Value) AS DNSSuffix

FROM f\_HardwareProperties('Win32\_NetworkAdapterConfiguration') nic

LEFT JOIN f\_HardwareValuesByPropertyName('DNSDomain') dns

ON nic.HardwareObjectID = dns.HardwareObjectID

WHERE nic.Property = 'IPEnabled' AND nic.Value = 'TRUE' AND dns.Value IS NOT NULL

GROUP BY nic.ComputerID) dns

ON c.ComputerID = dns.ComputerID

LEFT JOIN f\_HardwareProperties('CIM\_OperatingSystem') opCSName

ON opCSName.Property = 'CSNAME'

AND c.ComputerID = opCSName.ComputerID

## Powerstate

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

DROP TABLE #tmpVMwarePowerstate

SELECT

GuestComputerID,

PowerState,

CAST(null AS INT) AS [Count]

INTO #tmpVMwarePowerstate

FROM f\_Virtual\_VMware\_Guest(0) c

UPDATE #tmpVMwarePowerstate

SET [Count] = x2.[Count]

FROM #tmpVMwarePowerstate x

LEFT JOIN(SELECT tmp.GuestComputerID, COUNT(\*) AS COUNT FROM #tmpVMwarePowerstate tmp GROUP BY GuestComputerID) AS x2

ON x.GuestComputerID = x2.GuestComputerID

Sample to create Powerstate ID’s

SELECT

LEFT(COALESCE(fqdn.fqdn, c.ComputerName)

+ CASE

WHEN pst.[Count] > 1 THEN '\_' + CAST(pst.row# -1 AS NVARCHAR(2))

ELSE ''

END, 125) AS import\_id -- FQDN, ComputerName

-- PowerOff + \_0 , \_1, \_2

FROM f\_System\_Base() c

LEFT JOIN #tmpDeviceFQDN\_device fqdn

ON c.ComputerID = fqdn.ComputerID

LEFT JOIN (SELECT

GuestComputerID,

PowerState,

[Count],

ROW\_NUMBER() OVER(PARTITION BY GuestComputerID

ORDER BY GuestComputerID, PowerState DESC) [row#]

FROM #tmpVMwarePowerstate) pst

ON c.ComputerID = pst.GuestComputerID

## ISO Tag

WITH IsoTAG (SoftwareIsoTagFileID, Publisher, ProductName, ProductVersion,

SoftwareUniqueID, SoftwareTagID, LicensorID)

AS (SELECT

itf.SoftwareIsoTagFileID,

itfx.x.value('(/\*:SoftwareIdentity/\*:Entity/@name)[1]',

'nvarchar(256)') AS Publisher,

itfx.x.value('(/\*:SoftwareIdentity/@name)[1]',

'nvarchar(256)') AS ProductName,

itfx.x.value('(/\*:SoftwareIdentity/@version)[1]',

'nvarchar(256)') AS ProductVersion,

itfx.x.value('(/\*:SoftwareIdentity/@uniqueId)[1]',

'nvarchar(256)') AS SoftwareUniqueID,

itfx.x.value('(/\*:SoftwareIdentity/@tagId)[1]',

'nvarchar(256)') AS SoftwareTagID,

itfx.x.value('(/\*:SoftwareIdentity/\*:Entity/@regid)[1]',

'nvarchar(256)') AS LicensorID

FROM SoftwareIsoTagFile itf WITH (NOLOCK)

CROSS APPLY (SELECT CAST(CAST(

REPLACE(itf.TagContent,

'UTF-8', 'ISO-8859-1') AS VARCHAR(MAX)) AS XML) x) itfx

WHERE itf.TagContent LIKE '%SoftwareIdentity%')

SELECT

sf.ComputerID,

LEFT(sfp.Path, 255) AS FilePath,

sfn.Name AS [FileName],

sf.Size AS FileSize,

IIF(itf.MD5 = 'NO\_MD5', NULL, itf.MD5) AS FileMD5,

ISNULL(CONVERT(NVARCHAR(10),sf.DateTime,104),'') AS FileDate,

COALESCE(ite.Name, IsoTAG.Publisher, itf.OriginalArpPublisher, '')

AS Publisher,

COALESCE(itsv.ProductTitle, IsoTAG.ProductName, itf.OriginalArpDisplayName, '')

AS Product,

COALESCE(itsv.ProductVersionName, IsoTAG.ProductVersion,

itf.OriginalArpDisplayVersion, '') AS Version,

itf.OriginalArpGuid AS ProductGuid,

COALESCE(itsv.SoftwareUniqueID, IsoTAG.SoftwareUniqueID, IsoTAG.SoftwareTagID)

AS ProductUniqueID,

COALESCE(ite.RegID, IsoTAG.LicensorID) AS ProductLicensorID,

itf.TagContent AS SWIDTagContent

FROM SoftwareIsoTagFile itf WITH (NOLOCK)

INNER JOIN SoftwareFile sf WITH (NOLOCK)

ON itf.SoftwareIsoTagFileID = sf.SoftwareIsoTagFileID

LEFT JOIN dbo.SoftwareFilePath sfp WITH (NOLOCK)

ON sfp.SoftwareFilePathID = sf.SoftwareFilePathID

LEFT JOIN dbo.SoftwareFileName sfn WITH (NOLOCK)

ON sfn.SoftwareFileNameID = sf.SoftwareFileNameID

LEFT JOIN SoftwareIsoTagSoftwareVersion itsv WITH (NOLOCK)

ON itf.SoftwareIsoTagSoftwareVersionID = itsv.SoftwareIsoTagSoftwareVersionID

LEFT JOIN SoftwareIsoTagEntity ite WITH (NOLOCK)

ON itf.SoftwareCreatorEntityID = ite.SoftwareIsoTagEntityID

LEFT JOIN SoftwareVersion sv WITH (NOLOCK)

ON sf.SoftwareID = sv.SoftwareID

LEFT JOIN SoftwareOccurrence so WITH (NOLOCK)

ON sf.ComputerID = so.ComputerID

AND sf.SoftwareID = so.SoftwareID

LEFT JOIN SoftwareDetails sd WITH (NOLOCK)

ON so.SoftwareDetailsID = sd.SoftwareDetailsID

LEFT JOIN IsoTAG

ON itf.SoftwareIsoTagFileID = IsoTAG.SoftwareIsoTagFileID

## ISO Tag and all Files in one

WITH IsoTAG (SoftwareIsoTagFileID, Publisher, ProductName, ProductVersion,

SoftwareUniqueID, SoftwareTagID, LicensorID)

AS (SELECT

itf.SoftwareIsoTagFileID,

itfx.x.value('(/\*:SoftwareIdentity/\*:Entity/@name)[1]',

'nvarchar(256)') AS Publisher,

itfx.x.value('(/\*:SoftwareIdentity/@name)[1]',

'nvarchar(256)') AS ProductName,

itfx.x.value('(/\*:SoftwareIdentity/@version)[1]',

'nvarchar(256)') AS ProductVersion,

itfx.x.value('(/\*:SoftwareIdentity/@uniqueId)[1]',

'nvarchar(256)') AS SoftwareUniqueID,

itfx.x.value('(/\*:SoftwareIdentity/@tagId)[1]',

'nvarchar(256)') AS SoftwareTagID,

itfx.x.value('(/\*:SoftwareIdentity/\*:Entity/@regid)[1]',

'nvarchar(256)') AS LicensorID

FROM SoftwareIsoTagFile itf WITH (NOLOCK)

CROSS APPLY (SELECT CAST(CAST(

REPLACE(itf.TagContent,

'UTF-8', 'ISO-8859-1') AS VARCHAR(MAX)) AS XML) x) itfx

WHERE itf.TagContent LIKE '%SoftwareIdentity%')

SELECT

sf.ComputerID,

sfp.Path AS FilePath,

sfn.Name AS [FileName],

sf.Size AS FileSize,

CASE

WHEN sf.MD5 = 'NO\_MD5' THEN NULL

WHEN itf.MD5 = 'NO\_MD5' THEN NULL

ELSE COALESCE(sf.MD5, itf.MD5)

END AS FileMD5,

ISNULL(CONVERT(NVARCHAR(10),sf.DateTime,104),'') AS FileDate,

COALESCE(ite.Name, IsoTAG.Publisher,

itf.OriginalArpPublisher,sf.CompanyName) AS Publisher,

COALESCE(itsv.ProductTitle, IsoTAG.ProductName, itf.OriginalArpDisplayName,

sf.FileDescription) AS Product,

COALESCE(itsv.ProductVersionName, IsoTAG.ProductVersion,

itf.OriginalArpDisplayVersion, sf.FileVersion) AS Version,

itf.OriginalArpGuid AS ProductGuid,

COALESCE(itsv.SoftwareUniqueID, IsoTAG.SoftwareUniqueID,

IsoTAG.SoftwareTagID) AS SwidUniqueId,

COALESCE(ite.RegID, IsoTAG.LicensorID) AS SwidLicensorId,

itf.TagContent AS SWIDTagContent

FROM SoftwareFile sf WITH (NOLOCK)

LEFT JOIN SoftwareIsoTagFile itf WITH (NOLOCK)

ON itf.SoftwareIsoTagFileID = sf.SoftwareIsoTagFileID

LEFT JOIN dbo.SoftwareFilePath sfp WITH (NOLOCK)

ON sfp.SoftwareFilePathID = sf.SoftwareFilePathID

LEFT JOIN dbo.SoftwareFileName sfn WITH (NOLOCK)

ON sfn.SoftwareFileNameID = sf.SoftwareFileNameID

LEFT JOIN SoftwareIsoTagSoftwareVersion itsv WITH (NOLOCK)

ON itf.SoftwareIsoTagSoftwareVersionID = itsv.SoftwareIsoTagSoftwareVersionID

LEFT JOIN SoftwareIsoTagEntity ite WITH (NOLOCK)

ON itf.SoftwareCreatorEntityID = ite.SoftwareIsoTagEntityID

LEFT JOIN SoftwareVersion sv WITH (NOLOCK)

ON sf.SoftwareID = sv.SoftwareID

LEFT JOIN IsoTAG

ON itf.SoftwareIsoTagFileID = IsoTAG.SoftwareIsoTagFileID

-- WHERE itf.SoftwareIsoTagFileID > 0 OR LEN(sf.FileDescription) > 0 -- Only if Product Name exist

# Liste of Functions

## Base

### CIMBoolean(sValue)

Function get Boolean Value out of string

Parameter bValue:

* Boolean Value as String
  + True, 1, Yes is True
  + False, 0 No is False
  + All other is Null

Return:

BIT Value

You can as well use no or yes in following Languages:

* German, French, Italien, Spanish, Portoguese, Dutch, Turkish, Polish

Used by:

* Always to convert Boolean Strings to BIT Value

### CIMUniqueidentifier(sValue)

Function get Guid Value out of string

Parameter bValue:

* + Guid in String Form

Return:

Uniqueidentifier Value

Better to use TRY\_CAST(<Value> AS UNIQUEIDENTIFIER) this is faster

### f\_SQLDateTime(TimeStamp)

Function get DateTime from String.

It Understand following Date Formats:

* <YYYYMMDD>T<HHmmss>
* <YYYY-MM-DD>T<HH:mm:ss>
* DD.MM.YYYY HH:mm:ss
* DD.MM.YYYY HH:mm:ss\_

If the Date is older then 1900 it will set NULL.  
The last Convertation is a try\_convert.

This should work with the most time stamps in RayVentory Inventory Data like:

* Oracle DB
* Microsoft Windows
* Microsoft SQL

Parameter TimeStamp

* Time String

Return:

* DateTime Value

Used by:

* Always if a Datetime conversation are used.

### f\_StringSplit(TimeStamp)

Function Split String into a Table

Parameter sString

* String to Split

Parameter sSeperator

* Seperator to Split the String

Return:

* Return the Values per Row in a Table

### f\_GetInventoryDate(iType)

Function to get Inventory Date

Parameter iType

* null = from All Inventory Report Entries the MAX Date of HW SW oder FileDate
* 1 = Nur Machine Inventories MAX Date of HW SW oder FileDate
* 2 = Nur User Inventories MAX Date of HW SW oder FileDate

Return

* ComputerID
* UserID
* InventoyDate

Used by:

* f\_GetLastInventoryDate

### f\_GetLastInventoryDate(iType)

Function to get Last Inventory Date

Parameter iType

* null = from All Inventories the newest InventoryDate per ComputerID
* 1 = from all Machine Inventories the newest InventoryDate per ComputerID
* 2 = from all User Inventories the newest InventoryDate per ComputerID   
   (Last User per Inventory Report)

Return

* ComputerID
* UserID
* LastInventoyDate

Used by:

* All functions and store procedure in this document where we need LastInventoryDate like:
  + csp\_aspera\_....
  + sp\_GetSoftwareInstallInformation

## Device

### f\_GetDeviceFQDN(@bOnlyComputer)

Function to get all Devices with FQDN Information and Filtered on newest Inventory

Parameter @bOnlyComputer:

* 1 Only Machines with Computer Inventories
* 0 or null All Inventories

Return:

* ComputerID
* ComputerCN
* ComputerFQDN

Used by:

* All function and store procedure in this document where we need FQDN information like:
  + csp\_aspera\_...
  + some Oracle functions and store procedures

### f\_GetDeviceChassis(bAll)

Function to get all Device Chassis Information

Parameter bAll:

* 1 = all Chassis
* 0 or null = only lowest of device

Return:

* ComputerID
* TypeID
* Type

Used by:

* csp\_aspera\_connector\_device
* later for use with USU\_Export

### f\_GetDeviceDiskDriveBase()

Function to get all Disk Drive Objects unfiltered and can have Disk Drives and Partition Information

Return:

* ComputerID
* Class DiskDrive Object Class, like WIN32\_DiskDrive
* CIMDiskDriveID Hardware Object ID from CIM\_DiskDrive Object
* DiskDriveID This is DiskID for the Mappering to Logical Partition.
* DiskNumber Physical Index of the Disk Drive
* Manufacturer
* Model
* SerialNumber
* Interface
* TotalCapacity (in byte)

Used by:

* f\_GetDeviceDiskDrive

### f\_GetDeviceDiskDrive()

Function to get all Disk Drives. Partitions will be filtered out.

Return:

* ComputerID
* CIMDIskDriveID
* DiskDirveID
* DiskNumber
* Manufacturer
* Model
* SerialNumber
* Interface
* TotalCapacity (in byte)

Used by:

* csp\_aspera\_connector\_device
* later for use with USU\_Export
* later for sp\_rp\_infrastructure

### f\_GetDeviceLogicalDisk()

Function to get all Partitions from DiskDrives

Return:

* ComputerID
* CIMLogicalDiskID
* CIMDiskDriveID
* DriveDriveID
* DiskNumber
* PartitionNumber
* InterfaceName
* DriveLetter
* MediaType
* Size (in byte)
* FreeSpace (in byte)
* FileSystem
* VolumeName
* VolumeSerialNumber

Used by:

* later for use with USU\_Export

### f\_GetDeviceMemoryModul()

Function to get all Memory Modul Information

Return:

* ComputerID
* HardwareObjectID
* BankLocator (Slot of Modul)
* Manufacturer
* Size (in Bytes)
* Speed (in Mhz)
* Type
* TypeDetail
* SerialNumber
* PartNumber

Used by:

* csp\_aspera\_connector\_device
* later for use with USU\_Export
* later for sp\_rp\_Infrastrucre

### f\_GetDeviceNetworkInterface()

Function get all Network interfaces

Return:

* ComputerID
* InterfaceName
* Manufacturer
* Model
* MACAddress
* IPEnabled
* DHCPEnabled
* IPAddress
* IPAddressV6
* Subnetmask
* Broadcast
* DefaultGateway
* DNSServer
* DNSDomain
* DNSHostname
* Speed

Used by:

* f\_GetDeviceNetworkInfo
* later for use with USU Export

### f\_GetDeviceNetworkInfo()

Function get all IP Address and Macs separate with space, one Manufacturer and Model per Device

Return:

* ComputerID
* Manufacturer MAX of Manfucaturer
* Model MAX of NIC Model
* IPAddress All IP Addresses
* MACAddress All MAC Addresses

Used by:

* csp\_aspera\_connector\_device
* later for use with sp\_rp\_infrastructure

### f\_GetDeviceProcessors()

Get all Processor Informations

Return:

* ComputerID
* DeviceID
* Name (of the Processor)
* Manufacturer
* ClockSpeed in MHz
* L2CacheSize in KB
* CurrentVoltage
* CPUs Count of CPUs
* Cores Count of Cores per CPU
* Logical Count of Logical Cores per CPU
* TotalCPUs Total count of CPUs (Is the same as CPUs)
* TotalCores Total count of Cores (Is CPUs \* Cores)
* TotalLogical Total count of Logical cores (Is CPUs \* Logical)
* HyperThreading HyperThreading used

1 If Cores\*2 = Logical and Manufacturer Intel or AMD

0 off,

Used by:

* csp\_aspera\_connector\_device
* sp\_rp\_OracleVirtualInfrastructure
* later for use with USU\_Export
* later for use with sp\_rp\_infrastructure

### f\_GetDeviceSerialNumber()

Function to get all SerialNumbers from the Devices

Return:

* ComputerID
* CIMBIOSElementID HardwareObjectID from Class CIM\_BIOSElement
* CIMProductID HardwareObjectID from Class CIM\_Product
* SerialNumber

Used by:

* f\_GetDeviceVirtualVMwareUUID
* later for use with USU\_Export

### f\_GetDeviceVideoController()

Function get all Video Controllers

Return:

* ComputerID
* CIMPCVideoControllerID HardwareObjectID from Class CIM\_VideoController
* Manufacturer
* Model (Drivername)
* VideoProcessor
* AdapeterRAM (in Byte)
* DriverVersion
* DriverDate (as DateTime)

Used by:

* csp\_aspera\_connector\_device
* later for use with USU\_Export

### f\_GetDeviceModel()

Function to Get Model and Manufacturer of the Hardware and Hardware Type

Return:

* ComputerID
* CIMProductID HardwareObectID from CIMProduct
* Manufacturer
* Model
* Virtual 1 = Virtualmachine, 0 = Physic

Used by:

* csp\_aspera\_connector\_device
* f\_GetVirtualZoneHostInfo
* f\_GetDeviceInfo
* sp\_rp\_OracleReport
* later for use with f\_GetVirtualLPARHostInfo
* later for use with USU\_Export

### f\_GetDeviceInfo()

Function get from all Devices Base Device Informtions.

Return:

* ComputerID
* Manufacturer (this comes from f\_GetDeviceModel())
* Model (this comes from f\_GetDeviceModel())
* SerialNumber (this comes from f\_GetDeviceSerialNumber())
* UUID
* ChassisTypeID (this comes from f\_GetDeviceChassis(0))
* ChassisType (this comes from f\_GetDeviceChassis(0))
* BiosVersion
* Virtual (this comes from f\_GetDeviceModel())

This function need minimum 8 Logical Cores on SQL for performance on databases with more then 5000 Devices.

Used by:

* later for use with USU\_Export

### f\_GetDeviceOSLanguage()

Get Operating System Language per Device

Return:

* ComputerID
* LangaugeID Language Number like Microsoft
* Language Language as string

Used by:

* f\_GetDeviceOSInfo

### f\_GetDeviceOSLastLoginUser()

This will get by Last User Inventory or LastLoginUser in OS Inventory.   
Depend on following Information:

* Last known user from CIM\_UnitaryComputerSystem
* Last User Inventory Entry in Database

Return:

* ComputerID
* UserDomain Domainname from User if Domainuser
* UserName
* LastInventoryDate

Used by:

* f\_GetDeviceOSInfo

### f\_GetDeviceOSInfo()

Get OS Information per Device

Return:

* ComputerID
* SystemType Server / Client
* OSType Windows, Linux, HPUX
* Manufacturer
* ProductName
* Version
* Release On Windows like 2000, 2012 R2
* KernelVersion On Linux / Unix Systems
* Edition On Windows like Profesional, Enterprise, Datacenter
* ServicePack
* Architecture
* Language Comes from f\_GetDeviceOSLanguage
* LicenseKey
* InstallDate DateTime
* LastBoot DateTime
* LastLogin Comes from f\_GetDeviceOSLastLoginUser

ProductName use first Caption, if Caption not exists it use Name

Used by:

* csp\_aspera\_connector\_device
* csp\_aspera\_connector\_software\_generic\_os.sql
* f\_GetVirtualZoneHostInfo
* f\_GetDeviceInfo
* later for use with f\_GetVirtualLPARHostInfo
* later for use with USU\_Export
* later for use with sp\_rp\_infrastructure

### Deprecated Functions

#### f\_System\_MemoryDisk()

Give Memory and Disk Capacity per Device in GB

Please use 4.4.4 f\_GetDeviceDiskDrive() (erledigt) (Page 11)  
and 4.4.6 f\_GetDeviceMemoryModul() (erledigt) (Page 12)

#### f\_System\_Network()

Give all IP Addresses and MAC Addresses space sperated.

Please use 4.4.8 f\_GetDeviceNetworkInfo() (Page 13)

#### f\_System\_Operatingsystem()

Give OS Informations

Please use 4.4.17 f\_GetDeviceOSInfo() (Page 16)  
and for Computername 4.4.1 f\_GetDeviceFQDN(@bOnlyComputer) (erledigt) (Page 10)

#### f\_System\_Processor()

Give Processor Information

Please use 4.4.9 f\_GetDeviceProcessors() (erledigt) (Page 14)

#### f\_System\_Chassis()

Give Chassis Information

Please use 4.4.2 f\_GetDeviceChassis(bAll) (erledigt) (Page 10)

#### f\_System\_Base()

Function to get System Base Informationions, like Hardware Manufacturer, Model, usw.

We not more provide or fix this function, for better performance use store procedure and the catalog of functions in front of this documentation.

As sample look in store procedure csp\_aspera\_connector\_device.

## Virtualization

### VMware

#### f\_GetVirtualVMwareGuestUUID()

Get UUID from all Guest Machines from ESX Inventory

Return:

* ESXHostID ComputerID of ESX Host Inventory
* VMWAREVMComputerSystemID HardwareObjectID from Class VMWARE\_VMComputerSystem
* UUID UUID of the Guestsystem

Need SQL Server 2012 or higher

Used by:

* sp\_GetVirtualVMwareLinkGuest
* sp\_rp\_OracleVirtualInfrastructure

#### f\_GetDeviceVirtualVMwareUUID

Get UUID from all Inventories out of the Serialnumber, where the Serialnumber begins with VMWARE

Return:

* GuestID ComputerID of Guest Inventory
* CIMBIOSElementID HardwareObjectID from Class CIM\_BIOSElement
* UUID UUID of the Guestsystem on the ESX Host

Need SQL Server 2012 or higher

Used by:

* sp\_GetVirtualVMwareLinkGuest
* sp\_rp\_OracleVirtualInfrastructure

#### f\_GetVirtualVMwareGuestInfo()

Get all Guestinformation from ESX Inventory

Return:

* ESXHostID ComputerID of ESX Host Inventory
* VMWAREVMComputerSystemID HardwareobjectID of class VMWARE\_VMComputerSystem
* ElementName ElementName of the guest
* HostName (normally only if VMware Tools are installed)
* IPAddress (normally only if VMware Tools are installed)
* Memory max Memory usage what is assigned
* CPUs max CPU usage what is assigned
* GuestOS assigned OS, if VMware Tools installed and the machine is powerd on then it get the correct OS
* PowerState
* Annotation

Used by:

* csp\_aspera\_connector\_device\_relation

#### f\_GetVirtualVMwareLinkClusterName()

Get link between ESX Inventory and ClusterName

Return:

* ESXHostID ComputerID of ESX Host Inventory
* vCenterUUID UUID of the vCenter Object Element
* ParentID Parent ObjectID in vCenter
* ClusterName Name of the Cluster Object

Need SQL Server 2012 or higher

Used by:

* csp\_aspera\_connector\_device
* csp\_aspera\_connector\_device\_relation
* sp\_rp\_OracleVirtualInfrastructure
* Used later in sp\_rp\_infrastructure

#### f\_GetVirtualVMwareLinkVCenter()

For link between ESX Inventory and VCenter Inventory

Return:

* vCenterID ComputerID of vCenter Server Inventory
* vCenterCN Computername of vCenter Server
* vCenterFQDN FQDN of vCenter Server
* vCenterURLAddress SDK URL of vCenter
* vCenterUUID UUID of the vCenter

Need SQL Server 2012 or higher

Used by:

* Used later for sp\_rp\_infrastructure
* Used later for USU Export

### Hyper-V

#### f\_GetVirtualHyperVGuestInfo()

Get all Guest information from HyperV Host

Return:

* HyperVHostID ComputerID from HyperV Host
* SystemGuid UUID of the Guest in HyperV
* ElementName
* Description
* FQDN
* InstallDate (DATETIME)
* PowerState PowerState as string
* PowerStateID PowerState ID from Microsoft
* BIOSGuid UUID in the Guestsystem BIOS Information
* BIOSSerialNumber Seriennummer im Guestsystem BIOS
* Memory Assigned start Memory
* MemoryMin Minimum Memory
* MemoryMax Maximum of Dynamic Memory
* CPUs Assigned CPUs

Need Microsoft SQL 2012 or higher

Warning: BIOSGued and BIOSSerialNumber are only unique if you set this with powershell or create new Machines. By cloned machines, this will not changed by default.

Used by:

* sp\_GetVirtualHyperVLinkGuest
* sp\_rp\_OracleVirtualInfrastructure

#### f\_GetVirtualHyperVHostInfo()

Get all HyperV Host Information from HyperV Guest systems

Return:

* HyperVGuestID ComputerID form Guest Computer
* SystemGuid UUID of the Guest on the HyperV
* FQDN FQDN of the HyperV Host

Need Microsoft SQL 2012 or higher

Used by:

* sp\_GetVirtualHyperVLinkGuest
* sp\_rp\_OracleVirtualInfrastructure

#### f\_GetVirtualHyperVLinkClusterName()

Get Clusterinformation to the HyperV Host

Return:

* HyperVHostID ComputerID from the HyperV Host
* ClusterName Clustername on witch the HyperV Host run
* IsFailover Is a Failover Cluster

Used by:

* csp\_aspera\_connector\_device
* csp\_aspera\_connector\_device\_relation
* sp\_rp\_OracleVirtualInfrastructure

### Sun/Oracle Solaris Zone

#### f\_GetVirtualZoneLinkGuest()

Get link information between Zone Host and Zone Guest

Return:

* ZoneHostID ComputerID from Zone Host
* ZoneHostUnitaryComputerSystemID HardwareObjectID from Zone Host Class CIM\_UnitaryComputerSystem
* ProductHostID ProductHostID from Zone for the Host  
   (Matcher between Guests and Host)
* GuestID ComputerID from Guest System
* GuestUnitaryComputerSystemID HardwareObjectID from Zone Guest Class

CIM\_UnitaryComputerSystem

Used by:

* csp\_aspera\_connector\_device
* csp\_aspera\_connector\_device\_relation

#### f\_GetVirtualZoneHostInfo()

Get System Informations of Solaris Zone Host

Return:

* ZoneHostID ComputerID from Zone Host   
   (can be null, if no Hostsystem Inventory exists)
* ProductHostID ProductHostID from Zone for the Host  
   (Matcher between Guests and Host)
* Manufacturer
* Model
* CPUName
* TotalCPUs Total Number of CPUs
* TotalCores Total Number of Cores (Over all CPU’s)
* TotalLogical Total Number of Logical Cores (Over all CPU’s)
* ClockSpeed CPU ClockSpeed (in kHz)
* Memory (in Bytes)
* OperatingSystem
* LastInventoryDate

Note:

If no Inventory from Hostsystem exists, it create a sum of all virtual guest systems for the TotalCPUs, TotalCores , TotalLogical, Memory

Used by:

* csp\_aspera\_connector\_device
* csp\_aspera\_connector\_device\_relation

## Software

### f\_SoftwareValuesByPropertyName(@sProperty)

Get all SoftwareValues from a property name.

Return:

* SoftwareOccurenceID
* Property
* Value

Used by:

* sp\_GetSoftwareInstallInformation



### f\_GetSoftwareFileUsage ()

Function to get all File Usage Information with following Values

Return:

* ComputerID
* UserID
* SoftwareFileNameID
* Publisher
* ProductName
* ProductVersion
* FileDescription
* FileLongName
* FirstUsage
* LastUsage
* SessionPerMonth

Used by:

* sp\_GetSoftwareFiles

### f\_GetSoftwareTagFiles()

Function to get all ISO Tag Information with following Values

Return:

* SoftwareIsoTagFileID
* Publisher
* ProductName
* ProductVersion
* ProductVersionMajor
* ProductVersionMinor
* ProductVersionBuild
* ProductVersionReview
* ProductGuid
* SoftwareUniqueID
* SwidLicensorID
* MD5

Used by:

* f\_GetSoftwareFiles
* csp\_aspera\_connector\_software\_tag

### f\_GetSoftwareFiles(iType)

Function to get all Fileinformationtions for Software

Parameter iType can be:

* 0 or NULL = all Files
* 1 = only Files with Product Information
* 2 = only Files with Product Information, without ISO Tag Files
* 3 = only Tag Files
* 4 = only Tag Files and only with ProductName

Return:

* ComputerID
* SoftwareFileID
* SoftwareIsoTagFileID
* FilePath
* FileName
* FileSize
* FileMD5
* FileDate (DATETIME)
* Publisher Publisher of the Software
* ProductName
* ProductVersion
* ProductGuid
* SwidUniqueId
* SwidLicensorId

Used by:

* sp\_GetSoftwareFiles
* csp\_aspera\_connector\_software\_tag

### f\_GetSoftwareMicrosoftSQL()

Get Microsoft SQL Installation Information

Return:

* ComputerID ComputerID of the Inventory
* MGSSqlServiceID HardwareObjectID from class MGS\_SqlService
* Publisher Always Microsoft
* ProductName
* ProductVersion
* ProductEdition Edition like Express, Standard
* ProductRelease Release like 2005, 2012
* ServicePack
* InstanceName
* Clustered
* IsIntegratedSecurity
* DataPath
* ServiceName
* State
* StartMode
* InstanceID
* Sku
* SqlServiceType
* InstallLocation

Used by:

* csp\_aspera\_connector\_software\_gerneric\_microsoft

### f\_GetSoftwareMicrosoftExchange()

Get Microsoft Exchange Installation Information

Return:

* ComputerID ComputerID of the Inventory
* MGSExchangeServerID HardwareObjectID from class MGS\_ExchangeServer
* ComputerCN Hostname of the Exchange Server
* ComputerFQDN FQDN of the Exchange Server
* Publisher Always Microsoft
* ProductName Always Microsoft Exchange
* ProductVersion
* ProductEdition Standard, Enterprise
* ProductRelease 2016, 2003

It can be that more then one time the ComputerID exist with other Hostnames and FQDN, this is if more then one Exchange Server exists, but only from one a Computer Inventory exists.

Used by:

* csp\_aspera\_connector\_software\_gerneric\_microsoft

### f\_GetSoftwareMicrosoftSharePoint()

Get Microsoft SharePoint Installation Information

Return:

* ComputerID ComputerID of the Inventory
* MGSSPFarmID HardwareObjectID from class MGS\_SPFarm
* Publisher Always Microsoft
* ProductName
* ProductEdition
* ProductRelease
* ProductVersion
* FullProductName
* SKUID

Used by:

* csp\_aspera\_connector\_software\_gerneric\_microsoft

## Oracle

### f\_OracleRAC

Get Oracle RAC Server Information

Return:

* DBObjectID Database Inventory ComputerID
* DBName Database Name
* NodeInstanceID ComputerID des DBObjects der Instance -> muss nicht zwingend die selbe sein (2 Node eines Clusters)
* NodeComputerID ComputerID des Nodes (Cluster)
* NodeComputerCN ComputerCN des Nodes (Cluster)
* NodeHostname FQDN des Nodes (Cluster)
* NodeInstanceName InstanceName des Nodes
* NodeInstanceRole Role des Nodes Primary, Standby

Used by:

* f\_OracleNodes

### f\_OracleNodes

Get all Oracle Node Information

Return:

* DBObjectID Database Inventory ComputerID
* ClusterDBName Database Name from Cluster
* DBName Database Name
* NodeInstanceID ComputerID des DBObjects der Instance -> muss nicht zwingend die selbe sein (2 Node eines Clusters)
* NodeComputerID ComputerID des Nodes (Cluster)
* NodeComputerCN ComputerCN des Nodes (Cluster)
* NodeHostname FQDN des Nodes (Cluster)
* NodeInstanceName InstanceName des Nodes
* NodeInstanceRole Role des Nodes Primary, Standby

Used by:

* f\_OracleInstances

### f\_OracleInstances(@InstanceID)

Get all Oracle Nodes / Instances with Oracle DB Information

Parameter @InstanceID:

* with this you can filter to 1 instance by DBObjectID / InstanceID

Return:

* ClusterDBName Database Name from Cluster
* ComputerID ComputerID des Nodes
* ComputerCN ComputerCN des Nodes
* ComputerFQDN FQDN des Nodes
* InstanceID Database Inventory ComputerID
* Instance InstanceName des Nodes
* SystemDateTime Date and Time from the system at the inventory
* SystemDisplayName Full Product Name
* SystemDisplayVersion Full Product Version
* AppName Product Name
* AppVersion Product Version
* AppEdition Product Edition
* DatabaseRole Role of the Database
* DatabaseName Database Name of the Instance
* DatabaseNameUnique Unique DB Name (Cluster can be differend)
* DatabaseCreationDate When the Database was created
* OperatingSystem OS System

Used by:

* csp\_aspera\_connector\_software\_generic\_oracle
* f\_OracleFailures
* v\_OracleReport
* sp\_OracleFailures
* sp\_OracleRaynetOptionList
* sp\_OracleScriptOptionList
* sp\_rp\_OracleOptionList
* sp\_rp\_OracleReport

### f\_OracleDFUS (@DBObjectID)

Get all DFUS Inventory Table

Parameter @DBObjectID:

* with this you can filter to 1 instance by DBObjectID / InstanceID

Return:

* DBObjectID Database Inventory ComputerID
* FeatureName Featurename of Oracle Feature
* Version Oracle DB Version of the Feature
* CurrentlyUsed 1 If Used, 0 If not Used
* DetectedUsages How often it was used
* FirstUsageDate First used of the Feature
* LastUsageDate Last used of the Feature
* LastSampleDate Last Sample Date
* SampleInterval Interval for Sample Check runs
* TotalSamples How often the Sample Check runs
* FeatureInfo Extra Feature Information

Used by:

* Old sp\_OracleRaynetOptionList

Deprecated

* New sp\_OracleDFUS @DBObjectID

### f\_OracleOptions (@InstanceID, @ClassName)

Get all Oracle Option from old scan method

Parameter @InstanceID:

* with this you can filter to 1 instance by DBObjectID / InstanceID

Parameter @ClassName:

* Classname Filter with LIKE, need the Classname part on witch it should look

Return:

* ComputerID ComputerID of the Oracle Server
* ComputerCN ComputerCN of the Oracle Server
* InstanceID Database Inventory ComputerID
* Instance Database Instance Name
* OptionClassName Classname of the Option
* OptionName Name of the Option or Feature
* OptionInstalled 1 = Option is installed, 0 = is not installed
* OptionUsed 1 = Option is used, 0 = is not used

Used by:

* sp\_rp\_OracleOptionList

Deprecated

* New sp\_OracleOptions @DBObjectID, @ClassName

The sp\_rp\_OracleOptionList, should be change in a new Version to the store procedure instead of this function.

### f\_OracleUserCounts (@IncludeSystemUser)

Get user counts from user on Oracle DB

Parameter @IncludeSystemUser:

* If 1 all Users will count, if 0 only self-created user will counted

Return:

* DBObjectID Database Inventory ComputerID
* All Users All users
* Expired All expired users
* Locked All locked users
* Open All open users

Used by:

* sp\_rp\_OracleReport

### f\_OracleAccessCheck (@All)

Get check if we have enough rights on Oracle DB

Parameter @All:

* If 1 over all Elements, if 0 only over elements with ComputerID of the Oracle Server

Return:

* DBObjectID Database Inventory ComputerID
* TableName Table, View or Store what we check
* RightStatus 1 = Yes the right is ok, 0 = the right is defect

Used by:

* sp\_OracleFailures
* f\_OracleFailures (Deprecated)

### f\_OracleFailures ()

Check for Oracle Failures

Return:

* Instance Database Instance Name
* StatusID
* Status Status Message
* Version oratrack.xml.enc

Used by:

* v\_OracleReport (Deprecated, use by old RDL’s)

Deprecated:

* new sp\_OracleFailures

### v\_OracleReport

Get the Oracle Information for the old Oracle RDL Files

Return:

* ClusterDBName
* ComputerID
* ComputerCN
* InstanceID
* Instance
* SystemDateTime
* SystemDisplayName
* SystemDisplayVersion
* AppName
* AppVersion
* AppEdition
* DatabaseRole
* DatabaseName
* DatabaseNameUnique
* DatabaseCreationDate
* OperatingSystem
* UserCount

Used by:

* Report OracleServerWorksheetFramework

Deprecated:

new sp\_rp\_OracleReport

# List of Store Proceduren

## Base

### sp\_GetVirtualVMwareLinkGuest

For link between ESX, Guest, vCenter and Cluster

Return:

* ESXHostID ComputerID of the ESX Inventory
* GuestID ComputerID of the Guest Inventory
* GuestUUID Guest UUID in ESX
* VMWAREVMComputerSystemID HardwareObjectID from Class VMWARE\_VMComputerSystemID
* vCenterURLAddress SDK URL of vCenter
* vCenterUUID UUID of vCenter

Or can be returned directly to a Table sp\_GetVirtualVMwareLinkGuest for this create first a Table

CREATE TABLE #sp\_GetVirtualVMwareLinkGuest (

ESXHostID BIGINT NOT NULL,

GuestID BIGINT NULL,

GuestUUID UNIQUEIDENTIFIER NULL,

VMWAREVMComputerSystemID BIGINT NOT NULL,

vCenterURLAddress NVARCHAR(256) COLLATE database\_default NULL,

vCenterUUID UNIQUEIDENTIFIER NULL)

And start the procedure with EXEC sp\_GetVirtualVMwareLinkGuest

Then you can ask this with SELECT \* FROM #sp\_GetVirtualVMwareLinkGuest

Table can have more Columes as the defined in the front. This are only minimum.

Need SQL Server 2012 or higher

Used by:

* csp\_aspera\_connector\_device\_relation
* Used later in sp\_rp\_infrastructure

Sample, with add ClusterName and vCenterID:

CREATE TABLE #sp\_GetVirtualVMwareLinkGuest (

ESXHostID BIGINT NOT NULL,

GuestID BIGINT NULL,

GuestUUID UNIQUEIDENTIFIER NULL,

VMWAREVMComputerSystemID BIGINT NOT NULL,

vCenterURLAddress NVARCHAR(256) COLLATE database\_default NULL,

vCenterUUID UNIQUEIDENTIFIER NULL,

vCenterID BIGINT NULL,

ClusterName NVARCHAR(256) COLLATE database\_default NULL)

EXEC sp\_GetVirtualVMwareLinkGuest

UPDATE t

SET vCenterID = vc.vCenterID

FROM #sp\_GetVirtualVMwareLinkGuest t

LEFT JOIN f\_GetVirtualVMwareLinkVCenter() vc

ON t.vCenterURLAddress = vc.vCenterURLAddress

UPDATE t

SET ClusterName = cl.ClusterName

FROM #sp\_GetVirtualVMwareLinkGuest t

LEFT JOIN f\_GetVirtualVMwareLinkClusterName() cl

ON t.ESXHostID = cl.ESXHostID

SELECT \* FROM #sp\_GetVirtualVMwareLinkGuest

### sp\_GetVirtualHyperVLinkGuest @bWithSerial

Get Relation between HyperV Host and Guests

Parameter @bWithSerial: (Set the art to create the relation)

* 0 or null GuestFQDN = GuestFQDN (From Host Inventory)

HostFQDN = HostFQDN (From Guest Inventory)

BIOSSerialnumber only if one SerialNumber in Guest Inventories

* 1 Match all systems with Serialnumber  
   (Should only be used if you are 100% secure that your systems have a unique  
   serialnumber)

0 is default value, if the parameter not exists.

Return:

* HyperVHostID ComputerID of the HyperV Host
* HyperVHostCN Computername of the HyperV Host
* HyperVHostFQDN FQDN of the HyperV Host
* GuestID ComputerID of the Guest System
* GuestCN Computername of the Guest System
* GuestFQDN FQDN of the Guest System
* PowerState (Running, Off or NULL)

Or can be returned directly to a Table sp\_GetVirtualHyperVLinkGuest for this create first a Table

CREATE TABLE #sp\_GetVirtualHyperVLinkGuest (

HyperVHostID BIGINT NOT NULL,

HyperVHostCN NVARCHAR(256) COLLATE database\_default NULL,

HyperVHostFQDN NVARCHAR(256) COLLATE database\_default NULL,

GuestID BIGINT NULL,

GuestCN NVARCHAR(256) COLLATE database\_default NULL,

GuestFQDN NVARCHAR(256) COLLATE database\_default NULL,

PowerState NVARCHAR(30) COLLATE database\_default NULL)

And start the procedure with EXEC sp\_GetVirtualHyperVLinkGuest

Then you can ask this with SELECT \* FROM #sp\_GetVirtualHyperVLinkGuest

Table can have more Columes as the defined in the front. This are only minimum.

Used by:

* csp\_aspera\_connector\_device\_relation
* Used later in sp\_rp\_infrastructure

Sample, with add ClusterName:

CREATE TABLE #sp\_GetVirtualHyperVLinkGuest (

HyperVHostID BIGINT NOT NULL,

HyperVHostCN NVARCHAR(256) COLLATE database\_default NULL,

HyperVHostFQDN NVARCHAR(256) COLLATE database\_default NULL,

GuestID BIGINT NULL,

GuestCN NVARCHAR(256) COLLATE database\_default NULL,

GuestFQDN NVARCHAR(256) COLLATE database\_default NULL,

PowerState NVARCHAR(30) COLLATE database\_default NULL,

ClusterName NVARCHAR(256) COLLATE database\_default NULL)

EXEC sp\_GetVirtualHyperVLinkGuest

UPDATE t

SET ClusterName = hc.ClusterName

FROM #sp\_GetVirtualHyperVLinkGuest t

LEFT JOIN f\_GetVirtualHyperVLinkClusterName() hc

ON t.HyperVHostID = hc.HyperVHostID

SELECT \* FROM #sp\_GetVirtualHyperVLinkGuest

### sp\_GetSoftwareInstallInformation @iEvidenceGroup, @sEvidenceFilter, @bNoUpdate, @iLastSeenDays

Store Procedure to get Software List about Install Information like ARP, MSI, IA, ManageSoft

Parameter @iEvidenceGroup (Default null):

* null All Evidences or sEvidenceFilter
* 1 Only Windows ARP Information
* 2 Only Windows MSI
* 3 Other the Windows ARP or MSI

Parameter @sEvidenceFilter (Default null):

Comma Separated List of Evidences. If the Value is null then all  
If @iEvidenceGroup is set then this is value will be ignored

Evidences what exists:

|  |  |
| --- | --- |
| **Evidence** | **From where** |
| DPKG | Ubuntu Package Manager |
| ExeHdr | Windows EXE Handler |
| IA | InstallAnywhere |
| ISMP | InstallShield MultiPlatform |
| LPP | IBM Package Manager |
| ManageSoft | Managesoft Agent Installed |
| Managesoft + MSI | Managesoft Agent Installed MSI |
| MSI | MSI |
| OUI | Oracle Installer |
| RPM | RedHat Package Manager |
| SUNPKG | Sun Package Manager |
| Uninstall | Windows Uninstall Keys |
| VISDK | VMWare |

Parameter @bNoUpdate (Default 0):

* 0 or null All Packages are shown
* 1 Update Packages are filtered out

Parameter @iLastSeenDays (Default Null):

* null With out Filter
* 1 – xxx The days what the inventory maximum can be old

Return:

* ComputerID
* SoftwareOcurenceID
* Evidence
* Publisher
* ProductName
* ProductVersion
* ProductCode
* InstallationDate

Or can be returned directly to a Table sp\_GetSoftwareInstallInformation for this create first a Table

CREATE TABLE #sp\_GetSoftwareInstallInformation (

ComputerID BIGINT NOT NULL,

SoftwareOccurrenceID BIGINT NOT NULL,

SoftwareID BIGINT NULL,

SoftwareDetailsID BIGINT NULL,

Evidence VARCHAR(32) COLLATE database\_default NULL,

Publisher NVARCHAR(256) COLLATE database\_default NULL,

ProductName NVARCHAR(256) COLLATE database\_default NULL,

ProductVersion NVARCHAR(256) COLLATE database\_default NULL,

ProductCode NVARCHAR(256) COLLATE database\_default NULL,

Summary NVARCHAR(256) COLLATE database\_default NULL,

InstallationDate DATETIME NULL)

And start the procedure with EXEC sp\_GetSoftwareInstallInformation

Then you can ask this with SELECT \* FROM # sp\_GetSoftwareInstallInformation

Table can have more Columes as the defined in the front. This are only minimum.

Used by:

* csp\_aspera\_connector\_device\_arp
* csp\_aspera\_connector\_device\_msi
* csp\_aspera\_connector\_device\_generic\_othersw

### sp\_GetSoftwareFiles @iType, @noUsageData

Function to get all Fileinformationtions for Software

Parameter @iType (Default 0):

* 0 or NULL = all Files
* 1 = only Files with Product Information
* 2 = only Files with Product Information, without ISO Tag Files
* 3 = only Tag Files
* 4 = only Tag Files and only with ProductName

Parameter @noUsageData (Default 0)

* 0 or NULL = all Files
* 1 = only File Inventory, without Usage Data

Return:

* ComputerID
* SoftwareFileID
* SoftwareIsoTagFileID
* FilePath
* FileName
* FileSize
* FileMD5
* FileDate (DATETIME)
* Publisher Publisher of the Software
* ProductName
* ProductVersion
* ProductGuid
* SwidUniqueId
* SwidLicensorId
* LastUsage (DATETIME)
* SessionPerMonth

Or can be returned directly to a Table sp\_GetSoftwareFiles for this create first a Table

CREATE TABLE #sp\_GetSoftwareFiles (

ComputerID BIGINT NOT NULL,

SoftwareFileID BIGINT NULL,

SoftwareFileNameID BIGINT NULL,

SoftwareIsoTagFileID BIGINT NULL,

FilePath NVARCHAR(400) COLLATE database\_default NULL,

FileName NVARCHAR(256) COLLATE database\_default NULL,

FileDescription NVARCHAR(256) COLLATE database\_default NULL,

FileSize NVARCHAR(256) COLLATE database\_default NULL,

FileMD5 NVARCHAR(256) COLLATE database\_default NULL,

FileDate DATETIME NULL,

Publisher NVARCHAR(256) COLLATE database\_default NULL,

ProductName NVARCHAR(256) COLLATE database\_default NULL,

ProductVersion NVARCHAR(256) COLLATE database\_default NULL,

ProductGuid NVARCHAR(256) COLLATE database\_default NULL,

SwidUniqueId NVARCHAR(256) COLLATE database\_default NULL,

SwidLicensorID NVARCHAR(256) COLLATE database\_default NULL,

LastUsage DATETIME NULL,

SessionPerMonth BIGINT NULL)

And start the procedure with EXEC sp\_GetSoftwareFiles

Then you can ask this with SELECT \* FROM #sp\_GetSoftwareFiles

Table can have more Columes as the defined in the front. This are only minimum.

Used by:

* csp\_aspera\_connector\_device\_file

## Oracle Reports

### sp\_OracleDFUS @DBObjectID

Get the DFUS Table for Raynet Metric out of Database (Class RVSORACLE\_DFUS)

Parameter @DBObjectID:

Database Object ID from the DB Instance

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| DBObjectID | BIGINT | DB Instance ComputerID |
| RowID | NVARCHAR(256) | Line number |
| FeatureName | NVARCHAR(256) | Feature / Option Name from witch the product is part |
| Version | NVARCHAR(256) |  |
| CurrentlyUsed | BIT | If currently used |
| DetectedUsages | INT | How often used |
| FirstUsageDate | DATETIME | First usage |
| LastUsageDate | DATETIME | Last usage |
| LastSampleDate | DATETIME | Last sample Date |
| SampleInterval | BIGINT | Interval of samples |
| TotalSamples | INT | Count of samples since the DB exists |
| FeatureInfo | NVARCHAR(256) | Extra Information |

If Temp Table #sp\_OracleDFUS exist, the Data will not shown. It will put into this table.

Used by:

* sp\_OracleRaynetOptionList

### sp\_OracleOption @DBObjectID

Get all Oracle Options from Raynet type of inventarisation (Class RVSORACLE\_OPTIONS\_%).  
This is for the Options, what not comes out from DFUS Data.

Parameter @DBObjectID:

Database Object ID from the DB Instance

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| NodeComputerID | BIGINT | ComputerID of the Server Device |
| NodeComputerCN | NVARCHAR(256) | Computername from the Server |
| DBObjectID | BIGINT | DB Instance ComputerID |
| NodeInstanceName | NVARCHAR(256) | DB Instance Name |
| OptionClassName | NVARCHAR(256) | Class Name of the Option |
| OptionName | NVARCHAR(256) | Name of the Feature or Optin |
| OptionInstall | BIT | If the Option or Feature is Installed |
| OptionUsed | BIT | If the Option or Feature is Used |

If Temp Table #sp\_OracleOption exist, the Data will not shown. It will put into this table.

Used by:

* sp\_OracleRaynetOptionList

### sp\_OracleFailures @ComputerCN

Get all Oracle Inventory Failures, like last Script used, rights correct etc.

Parameter @ComputerCN:

ComputerName of the Database Server to Filter. If null, all Devices. Can be a value for LIKE

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| ComputerID | INT | ComputerID of the Server Device |
| Instance | NVARCHAR(256) | DB Instance Name |
| InstanceID | INT | DB Instance ComputerID |
| StatusID | INT | Id for Status 0 Missing Inventory  1 Failed connection  2 not enough rights  3 old oratrack.xml.enc  4 No Hardware Inventory |
| [Status] | NVARCHAR(256) | Depends on StatusID in combination of tmp\_MessageID  Look extra Table |
| [Version] | NVARCHAR(256) | Version of query.xml.enc |
| tmp\_MessageID | INT | Used to create Status Message ([Status]) |
| tmp\_VerMajor | INT | Major Version of query.xml.enc |
| tmp\_VerMinor | INT | Minor Version of query.xml.enc |

If Temp Table #sp\_OracleFailures exist, the Data will not shown. It will put into this table.

Used by:

* sp\_rp\_OracleReport

### sp\_OracleRaynetOptionList @DBObjectID

Get all Oracle Options from Raynet type of inventarisation

Parameter @DBObjectID:

Database Object ID from the DB Instance

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| DBObjectID | BIGINT | DB Instance ComputerID |
| Product | NVARCHAR(256) | Product Name |
| ProductType | NVARCHAR(256) | Option or Feature |
| FeatureName | NVARCHAR(256) | Feature / Option Name from witch the product is part |
| Version | NVARCHAR(256) |  |
| FirstUsageDate | DATETIME | First usage |
| LastUsageDate | DATETIME | Last usage |
| Used | BIT | 1 = Used |
| Active | BIT |  |
| NeedLicense | BIT | License Needed |
| DependFeatureName | NVARCHAR(256) | If used the FeatureName what is set here needs as well a license |
| Comment | NVARCHAR(256) |  |

If Temp Table #sp\_OracleRaynetOptionList exist, the Data will not shown. It will put into this table.

Used by:

* sp\_rp\_OracleOptionList

### sp\_OracleScriptOptionList @DBObjectID

Get all Oracle Options from Oracle DBFUS Script

Parameter @DBObjectID:

Database Object ID from the DB Instance

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| DBObjectID | BIGINT | DB Instance ComputerID |
| ContainerName | NVARCHAR(256) | Container Name if Container Database |
| Product | NVARCHAR(256) | Product Name |
| ProductType | NVARCHAR(256) | Option or Feature |
| FeatureName | NVARCHAR(256) | Feature / Option Name from witch the product is part |
| Version | NVARCHAR(256) |  |
| Usage | NVARCHAR(32) | Info like PAST\_USAGE, CURRENT\_USAGE |
| CurrentlyUsed | BIT | 1 if Usage = CURRENT\_USAGE |
| DetectedUsages | INTEGER | Count of usage |
| FirstUsageDate | DATETIME | First used of the product |
| LastUsageDate | DATETIME | Last used of the product |
| LastSampleDate | DATETIME | Last check |
| TotalSamples | INTEGER | Count of cheks |
| FeatureInfo | NVACHAR(max) | Detail information like compression count |
| Used | BIT | 1 if used |
| NeedLicense | BIT | 1 license is needed |
| DependFeatureName | NVARCHAR(256) | If used the FeatureName what is set here needs as well a license |
| Comment | NVARCHAR(256) |  |

If Temp Table #sp\_OracleScriptOptionList exist, the Data will not shown. It will put into this table.

Used by:

* sp\_rp\_OracleOptionList

### sp\_rp\_OracleOptionList @DBObjectID, @ComputerID, @OnlyProduct

Get all Oracle Option License Information

Parameter @DBObjectID:

* ComputerID of the DB Instance Inventory

Parameter @ComputerID

* ComputerID of the Oracle DB Server Hardware Inventory

Parameter @OnlyProduct

* 1 = SUM all Products over all Instances of @ComputerID (If @DBObjectID is null)
* 0 = all Products split in the Instances

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| DBObjectID | BIGINT | DB Instance ComputerID |
| Product | NVARCHAR(256) | Product Name |
| FeatureName | NVARCHAR(256) | Feature / Option Name from witch the product is part |
| Installed | BIT | Always 1 |
| RaynedUsed | BIT | If Raynet Inventory says used |
| OracleUsed | BIT | If Oracle Inventory says used |
| TechnicalLicense | INT | 2 = both says used, or if only raynet inventory exists and say used  1 = only one says used  0 = both says no used |
| RaynetComment | NVARCHAR(256) | Comment from Raynet Inventory |
| OracleComment | NVARCHAR(256) | Comment from Oracle DBFUS Data |
| CustomLicense | BIT | For future |
| CustomComment | NARCHAR(256) | For future |
| LicenseResult | BIT | Say used depend on settings for the Report in ReportingConfiguration |

If Temp Table #sp\_rp\_OracleOptionList exist, the Data will not shown. It will put into this table.

Used by:

* sp\_rp\_OracleReport
* csp\_aspera\_connector\_software\_generic\_oracle
* Report Sub\_OracleOption
* Report Sub\_OracleLicenseSummary
* Report Sub\_OracleInstance

### sp\_rp\_OracleVirtualInfrastructure @DBObjectID

Get all Virtual Infrastructur Information for Oracle and the total amount of cores for the ESX Hosts.

Parameter @ComputerID:

ComputerID des Oracle DB Servers

Parameter @HyperVWithSerial

Matching criterium for HyperV Default is 0

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| Virtualization | NVARCHAR(256) | Virtualization Type, VMware, HyperV |
| HostID | BIGINT | ComputerID of ESX Host, on witch the Oracle Server is |
| HostCN | NVARCHAR(256) | Computername of ESX Host on witch the Oracle Server is |
| ClusterName | NVARCHAR(256) | Clustername if exist |
| GuestID | BIGINT | ComputerID of Oracle Server |
| GuestCN | NVARCHAR(256) | Computername of Oracle Server |
| ProcHostID | BIGINT | ComputerID of ESX Host from witch the Processor Information is, give for every ESX Host of the Cluster |
| ProcHostCN | NVARCHAR(256) | Computername of the ESX Host to the ProcHostID |
| ProcName | NVARCHAR(256) | Processor Name |
| TotalCores | INT | Total amount of cores from ESX ESX Host to the ProcHostID |

Sum of TotalCores of all rows on GuestID is the amount to license.

This Store Procedure has not the ability to put the information directly in a Temp Table

Used by:

* Report OracleGuestOverview

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### sp\_rp\_OracleReport @ComputerCN, @InstanceName

Get Oracle DB Information for Oracle Report.

Parameter @ComputerCN:

Server Hostname

Parameter @InstanceName

DB Instance Name

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| ClusterDBName | NVARCHAR(256) | Cluster Database Name |
| ComputerID | BIGINT | ComputerID of the Oracle Server |
| ComputerCN | NVARCHAR(256) | Computername of the Oracle Server |
| InstanceID | BIGINT | ComputerID of the Oracle DB Inventory |
| Instance | NVARCHAR(256) | Oracle Database Instance Name |
| SystemDateTime | DATETIME | System time of the Oracle Server at Inventory time |
| SystemDisplayName | NVARCHAR(256) | Full Oracle Display Name |
| SystemDisplayVersion | NVARCHAR(256) | Full Oracle Display Version |
| AppName | NVARCHAR(256) | Oracle DB Product Name |
| AppVersion | NVARCHAR(256) | Oracle DB Product Version |
| AppEdition | NVARCHAR(256) | Oracle DB Product Edition |
| DatabaseRole | NVARCHAR(256) | Role of the Database |
| DatabaseName | NVARCHAR(256) | Name of the Oracle Database |
| DatabaseNameUnique | NVARCHAR(256) |  |
| DatabaseCreationDate | DATETIME | When the Database was created |
| OperatingSystem | NVARCHAR(256) | Operating System out of the Oracle DB Inventory |
| UserCount | INT | Count of User on the Database |
| MakeModel | NVARCHAR(256) | Hardware Model |
| MaxClockSpeed | NVARCHAR(256) | Max Speed of the CPU |
| NumberOfProcessors | INT | Total number of CPUs of the Oracle Server |
| NumberOfLogicalProcessors | INT | Total number of Logical Cores of the Oracle Server |
| Status | NVARCHAR(256) | Status of the Inventory (Failures) |
| XMLVersion | NVARCHAR(256) | Version of the oratrack.xml.enc |
| OracleOption | VARCHAR(MAX) | List of used Options in a special format for the SubRepot sub\_OracleOption |

This Store Procedure has not the ability to put the information directly in a Temp Table

Used by:

* Report OracleServerOverview
* Used later in Report OracleServerWorksheetFramework

## Aspera Connector

### csp\_aspera\_connector\_device

Store Procedure Used by Aspera Connector 400\_raynet\_devices.xml.

Get all Device Information

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @sOrgLevel2ID | unknown\_costcenter | Default costcenter |
| @bImportIDFQDN | 1 | 0 = Use Hostname  1 = Use FQDN |
| @bChassisTypeID | 0 | 0 = Chassis Description  1 = chassis\_{ID from WMI} |
| @bVMwareRelation | 1 | 0 = no VMware Relation  1 = VMware Relation |
| @bHyperVRelation | 1 | 0 = no HyperV Relation  1 = HyperV Relation |
| @bHWPartRelation | 1 | 0 = no Hardware Partition like LPAR, Zone  1 = Hardware Partition |
| @iLastSeenDays | 90 | Last seen days, by default Inventories of the last 90 days |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(125) | Hostname or FQDN depend on @bImportIDFQDN |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| device\_key | NVARCHAR(255) |  |
| import\_org\_level\_2\_id | NVARCHAR(50) | Value of @sOrgLevel2ID |
| device\_name | NVARCHAR(255) |  |
| device\_manufacturer | NVARCHAR(255) |  |
| device\_model | NVARCHAR(255) |  |
| serial\_number | NVARCHAR(255) |  |
| import\_operating\_system\_id | NVARCHAR(255) |  |
| import\_device\_type\_id | NVARCHAR(50) |  |
| import\_device\_status\_id | NVARCHAR(50) |  |
| import\_domain\_id | NVARCHAR(50) |  |
| Import\_user\_id | NVARCHAR(50) |  |
| ip\_address | NVARCHAR(255) |  |
| mac\_address | NVARCHAR(255) |  |
| fqdn | NVARCHAR(255) |  |
| inventory\_date | NVARCHAR(10) | Format dd.mm.YYYY |
| cpu\_chip\_count | DECIMAL(15,2) |  |
| cpu\_core\_count | DECIMAL(15,2) |  |
| cpu\_speed | INT |  |
| import\_cpu\_type\_id | NVARCHAR(255) |  |
| ram | DECIMAL(15,2) | In MB |
| storage | DECIMAL(15,2) | In GB |
| graphics | NVARCHAR(255) |  |
| network | NVARCHAR(255) |  |
| bios | NVARCHAR(255) |  |
| tmp\_computerid | INT |  |

If Temp Table #csp\_aspera\_connector\_device exist, the Data will not shown. It will put into this table.

Used by:

* Store Procedure Used by Aspera Connector 400\_raynet\_devices.xml.

### csp\_aspera\_connector\_device\_types

Get list of all device types, used by Apsera Connector 400\_raynet\_devices.xml

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @bChassisTypeID | 0 | 0 = Chassis Description  1 = chassis\_{ID from WMI} |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(50) | Chassis import ID look in Chassis List |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| name | NVARCHAR(255) | Name of Chassi, look in Chassis List |
| remarks | NVARCHAR(4000) |  |
| virtualization\_type | NVARCHAR(4000) | Type of Virtualisation |

Chassis List:

|  |  |  |  |
| --- | --- | --- | --- |
| import\_id | name | virtualization\_type | Description |
| dummy\_host | Dummy Host | NULL | HW Partition Host Systems (Zone, LPAR, usw.) |
| zone | Zone | partition | Zone Guest System |
| lpar | LPAR | partition | IBM LPAR Guest System |
| vmware\_esx\_cluster | VMware ESX Cluster | cluster | ESX Clusterelement |
| vmware\_esx\_host | VMware ESX Host | NULL | ESX Host System |
| hyperv\_cluster | HyperV Cluster | cluster | HyperV Clusterelement |
| hyperv\_host | HyperV Host | NULL | HyperV Host System |
| virtual\_machine | Virtual Machine | partition | VMware oder HyperV Guest System |

If @bChassisTypeID = 0

|  |  |  |  |
| --- | --- | --- | --- |
| import\_id | name | virtualization\_type | Description |
| [chassistype] | [chassistype] | NULL | This is the Value from all used chassistypes of RayVentory Devices |

If @bChassiTypeID = 1

|  |  |  |  |
| --- | --- | --- | --- |
| import\_id | name | virtualization\_type | Description |
| chassis\_[chassisid] | [chassistype] | NULL | This is the Value from all used chassisid and types of RayVentory Devices |

Used by:

* Store Procedure Used by Aspera Connector 400\_raynet\_devices.xml.

### csp\_aspera\_connector\_device\_relation

Get all Virtual Device Relation information for Aspera

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @bImportIDFQDN | 1 | 0 = Use Hostname  1 = Use FQDN |
| @bVMwareRelation | 1 | 0 = no VMware Relation  1 = VMware Relation |
| @bHyperVRelation | 1 | 0 = no HyperV Relation  1 = HyperV Relation |
| @bHWPartRelation | 1 | 0 = no Hardware Partition like LPAR, Zone  1 = Hardware Partition |
| @iLastSeenDays | 90 | Last seen days, by default Inventories of the last 90 days |
| @bHyperVwithSerial | 0 | 0 = Special Logic  1 = always user serial number |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(125) | Hostname or FQDN depend on @bImportIDFQDN to Hostname or FQDN depend on @bImportIDFQDN or CluterName and more |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| import\_device\_rel\_type\_id | NVARCHAR(50) |  |
| import\_from\_device\_id | NVARCHAR(125) |  |
| import\_to\_device\_id | NVARCHAR(125) |  |
| tmp\_hostid | BIGINT | The tmp\_ values are used to |
| tmp\_inventory\_date\_host | DATETIME | match from other Importer, to |
| tmp\_ipaddress | NVARCHAR(255) | create relations |
| tmp\_uuid | NVARCHAR(40) |  |
| tmp\_elementname | NVARCHAR(125) |  |
| tmp\_powerstate | NVARCHAR(255) |  |

If Temp Table #csp\_aspera\_connector\_device\_relation exist, the Data will not shown. It will put into this table.

Used by:

### csp\_aspera\_connector\_device\_relation\_types

Get all Device Relation Types what are used by RayVentory by default.

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(50) | Relation Type ImportID, look Relation Type List |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| name | NVARCHAR(255) | Name of Relation Type, look Relation Type List |
| virtualization\_type | NVARCHAR(4000) | Type of Virtualisation |

Relation Type List:

|  |  |  |  |
| --- | --- | --- | --- |
| import\_id | name | virtualization\_type | Description |
| is\_running\_on\_zone | Is running on Zone | hw\_partition | HW Partition Zone |
| is\_running\_on\_lpar | Is running on LPAR | hw\_partition | HW Partition LPAR |
| is\_part\_of\_esx\_cluster | Is running on ESX Cluster | part\_of\_cluster | ESX Host - Cluster Relation |
| is\_running\_on\_esx | Is running on ESX | sw\_partition | ESX Guest SW Partition |
| is\_part\_of\_hyperv\_cluster | Is running on HyperV Cluster | part\_of\_cluster | HyperV Host – Cluster Relation |
| Is\_running\_on\_hyperv | Is running on HyperV | sw\_partition | HyperV Guest SW Partition |

Used by:



### csp\_aspera\_connector\_software\_arp

Get all ARP Information for Aspera

This shows all ARP Information

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @bImportIDFQDN | 1 | 0 = Use Hostname  1 = Use FQDN |
| @bNoUpdate | 1 | 0 = All Update Packages are shown  1 = no Update Packages are shown |
| @iLastSeenDays | 90 | Last seen days, by default Inventories of the last 90 days |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(255) | Hostname or FQDN depend on @bImportIDFQDN to Hostname or FQDN depend on @bImportIDFQDN or CluterName and more |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| publisher | NVARCHAR(255) |  |
| product | NVARCHAR(255) |  |
| product\_version | NVARCHAR(255) |  |
| import\_device\_id | NVARCHAR(125) | Import\_device\_id from witch this sowfware element is. |
| installation\_date | NVARCHAR(10) | Format dd.mm.YYYY |

If Temp Table #csp\_aspera\_connector\_software\_arp exist, the Data will not shown. It will put into this table.

Used by:

### csp\_aspera\_connector\_software\_msi

Get all MSI Software Information for Aspera

This shows all MSI Information

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @bImportIDFQDN | 1 | 0 = Use Hostname  1 = Use FQDN |
| @bNoUpdate | 1 | 0 = All Update Packages are shown  1 = no Update Packages are shown |
| @iLastSeenDays | 90 | Last seen days, by default Inventories of the last 90 days |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(255) | Hostname or FQDN depend on @bImportIDFQDN to Hostname or FQDN depend on @bImportIDFQDN or CluterName and more |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| Publisher | NVARCHAR(255) |  |
| Product | NVARCHAR(255) |  |
| product\_version | NVARCHAR(255) |  |
| guid | NVARCHAR(255) |  |
| import\_device\_id | NVARCHAR(125) | Import\_device\_id from witch this sowfware element is. |
| installation\_date | NVARCHAR(10) | Format dd.mm.YYYY |

If Temp Table #csp\_aspera\_connector\_software\_msi exist, the Data will not shown. It will put into this table.

Used by:

### csp\_aspera\_connector\_software\_file

Get all File Information for Aspera.

This shows all File Information

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @bImportIDFQDN | 1 | 0 = Use Hostname  1 = Use FQDN |
| @noUsageData | 0 | 0 = include Meetering Data  1 = with out Meetering Data |
| @iLastSeenDays | 90 | Last seen days, by default Inventories of the last 90 days |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(255) | Hostname or FQDN depend on @bImportIDFQDN to Hostname or FQDN depend on @bImportIDFQDN or CluterName and more |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| file\_name | NVARCHAR(255) |  |
| file\_size | NVARCHAR(255) |  |
| file\_path | NVARCHAR(255) |  |
| Publisher | NVARCHAR(255) |  |
| Product | NVARCHAR(255) |  |
| product\_version | NVARCHAR(255) |  |
| import\_device\_id | NVARCHAR(125) | Import\_device\_id from witch this sowfware element is. |
| installation\_date | NVARCHAR(10) | Format dd.mm.YYYY |

If Temp Table #csp\_aspera\_connector\_software\_file exist, the Data will not shown. It will put into this table.

Used by:



### csp\_aspera\_connector\_software\_tag

Get all File Tag Information for Aspera

This shows all File Information

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @bImportIDFQDN | 1 | 0 = Use Hostname  1 = Use FQDN |
| @iLastSeenDays | 90 | Last seen days, by default Inventories of the last 90 days |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(255) | Hostname or FQDN depend on @bImportIDFQDN to Hostname or FQDN depend on @bImportIDFQDN or CluterName and more |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| product\_title | NVARCHAR(255) |  |
| product\_ver\_name | NVARCHAR(255) |  |
| product\_ver\_major | NVARCHAR(255) |  |
| product\_ver\_minor | NVARCHAR(255) |  |
| manufacturer\_name | NVARCHAR(255) |  |
| manufacturer\_guid | NVARCHAR(255) |  |
| import\_device\_id | NVARCHAR(125) | Import\_device\_id from witch this sowfware element is. |
| installation\_date | NVARCHAR(10) | Format dd.mm.YYYY |

If Temp Table #csp\_aspera\_connector\_software\_tag exist, the Data will not shown. It will put into this table.

Used by:

### csp\_aspera\_connector\_software\_generic\_othersw

Get all generic software, this means Linux Software or other Package Manager like Install Anywhere.

This shows all other software catalog information, like Linux Package Manager

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @bImportIDFQDN | 1 | 0 = Use Hostname  1 = Use FQDN |
| @bNoUpdate | 1 | 0 = All Update Packages are shown  1 = no Update Packages are shown |
| @iLastSeenDays | 90 | Last seen days, by default Inventories of the last 90 days |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(255) | Hostname or FQDN depend on @bImportIDFQDN to Hostname or FQDN depend on @bImportIDFQDN or CluterName and more |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| generic\_key | NVARCHAR(255) |  |
| description | NVARCHAR(255) |  |
| Publisher | NVARCHAR(255) |  |
| Product | NVARCHAR(255) |  |
| product\_version | NVARCHAR(255) |  |
| quantity | NVARCHAR(3) |  |
| import\_device\_id | NVARCHAR(125) | Import\_device\_id from witch this sowfware element is. |
| import\_account\_id | NVARCHAR(125) |  |
| installation\_date | NVARCHAR(10) | Format dd.mm.YYYY |
| instance\_name | NVARCHAR(255) |  |

If Temp Table #csp\_aspera\_connector\_software\_generic\_othersw exist, the Data will not shown. It will put into this table.

Used by:



### csp\_aspera\_connector\_software\_generic\_os

Get all OS Software Information for Aspera, this will put as Generic Information.

This shows all OS Information

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @bImportIDFQDN | 1 | 0 = Use Hostname  1 = Use FQDN |
| @iLastSeenDays | 90 | Last seen days, by default Inventories of the last 90 days |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(255) | Hostname or FQDN depend on @bImportIDFQDN to Hostname or FQDN depend on @bImportIDFQDN or CluterName and more |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| generic\_key | NVARCHAR(255) |  |
| description | NVARCHAR(255) |  |
| publisher | NVARCHAR(255) |  |
| product | NVARCHAR(255) |  |
| product\_version | NVARCHAR(255) |  |
| quantity | NVARCHAR(3) |  |
| import\_device\_id | NVARCHAR(125) | Import\_device\_id from witch this sowfware element is. |
| import\_account\_id | NVARCHAR(125) |  |
| installation\_date | NVARCHAR(10) | Format dd.mm.YYYY |
| instance\_name | NVARCHAR(255) |  |

If Temp Table #csp\_aspera\_connector\_software\_generic\_os exist, the Data will not shown. It will put into this table.

Used by:

* Store Procedure Used by Aspera Connector 500\_raynet\_software\_generic.xml

### csp\_aspera\_connector\_software\_generic\_microsoft

Get all special Information for Microsoft products:

* Microsoft SQL
* Microsoft Exchange
* Microsoft SharePoint

This shows all Special Microsoft Products like SQL, Exchange and Sharepoint

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @bImportIDFQDN | 1 | 0 = Use Hostname  1 = Use FQDN |
| @iLastSeenDays | 90 | Last seen days, by default Inventories of the last 90 days |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(255) | Hostname or FQDN depend on @bImportIDFQDN to Hostname or FQDN depend on @bImportIDFQDN or CluterName and more |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| generic\_key | NVARCHAR(255) |  |
| description | NVARCHAR(255) |  |
| publisher | NVARCHAR(255) |  |
| product | NVARCHAR(255) |  |
| product\_version | NVARCHAR(255) |  |
| quantity | NVARCHAR(3) |  |
| import\_device\_id | NVARCHAR(125) | Import\_device\_id from witch this sowfware element is. |
| import\_account\_id | NVARCHAR(125) |  |
| installation\_date | NVARCHAR(10) | Format dd.mm.YYYY |
| instance\_name | NVARCHAR(255) |  |

If Temp Table #csp\_aspera\_connector\_software\_generic\_microsoft exist, the Data will not shown. It will put into this table.

Used by:

* Store Procedure Used by Aspera Connector 500\_raynet\_software\_generic.xml

### csp\_aspera\_connector\_software\_generic\_oracle

Get all Oracle DB Information:

* Oracle DB
* Oracle DB Feature and Options

This shows all Special Oracle Products like DB and his features and option packs

Parameter:

|  |  |  |
| --- | --- | --- |
| Parameter | Default value | Description |
| @sDataSourceID | RayVentory | Smarttrack Connector SourceID |
| @bImportIDFQDN | 1 | 0 = Use Hostname  1 = Use FQDN |
| @iLastSeenDays | 90 | Last seen days, by default Inventories of the last 90 days |

Return:

|  |  |  |
| --- | --- | --- |
| Colum name | Type | Description |
| import\_id | NVARCHAR(255) | Hostname or FQDN depend on @bImportIDFQDN to Hostname or FQDN depend on @bImportIDFQDN or CluterName and more |
| import\_data\_source\_id | NVARCHAR(50) | Value of @sDataSourceID |
| generic\_key | NVARCHAR(255) |  |
| description | NVARCHAR(255) |  |
| publisher | NVARCHAR(255) |  |
| product | NVARCHAR(255) |  |
| product\_version | NVARCHAR(255) |  |
| quantity | NVARCHAR(3) |  |
| import\_device\_id | NVARCHAR(125) | Import\_device\_id from witch this sowfware element is. |
| import\_account\_id | NVARCHAR(125) |  |
| installation\_date | NVARCHAR(10) | Format dd.mm.YYYY |
| instance\_name | NVARCHAR(255) |  |

If Temp Table #csp\_aspera\_connector\_software\_generic\_oracle exist, the Data will not shown. It will put into this table.

Used by:

* Store Procedure Used by Aspera Connector 500\_raynet\_software\_generic.xml

# Configuration

## Oracle

For Oracle it give a Property what we read out of ReportConfiguration Table.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |