

# **FlexNet Manager Suite Schema Reference**

# Contents

<b>Preface: Overview.....</b>	<b>V</b>
-------------------------------	----------

<b>Chapter 1: Compliance Database Schema.....</b>	<b>7</b>
---------------------------------------------------	----------

Information Structure.....	8
BatchProcessing.Common Tables.....	8
BatchProcessing Tables.....	15
Compliance.Logic.Administration Tables.....	16
Compliance.Logic.Assets Tables.....	32
Compliance.Logic.Beacon Tables.....	49
Compliance.Logic.Core Tables.....	92
Compliance.Logic.Discovery Tables.....	259
Compliance.Logic.Licensing Tables.....	273
Compliance.Logic.Structure Tables.....	459
Compliance.Logic.Users Tables.....	465
Compliance.SAP Tables.....	473
ManageSoft Tables.....	542
ReferenceData Tables.....	543
Rights Tables.....	545
Targeting Tables.....	547
Tenants Tables.....	548

<b>Chapter 2: Compliance Reader Database Schema.....</b>	<b>551</b>
----------------------------------------------------------	------------

Information Structure.....	552
Compliance.InventoryReader.Logic Tables.....	552
Compliance.InventoryWriter.Matching Tables.....	629

<b>Chapter 3: Inventory Database Schema.....</b>	<b>634</b>
--------------------------------------------------	------------

Information Structure.....	635
AD Tables.....	635
Allocation Tables.....	638
DirectoryObjects Tables.....	645
Directory Tables.....	647
Distribution Tables.....	653
IM_Right Tables.....	658
Installation Tables.....	659
Inventory Tables.....	663
Licensing Tables.....	691
ManageSoft Tables.....	696
Networking Tables.....	697
Packaging Tables.....	698
ReferenceData Tables.....	707
Rights Tables.....	710
Status Tables.....	711
Targeting Tables.....	713
Tenants Tables.....	714

Usage Tables.....	716
WakeOnLAN Tables.....	719
WorkFlow Tables.....	722
 <b>Chapter 4: License Portal Database Schema.....</b>	 <b>729</b>
Information Structure.....	730
Compliance.ECM.Logic Tables.....	730
 <b>Chapter 5: Inventory Spreadsheet Templates.....</b>	 <b>739</b>
Information Structure for Spreadsheet Inventory Imports.....	740
Compliance.InventoryReader.Logic Tables.....	742

# Legal Information

**Document Name:** FlexNet Manager Suite 2015 R2 SP5 Schema Reference (for on premises and cloud implementations)

**Part Number:** FMS-11.5.0-DR01

**Product Release Date:** May 18, 2016

## Copyright Notice

Copyright © 2016 Flexera Software LLC. All Rights Reserved.

This publication contains proprietary and confidential technology, information and creative works owned by Flexera Software LLC and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such publication in whole or in part in any form or by any means without the prior express written permission of Flexera Software LLC is strictly prohibited. Except where expressly provided by Flexera Software LLC in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera Software LLC intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera Software LLC, must display this notice of copyright and ownership in full.

FlexNet Manager Suite incorporates software developed by others and redistributed according to license agreements. Copyright notices and licenses for this externally-developed software are provided in the link below.

## Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see <http://www.flexerasoftware.com/intellectual-property>. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

## Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.

---

## Preface

---

# Overview

The data underlying FlexNet Manager Suite is arranged in a number of distinct databases. Most frequently these databases live within a single database server, although in very large scale implementations, it is possible to implement them across multiple servers.

The most fundamental distinction is between:

- Data imported from various instances of the FlexNet inventory agent as software and hardware inventory from individual computers within the enterprise (see *Inventory Database Schema* on page 634)
- Data used to calculate license positions, combining the software applications recognized from the imported inventory, the license entitlements collated from purchase records and other sources, structural information about the enterprise itself, and so on (see *Compliance Database Schema* on page 7).

A small set of tables is common to both these databases. These shared tables are documented within each of the above chapters.

In support of this basic structure, there are also the following major aspects:

- Staging tables used to rationalize data being imported into the main compliance database by `ComplianceReader.exe` (see *Compliance Reader Database Schema* on page 551)
- A separate schema for presenting summarized license information on a once-separate web portal (see *License Portal Database Schema* on page 729).

Each of the chapters covering these schemata has a common structure:

- The chapter header includes a list of different *aspects* of the data described in the chapter. (These aspects are also the lowest level included in the summary table of contents for the entire volume.)
- The chapter header is followed by a reminder of the information structure in each of the database table descriptions.
- Each aspect then has a section header page listing all the individual database tables contained within that aspect.
- Finally, the detailed topics, one for each database table, listing all the properties (columns) in the table and various attributes of each one.

This structure makes it easy to drill down from a high-level understanding of the data structure to an individual table. Conversely, if you know a table name, use the index at the end of this document to jump directly to its

description. Finally, use the PDF search mechanism in your reader software to locate individual properties within tables, when you don't know their provenance.

One final chapter takes a slightly different approach. Rather than documenting an internal schema, it covers the schema used for spreadsheets importing inventory information, and the mapping of those columns to the relevant database tables and column.

This document is not an exhaustive description of the entire database structure. For example, the system makes widespread use of views extracted from these underlying tables for (amongst other reasons) performance improvements. These views are not documented here. Nor are the mechanisms used in a multi-tenant implementation for partitioning each tenant's data made explicit in this document. However, this is a complete description of all the basic data tables from which all else is derived.

Furthermore, the descriptions of each database table are compiled automatically using the same mechanism that generates the database schemata themselves. This process guarantees complete coverage of all tables at each release.

---

# 1

---

## Compliance Database Schema

### Topics:

- *Information Structure*
- *BatchProcessing.Common Tables*
- *BatchProcessing Tables*
- *Compliance.Logic.Administration Tables*
- *Compliance.Logic.Assets Tables*
- *Compliance.Logic.Beacon Tables*
- *Compliance.Logic.Core Tables*
- *Compliance.Logic.Discovery Tables*
- *Compliance.Logic.Licensing Tables*
- *Compliance.Logic.Structure Tables*
- *Compliance.Logic.Users Tables*
- *Compliance.SAP Tables*
- *ManageSoft Tables*
- *ReferenceData Tables*
- *Rights Tables*
- *Targeting Tables*
- *Tenants Tables*

This chapter describes the schema for the main database underlying FlexNet Manager Suite.

Separately documented is the schema for the inventory tables for inventory gathered by the FlexNet inventory agent, either when installed on 'adopted' devices, or when executing a remote, zero-touch inventory (see *Inventory Database Schema* on page 634).

Some tables from that inventory database are (correctly) duplicated in this compliance database, and these shared tables are also listed toward the end of this chapter.

# Information Structure

The following information is provided about database tables. Items appear only when relevant to the database column, and are suppressed where they do not apply. Two of these items (shown bold) are columns in the following pages, and the remainder are displayed within the **Details**.

Item	Comment
<b>Database Column</b>	The name of the column in the SQL table.
<i>Type</i>	The data type of the contents of the database column.
Size	For types that have a maximum capacity, the upper limit is provided in parentheses.
Key	The word "Key" appears when a column is a unique key field within the table. It is possible for several database columns to be part of the key, so that this indicator may appear for several columns in a table.
Generated ID	This indicates that a numeric ID is assigned by the database.
Nullable	If this indicator is present, the database column permits nulls.
Computed	This indicator appears for columns that are automatically computed by the database.
Default	If a column has a default value declared in the schema, this is specified at the end of the first set of details for the column.
<b>Details</b>	Describes the data stored in the database column, including many of the indicators described above.

## BatchProcessing.Common Tables

The complete set of database tables documented here includes:

- BatchProcessExecution table (see *BatchProcessExecution Table* on page 9)
- BatchProcessExecutionData table (see *BatchProcessExecutionData Table* on page 10)
- BatchProcessExecutionDataName table (see *BatchProcessExecutionDataName Table* on page 11)
- BatchProcessSchedule table (see *BatchProcessSchedule Table* on page 11)
- BatchProcessStatus table (see *BatchProcessStatus Table* on page 12)
- BatchProcessType table (see *BatchProcessType Table* on page 13)
- BatchProcessTypeLimit table (see *BatchProcessTypeLimit Table* on page 14)



# BatchProcessExecution Table

`BatchProcessExecution` is a table storing the details of batch processes requested and executed.

**Table 1: Database columns for `BatchProcessExecution` table**

Database Column	Details
<code>BatchProcessExecutionID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a batch processor.
<code>GUID</code>	<i>Type:</i> unique identifier. Key The GUID identifying a batch process execution.
<code>BatchProcessTypeID</code>	<i>Type:</i> integer. Key The type of this batch process execution. Foreign key to the <code>BatchProcessType</code> table.
<code>Submitted</code>	<i>Type:</i> datetime The date and time at which this batch process execution was submitted.
<code>OperatorLogin</code>	<i>Type:</i> text (max 512 characters). Nullable The login name of the operator requesting the batch process, NULL indicates a system request.
<code>BeaconID</code>	<i>Type:</i> integer. Key. Nullable The ID of the beacon which requested a batch process execution. Foreign key to the <code>Beacon</code> table.
<code>BatchProcessorHostname</code>	<i>Type:</i> text (max 128 characters). Key. Nullable The batch processor responsible for the execution of this batch process. A processor by this name may be in the <code>BatchProcessor</code> table, but this is not required.
<code>BatchProcessStatusID</code>	<i>Type:</i> integer Status of the batch process execution. Foreign key to the <code>BatchProcessStatus</code> table.
<code>StartTime</code>	<i>Type:</i> datetime. Nullable The date and time the batch process execution was started.
<code>FinishTime</code>	<i>Type:</i> datetime. Key. Nullable The date and time the batch process execution finished.

Database Column	Details
Progress	<i>Type:</i> integer Percentage indicator of how far through the batch process execution is.
ReturnCode	<i>Type:</i> integer. Nullable The return code of the batch process execution.
Output	<i>Type:</i> text. Nullable Contains any output reported by a batch process execution.
GroupName	<i>Type:</i> text (max 50 characters). Nullable The group name used to partition this batch process. Only relevant for types that require separation by group.
TenantUID	<i>Type:</i> text (max 40 characters). Nullable The tenant UID for this batch process. Only relevant for types that require separation by tenant.
RawMessage	<i>Type:</i> text. Nullable The raw, serialized message. Used for pending messages to reconstruct the queue when the batch processor restarts.

## BatchProcessExecutionData Table

This table stores any extra data needed for a `BatchProcessExecution` record.

**Table 2: Database columns for BatchProcessExecutionData table**

Database Column	Details
BatchProcessExecutionDataID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this table.
BatchProcessExecutionID	<i>Type:</i> integer. Key The ID of the <code>BatchProcessExecution</code> record this data is associated with. Foreign key to the <code>BatchProcessExecution</code> table.
BatchProcessExecutionDataNameID	<i>Type:</i> integer. Key An identifier for the data being stored in this row
DataValue	<i>Type:</i> text

Database Column	Details
	The value being stored in this row

## BatchProcessExecutionDataName Table

This table holds a list of the different types of data that can be stored in `BatchProcessExecutionData`.

**Table 3: Database columns for `BatchProcessExecutionDataName` table**

Database Column	Details
<code>BatchProcessExecutionDataNameID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for this table.
<code>Name</code>	<i>Type:</i> text (max 128 characters). Key Name of the setting.

## BatchProcessSchedule Table

`BatchProcessSchedule` stores the schedule of a batch process.

**Table 4: Database columns for `BatchProcessSchedule` table**

Database Column	Details
<code>BatchProcessScheduleID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for this table.
<code>BatchProcessTypeID</code>	<i>Type:</i> integer. Key The process type ID this schedule belongs to. Foreign key to the <code>BatchProcessType</code> table.
<code>TenantUID</code>	<i>Type:</i> text (max 40 characters). Key. Nullable The tenant UID for this batch schedule.
<code>BatchProcessScheduleData</code>	<i>Type:</i> text The Quartz scheduler data
<code>UpdatedBy</code>	<i>Type:</i> text (max 200 characters). Nullable The last operator to update the event.

Database Column	Details
UpdatedDate	<i>Type:</i> datetime. Nullable The date the event was last updated.
GUID	<i>Type:</i> unique identifier. Key Unique identifier for schedule.
LastRun	<i>Type:</i> datetime. Nullable The datetime this schedule was last executed.
Enabled	<i>Type:</i> boolean The datetime this schedule was last executed.

## BatchProcessStatus Table

`BatchProcessStatus` is a static table listing status values for batch process execution.

**Table 5: Database columns for `BatchProcessStatus` table**

Database Column	Details
BatchProcessStatusID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>BatchProcessStatus</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Submitted</li> <li>• 2 = Queued</li> <li>• 3 = Processing</li> <li>• 4 = Success</li> <li>• 5 = Error</li> <li>• 6 = Duplicate</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing an status of batch process execution. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the status resource string has no translation.

# BatchProcessType Table

`BatchProcessType` is a static table storing the types of batch processes

**Table 6: Database columns for `BatchProcessType` table**

Database Column	Details
<code>BatchProcessTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>BatchProcessType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = License reconcile</li> <li>• 2 = PO line import</li> <li>• 3 = Enterprise group import</li> <li>• 4 = User assignment import</li> <li>• 5 = Inventory import</li> <li>• 6 = Active directory import</li> <li>• 7 = Entitlement recommendations recalculation</li> <li>• 8 = SAP user recommendations export</li> <li>• 9 = Business adapter import</li> <li>• 10 = Generate business adapter config</li> <li>• 15 = ServiceNow export</li> <li>• 16 = FNMEA enterprise groups export</li> <li>• 17 = IBM Passport Advantage import</li> <li>• 18 = Data Warehouse access rights update</li> <li>• 19 = Update license consumption of IBM PVU licenses</li> <li>• 20 = Data Warehouse export</li> <li>• 21 = Import SAP inventories</li> <li>• 22 = Import SAP package license</li> <li>• 23 = Inventory import and license reconcile</li> <li>• 24 = Recognition data import</li> <li>• 25 = Inventory manager compliance import</li> <li>• 26 = Compliance import readers only</li> <li>• 27 = Compliance import writers only</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 28 = Recognition data download</li> <li>• 29 = Recognition data cleanup</li> <li>• 30 = IM Data maintenance</li> <li>• 31 = SAP user and activity information import</li> <li>• 32 = Inventory import spreadsheet and license reconcile</li> <li>• 33 = FNMP Data maintenance</li> <li>• 34 = FNMP software usage history update</li> <li>• 35 = Delete activity log history</li> <li>• 36 = Baseline import processing</li> </ul>
TypeName	<i>Type:</i> text (max 256 characters). Key The unique name of the batch process type.
ResourceName	<i>Type:</i> text (max 256 characters) The unique name of the localizable resource string representing a batch process type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.
StarvedAt	<i>Type:</i> integer. Nullable The age, in minutes, after which a task of this type will be given priority over other tasks to avoid starvation.
Timeout	<i>Type:</i> integer. Nullable The age, in minutes, after which a task of this type will be regarded as failed if its processor becomes unresponsive.
BatchProcessTypeLimitID	<i>Type:</i> integer. Nullable An optional reference to a limit that will restrict the number of items of this type that can execute at the same time.

## BatchProcessTypeLimit Table

`BatchProcessTypeLimit` is a table storing the limits placed on the parallel execution of tasks within the Batch Processor. A limit is associated with one or more `BatchProcessTypes`. The limit value is the number of tasks of the associated types that may be executed at any one time.

Note that these limits are applied after the standard parallel execution restrictions are applied. This means that these limits will generally affect a single tenant system. They will take effect only if the limit is applied to types that are allowed to run in parallel for a tenant. For example, if a limit is applied to a types that run the `ComplianceReader` executable, the Business importer and the ARL import, it may be possible to reach the limit.

In a multi-tenant system, the limits allow the system administrator to define reasonable limits to try to ensure that the Batch scheduler does not overload the hardware it is allotted.

**Table 7: Database columns for `BatchProcessTypeLimit` table**

Database Column	Details
<code>BatchProcessTypeLimitID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a <code>BatchProcessTypeLimit</code> .
<code>Name</code>	<i>Type:</i> text (max 128 characters). Key The name of this <code>BatchProcessTypeLimit</code> . This name will be used internally to reference the limit, and will be shown in the tracing output.
<code>MaxTasks</code>	<i>Type:</i> integer The number of tasks associated with this limit that may be executed in parallel by the Batch scheduler. A zero or negative value in this column will cause the limit to be ignored.

## BatchProcessing Tables

The complete set of database tables documented here includes:

- `BatchProcessor` table (see *BatchProcessor Table* on page 15)
- `BatchProcessorProcessType` table (see *BatchProcessorProcessType Table* on page 16)

## BatchProcessor Table

`BatchProcessor` is a table storing the machines responsible for executing batch processes.

**Table 8: Database columns for `BatchProcessor` table**

Database Column	Details
<code>BatchProcessorID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a batch processor.
<code>Hostname</code>	<i>Type:</i> text (max 128 characters). Key

Database Column	Details
	The host name of this batch processor.
LastHeartbeat	Type: datetime. Nullable The UTC date and time this batch processor configured.
LastExecution	Type: datetime. Nullable The UTC date and time this batch processor last executed a batch process.

## BatchProcessorProcessType Table

This table records the mapping of process types to batch processors.

**Table 9: Database columns for BatchProcessorProcessType table**

Database Column	Details
BatchProcessorID	Type: integer. Key The ID of the BatchProcessor record this data is associated with. Foreign key to the BatchProcessor table.
BatchProcessTypeID	Type: integer. Key The ID of the BatchProcessType record this data is associated with. Foreign key to the BatchProcessorType table.

## Compliance.Logic.Administration Tables

The complete set of database tables documented here includes:

- APIServiceAccount table (see *APIServiceAccount Table* on page 17)
- ComplianceConnection table (see *ComplianceConnection Table* on page 18)
- ComplianceCultureType table (see *ComplianceCultureType Table* on page 21)
- ComplianceOperator table (see *ComplianceOperator Table* on page 21)
- ComplianceOperatorTenant table (see *ComplianceOperatorTenant Table* on page 23)
- ComplianceResourceString table (see *ComplianceResourceString Table* on page 24)
- ComplianceSetting table (see *ComplianceSetting Table* on page 24)
- ComplianceTenantSetting table (see *ComplianceTenantSetting Table* on page 24)



- ConfigurationFile table (see *ConfigurationFile Table* on page 25)
- ConfigurationFileType table (see *ConfigurationFileType Table* on page 26)
- ConnectionType table (see *ConnectionType Table* on page 26)
- Currency table (see *Currency Table* on page 27)
- MasterConfigurationFile table (see *MasterConfigurationFile Table* on page 28)
- OperatorTenantSetting table (see *OperatorTenantSetting Table* on page 29)
- ResourceStringCultureType table (see *ResourceStringCultureType Table* on page 30)
- RightDefinition table (see *RightDefinition Table* on page 30)
- SettingName table (see *SettingName Table* on page 31)
- TimezoneType table (see *TimezoneType Table* on page 31)

## APIServiceAccount Table

Stores a collection of external API service accounts.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 10: Database columns for APIServiceAccount table**

Database Column	Details
APIServiceAccountID	Type: integer. Key. Generated ID Unique identifier for a API service account.
ComplianceOperatorID	Type: integer. Key. Nullable Reference to a compliance operator.
AccessThreshold	Type: integer. Nullable API access alert threshold
AccessCount	Type: integer API access count.
LastSync	Type: datetime. Nullable

Database Column	Details
	Indicates the last datetime this account is synced with FNOOD or validateToken API is called.
Description	Type: text (max 256 characters). Nullable Description for this service account.
CreationUser	Type: text (max 256 characters). Nullable Created by.
CreationDate	Type: datetime. Nullable Creation date.
UpdatedUser	Type: text (max 256 characters). Nullable Operator who made the latest change to the currency record.
UpdatedDate	Type: datetime. Nullable Updated date

## ComplianceConnection Table

The `ComplianceConnection` table stores details about databases configured for use in compliance imports, such as Microsoft SMS.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 11: Database columns for `ComplianceConnection` table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key. Generated ID A unique identifier for a compliance connection.
ConnectionTypeID	Type: integer. Key The compliance connection type. Foreign key to the <code>ConnectionType</code> table.
ConnectionName	Type: text (max 128 characters). Key The internal, unique name of the connection.

Database Column	Details
ConnectionNameDisplayName	<p><i>Type:</i> text (max 64 characters)</p> <p>The name of the connection for display purposes.</p>
UseFnmpDbServerAsSource	<p><i>Type:</i> boolean</p> <p>Use the FNMP database server as the source.</p>
Server	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The name of the SQL Server.</p>
UseWindowsAuth	<p><i>Type:</i> boolean. Nullable</p> <p>If this field is set to <code>True</code>, the connection will use Windows authentication when connecting to the database. If <code>False</code>, SQL authentication will be used.</p>
Username	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The username to use when connecting with SQL authentication.</p>
Password	<p><i>Type:</i> text. Nullable</p> <p>The password to use when connecting with SQL authentication.</p>
DatabaseName	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The name of the database to connect to.</p>
ConnectionString	<p><i>Type:</i> text. Nullable</p> <p>The connection string used to connect to a datasource.</p>
LastImportDate	<p><i>Type:</i> datetime. Nullable</p> <p>Date and time when data from this data source was successfully imported into the staging area (reader execution). The imported data may not have been applied to the core tables.</p>
LastImportStarted	<p><i>Type:</i> datetime. Nullable</p> <p>Date and time when the import from this data source started.</p>
LastImportEnded	<p><i>Type:</i> datetime. Nullable</p> <p>Date and time when the import from this data source ended.</p>
LastImportSuccessful	<p><i>Type:</i> boolean</p> <p>Whether or not the last import attempted for this datasource succeeded or failed.</p>
SourceType	<p><i>Type:</i> text (max 256 characters)</p>

Database Column	Details
	The source database type (one of several predefined values, such as <code>ManageSoft</code> or <code>SMS</code> ).
<code>SourceTypeDisplayName</code>	<p><i>Type:</i> text (max 128 characters)</p> <p>A version of the <code>SourceType</code> field, that has been scoped to be specific to this connection.</p>
<code>Signature</code>	<p><i>Type:</i> text (max 128 characters)</p> <p>A connection signature optionally given by the source database. This allows the source database to identify its connection.</p>
<code>PrimaryConnection</code>	<p><i>Type:</i> boolean. Key</p> <p>Set this to <code>True</code> if this is the primary data source to import from. If computers or users exist in multiple connections, data from the primary connection is always given precedence.</p>
<code>TestConnection</code>	<p><i>Type:</i> boolean</p> <p>Indicate if this connection is a test connection. If this is set to <code>True</code> writer will not populate target FNMP tables with data in the imported tables from this connection. If this is set to <code>False</code> writer will populate data from this connection as is. Compliance Reader Editor UI sets connection as test so that test data would not accidentally be written to target FNMP tables.</p>
<code>Enabled</code>	<p><i>Type:</i> boolean</p> <p>Indicate if this connection is enabled. If this is set to <code>False</code> reader will not import data from this connection.</p>
<code>GroupName</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The <code>GroupName</code> represents subgroups of data from the source. For example, for a citrix connection, this stores a farm name. If this is Null, then there is no sub-grouping (import all).</p>
<code>ExpiryPeriod</code>	<p><i>Type:</i> integer. Nullable</p> <p>The number of days before considering records in <code>ImportedComputer</code> to be out of date and should be considered stale. NULL means use the Compliance Setting value <code>StaleInventoryThreshold</code>. 0 means always include device data regardless of age.</p>
<code>PerformStaleInventoryCheck</code>	<p><i>Type:</i> boolean</p> <p>Indicates if this connection needs to have the inventory checked to see if data is considered stale. It is reset to 1 after completing the reader's step of an import.</p>

Database Column	Details
IsRemote	<i>Type:</i> boolean Is this a remote connection, where the source side of the readers are running on a remote location (an Inventory Beacon)?
ConnectionExID	<i>Type:</i> unique identifier. Key The externally unique identifier for this connection, that can be used by both an Inventory Beacon and the server to track a connection.
BeaconUID	<i>Type:</i> unique identifier. Key. Nullable The unique ID of the beacon where this connection is running.

## ComplianceCultureType Table

The `ComplianceCultureType` table holds all the different languages that FlexNet Manager Suite supports.

**Table 12: Database columns for `ComplianceCultureType` table**

Database Column	Details
CultureType	<i>Type:</i> text (max 12 characters). Key A unique identifier for a culture type.
DefaultCulture	<i>Type:</i> boolean Indicates whether this language is a default language on the system.
Installed	<i>Type:</i> boolean Indicates whether string for this language are installed.
DisplayName	<i>Type:</i> text (max 80 characters) The display name for this culture.

## ComplianceOperator Table

`ComplianceOperator` stores the list of people (operators) authorized to use FlexNet Manager Suite. Operators need not be end-users of the enterprise.

**Table 13: Database columns for ComplianceOperator table**

Database Column	Details
ComplianceOperatorID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the operator.
OperatorLogin	<i>Type:</i> text (max 256 characters). Key The login (account name) of the operator. Usually of the form [domain \account].
OperatorName	<i>Type:</i> text (max 512 characters). Nullable The name of the operator.
IsEnabled	<i>Type:</i> boolean When <code>False</code> , this operator may not use FlexNet Manager Suite, even if he or she is assigned to roles granting them access.
Email	<i>Type:</i> text (max 200 characters). Nullable The operator's email address.
JobTitle	<i>Type:</i> text (max 128 characters). Nullable The job title of the end-user.
ComplianceUserID	<i>Type:</i> integer. Key. Nullable An optional link to an end-user in the system. Foreign key to the <code>ComplianceUser</code> table.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.
BusinessReportingToken	<i>Type:</i> text (max 256 characters). Nullable A token that is issued to an operator to allow them to authenticate with the business reporting framework.

Database Column	Details
TenantID	<i>Type:</i> small integer. Nullable The default tenant that this operator works on. Note that there is no tenant-filtered view on this table.
GlobalOperator	<i>Type:</i> boolean Allows an operator to access all tenants.
Interactive	<i>Type:</i> boolean Non-interactive accounts are service accounts.
LastLogin	<i>Type:</i> datetime. Nullable Last login datetime.
LastLogout	<i>Type:</i> datetime. Nullable Last logout datetime.

## ComplianceOperatorTenant Table

ComplianceOperatorTenant stores the list of people (operators) authorized to access a tenant.

**Table 14: Database columns for ComplianceOperatorTenant table**

Database Column	Details
ComplianceOperatorID	<i>Type:</i> integer. Key The operatorID that the permission will be granted for.
TenantId	<i>Type:</i> small integer. Key The tenantID that the operator will be granted access for.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
IsEnabled	<i>Type:</i> boolean When <i>False</i> , this operator may not use FlexNet Manager Suite, even if he or she is assigned to roles granting them access.

## ComplianceResourceString Table

The `ComplianceResourceString` table holds all the strings that require translation.

**Table 15: Database columns for `ComplianceResourceString` table**

Database Column	Details
<code>ResourceString</code>	<i>Type:</i> text (max 256 characters). Key A unique identifier for a string.

## ComplianceSetting Table

The `ComplianceSetting` table holds the settings for the configuration and business rules of the application. With the introduction of `SettingName`, `ComplianceTenantSetting` and `OperatorTenantSetting` tables, if new global setting is to be added to `ComplianceSetting` table, the `ComplianceSettingID` must not overlap with those defined in `SettingName` table.

**Table 16: Database columns for `ComplianceSetting` table**

Database Column	Details
<code>ComplianceSettingID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a setting.
<code>SettingName</code>	<i>Type:</i> text (max 128 characters). Key A primary key for the setting.
<code>SettingValue</code>	<i>Type:</i> text (max 512 characters) The setting that indicates specified behavior.

## ComplianceTenantSetting Table

`ComplianceTenantSetting` is a multi-tenant table that stores configuration and business rules specific to each tenant.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.



**Table 17: Database columns for ComplianceTenantSetting table**

Database Column	Details
SettingNameID	Type: integer. Key ID of the setting name. Foreign key to the SettingName table.
SettingValue	Type: text (max 512 characters). Nullable Value of the setting.

## ConfigurationFile Table

The ConfigurationFile table stores configuration files generated from the master configuration files used by FlexNet Manager Suite.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 18: Database columns for ConfigurationFile table**

Database Column	Details
ConfigurationFileID	Type: integer. Key. Generated ID A unique identifier for a configuration file.
ConfigurationFileTypeID	Type: integer. Key The configuration file type. Foreign key to the ConfigurationFileType table.
Name	Type: text (max 100 characters) The name of the configuration file.
Revision	Type: integer The revision of the configuration file.
XMLFile	Type: text The content of the configuration file.

## ConfigurationFileType Table

`ConfigurationFileType` is a static table storing the types of configuration files used by FlexNet Manager Suite.

**Table 19: Database columns for ConfigurationFileType table**

Database Column	Details
<code>ConfigurationFileTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ConfigurationFileType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = SQL Server</li> <li>• 2 = Other (the inventory source is another type of data store, like an Excel sheet or MS Access database).</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a configuration file type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## ConnectionType Table

`ConnectionType` is a static table storing the types of connection that can be used to import data into FlexNet Manager Suite.

**Table 20: Database columns for ConnectionType table**

Database Column	Details
<code>ConnectionTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ConnectionType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = SQL Server</li> <li>• 2 = Other (the inventory source is another type of data store, like an Excel sheet or MS Access database).</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p>

Database Column	Details
	The unique name of the localizable resource string representing a connection type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	Type: text (max 100 characters) The text to display if the type resource string has no translation.

## Currency Table

Currency stores a collection of currencies that can be used for money values.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 21: Database columns for Currency table**

Database Column	Details
CurrencyID	Type: integer. Key. Generated ID Unique identifier for a currency.
CurrencyName	Type: text (max 256 characters) Name of currency.
CurrencyResourceID	Type: text (max 64 characters). Nullable The resource string containing the name of this currency to display on the user interface.
CurrencyCode	Type: text (max 32 characters). Key Code assigned to currency.
LongPrefix	Type: text (max 32 characters) Long prefix to display in front of the money value.
LongSuffix	Type: text (max 32 characters) Long suffix to display after the money value.
LongFormat	Type: text (max 80 characters). Nullable Long format of the currency. This is a calculated field.

Database Column	Details
ShortPrefix	<i>Type:</i> text (max 32 characters) Short prefix to display in front of the money value.
ShortSuffix	<i>Type:</i> text (max 32 characters) Short suffix to display after the money value.
ShortFormat	<i>Type:</i> text (max 80 characters). Nullable Short format of the currency. This is a calculated field.
IsActive	<i>Type:</i> boolean. Key Indicates whether this currency is enabled.
Comments	<i>Type:</i> text. Nullable Operator comments about this currency.
Countries	<i>Type:</i> text (max 2048 characters). Nullable A semicolon-separated list of the country codes for countries to which this currency is applicable.
ActivationDate	<i>Type:</i> datetime. Nullable Date currency was enabled.
RetirementDate	<i>Type:</i> datetime. Nullable Date that currency was retired.
UpdatedUser	<i>Type:</i> text (max 256 characters). Nullable Operator who made the latest change to the currency record.
UpdatedDate	<i>Type:</i> datetime. Nullable Date that the currency record was changed.

## MasterConfigurationFile Table

The `MasterConfigurationFile` table stores master configuration files used by FlexNet Manager Suite.

**Table 22: Database columns for MasterConfigurationFile table**

Database Column	Details
MasterConfigurationFileID	<i>Type:</i> integer. Key. Generated ID

Database Column	Details
	A unique identifier for a configuration file.
ConfigurationFileTypeID	<i>Type:</i> integer. Key The configuration file type. Foreign key to the <code>ConfigurationFileType</code> table.
Name	<i>Type:</i> text (max 100 characters) The name of the configuration file.
Revision	<i>Type:</i> integer The revision of the configuration file.
XMLFile	<i>Type:</i> text The content of the configuration file.

## OperatorTenantSetting Table

`OperatorTenantSetting` is a multi-tenant table that stores configuration and preferences for each operator per tenant



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 23: Database columns for `OperatorTenantSetting` table**

Database Column	Details
OperatorTenantSettingID	<i>Type:</i> integer. Key. Generated ID Unique identifier of an operator tenant setting, this is a primary key.
ComplianceOperatorID	<i>Type:</i> integer. Key The operator of the setting. Foreign key to the <code>ComplianceOperator</code> table.
SettingNameID	<i>Type:</i> integer. Key ID of the setting name. Foreign key to the <code>SettingName</code> table.
SettingValue	<i>Type:</i> text (max 512 characters). Nullable Value of the setting.

## ResourceStringCultureType Table

The `ResourceStringCultureType` table holds all translations of all the resource strings.

**Table 24: Database columns for `ResourceStringCultureType` table**

Database Column	Details
<code>ResourceString</code>	<i>Type:</i> text (max 256 characters). Key A unique identifier for a resource string. Foreign key to the <code>ComplianceResourceString</code> table.
<code>CultureType</code>	<i>Type:</i> text (max 12 characters). Key A unique identifier for a culture type. Foreign key to the <code>ComplianceCultureType</code> table.
<code>ResourceValue</code>	<i>Type:</i> text (max 1000 characters) A translated resource string.

## RightDefinition Table

`RightDefinition` defines additional access rights that supplement the built-in rights.

**Table 25: Database columns for `RightDefinition` table**

Database Column	Details
<code>RightDefinitionID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a right definition.
<code>ResourceName</code>	<i>Type:</i> text (max 16 characters). Key Resource (such as inventory, usage tracking, and so on) that access right relates to. Foreign key to the <code>Resource</code> table.
<code>ActionClassName</code>	<i>Type:</i> text (max 16 characters). Key Action class (such as modify, read, and so on) of access right. Foreign key to the <code>ActionClass</code> table.
<code>ParentFeature</code>	<i>Type:</i> text (max 50 characters) The product feature to which this access right applies.
<code>Title</code>	<i>Type:</i> text (max 1000 characters)

Database Column	Details
	Default value for access right title.
TitleResourceString	Type: text (max 256 characters). Key. Nullable The unique name of the localizable resource string representing an access right. Foreign key to the ComplianceResourceString table.
MinAccessType	Type: text (max 50 characters). Nullable Minimum access type that allows this right. Possible values include NoAccess, ReadOnlyAccess, NormalAccess, AdministratorAccess and CustomAccess.
DisplayIndex	Type: integer. Nullable Order in which rights are displayed (smaller numbers are displayed first). FlexNet Manager Suite built-in rights have the value 100.

## SettingName Table

SettingName is a static table containing ids of setting names that are referenced by ComplianceTenantSetting and OperatorTenantSetting tables. As settings in ComplianceSetting table is migrated to either ComplianceTenantSetting and OperatorTenantSetting, ComplianceSetting table will be changed to refer to this table as well for global settings.

**Table 26: Database columns for SettingName table**

Database Column	Details
SettingNameID	Type: integer. Key. Generated ID A unique identifier for a setting name.
Name	Type: text (max 128 characters). Key Name of the setting.

## TimezoneType Table

This table stores a collection of timezonetypes.

**Table 27: Database columns for TimezoneType table**

Database Column	Details
TimezoneTypeID	<i>Type:</i> integer. Key. Generated ID Unique identifier for a TimezoneType.
TimezoneID	<i>Type:</i> text (max 128 characters) The .NET representation of the time zone id.
ResourceName	<i>Type:</i> text (max 256 characters). Nullable The unique name of the localizable resource string representing a timezone type. Foreign key to the <i>ComplianceResourceString</i> table.
DefaultValue	<i>Type:</i> text (max 256 characters) The default display timezone name

## Compliance.Logic.Assets Tables

The complete set of database tables documented here includes:

- AcquisitionMode table (see *AcquisitionMode Table* on page 33)
- Asset table (see *Asset Table* on page 33)
- AssetComplianceColumn table (see *AssetComplianceColumn Table* on page 39)
- AssetComplianceStatus table (see *AssetComplianceStatus Table* on page 41)
- AssetContract table (see *AssetContract Table* on page 41)
- AssetPropertyValue table (see *AssetPropertyValue Table* on page 42)
- AssetPurchaseOrder table (see *AssetPurchaseOrder Table* on page 43)
- AssetStatus table (see *AssetStatus Table* on page 43)
- AssetType table (see *AssetType Table* on page 44)
- AssetTypeProperty table (see *AssetTypeProperty Table* on page 45)
- AssetWarrantyType table (see *AssetWarrantyType Table* on page 46)
- DepreciationMethod table (see *DepreciationMethod Table* on page 47)
- EndOfLifeReason table (see *EndOfLifeReason Table* on page 47)
- LeaseEndReason table (see *LeaseEndReason Table* on page 48)



## AcquisitionMode Table

`AcquisitionMode` is a static table listing all the methods by which a company may obtain an asset.

**Table 28: Database columns for `AcquisitionMode` table**

Database Column	Details
<code>AcquisitionModeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>AcquisitionMode</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Purchased</li> <li>• 2 = Leased</li> <li>• 3 = Rented</li> <li>• 4 = Loaned.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an acquisition mode. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the mode resource string has no translation.</p>

## Asset Table

The `Asset` table contains details of all the assets being managed within FlexNet Manager Suite.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 29: Database columns for `Asset` table**

Database Column	Details
<code>AssetID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for an asset.</p>
<code>ParentAssetID</code>	<p><i>Type:</i> integer. Key. Nullable</p>

Database Column	Details
	The parent asset. Foreign key to another asset in this same <code>Asset</code> table.
ShortDescription	<i>Type:</i> text (max 256 characters) A brief description of the asset.
SerialNumber	<i>Type:</i> text (max 150 characters). Key. Nullable The serial number of the asset.
AssetTypeID	<i>Type:</i> integer. Key The asset type. Foreign key to the <code>AssetType</code> table.
AssetTag	<i>Type:</i> text (max 256 characters). Nullable A user-defined asset tag for a particular asset. This may be a barcode number.
AssetStatusID	<i>Type:</i> integer. Key The status of the asset. Defaults to <code>Purchased</code> . Foreign key to the <code>AssetStatus</code> table.
PurchasePrice	<i>Type:</i> currency. Nullable The purchase price of the asset.
PurchasePriceRateID	<i>Type:</i> integer. Nullable The currency rate to apply to the purchase price of the asset. Foreign key to the <code>CurrencyRate</code> table.
AcquisitionModeID	<i>Type:</i> integer. Nullable The method of acquisition used for the asset. Defaults to <code>Purchased</code> . Foreign key to the <code>AcquisitionMode</code> table.
PrimaryPurchaseOrderNo	<i>Type:</i> text (max 50 characters). Nullable The purchase order number which was used to purchase the asset.
PrimaryPurchaseOrderDate	<i>Type:</i> datetime. Nullable The date the primary purchase order was made.
VendorID	<i>Type:</i> integer. Key. Nullable The vendor from whom the asset was purchased. Foreign key to the <code>Vendor</code> table.
Manufacturer	<i>Type:</i> text (max 200 characters). Nullable The manufacturer of the asset.

Database Column	Details
ManufacturerPartNo	<i>Type:</i> text (max 100 characters). Nullable The manufacturer's part number for this asset.
ModelNo	<i>Type:</i> text (max 200 characters). Nullable The model number of the asset.
DeliveryDate	<i>Type:</i> datetime. Nullable The date the asset was received.
AssetWarrantyTypeID	<i>Type:</i> integer The type of warranty for the asset. Defaults to <code>None</code> . Foreign key to the <code>AssetWarrantyType</code> table.
WarrantyExpirationDate	<i>Type:</i> datetime. Nullable The date the warranty expires.
InstallationDate	<i>Type:</i> datetime. Nullable The date the asset was installed.
RetirementDate	<i>Type:</i> datetime. Nullable The date the asset was retired.
DisposalDate	<i>Type:</i> datetime. Nullable The date the asset was disposed of.
DeletionDate	<i>Type:</i> datetime. Nullable The date the asset was deleted.
InventoryDate	<i>Type:</i> datetime. Nullable The date the asset last had inventory reported.
InventoryAgent	<i>Type:</i> text (max 64 characters). Nullable The name of the person or tool that performed the last inventory.
InventoryDateManual	<i>Type:</i> datetime. Nullable The date the asset last had inventory updated (entered) manually.
InventoryAgentManual	<i>Type:</i> text (max 64 characters). Nullable The name of the person or tool that performed the last manual inventory.
RequestNo	<i>Type:</i> text (max 60 characters). Nullable

Database Column	Details
	The request number for the asset.
PartNo	<i>Type:</i> text (max 100 characters). Nullable The vendor's part number for this asset.
IsLeased	<i>Type:</i> boolean Flag to indicate if this asset is leased. This field is no longer in use in FlexNet Manager Suite.
LeaseNo	<i>Type:</i> text (max 60 characters). Nullable The contract number of the lease agreement for this asset.
LeaseName	<i>Type:</i> text (max 100 characters). Nullable A contract name of the lease agreement for this asset.
LeaseStartDate	<i>Type:</i> datetime. Nullable The start date of the lease for this asset.
LeaseEndDate	<i>Type:</i> datetime. Nullable The end date of the lease for this asset.
LeaseTerminationDate	<i>Type:</i> datetime. Nullable The date that the lease for this asset is terminated.
LeaseEndReasonID	<i>Type:</i> integer The reason for the end of lease for this asset.
LeasePrice	<i>Type:</i> currency. Nullable The purchase price of the lease for this individual asset.
LeasePriceRateID	<i>Type:</i> integer. Nullable The purchase price of the lease currency rate for this individual asset.
LeasePeriodicPayment	<i>Type:</i> currency. Nullable The price of periodic payments associated with this contract.
LeasePeriodicPaymentRateID	<i>Type:</i> integer. Nullable The price of periodic payments currency rate associated with this contract.
LeasePeriodTypeID	<i>Type:</i> integer The frequency with which the lease payments are applicable.

Database Column	Details
LeaseBuyoutCost	<i>Type:</i> currency. Nullable The buyout cost of the lease for this asset.
LeaseBuyoutCostRateID	<i>Type:</i> integer. Nullable The buyout cost of the lease currency rate associated for this asset.
LeaseComments	<i>Type:</i> text. Nullable Comments recorded about the lease for this asset. This field is no longer in use in FlexNet Manager Suite.
AssignToUserID	<i>Type:</i> integer. Key. Nullable The end-user the asset has been assigned to. Foreign key to the <code>ComplianceUser</code> table.
Comments	<i>Type:</i> text. Nullable Comments entered about the asset.
ChargeBackPrice	<i>Type:</i> currency. Nullable Amount to be charged back for the use of this asset. No calculations based on this charge and the frequency will be provided.
ChargeBackPriceRateID	<i>Type:</i> integer. Nullable The currency rate to be applied to the charge back value of the asset. Foreign key to the <code>CurrencyRate</code> table.
ChargeBackPeriodTypeID	<i>Type:</i> integer The frequency with which the charge back price is charged. Defaults to <code>None</code> . Foreign key to the <code>PeriodType</code> table.
EndOfLifeRecipient	<i>Type:</i> text (max 128 characters). Nullable The person or organization who received the asset when it was disposed of.
EndOfLifeReasonID	<i>Type:</i> integer The reason the asset was disposed of. Foreign key to the <code>EndOfLifeReason</code> table.
ResalePrice	<i>Type:</i> currency. Nullable The amount the asset was sold for.
ResalePriceRateID	<i>Type:</i> integer. Nullable

Database Column	Details
	The currency rate to be applied to the resale price of the asset. Foreign key to the <code>CurrencyRate</code> table.
<code>CreationUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
<code>CreationDate</code>	<i>Type:</i> datetime The date the record was created.
<code>UpdatedUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
<code>UpdatedDate</code>	<i>Type:</i> datetime. Nullable The date the record was last updated.
<code>LocationID</code>	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise location associated with this asset. Foreign key to the <code>GroupEx</code> table.
<code>BusinessUnitID</code>	<i>Type:</i> text (max 128 characters). Key. Nullable Any corporate unit in the enterprise associated with this asset. Foreign key to the <code>GroupEx</code> table.
<code>CostCenterID</code>	<i>Type:</i> text (max 128 characters). Key. Nullable Any cost center in the enterprise associated with this asset. Foreign key to the <code>GroupEx</code> table.
<code>CategoryID</code>	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise category associated with this asset. Foreign key to the <code>GroupEx</code> table.
<code>DepreciationCurrentValue</code>	<i>Type:</i> currency. Nullable The current value of the asset, after depreciation has been applied.
<code>DepreciationCurrent ValueRateID</code>	<i>Type:</i> integer. Nullable The currency rate to be applied to the depreciation current value of the asset. Foreign key to the <code>CurrencyRate</code> table.
<code>DepreciationResidualValue</code>	<i>Type:</i> currency. Nullable The residual value of the asset (value when fully depreciated).

Database Column	Details
DepreciationResidualValueRateID	<i>Type:</i> integer. Nullable The currency rate to be applied to the residual value of the asset. Foreign key to the <code>CurrencyRate</code> table.
DepreciationMethodID	<i>Type:</i> integer. Nullable The depreciation method (straight line or residual value). Foreign key to the <code>DepreciationMethod</code> table.
DepreciationPeriod	<i>Type:</i> integer The depreciation period (in years), for customers to use for straight line depreciation.
DepreciationRate	<i>Type:</i> decimal. Nullable The annual depreciation rate (as a percentage - like 50% per year), for customers to use for residual value depreciation. Stored as a value between 0 (for 0%) and 1 (for 100%).
WrittenOffValue	<i>Type:</i> currency. Nullable The written-off value is the value of the asset at the time of retirement/disposal.
WrittenOffValueRateID	<i>Type:</i> integer. Nullable The currency rate to be applied to the written-off value of the asset. Foreign key to the <code>CurrencyRate</code> table.

## AssetComplianceColumn Table

The `AssetComplianceColumn` table lists the columns (or aspects of the asset record) for which compliance changes can be tracked.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 30: Database columns for `AssetComplianceColumn` table**

Database Column	Details
AssetComplianceColumnID	<i>Type:</i> integer. Key. Generated ID

Database Column	Details
	<p>A unique identifier for each <code>AssetComplianceColumn</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Operating System</li> <li>• 2 = Service Pack</li> <li>• 3 = Number of Processors</li> <li>• 4 = Processor Type</li> <li>• 5 = Max Clock Speed</li> <li>• 6 = Total Memory</li> <li>• 7 = Chassis Type</li> <li>• 8 = Number of Hard Drives</li> <li>• 9 = Total Disk Size</li> <li>• 10 = Number of Network Cards</li> <li>• 11 = Number of Display Adapters</li> <li>• 12 = IP Address</li> <li>• 13 = MAC Address</li> <li>• 14 = Host</li> <li>• 15 = Number of Cores</li> <li>• 16 = Number of Threads.</li> </ul>
<code>ColumnNameResourceName</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The unique name of the localizable resource string representing a compliance-tracked column. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>ColumnName</code>	<p><i>Type:</i> text (max 128 characters). Key</p> <p>The text to display if the column resource string has no translation.</p>
<code>IsColumnNumeric</code>	<p><i>Type:</i> boolean</p> <p>Indicates whether the column is numeric (<code>True</code>) or a string (<code>False</code>).</p>
<code>ComplianceAction</code>	<p><i>Type:</i> integer</p> <p>Bitwise value to indicate what type of action to track change on.</p>
<code>TrackComplianceBitwiseValue</code>	<p><i>Type:</i> integer</p> <p>Bitwise value indicating which asset types compliance tracking is turned on for.</p>



## AssetComplianceStatus Table

`AssetComplianceStatus` is a static table listing possible asset compliance states, such as compliant, new, changed, or ignored.

**Table 31: Database columns for `AssetComplianceStatus` table**

Database Column	Details
<code>AssetComplianceStatusID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>AssetComplianceStatus</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = New</li> <li>• 2 = Compliant</li> <li>• 3 = Changed</li> <li>• 4 = Ignore.</li> </ul>
<code>StatusResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an asset compliance status. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>StatusDefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the status resource string has no translation.</p>

## AssetContract Table

The `AssetContract` table links assets to related contracts.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 32: Database columns for `AssetContract` table**

Database Column	Details
<code>AssetContractID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for this record.</p>
<code>ContractID</code>	<p><i>Type:</i> integer. Key</p>

Database Column	Details
	The contract linked to the asset. Foreign key to the <code>Contract</code> table.
AssetID	Type: integer. Key The asset linked to the contract. Foreign key to the <code>Asset</code> table.

## AssetPropertyValue Table

For each asset, `AssetPropertyValue` stores the values for the custom properties defined in `AssetTypeProperty`.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 33: Database columns for AssetPropertyValue table**

Database Column	Details
AssetPropertyValueID	Type: integer. Key. Generated ID A unique identifier for this record.
AssetTypePropertyID	Type: integer. Key The property whose value is being stored. The type of the asset should match the type that the property is associated with. Foreign key to the <code>AssetTypeProperty</code> table.
AssetID	Type: integer. Key The asset associated with the property value. Foreign key to the <code>Asset</code> table.
PropertyValue	Type: text (max 4000 characters) The value of the property for the specified <code>Asset</code> .
CreationUser	Type: text (max 128 characters). Nullable The operator who created the record.
CreationDate	Type: datetime The date the record was created.
UpdatedUser	Type: text (max 128 characters). Nullable

Database Column	Details
	The operator who last updated the record.
UpdatedDate	Type: datetime. Nullable The date the record was last updated.

## AssetPurchaseOrder Table

The `AssetPurchaseOrder` table links assets to related purchase order lines.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 34: Database columns for AssetPurchaseOrder table**

Database Column	Details
AssetID	Type: integer. Key The asset linked to a purchase order. Foreign key to the <code>Asset</code> table.
PurchaseOrderDetailID	Type: integer. Key The purchase order line linked to an asset. Foreign key to the <code>PurchaseOrderDetail</code> table.

## AssetStatus Table

`AssetStatus` is a static table storing a list of possible asset states, such as purchased, in storage, installed, retired, disposed and other.

**Table 35: Database columns for AssetStatus table**

Database Column	Details
AssetStatusID	Type: integer. Key. Generated ID A unique identifier for each <code>AssetStatus</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>1 = Purchased</li> <li>2 = In Storage</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 3 = Installed</li> <li>• 4 = Retired</li> <li>• 5 = Disposed</li> <li>• 6 = Other.</li> </ul>
StatusResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an asset status. Foreign key to the <code>ComplianceResourceString</code> table.</p>
StatusDefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the status resource string has no translation.</p>

## AssetType Table

`AssetType` stores the collection of the types of assets that can be created in FlexNet Manager Suite.

**Table 36: Database columns for `AssetType` table**

Database Column	Details
AssetTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>AssetType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Workstation</li> <li>• 2 = Server</li> <li>• 3 = Monitor</li> <li>• 4 = Desk</li> <li>• 5 = Chair</li> <li>• 6 = Printer</li> <li>• 7 = Router</li> <li>• 8 = Switch</li> <li>• 9 = Telephone</li> <li>• 10 = Cell phone</li> <li>• 11 = Laptop.</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>12 = Mobile Device.</li> </ul>
AssetTypeResourceName	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The unique name of the localizable resource string representing a document type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
AssetTypeName	<p><i>Type:</i> text (max 64 characters). Key</p> <p>The text to display if the type resource string has no translation.</p>
XMLFile	<p><i>Type:</i> text. Nullable</p> <p>The layout of the property dialog for this type of asset, stored in XML format.</p>
ParentAssetTypeID	<p><i>Type:</i> integer. Nullable</p> <p>An asset type which is a parent of this asset type. Foreign key to the same <code>AssetType</code> table.</p>
ManagedType	<p><i>Type:</i> boolean. Key</p> <p>Set this field to <code>True</code> if this type of asset is directly managed by FlexNet Manager Suite (for example, laptops, servers and workstations).</p>
BitwiseValue	<p><i>Type:</i> integer</p> <p>The bitwise value of the asset type. This value is used when tracking compliance changes for assets linked to computers.</p>

## AssetTypeProperty Table

`AssetTypeProperty` defines extra custom properties for all assets.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 37: Database columns for `AssetTypeProperty` table**

Database Column	Details
AssetTypePropertyID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for a property of an asset type.</p>

Database Column	Details
PropertyName	<i>Type:</i> text (max 256 characters). Key The name of the property.
AssetTypeID	<i>Type:</i> integer. Key. Nullable Asset type with which this property is associated. Foreign key to the <code>AssetType</code> table.
HardwareClassName	<i>Type:</i> text (max 256 characters). Nullable The WMI class name associated with this property. This field applies for hardware properties that are mapped to hardware inventory tables.
HardwarePropertyName	<i>Type:</i> text (max 256 characters) The WMI property name associated with this property. This field applies for hardware properties that are mapped to hardware inventory tables.
CustomPropertyDisplayXMLID	<i>Type:</i> integer. Nullable Foreign key to a record in the <code>CustomPropertyDisplayXML</code> table, describing how to show the property on a property dialog.

## AssetWarrantyType Table

`AssetWarrantyType` is a static table listing all the types of warranties.

**Table 38: Database columns for `AssetWarrantyType` table**

Database Column	Details
AssetWarrantyTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>AssetWarrantyType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = None</li> <li>• 2 = One year on site</li> <li>• 3 = Three years on site.</li> </ul>
WarrantyTypeResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing an asset warranty type. Foreign key to the <code>ComplianceResourceString</code> table.
WarrantyTypeDefaultValue	<i>Type:</i> text (max 100 characters)

Database Column	Details
	The text to display if the type resource string has no translation.

## DepreciationMethod Table

`DepreciationMethod` is a static table storing the collection of available depreciation methods.

**Table 39: Database columns for `DepreciationMethod` table**

Database Column	Details
<code>DepreciationMethodID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>DepreciationMethod</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>1 = Straight line</li> <li>2 = Residual value.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 50 characters). Key</p> <p>The unique name of the localizable resource string representing a depreciation method. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the method resource string has no translation.</p>

## EndOfLifeReason Table

`EndOfLifeReason` is a static table storing the collection of all reasons for disposing of an asset.

**Table 40: Database columns for `EndOfLifeReason` table**

Database Column	Details
<code>EndOfLifeReasonID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>EndOfLifeReason</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>1 = [empty string]</li> <li>2 = Lost</li> <li>3 = Stolen</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 4 = Disposed</li> <li>• 5 = Sold</li> <li>• 6 = Donated</li> <li>• 7 = Broken.</li> </ul>
ResourceName	<i>Type:</i> text (max 50 characters). Key The unique name of the localizable resource string representing an end-of-life reason. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the reason resource string has no translation.

## LeaseEndReason Table

`LeaseEndReason` is a static table listing all the reasons that a company terminates a lease.

**Table 41: Database columns for `LeaseEndReason` table**

Database Column	Details
LeaseEndReasonID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>LeaseEndReason</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = [empty string]</li> <li>• 2 = Lease Ended - Asset Returned</li> <li>• 3 = Early Termination - Asset Returned</li> <li>• 4 = Buyout</li> <li>• 5 = Early Buyout</li> <li>• 6 = Trade.</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a lease-end reason. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the reason resource string has no translation.



# Compliance.Logic.Beacon Tables

The complete set of database tables documented here includes:

- ActiveDirectoryComputer table (see *ActiveDirectoryComputer Table* on page 51)
- ActiveDirectoryDomain table (see *ActiveDirectoryDomain Table* on page 51)
- ActiveDirectoryGroup table (see *ActiveDirectoryGroup Table* on page 52)
- ActiveDirectoryMember table (see *ActiveDirectoryMember Table* on page 53)
- ActiveDirectoryUser table (see *ActiveDirectoryUser Table* on page 53)
- AdministrationAccount table (see *AdministrationAccount Table* on page 54)
- AppVPackageMapping table (see *AppVPackageMapping Table* on page 54)
- AvailablePackage table (see *AvailablePackage Table* on page 55)
- AvailablePackageType table (see *AvailablePackageType Table* on page 56)
- BaselineImport table (see *BaselineImport Table* on page 56)
- Beacon table (see *Beacon Table* on page 57)
- BeaconActivityStatus table (see *BeaconActivityStatus Table* on page 59)
- BeaconAdministrationAccount table (see *BeaconAdministrationAccount Table* on page 60)
- BeaconAgentEvent table (see *BeaconAgentEvent Table* on page 60)
- BeaconDiscoveryStatus table (see *BeaconDiscoveryStatus Table* on page 61)
- BeaconDiscoveryTaskSummaryStatus table (see *BeaconDiscoveryTaskSummaryStatus Table* on page 62)
- BeaconDownloadedPolicy table (see *BeaconDownloadedPolicy Table* on page 62)
- BeaconExecutionStatusType table (see *BeaconExecutionStatusType Table* on page 63)
- BeaconFilter table (see *BeaconFilter Table* on page 64)
- BeaconIssueStatus table (see *BeaconIssueStatus Table* on page 64)
- BeaconIssueStatusType table (see *BeaconIssueStatusType Table* on page 65)
- BeaconPolicy table (see *BeaconPolicy Table* on page 66)
- BeaconPolicyPropertyValue table (see *BeaconPolicyPropertyValue Table* on page 67)
- BeaconPropertyValue table (see *BeaconPropertyValue Table* on page 68)
- BeaconRule table (see *BeaconRule Table* on page 68)
- BeaconRuleAction table (see *BeaconRuleAction Table* on page 69)
- BeaconRuleActionPropertyValue table (see *BeaconRuleActionPropertyValue Table* on page 70)
- BeaconRuleBeaconTargetMapping table (see *BeaconRuleBeaconTargetMapping Table* on page 71)

- BeaconSiteSubnetMapping table (see *BeaconSiteSubnetMapping Table* on page 71)
- BeaconTarget table (see *BeaconTarget Table* on page 72)
- BeaconTargetAgentEvent table (see *BeaconTargetAgentEvent Table* on page 73)
- BeaconTargetDiscoveredDeviceMapping table (see *BeaconTargetDiscoveredDeviceMapping Table* on page 73)
- BeaconTargetPropertyValue table (see *BeaconTargetPropertyValue Table* on page 74)
- BeaconTargetSiteMapping table (see *BeaconTargetSiteMapping Table* on page 74)
- BeaconTargetSiteSubnetMapping table (see *BeaconTargetSiteSubnetMapping Table* on page 75)
- BeaconUpgradeMode table (see *BeaconUpgradeMode Table* on page 75)
- BeaconUpgradeStatus table (see *BeaconUpgradeStatus Table* on page 76)
- BeaconWebServerStatus table (see *BeaconWebServerStatus Table* on page 76)
- DiscoveredDeviceDiscoveredBy table (see *DiscoveredDeviceDiscoveredBy Table* on page 77)
- DiscoveredDeviceDiscoveryStatus table (see *DiscoveredDeviceDiscoveryStatus Table* on page 78)
- DiscoveredDeviceInventoryStatus table (see *DiscoveredDeviceInventoryStatus Table* on page 78)
- DiscoveredDeviceTaskDetailedError table (see *DiscoveredDeviceTaskDetailedError Table* on page 79)
- DiscoveredDeviceTaskStatus table (see *DiscoveredDeviceTaskStatus Table* on page 80)
- DiscoveredDeviceTaskStatusHistory table (see *DiscoveredDeviceTaskStatusHistory Table* on page 81)
- DiscoveredDeviceTaskType table (see *DiscoveredDeviceTaskType Table* on page 83)
- ErrorCategory table (see *ErrorCategory Table* on page 83)
- FNMEAAgent table (see *FNMEAAgent Table* on page 83)
- IncomingBaseline table (see *IncomingBaseline Table* on page 84)
- ReconcileSoftwareLicenseReconcileExemptionReason table (see *ReconcileSoftwareLicenseReconcileExemptionReason Table* on page 85)
- RuleDiscoveryActionSummary table (see *RuleDiscoveryActionSummary Table* on page 86)
- RuleInventoryActionSummary table (see *RuleInventoryActionSummary Table* on page 86)
- SoftwareLicenseReconcileExemptionReasonData table (see *SoftwareLicenseReconcileExemptionReasonData Table* on page 87)
- StatusCodeCategory table (see *StatusCodeCategory Table* on page 88)
- UIAlignmentType table (see *UIAlignmentType Table* on page 88)
- UIFieldType table (see *UIFieldType Table* on page 89)
- UIInsertType table (see *UIInsertType Table* on page 89)
- UIItem table (see *UIItem Table* on page 90)

- UllItemTargetSubType table (see *UllItemTargetSubType Table* on page 91)

## ActiveDirectoryComputer Table

The `ActiveDirectoryComputer` table stores the active directory data for computers.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 42: Database columns for ActiveDirectoryComputer table**

Database Column	Details
<code>ActiveDirectoryComputerID</code>	Type: integer. Key. Generated ID Auto-generated Active Directory computer ID
<code>GUID</code>	Type: unique identifier. Key The GUID of the computer.
<code>ComputerName</code>	Type: text (max 64 characters) The computer name.
<code>ActiveDirectoryDomainID</code>	Type: integer. Key Foreign key to the <code>ActiveDirectoryDomain</code> table
<code>SID</code>	Type: text (max 256 characters). Key. Nullable The SID of the computer.

## ActiveDirectoryDomain Table

The `ActiveDirectoryDomain` table stores the active directory domains.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 43: Database columns for ActiveDirectoryDomain table**

Database Column	Details
ActiveDirectoryDomainID	Type: integer. Key. Generated ID Auto-generated Active Directory Domain ID
QualifiedName	Type: text (max 100 characters). Key The fully qualified domain name
FlatName	Type: text (max 32 characters) The domain flat name

## ActiveDirectoryGroup Table

The ActiveDirectoryGroup table stores the active directory data.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 44: Database columns for ActiveDirectoryGroup table**

Database Column	Details
ActiveDirectoryGroupID	Type: integer. Key. Generated ID Auto-generated Active Directory Group ID
GUID	Type: unique identifier. Key The GUID of the AD group.
SID	Type: text (max 256 characters). Key. Nullable The SID of the AD group.
Name	Type: text (max 128 characters). Nullable The AD group name
ActiveDirectoryDomainID	Type: integer. Key Foreign key to the ActiveDirectoryDomain table

## ActiveDirectoryMember Table

The `ActiveDirectoryMember` table stores the active directory data for AD member objects.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 45: Database columns for ActiveDirectoryMember table**

Database Column	Details
GUID	Type: unique identifier. Key The GUID of the member object.
ParentGroupGUID	Type: unique identifier. Key The parent AD group GUID.

## ActiveDirectoryUser Table

The `ActiveDirectoryUser` table stores the active directory data for users.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 46: Database columns for ActiveDirectoryUser table**

Database Column	Details
ActiveDirectoryUserID	Type: integer. Key. Generated ID Auto-generated Active Directory user ID
GUID	Type: unique identifier. Key The GUID of the user.
SAMAccountName	Type: text (max 20 characters). Key The user name.
ActiveDirectoryDomainID	Type: integer. Key

Database Column	Details
	Foreign key to the <code>ActiveDirectoryDomain</code> table
<code>Sid</code>	<i>Type:</i> text (max 256 characters). Key. Nullable The SID of the user.

## AdministrationAccount Table

Records the complete set of administration accounts configured on inventory beacons.

**Table 47: Database columns for AdministrationAccount table**

Database Column	Details
<code>AccountID</code>	<i>Type:</i> integer. Key. Generated ID Unique id for the account.
<code>AccountName</code>	<i>Type:</i> text (max 256 characters). Key The logical name of the account.

## AppVPackageMapping Table

The `AppVPackageMapping` table is a table that maps App-V 4.6 packages to installer evidence.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 48: Database columns for AppVPackageMapping table**

Database Column	Details
<code>AppVPackageMappingID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated App-V 4.6 package mapping ID.
<code>PackageName</code>	<i>Type:</i> text (max 256 characters). Key The App-V 4.6 package name.
<code>PackageVersion</code>	<i>Type:</i> text (max 128 characters). Key

Database Column	Details
	The App-V 4.6 package version.
DisplayName	Type: text (max 256 characters) The display name of the software as reported by the installer evidence.
Version	Type: text (max 72 characters) The version of the software as reported by the installer evidence.
Publisher	Type: text (max 200 characters) The publisher of the software as reported by the installer evidence.

## AvailablePackage Table

Packages which are available to beacons.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 49: Database columns for AvailablePackage table**

Database Column	Details
AvailablePackageID	Type: integer. Key. Generated ID The ID of the available package.
FullName	Type: text (max 256 characters). Key The full path of the package within the repository.
Version	Type: text (max 32 characters). Key The version of the package.
AvailablePackageTypeID	Type: integer. Key The type of the package. Foreign key to the AvailablePackageType table.
RelativeURLToOSD	Type: text (max 256 characters) The relative URL to the OSD of the package for use in inventory agent policy.
UseInAgentPolicy	Type: boolean

Database Column	Details
	Whether the package should be added to policy for inventory agents.
Build	<i>Type:</i> text (max 8 characters). Key The build number of the package, necessary for choosing between patched versions of the same release.
WebUIRelativeURL	<i>Type:</i> text (max 256 characters). Nullable The relative URL to download the package from WebUI

## AvailablePackageType Table

**Table 50: Database columns for AvailablePackageType table**

Database Column	Details
AvailablePackageTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each AvailablePackageType. Possible values are: <ul style="list-style-type: none"> <li>• 1 = Adoption</li> <li>• 2 = Upgrade</li> <li>• 3 = Inventory agent plugin</li> <li>• 4 = Software</li> <li>• 5 = Other</li> <li>• 6 = Inventory beacon upgrade</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a purchase order line item type. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.

## BaselineImport Table





**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 51: Database columns for BaselineImport table**

Database Column	Details
BaselineImportID	Type: integer. Key. Generated ID The baseline import ID
Type	Type: text (max 16 characters) The baseline type
Date	Type: datetime The date of the baseline import
PurchaseOrderID	Type: integer. Key. Nullable The purchase order for the baseline import
ComplianceOperatorID	Type: integer. Key The compliance operator who performed the baseline import

## Beacon Table

The Beacon table contains beacon definition.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 52: Database columns for Beacon table**

Database Column	Details
BeaconID	Type: integer. Key. Generated ID Unique ID assigned to each beacon.
BeaconUID	Type: unique identifier. Key Unique string ID of the beacon.

Database Column	Details
BeaconName	<i>Type:</i> text (max 64 characters) Name of the beacon.
BeaconDescription	<i>Type:</i> text (max 256 characters). Nullable Description of the beacon.
BeaconStatus	<i>Type:</i> boolean Boolean indicating to include or exclude Beacon.
LastKnownActivityTime	<i>Type:</i> datetime. Nullable Last known time that communication has been received from the beacon.
ActivityStatusID	<i>Type:</i> integer Last known activity status reported by the beacon.
PolicyDownloadedTime	<i>Type:</i> datetime. Nullable Policy downloaded time
CurrentPolicyRevisionNo	<i>Type:</i> integer. Nullable Last downloaded policy revision number
BeaconLocation	<i>Type:</i> text (max 256 characters). Nullable Location field for Beacon.
PrimaryParentUID	<i>Type:</i> unique identifier. Nullable The parent of the Beacon. For the core Beacon, the PrimaryParentUID is NULL.
BeaconPassword	<i>Type:</i> text (max 64 characters). Nullable The password used by the beacon to authenticate with.
HTTPAccessData	<i>Type:</i> text. Nullable The HTTPEndPointStatus object, used for storing a summary of how to access the sahres on this beacon.
UpgradeModeID	<i>Type:</i> integer The upgrade mode selected for this beacon.
UpgradeStatusID	<i>Type:</i> integer The latest information reported by a beacon about any upgrade activity or changes.

Database Column	Details
LastKnownPolicy	<i>Type:</i> datetime. Nullable The last known time that the beacon has communicated with the server.
Version	<i>Type:</i> text (max 50 characters). Nullable Version of installed beacon on the server
WebServerStatusID	<i>Type:</i> integer The last known time that the beacon has communicated with the server.
UpgradeStatusTime	<i>Type:</i> datetime. Nullable The time the last upgrade status was reported.
AvailablePackageID	<i>Type:</i> integer. Key. Nullable If the beacon upgrade mode is set to specific version, then this stored the specific package to upgrade to.
ParentServerURL	<i>Type:</i> text. Nullable The parent to which this beacon will communicate with.
DownloadURL	<i>Type:</i> text. Nullable The download URL of the parent.
UploadURL	<i>Type:</i> text. Nullable The upload URL of the parent.

## BeaconActivityStatus Table

BeaconActivityStatus is a static table listing all of the states of a beacon.

**Table 53: Database columns for BeaconActivityStatus table**

Database Column	Details
BeaconActivityStatusID	<i>Type:</i> integer. Key. Generated ID
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing the BeaconActivityStatus record. Foreign key to the ComplianceResourceString table.

Database Column	Details
DefaultValue	Type: text (max 256 characters) The text to display if the state resource string has no translation.

## BeaconAdministrationAccount Table

Records an administration account discovered on an inventory beacon.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 54: Database columns for BeaconAdministrationAccount table**

Database Column	Details
AccountID	Type: integer. Key Unique id for the account.
ServerUID	Type: unique identifier. Key Identifies the distribution server which discovered the account.

## BeaconAgentEvent Table

The `BeaconAgentEvent` table contains a list of events that can be included in agent schedules.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 55: Database columns for BeaconAgentEvent table**

Database Column	Details
BeaconAgentEventID	Type: integer. Key. Generated ID Unique ID assigned to each beacon agent event.

Database Column	Details
EventName	Type: text (max 256 characters). Key Event name.
EventUID	Type: unique identifier. Key Event uid.
Value	Type: text An XML representation of the agent event data.

## BeaconDiscoveryStatus Table

Discovery and remote execution status of Beacon



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 56: Database columns for BeaconDiscoveryStatus table**

Database Column	Details
BeaconDiscoveryStatusID	Type: integer. Key. Generated ID The ID of the beacon discovery status.
ServerUID	Type: unique identifier. Key The inventory beacon that has run the task.
State	Type: text (max 256 characters) State of the discovery/execution - Running/Finished.
StartDateTime	Type: datetime Execution start time.
Duration	Type: integer Duration in Seconds of the discovery execution.
DiscoveredCount	Type: integer Total number of devices discovered.

Database Column	Details
ExecutionSuccess	Type: integer Total number successful remote executions.
ExecutionFailure	Type: integer Total number failed remote executions.

## BeaconDiscoveryTaskSummaryStatus Table

Task summary list for a particular beacon



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 57: Database columns for BeaconDiscoveryTaskSummaryStatus table**

Database Column	Details
TaskSummaryStatusID	Type: integer. Key. Generated ID The ID of the device status.
BeaconDiscoveryStatusID	Type: integer. Key The beacon discovery status table which this refers to.
TaskTypeID	Type: integer The type of task which was run.
SuccessCount	Type: integer Success count in this particular execution.
FailureCount	Type: integer Failure count in this particular execution.

## BeaconDownloadedPolicy Table

The `BeaconDownloadedPolicy` table contains policies downloaded by inventory beacons.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 58: Database columns for BeaconDownloadedPolicy table**

Database Column	Details
BeaconDownloadedPolicyID	<i>Type:</i> integer. Key. Generated ID The ID of the downloaded beacon policy.
RevisionNumber	<i>Type:</i> integer. Key The revision number of this policy.
PolicyXML	<i>Type:</i> XML The beacon policy xml downloaded by inventory beacons.

## BeaconExecutionStatusType Table

BeaconExecutionStatusType is a static table listing possible beacon status values.

**Table 59: Database columns for BeaconExecutionStatusType table**

Database Column	Details
BeaconExecutionStatusTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each BeaconExecutionStatusType. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Unknown</li> <li>• 2 = Started</li> <li>• 3 = Not configured</li> <li>• 4 = Running</li> <li>• 5 = Finished</li> <li>• 6 = Stopped</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a batch process type. Foreign key to the ComplianceResourceString table.

Database Column	Details
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.

## BeaconFilter Table

The `BeaconFilter` table contains target filters.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 60: Database columns for BeaconFilter table**

Database Column	Details
BeaconFilterID	<i>Type:</i> integer. Key. Generated ID Unique ID automatically assigned to each beacon target filters.
BeaconTargetID	<i>Type:</i> integer. Key Target this filter refers to.
Include	<i>Type:</i> boolean Boolean string indicating to include or exclude filter value.
IsLinked	<i>Type:</i> boolean Boolean indicating if the filter is linked to site/subnet/device or an independent filter.
Value	<i>Type:</i> text (max 256 characters) Filter value.
FilterType	<i>Type:</i> text (max 64 characters) Filter type set for this filter.

## BeaconIssueStatus Table

Records beacon issue detail information.





**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 61: Database columns for BeaconIssueStatus table**

Database Column	Details
BeaconIssueStatusID	Type: integer. Key. Generated ID Unique id for the BeaconIssueStatus.
BeaconID	Type: integer. Key Beacon that this issue status relates to
BeaconIssueStatusTypeID	Type: integer. Key Issue type
IsActive	Type: boolean Policy downloaded time
IssueDetail	Type: text. Nullable Detail information about the issue

## BeaconIssueStatusType Table

BeaconIssueStatusType is a static table listing possible beacon alerts.

**Table 62: Database columns for BeaconIssueStatusType table**

Database Column	Details
BeaconIssueStatusTypeID	Type: integer. Key. Generated ID A unique identifier for each BeaconIssueStatusType. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 0 = Unknown</li> <li>• 1 = Policy load</li> <li>• 2 = Policy download</li> <li>• 3 = Discovery execution</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 4 = Action execution</li> <li>• 5 = Self hosted web server</li> <li>• 6 = Service exit</li> <li>• 7 = Package download</li> <li>• 8 = Active Directory import</li> <li>• 9 = SAP Inventory import</li> <li>• 10 = SAP recommendation set download</li> <li>• 11 = Beacon self upgrade</li> <li>• 12 = Beacon Parent Configuration</li> </ul>
ResourceName	<p>Type: text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a batch process type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p>Type: text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## BeaconPolicy Table

The `BeaconPolicy` table contains the beacon policy.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 63: Database columns for BeaconPolicy table**

Database Column	Details
BeaconPolicyID	<p>Type: integer. Key. Generated ID</p> <p>The ID of the beacon policy.</p>
RevisionNumber	<p>Type: integer</p> <p>The revision number of this policy.</p>
AgentScheduleData	<p>Type: text. Nullable</p>

Database Column	Details
	The Schedule object, used for storing the global schedule for managed devices.
CreationDate	<i>Type:</i> datetime Date and time when the policy was created.
LastChangedOn	<i>Type:</i> datetime. Nullable Date and time that the policy was last modified.
ApprovedBeaconPackageID	<i>Type:</i> integer. Key. Nullable The beacon upgrade package that has been approved by the customer. NULL indicates to stay always on the latest.
LastDiscoveryFullExportTime	<i>Type:</i> datetime. Nullable The last time a discovery export was generated.
LastDiscoveryFullExportVersion	<i>Type:</i> integer. Nullable The revision number of the last full discovery export.
LastTargetRefreshTime	<i>Type:</i> datetime. Nullable The last time special internal targets were recalculated and refreshed.

## BeaconPolicyPropertyValue Table

The BeaconPolicyPropertyValue table contains beacon policy property value elements.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 64: Database columns for BeaconPolicyPropertyValue table**

Database Column	Details
BeaconPolicyPropertyID	<i>Type:</i> integer. Key. Generated ID Unique ID assigned to each beacon policy property.
KeyName	<i>Type:</i> text (max 256 characters). Key Property Key.

Database Column	Details
Value	Type: text (max 256 characters) Property Value.

## BeaconPropertyValue Table

The `BeaconPropertyValue` table contains beacon property value elements.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 65: Database columns for BeaconPropertyValue table**

Database Column	Details
BeaconPropertyID	Type: integer. Key. Generated ID Unique ID assigned to each beacon property.
BeaconID	Type: integer. Key Beacon this property refers to.
KeyName	Type: text (max 256 characters). Key Property Key.
Value	Type: text (max 256 characters) Property Value.

## BeaconRule Table

The `BeaconRule` table contains the details of beacon rules.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 66: Database columns for BeaconRule table**

Database Column	Details
BeaconRuleID	<i>Type:</i> integer. Key. Generated ID The ID of the beacon rule.
BeaconRuleActionID	<i>Type:</i> integer. Key The reference of Action from the beacon rule.
RuleName	<i>Type:</i> text (max 128 characters) The name of the rule.
RulePriority	<i>Type:</i> small integer Beacon rules are prioritised according to the rule priority. Higher priority takes precedence over lower priorities.
MaximumAge	<i>Type:</i> integer. Nullable Maximum age of the rule before it is re-scheduled.
ExternalID	<i>Type:</i> unique identifier. Key The ID that exists externally.
BeaconScheduleData	<i>Type:</i> text The Schedule object.
Include	<i>Type:</i> boolean Boolean string indicating to include or exclude rule.
Internal	<i>Type:</i> boolean Is this rule used internally, or managed by the user.
NameResourceName	<i>Type:</i> text (max 256 characters). Nullable Resource for translation of Name column. Foreign key to <code>ComplianceResourceString</code> table.

## BeaconRuleAction Table

The `BeaconRuleAction` table contains beacon rule action.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 67: Database columns for BeaconRuleAction table**

Database Column	Details
BeaconRuleActionID	Type: integer. Key. Generated ID Unique ID automatically assigned to each beacon actions.
Name	Type: text (max 100 characters). Key Name of Action.
Description	Type: text (max 256 characters). Nullable Description of Action.
NameResourceName	Type: text (max 256 characters). Nullable Resource for translation of Name column. Foreign key to the ComplianceResourceString table.
DescriptionResourceName	Type: text (max 256 characters). Nullable Resource for translation of Description column. Foreign key to the ComplianceResourceString table.
Internal	Type: boolean Is this action used internally, or managed by the user.

## BeaconRuleActionPropertyValue Table

The BeaconRuleActionPropertyValue table contains beacon action property value elements.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 68: Database columns for BeaconRuleActionPropertyValue table**

Database Column	Details
BeaconRuleAction PropertyID	Type: integer. Key. Generated ID Unique ID assigned to each beacon action property.
BeaconRuleActionID	Type: integer. Key Beacon action this property refers to.
KeyName	Type: text (max 256 characters). Key Property Key.
Value	Type: text Property Value.

## BeaconRuleBeaconTargetMapping Table

Table that maps targets to rule.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 69: Database columns for BeaconRuleBeaconTargetMapping table**

Database Column	Details
BeaconRuleID	Type: integer. Key Foreign key to the BeaconRule table.
BeaconTargetID	Type: integer. Key Foreign key to the BeaconTarget table.

## BeaconSiteSubnetMapping Table

Table that maps site to Beacons.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 70: Database columns for BeaconSiteSubnetMapping table**

Database Column	Details
BeaconID	Type: integer. Key Foreign key to the Beacon table.
SubnetID	Type: integer. Key Foreign key to the SiteSubnet table.

## BeaconTarget Table

The BeaconTarget table contains beacon rule targets.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 71: Database columns for BeaconTarget table**

Database Column	Details
BeaconTargetID	Type: integer. Key. Generated ID Unique ID automatically assigned to each beacon targets.
Name	Type: text (max 100 characters). Key Name identifying the target.
Description	Type: text (max 256 characters). Nullable Name identifying the target.
Internal	Type: boolean. Key Is this target used internally, or managed by the user.
Visible	Type: boolean



Database Column	Details
	Can this target be displayed to the user for selection etc. This does not apply to the actual Targets page.

## BeaconTargetAgentEvent Table

Table that maps agent events to targets.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 72: Database columns for BeaconTargetAgentEvent table**

Database Column	Details
BeaconTargetID	Type: integer. Key Foreign key to the BeaconTarget; table.
BeaconAgentEventUID	Type: unique identifier. Key Foreign key to the BeaconAgentEvent table.

## BeaconTargetDiscoveredDeviceMapping Table

Table that maps site to targets.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 73: Database columns for BeaconTargetDiscoveredDeviceMapping table**

Database Column	Details
BeaconTargetID	Type: integer. Key Foreign key to the BeaconTarget table.
DeviceID	Type: integer. Key

Database Column	Details
	Foreign key to the <code>DiscoveredDevice</code> table.
Include	Type: boolean Boolean string indicating to include or exclude Device.

## BeaconTargetPropertyValue Table

The `BeaconTargetPropertyValue` table contains beacon target property value elements.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 74: Database columns for `BeaconTargetPropertyValue` table**

Database Column	Details
<code>BeaconTargetPropertyID</code>	Type: integer. Key. Generated ID Unique ID assigned to each beacon target property.
<code>BeaconTargetID</code>	Type: integer. Key Beacon target this property refers to.
<code>KeyName</code>	Type: text (max 256 characters). Key Property Key.
<code>Value</code>	Type: text (max 256 characters) Property Value.

## BeaconTargetSiteMapping Table

Table that maps site to targets.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 75: Database columns for BeaconTargetSiteMapping table**

Database Column	Details
BeaconTargetID	Type: integer. Key Foreign key to the BeaconTarget table.
SiteID	Type: integer. Key Foreign key to the Site table.
Include	Type: boolean Boolean string indicating to include or exclude Device.

## BeaconTargetSiteSubnetMapping Table

Table that maps site to targets.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 76: Database columns for BeaconTargetSiteSubnetMapping table**

Database Column	Details
BeaconTargetID	Type: integer. Key Foreign key to the BeaconTarget table.
SubnetID	Type: integer. Key Foreign key to the SiteSubnet table.
Include	Type: boolean Boolean string indicating to include or exclude Device.

## BeaconUpgradeMode Table

BeaconUpgradeMode is a static table listing all of the styles of upgrade that a beacon can follow.

**Table 77: Database columns for BeaconUpgradeMode table**

Database Column	Details
BeaconUpgradeModeID	<i>Type:</i> integer. Key. Generated ID
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing the BeaconUpgradeMode record. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 256 characters) The text to display if the state resource string has no translation.

## BeaconUpgradeStatus Table

BeaconUpgradeStatus is a static table listing all of the upgrade states that a beacon can be in.

**Table 78: Database columns for BeaconUpgradeStatus table**

Database Column	Details
BeaconUpgradeStatusID	<i>Type:</i> integer. Key. Generated ID
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing the BeaconUpgradeStatus record. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 256 characters) The text to display if the state resource string has no translation.

## BeaconWebServerStatus Table

BeaconWebServerStatus is a static table listing all of the states of the beacons web server.

**Table 79: Database columns for BeaconWebServerStatus table**

Database Column	Details
BeaconWebServerStatusID	<i>Type:</i> integer. Key. Generated ID
ResourceName	<i>Type:</i> text (max 256 characters). Key

Database Column	Details
	The unique name of the localizable resource string representing the BeaconWebServerStatus record. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 256 characters) The text to display if the state resource string has no translation.

## DiscoveredDeviceDiscoveredBy Table

By which inventory beacon was this device discovered? Sometimes useful when other identifying features are duplicated, and when the distribution server should do something to the device.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 80: Database columns for DiscoveredDeviceDiscoveredBy table**

Database Column	Details
DeviceID	<i>Type:</i> integer. Key The id of the device discovered.
ServerUID	<i>Type:</i> unique identifier. Key The inventory beacon that discovered it.
RuleID	<i>Type:</i> integer. Key. Nullable The RuleID executed on the beacon that discovered the device.
CanAdminister	<i>Type:</i> boolean. Nullable Does the distribution server have administrative privileges for the device?
LastUpdate	<i>Type:</i> datetime The date and time that the distribution server last reported its discovery of this device.
AccountID	<i>Type:</i> integer. Key. Nullable Account that can administer the device.

Database Column	Details
AccountIDOverride	Type: integer. Key. Nullable Account that can administer the device, overridden by the user.

## DiscoveredDeviceDiscoveryStatus Table



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 81: Database columns for DiscoveredDeviceDiscoveryStatus table**

Database Column	Details
DeviceID	Type: integer. Key
TaskTypeID	Type: integer. Key
BeaconRuleID	Type: integer. Key Rule that executed this task.
BeaconPolicyRevision Number	Type: integer The beacon policy revision number where rule is found
SessionUID	Type: unique identifier. Nullable
DiscoveryDate	Type: datetime. Nullable
RuleDiscoveryAction SummaryID	Type: integer Rule discovery summary.
BeaconUID	Type: unique identifier. Key. Nullable The inventory beacon that ran the task.

## DiscoveredDeviceInventoryStatus Table



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 82: Database columns for DiscoveredDeviceInventoryStatus table**

Database Column	Details
DeviceID	Type: integer. Key
TaskTypeID	Type: integer. Key
BeaconRuleID	Type: integer. Key. Nullable Rule that executed this task.
BeaconPolicyRevision Number	Type: integer. Nullable The beacon policy revision number where rule is found
SessionUID	Type: unique identifier. Nullable
InventoryDate	Type: datetime. Nullable
RuleInventoryAction SummaryID	Type: integer Rule action summary.
BeaconUID	Type: unique identifier. Key. Nullable The inventory beacon that ran the task.

## DiscoveredDeviceTaskDetailedError Table



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 83: Database columns for DiscoveredDeviceTaskDetailedError table**

Database Column	Details
DiscoveredDeviceTask DetailedErrorID	Type: integer. Key. Generated ID The ID of the discovered device error.

Database Column	Details
DiscoveredDeviceTask StatusHistoryID	Type: integer. Key Discovered device task status.
Status	Type: text (max 256 characters). Key The status code of task.
DetailedStatus	Type: text. Nullable The detailed error status.

## DiscoveredDeviceTaskStatus Table

Records any task status information for `DiscoveredDevice`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 84: Database columns for `DiscoveredDeviceTaskStatus` table**

Database Column	Details
DiscoveredDeviceTask StatusID	Type: integer. Key. Generated ID The ID of the discovered device task.
DeviceID	Type: integer. Key Device identity number.
TaskTypeID	Type: integer. Key The type of task which was run on the device.
BeaconUID	Type: unique identifier. Key. Nullable The inventory beacon that has run the task.
BeaconRuleID	Type: integer. Key. Nullable Rule that executed this task.
Success	Type: boolean. Key Status of the task. It can be Success OR Failed



Database Column	Details
Credential	<i>Type:</i> text (max 256 characters). Nullable The credential name for the task performed.
Status	<i>Type:</i> text (max 256 characters) The status code of task.
DetailedStatus	<i>Type:</i> text. Nullable The detailed error status.
StartDateTime	<i>Type:</i> datetime Date and time the task was started.
BeaconPolicyRevision Number	<i>Type:</i> integer. Nullable The beacon policy revision number where rule is found
SessionUID	<i>Type:</i> unique identifier. Nullable An identifier <code>TaskExecutionStatus</code> table
IsSkipTask	<i>Type:</i> boolean Determines whether the task status is a skip task
IsDiscoveryTask	<i>Type:</i> boolean Determines whether the task status is a discovery task

## DiscoveredDeviceTaskStatusHistory Table

Records any task status information for `DiscoveredDevice`.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 85: Database columns for `DiscoveredDeviceTaskStatusHistory` table**

Database Column	Details
DiscoveredDeviceTask StatusHistoryID	<i>Type:</i> integer. Key. Generated ID The ID of the discovered device task.

Database Column	Details
DeviceID	<i>Type:</i> integer. Key Device identity number.
TaskTypeID	<i>Type:</i> integer. Key The type of task which was run on the device.
SessionUID	<i>Type:</i> unique identifier. Key An identifier <code>TaskExecutionStatus</code> table
BeaconUID	<i>Type:</i> unique identifier. Key The inventory beacon that has run the task.
BeaconRuleID	<i>Type:</i> integer. Key. Nullable Rule that executed this task.
Success	<i>Type:</i> boolean. Key Status of the task. It can be Success OR Failed
Credential	<i>Type:</i> text (max 256 characters). Nullable The credential name for the task performed.
Status	<i>Type:</i> text (max 256 characters) The status code of task.
DetailedStatus	<i>Type:</i> text. Nullable The detailed error status.
StartDateTime	<i>Type:</i> datetime Date and time the task was started.
BeaconPolicyRevision Number	<i>Type:</i> integer. Nullable The beacon policy revision number where rule is found
IsSkipTask	<i>Type:</i> boolean. Key Determines whether the task status is a skip task
IsDiscoveryTask	<i>Type:</i> boolean. Key Determines whether the task status is a discovery task

## DiscoveredDeviceTaskType Table

This table stores the information about different types of tasks executed on a discovered device and their associated IDs.

**Table 86: Database columns for DiscoveredDeviceTaskType table**

Database Column	Details
TaskTypeID	<i>Type:</i> integer. Key. Generated ID The id for the task.
TaskTypeName	<i>Type:</i> text (max 32 characters). Key The name of the task.

## ErrorCategory Table

Reported error category

**Table 87: Database columns for ErrorCategory table**

Database Column	Details
ErrorCategoryID	<i>Type:</i> integer. Key. Generated ID The ID of the error category.
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a error category name. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.

## FNMEAAgent Table

The `FNMEAAgent` table stores the FNM-EA connection defined in inventory beacons.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 88: Database columns for FNMEAAgent table**

Database Column	Details
FNMEAAgentID	Type: integer. Key. Generated ID Auto-generated FNMEA agent connection ID
BeaconID	Type: integer. Key. Nullable Beacon where the FNM-EA agent connection is defined
AgentIdentifier	Type: unique identifier. Key The GUID of the FNM-EA agent defined on inventory beacon.
AgentName	Type: text (max 128 characters) The FNM-EA agent name defined on inventory beacon.
LastReportedLogRotation	Type: datetime. Nullable Date time of the last report log rotation.
LastReportedAgentStatus	Type: datetime. Nullable Date time of the last reported status.

## IncomingBaseline Table



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 89: Database columns for IncomingBaseline table**

Database Column	Details
Type	Type: text (max 16 characters). Key The baseline type
Date	Type: datetime. Key The date of the baseline data
ProductPool	Type: text (max 128 characters). Key

Database Column	Details
	The license product pool
ProductFamily	Type: text (max 256 characters). Key The license product family
ProductVersion	Type: text (max 50 characters). Key The license product version
EffectiveQuantity	Type: integer The effective quantity of the license
UpgradeQuantity	Type: integer The upgrade quantity of the license
UpgradeWithMaintenanceQuantity	Type: integer The upgrade with maintenance quantity of the license
ActiveSAQuantity	Type: integer The active software assurance quantity of the license
ExpiringSA0To12Months	Type: integer The software assurance quantity expiring within 0-12 months
ExpiringSA12To24Months	Type: integer The software assurance quantity expiring within 12-24 months
ExpiringSA24PlusMonths	Type: integer The software assurance quantity expiring greater than 24 months

## ReconcileSoftwareLicenseReconcileExemptionReason Table

The `ReconcileSoftwareLicenseReconcileExemptionReason` table stores the staging license reconcile generated exemption reasons.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 90: Database columns for ReconcileSoftwareLicenseReconcileExemptionReason table**

Database Column	Details
SoftwareLicenseID	Type: integer. Key Foreign key to the SoftwareLicense table
ComplianceComputerID	Type: integer. Key. Nullable Foreign key to the ComplianceComputer table
ComplianceUserID	Type: integer. Key. Nullable Foreign key to the ComplianceUser table
SoftwareLicenseExemptionReasonID	Type: integer. Key Foreign key to the SoftwareLicenseExemptionReason table

## RuleDiscoveryActionSummary Table

Summary of the discovery action.

**Table 91: Database columns for RuleDiscoveryActionSummary table**

Database Column	Details
RuleDiscoveryActionSummaryID	Type: integer. Key. Generated ID The ID of the discovery action summary.
ResourceName	Type: text (max 256 characters). Key The unique name of the localizable resource string representing a discovery action summary. Foreign key to the ComplianceResourceString table.
DefaultValue	Type: text (max 100 characters) The text to display if the type resource string has no translation.

## RuleInventoryActionSummary Table

Summary of the inventory gathering action.

**Table 92: Database columns for RuleInventoryActionSummary table**

Database Column	Details
RuleInventoryActionSummaryID	Type: integer. Key. Generated ID The ID of the inventory gathering action summary.
ResourceName	Type: text (max 256 characters). Key The unique name of the localizable resource string representing a discovery action summary. Foreign key to the ComplianceResourceString table.
DefaultValue	Type: text (max 100 characters) The text to display if the type resource string has no translation.

## SoftwareLicenseReconcileExemptionReasonData Table

The SoftwareLicenseReconcileExemptionReasonData table stores the exemption reasons generated by the license reconcile.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 93: Database columns for SoftwareLicenseReconcileExemptionReasonData table**

Database Column	Details
SoftwareLicenseID	Type: integer. Key Foreign key to the SoftwareLicenseSnapshot table
ComplianceComputerID	Type: integer. Key. Nullable Foreign key to the ComplianceComputerSnapshot table
ComplianceUserID	Type: integer. Key. Nullable Foreign key to the ComplianceUserSnapshot table
SoftwareLicenseExemptionReasonID	Type: integer. Key Foreign key to the SoftwareLicenseExemptionReason table
LicenseMeasurementID	Type: integer. Key

Database Column	Details
	The snapshot ID. Foreign key to the <code>LicenseMeasurement</code> table.

## StatusCodeCategory Table

Reported error category

**Table 94: Database columns for StatusCodeCategory table**

Database Column	Details
<code>StatusCodeCategoryID</code>	<i>Type:</i> integer. Key. Generated ID The ID of the error category.
<code>StatusCode</code>	<i>Type:</i> text (max 256 characters). Key Status code.
<code>ErrorCategoryID</code>	<i>Type:</i> integer. Nullable An identifier <code>ErrorCategory</code> table

## UIAlignmentType Table

**Table 95: Database columns for UIAlignmentType table**

Database Column	Details
<code>UIAlignmentTypeID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>UIAlignmentType</code> . Possible values are: <ul style="list-style-type: none"> <li>• 1 = UseAvailableSpace</li> <li>• 2 = ForceLeft</li> <li>• 3 = ForceRight</li> </ul>
<code>ResourceName</code>	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a insert type. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DefaultValue</code>	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.



## UIFieldType Table

**Table 96: Database columns for UIFieldType table**

Database Column	Details
UIFieldTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each UIFieldType. Possible values are:</p> <ul style="list-style-type: none"> <li>• 1 = Tab</li> <li>• 2 = Section</li> <li>• 3 = Integer</li> <li>• 4 = Text box</li> <li>• 5 = Text area</li> <li>• 6 = Date</li> <li>• 7 = Date and time</li> <li>• 8 = Combo box</li> <li>• 9 = Check box</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a connection type. Foreign key to the ComplianceResourceString table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## UIInsertType Table

**Table 97: Database columns for UIInsertType table**

Database Column	Details
UIInsertTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each UIInsertType. Possible values are:</p> <ul style="list-style-type: none"> <li>• 1 = Before</li> <li>• 2 = After</li> <li>• 3 = Start of</li> </ul>

Database Column	Details
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a insert type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.

## UIItem Table



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 98: Database columns for UIItem table**

Database Column	Details
UIItemID	<i>Type:</i> integer. Key. Generated ID
TargetTypeID	<i>Type:</i> integer. Key Type of object. Foreign key to the <code>TargetType</code> table.
ItemResourceName	<i>Type:</i> text (max 256 characters). Key Name of the item
ItemName	<i>Type:</i> text (max 256 characters) Name of the item
UIFieldTypeID	<i>Type:</i> integer. Nullable UI field type if the elemet type is of type 'field'. Foreign key to the <code>UIFieldType</code> table.
UIInsertTypeID	<i>Type:</i> integer Insert type. Foreign key to <code>UIInsertType</code> table.
UIAlignmentTypeID	<i>Type:</i> integer. Nullable Alignment type. Foreign key to <code>UIAlignmentType</code> table.

Database Column	Details
TabName	<i>Type:</i> text (max 80 characters) Name of the object to place the UI item.
RelativePositionTo	<i>Type:</i> text (max 80 characters) Name of the object to place the UI item.
Position	<i>Type:</i> integer
Width	<i>Type:</i> integer
DataSource	<i>Type:</i> XML. Nullable Date source for item of element type 'field' and of field type combo box . .
SequenceNumber	<i>Type:</i> integer Sequence where items to be added into UI
FromTable	<i>Type:</i> text. Nullable The name of the database table where the field can be found.
SelectName	<i>Type:</i> text. Nullable The name of the field in the database.
WhereClause	<i>Type:</i> text. Nullable The SQL "WHERE" statement that limits the information returned.
Required	<i>Type:</i> boolean Is the field a mandatory field.
StringLength	<i>Type:</i> integer String length.
ReadOnly	<i>Type:</i> boolean Is the field a readonly field.

## UIItemTargetSubType Table



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 99: Database columns for UIItemTargetSubType table**

Database Column	Details
UIItemTargetSubTypeID	Type: integer. Key. Generated ID
UIItemID	Type: integer. Key Type of object. Foreign key to the UIItem table.
TargetSubTypeID	Type: integer. Key object subtype. Foreign key to the various object type tables.

## Compliance.Logic.Core Tables

The complete set of database tables documented here includes:

- Activity table (see *Activity Table* on page 97)
- ActivitySource table (see *ActivitySource Table* on page 98)
- ActivityTraceLog table (see *ActivityTraceLog Table* on page 99)
- ActivityType table (see *ActivityType Table* on page 99)
- Alert table (see *Alert Table* on page 100)
- AlertCategory table (see *AlertCategory Table* on page 101)
- AlertTarget table (see *AlertTarget Table* on page 101)
- AlertType table (see *AlertType Table* on page 102)
- AssetContractPaymentSchedule table (see *AssetContractPaymentSchedule Table* on page 102)
- Attribute table (see *Attribute Table* on page 103)
- BusinessImportLogDetail table (see *BusinessImportLogDetail Table* on page 104)
- BusinessImportLogObject table (see *BusinessImportLogObject Table* on page 105)
- BusinessImportLogSummary table (see *BusinessImportLogSummary Table* on page 106)
- BusinessImportResult table (see *BusinessImportResult Table* on page 107)
- ComplianceComputer table (see *ComplianceComputer Table* on page 108)
- ComplianceComputerConnection table (see *ComplianceComputerConnection Table* on page 114)

- ComplianceComputerContract table (see *ComplianceComputerContract Table* on page 114)
- ComplianceComputerInventorySourceType table (see *ComplianceComputerInventorySourceType Table* on page 115)
- ComplianceComputerPropertyValue table (see *ComplianceComputerPropertyValue Table* on page 116)
- ComplianceComputerRole table (see *ComplianceComputerRole Table* on page 117)
- ComplianceComputerStatus table (see *ComplianceComputerStatus Table* on page 117)
- ComplianceComputerType table (see *ComplianceComputerType Table* on page 118)
- ComplianceComputerTypeProperty table (see *ComplianceComputerTypeProperty Table* on page 119)
- ComplianceComputerUsage table (see *ComplianceComputerUsage Table* on page 120)
- ComplianceEvent table (see *ComplianceEvent Table* on page 120)
- ComplianceEventAction table (see *ComplianceEventAction Table* on page 121)
- ComplianceEventHistory table (see *ComplianceEventHistory Table* on page 122)
- ComplianceEventState table (see *ComplianceEventState Table* on page 123)
- ComplianceEventType table (see *ComplianceEventType Table* on page 123)
- ComplianceHistory table (see *ComplianceHistory Table* on page 124)
- ComplianceHistoryColumn table (see *ComplianceHistoryColumn Table* on page 127)
- ComplianceHistoryType table (see *ComplianceHistoryType Table* on page 127)
- ComplianceImage table (see *ComplianceImage Table* on page 129)
- ComplianceLicenseUser table (see *ComplianceLicenseUser Table* on page 129)
- CompliancePredefinedSearch table (see *CompliancePredefinedSearch Table* on page 130)
- ComplianceResponsibility table (see *ComplianceResponsibility Table* on page 131)
- ComplianceSavedSearch table (see *ComplianceSavedSearch Table* on page 132)
- ComplianceSchedule table (see *ComplianceSchedule Table* on page 134)
- ComplianceSearchFolder table (see *ComplianceSearchFolder Table* on page 135)
- ComplianceSearchType table (see *ComplianceSearchType Table* on page 136)
- ComplianceSearchTypeColumn table (see *ComplianceSearchTypeColumn Table* on page 138)
- ComplianceSearchTypeRelation table (see *ComplianceSearchTypeRelation Table* on page 140)
- ComplianceTask table (see *ComplianceTask Table* on page 141)
- ComplianceUserPropertyValue table (see *ComplianceUserPropertyValue Table* on page 142)
- ComplianceUserTypeProperty table (see *ComplianceUserTypeProperty Table* on page 143)
- ComputerChassisType table (see *ComputerChassisType Table* on page 144)

- ConsolidatedLicenseUser table (see *ConsolidatedLicenseUser Table* on page 145)
- ConsolidationType table (see *ConsolidationType Table* on page 146)
- Contract table (see *Contract Table* on page 146)
- ContractNote table (see *ContractNote Table* on page 153)
- ContractNotification table (see *ContractNotification Table* on page 154)
- ContractNotificationResponsibility table (see *ContractNotificationResponsibility Table* on page 155)
- ContractProperty table (see *ContractProperty Table* on page 156)
- ContractPropertyValue table (see *ContractPropertyValue Table* on page 156)
- ContractScopingData table (see *ContractScopingData Table* on page 157)
- ContractSecurityUser table (see *ContractSecurityUser Table* on page 158)
- ContractState table (see *ContractState Table* on page 159)
- ContractStatus table (see *ContractStatus Table* on page 159)
- ContractType table (see *ContractType Table* on page 160)
- ContractUseRight table (see *ContractUseRight Table* on page 161)
- ContractUseRightIBM table (see *ContractUseRightIBM Table* on page 164)
- ContractVendor table (see *ContractVendor Table* on page 165)
- CurrencyRate table (see *CurrencyRate Table* on page 165)
- CurrencyRateSnapshot table (see *CurrencyRateSnapshot Table* on page 166)
- CustomPropertyDisplayXML table (see *CustomPropertyDisplayXML Table* on page 167)
- DisplayXML table (see *DisplayXML Table* on page 168)
- Document table (see *Document Table* on page 169)
- DocumentHistory table (see *DocumentHistory Table* on page 171)
- DocumentNote table (see *DocumentNote Table* on page 173)
- DocumentType table (see *DocumentType Table* on page 173)
- Event table (see *Event Table* on page 174)
- EventLogCategory table (see *EventLogCategory Table* on page 175)
- EventLogDetail table (see *EventLogDetail Table* on page 175)
- EventLogLevel table (see *EventLogLevel Table* on page 176)
- EventLogStatus table (see *EventLogStatus Table* on page 177)
- EventLogSummary table (see *EventLogSummary Table* on page 177)
- EventParameter table (see *EventParameter Table* on page 178)

- EventParameterType table (see *EventParameterType Table* on page 179)
- EventSeverity table (see *EventSeverity Table* on page 179)
- EventTarget table (see *EventTarget Table* on page 180)
- EventType table (see *EventType Table* on page 180)
- EventTypeStatus table (see *EventTypeStatus Table* on page 181)
- ILMTPVUCounts table (see *ILMTPVUCounts Table* on page 182)
- ImportResolverErrorResult table (see *ImportResolverErrorResult Table* on page 183)
- ImportResolverType table (see *ImportResolverType Table* on page 183)
- InstalledSoftwareAttribute table (see *InstalledSoftwareAttribute Table* on page 184)
- Instance table (see *Instance Table* on page 184)
- InstanceAttribute table (see *InstanceAttribute Table* on page 187)
- InstanceEnvironment table (see *InstanceEnvironment Table* on page 187)
- InstancePropertyValue table (see *InstancePropertyValue Table* on page 188)
- InstanceRole table (see *InstanceRole Table* on page 189)
- InstanceType table (see *InstanceType Table* on page 190)
- InstanceTypeProperty table (see *InstanceTypeProperty Table* on page 191)
- InstanceUser table (see *InstanceUser Table* on page 191)
- IntervalType table (see *IntervalType Table* on page 192)
- LicenseUser table (see *LicenseUser Table* on page 193)
- LicenseUserConnection table (see *LicenseUserConnection Table* on page 194)
- LicenseUserExcluded table (see *LicenseUserExcluded Table* on page 195)
- LicenseUserType table (see *LicenseUserType Table* on page 195)
- LogFile table (see *LogFile Table* on page 196)
- MSEAARLSoftwareTitleEdition table (see *MSEAARLSoftwareTitleEdition Table* on page 197)
- MSSelectLevel table (see *MSSelectLevel Table* on page 197)
- MSSelectPool table (see *MSSelectPool Table* on page 198)
- MobileDevice table (see *MobileDevice Table* on page 198)
- NotificationItem table (see *NotificationItem Table* on page 199)
- NotificationTemplate table (see *NotificationTemplate Table* on page 200)
- NotificationType table (see *NotificationType Table* on page 201)
- OperatorManageState table (see *OperatorManageState Table* on page 202)

- OperatorTaskTypeSetting table (see *OperatorTaskTypeSetting Table* on page 202)
- OracleInstance table (see *OracleInstance Table* on page 203)
- PaymentSchedule table (see *PaymentSchedule Table* on page 204)
- PaymentScheduleCategory table (see *PaymentScheduleCategory Table* on page 206)
- PaymentScheduleDetail table (see *PaymentScheduleDetail Table* on page 207)
- PaymentScheduleDetailPaymentStatus table (see *PaymentScheduleDetailPaymentStatus Table* on page 209)
- PaymentScheduleTerm table (see *PaymentScheduleTerm Table* on page 210)
- PaymentScheduleType table (see *PaymentScheduleType Table* on page 210)
- Project table (see *Project Table* on page 211)
- PurchaseOrder table (see *PurchaseOrder Table* on page 212)
- PurchaseOrderDetail table (see *PurchaseOrderDetail Table* on page 215)
- PurchaseOrderDetailProperty table (see *PurchaseOrderDetailProperty Table* on page 221)
- PurchaseOrderDetailPropertyValue table (see *PurchaseOrderDetailPropertyValue Table* on page 222)
- PurchaseOrderDetailStatus table (see *PurchaseOrderDetailStatus Table* on page 223)
- PurchaseOrderDetailType table (see *PurchaseOrderDetailType Table* on page 223)
- PurchaseOrderProperty table (see *PurchaseOrderProperty Table* on page 224)
- PurchaseOrderPropertyValue table (see *PurchaseOrderPropertyValue Table* on page 225)
- PurchaseOrderStatus table (see *PurchaseOrderStatus Table* on page 226)
- PurchaseOrderType table (see *PurchaseOrderType Table* on page 226)
- PurchaseProgram table (see *PurchaseProgram Table* on page 227)
- QuerySnapshot table (see *QuerySnapshot Table* on page 229)
- RelationType table (see *RelationType Table* on page 230)
- ResponsibilityType table (see *ResponsibilityType Table* on page 230)
- RestrictedAccessType table (see *RestrictedAccessType Table* on page 231)
- RulesEngineRuleDefinition table (see *RulesEngineRuleDefinition Table* on page 231)
- RulesEngineRuleType table (see *RulesEngineRuleType Table* on page 232)
- SecurityType table (see *SecurityType Table* on page 233)
- SerialNumberBlackList table (see *SerialNumberBlackList Table* on page 234)
- SessionUIDBeacon table (see *SessionUIDBeacon Table* on page 234)
- ShippingMethod table (see *ShippingMethod Table* on page 234)



- SoftwareLicenseContractPaymentSchedule table (see *SoftwareLicenseContractPaymentSchedule Table* on page 235)
- SystemShutdown table (see *SystemShutdown Table* on page 235)
- TaskExecutionStatus table (see *TaskExecutionStatus Table* on page 236)
- TaskExecutionStatusStep table (see *TaskExecutionStatusStep Table* on page 237)
- TaskStep table (see *TaskStep Table* on page 238)
- TaskStepEventType table (see *TaskStepEventType Table* on page 239)
- TermAndCondition table (see *TermAndCondition Table* on page 239)
- TermAndConditionTask table (see *TermAndConditionTask Table* on page 241)
- TermAndConditionType table (see *TermAndConditionType Table* on page 242)
- UserNameBlacklist table (see *UserNameBlacklist Table* on page 243)
- VMEnabledState table (see *VMEnabledState Table* on page 244)
- VMHostManagedBySoftware table (see *VMHostManagedBySoftware Table* on page 244)
- VMPool table (see *VMPool Table* on page 245)
- VMPoolType table (see *VMPoolType Table* on page 246)
- VMSourceType table (see *VMSourceType Table* on page 247)
- VMState table (see *VMState Table* on page 248)
- VMType table (see *VMType Table* on page 248)
- Vendor table (see *Vendor Table* on page 249)
- VendorContact table (see *VendorContact Table* on page 251)
- VendorProperty table (see *VendorProperty Table* on page 254)
- VendorPropertyValue table (see *VendorPropertyValue Table* on page 254)
- VirtualMachine table (see *VirtualMachine Table* on page 255)
- XMLInsertType table (see *XMLInsertType Table* on page 258)

## Activity Table

The `Activity` table stores errors and events processed by the beacon, devices, rules etc.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 100: Database columns for Activity table**

Database Column	Details
ActivityID	<i>Type:</i> integer. Key. Generated ID Synthetic key for this table.
SourceTypeID	<i>Type:</i> integer The source type ID such as Beacon, External and so on
SourceTypeName	<i>Type:</i> text (max 256 characters) The source type name such as Beacon, External and so on
ActivityTypeID	<i>Type:</i> integer. Key Foreign key to the ActivityType table.
ActivityUID	<i>Type:</i> unique identifier. Key UID to uniquely identify the activity.
DateCreated	<i>Type:</i> datetime Time that the activity is created in the database.

## ActivitySource Table

ActivitySource is a static table listing all of the Sources that can generate the activity logs.

**Table 101: Database columns for ActivitySource table**

Database Column	Details
ActivitySourceID	<i>Type:</i> integer. Key. Generated ID
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing the ActivitySource record. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 256 characters) The text to display if the state resource string has no translation.

## ActivityTraceLog Table

The ActivityTraceLog table stores the logs generated by the trace logger for the corresponding activity.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 102: Database columns for ActivityTraceLog table**

Database Column	Details
TraceID	Type: integer. Key. Generated ID The unique row identifier.
ActivityUID	Type: unique identifier. Nullable The Guid of the activity that trace logger is logging the events for.
DateCreated	Type: datetime. Nullable The date and time when the event occurred.
LogMessage	Type: text. Nullable The actual message logged by the trace logger.
LogLevel	Type: integer. Nullable The log level that the trace logger is logging to.
EventID	Type: integer. Key. Nullable The unique row identifier in negative form.

## ActivityType Table

The ActivityType table stores details about the different types of Activities.

**Table 103: Database columns for ActivityType table**

Database Column	Details
ActivityTypeID	Type: integer. Key. Generated ID Synthetic key for this table.

Database Column	Details
ActivityTypeName	<i>Type:</i> text (max 256 characters). Key A short piece of text representing the Activity Type. Internal use only- not to be displayed to the operator.
ActivityMessageResource	<i>Type:</i> text (max 256 characters) A resource name used to look up a description for this Activity
IsMonitored	<i>Type:</i> boolean Flag that determines whether to track this activity

## Alert Table

The `Alert` table stores alerts and notifications that the system can attach to different objects to be displayed to the operator.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 104: Database columns for Alert table**

Database Column	Details
AlertID	<i>Type:</i> integer. Key. Generated ID Synthetic key for this table.
AlertTypeID	<i>Type:</i> integer. Key Foreign key to the <code>AlertType</code> table.
Ignored	<i>Type:</i> boolean. Key This flag indicates whether this alert has been ignored by an operator. If so, then the <code>IgnoredDate</code> and <code>IgnoredOperator</code> values will be populated.
IgnoredDate	<i>Type:</i> datetime. Nullable If the alert has been ignored by an operator, then this field shows the date when this was done.
IgnoredOperator	<i>Type:</i> text (max 256 characters). Nullable

Database Column	Details
	If the alert has been ignored by an operator, then this field shows which operator ignored the alert.
CreationDate	Type: datetime Date and time (UTC) when alert was created.

## AlertCategory Table

The `AlertCategory` table stores the different categories of alerts.

**Table 105: Database columns for AlertCategory table**

Database Column	Details
AlertCategoryID	Type: integer. Key. Generated ID Synthetic key for this table.
DefaultName	Type: text (max 128 characters) The default name for this alert category
ResourceName	Type: text (max 128 characters). Key A resource name used to look up a description for this alert category

## AlertTarget Table

The `AlertTarget` table stores the links between alerts and other tables in the database.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 106: Database columns for AlertTarget table**

Database Column	Details
AlertID	Type: integer. Key Link to the <code>Alert</code> table

Database Column	Details
TargetTypeID	<i>Type:</i> integer. Key A link the the <code>TargetType</code> table. this value specifies which kind of object the alert is linked to.
TargetID	<i>Type:</i> integer. Key used to attach the <code>Alert</code> to its target. The target table depends on the TargetTypeID of the linked <code>AlertType</code> .
FieldName	<i>Type:</i> text. Nullable A semi-colon separated list of view-model names that represent the fields that the alert is attached to. A null value indicates that the alert applies to the overall object as a whole.

## AlertType Table

The `AlertType` table stores details about the different types of alerts.

**Table 107: Database columns for `AlertType` table**

Database Column	Details
AlertTypeID	<i>Type:</i> integer. Key. Generated ID Synthetic key for this table.
AlertTypeName	<i>Type:</i> text (max 256 characters). Key A short piece of text representing the Alert Type. Internal use only- not to be displayed to the operator.
AlertMessageResource	<i>Type:</i> text (max 256 characters) A resource name used to look up a description for this alert
AlertCategoryID	<i>Type:</i> integer The category of this type of alert

## AssetContractPaymentSchedule Table

`AssetContractPaymentSchedule` links a payment schedule to an asset, via a link from that asset to a contract.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 108: Database columns for AssetContractPaymentSchedule table**

Database Column	Details
AssetContractPaymentScheduleID	<i>Type:</i> integer. Key. Generated ID Unique identifier to represent a link between a payment schedule and an asset. This allows an asset to link multiple times to a payment schedule, each time with its own start and end dates.
AssetContractID	<i>Type:</i> integer. Key Identifies a link between an asset and a contract. Foreign key to the AssetContract table.
PaymentScheduleID	<i>Type:</i> integer. Key Identifies a payment schedule. Foreign key to the PaymentSchedule table.
ActiveStartDate	<i>Type:</i> datetime Start date of the association between the payment schedule and asset.
ActiveEndDate	<i>Type:</i> datetime. Nullable End date of the association between the payment schedule and asset.

## Attribute Table

Attribute holds the collection of possible attributes of database instances.

**Table 109: Database columns for Attribute table**

Database Column	Details
AttributeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for an attribute.
AttributeName	<i>Type:</i> text (max 256 characters). Key The name of the attribute.

## BusinessImportLogDetail Table

The `BusinessImportLogDetail` table stores per record import execution details for a business import execution.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 110: Database columns for `BusinessImportLogDetail` table**

Database Column	Details
<code>ImportDetailID</code>	<i>Type:</i> integer. Key. Generated ID Surrogate ID that uniquely identifies an import execution detail.
<code>ImportID</code>	<i>Type:</i> integer. Key Business import ID this execution detail relates to, foreign key to <code>BusinessImportLogSummary</code> table.
<code>RecordNumber</code>	<i>Type:</i> integer. Nullable Row number of source data in staging table that this execution detail related to.
<code>Action</code>	<i>Type:</i> text (max 10 characters). Nullable The trace action of the import execution detail.
<code>MGSRecordKey</code>	<i>Type:</i> text (max 50 characters). Nullable ID of matching FNMS table record the Record Number is matched against.
<code>ImportObjectID</code>	<i>Type:</i> integer. Key. Nullable Import object that this execution detail is related to, foreign key to <code>BusinessImportLogObject</code> table.
<code>RecordDescription</code>	<i>Type:</i> text (max 255 characters). Nullable Value of the trace field specified in the import element of business adapter xml if any.
<code>Message</code>	<i>Type:</i> text (max 3000 characters). Nullable Messages related to this import execution detail.



# BusinessImportLogObject Table

The `BusinessImportLogObject` table stores summary data for the execution of individual object imports within a business import execution.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 111: Database columns for `BusinessImportLogObject` table**

Database Column	Details
<code>ImportObjectID</code>	<i>Type:</i> integer. Key. Generated ID Surrogate ID that uniquely identifies an object in a business import execution.
<code>ImportID</code>	<i>Type:</i> integer. Key Business import ID this object belongs, foreign key to <code>BusinessImportLogSummary</code> table.
<code>ObjectName</code>	<i>Type:</i> text (max 50 characters). Nullable Name of the business import object.
<code>ObjectType</code>	<i>Type:</i> text (max 50 characters). Nullable Type of the business import object.
<code>StartDate</code>	<i>Type:</i> datetime. Nullable Date and time when the object began to be imported on FNMS server.
<code>EndDate</code>	<i>Type:</i> datetime. Nullable Date and time when import of the object is completed on FNMS server.
<code>Status</code>	<i>Type:</i> integer. Nullable Status of object import: 0 - Not completed, 1 - Completed.
<code>Processed</code>	<i>Type:</i> integer. Nullable Number of rows from data source that are processed for the object import.
<code>Matched</code>	<i>Type:</i> integer. Nullable Number of rows in the staging table that match records in the corresponding FNMS table for the object.

Database Column	Details
Rejected	<i>Type:</i> integer. Nullable Number of rows in the staging table that are rejected for the object import.
Updated	<i>Type:</i> integer. Nullable Number of rows in the staging table that are updated for the object import.
Created	<i>Type:</i> integer. Nullable Number of rows in the staging table that are created for the object import.
Deleted	<i>Type:</i> integer. Nullable Number of rows in the staging table that are deleted for the object import.

## BusinessImportLogSummary Table

The `BusinessImportLogSummary` table stores summary data for each business import execution.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 112: Database columns for `BusinessImportLogSummary` table**

Database Column	Details
ImportID	<i>Type:</i> integer. Key. Generated ID Surrogate ID that uniquely identifies a business import.
ImportName	<i>Type:</i> text (max 255 characters). Nullable Import name of the business import.
ImportType	<i>Type:</i> text (max 50 characters). Nullable Import type of the business import.
Action	<i>Type:</i> text (max 20 characters). Nullable The mode the business import is operating in e.g. Import, Simulation.
StartDate	<i>Type:</i> datetime. Nullable Date and time when the business import is started on FNMS server.

Database Column	Details
EndDate	Type: datetime. Nullable Date and time when the business import is completed on FNMS server.
Status	Type: integer. Nullable Status of the business import: 0 - Not completed, 1 - Completed.
Processed	Type: integer. Nullable Number of rows from data source that are processed for import.
Rejected	Type: integer. Nullable Number of rows from data source that are rejected from importing.
SessionUID	Type: unique identifier. Key. Nullable Unique task run identifier of the business import, nullable for business import initiated on the server.

## BusinessImportResult Table

The `BusinessImportResult` table contains the results of all business imports executed on the batch server.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 113: Database columns for `BusinessImportResult` table**

Database Column	Details
BusinessImportResultID	Type: integer. Key. Generated ID A unique identifier for the business import result.
ImportName	Type: text (max 256 characters) The name of the business import.
BeaconID	Type: integer. Key A link to <code>Beacon</code> from which this import was uploaded.
ImportStarted	Type: datetime

Database Column	Details
	The time at which the import was executed.
ImportEnded	Type: datetime The time at which the import was completed.
Result	Type: boolean Whether the import succeeded.

## ComplianceComputer Table

ComplianceComputer stores information about computers used in the enterprise, including hardware details, inventory source information and computer types.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 114: Database columns for ComplianceComputer table**

Database Column	Details
ComplianceComputerID	Type: integer. Key. Generated ID A unique identifier for a ComplianceComputer.
ComplianceComputerTypeID	Type: integer. Key A unique identifier for the type of computer. Foreign key to the ComplianceComputerType table.
IsComplianceComputerTypeIDFromInventory	Type: boolean This is true for records sourced from inventory, where the inventory source has specified the value of the ComplianceComputerTypeID. A true value will exclude this record from some processes that infer the type of a record. This value is set by the import process.
ComputerName	Type: text (max 256 characters). Key. Nullable The name of the computer.
ComplianceDomainID	Type: integer. Key. Nullable

Database Column	Details
	The domain to which the computer belongs. Foreign key to the ComplianceDomain table.
ComplianceComputerStatusID	Type: integer. Key The last recorded status for this computer. Foreign key to the ComplianceComputerStatus table.
ComplianceComputerRoleID	Type: integer. Key The functional role of this computer. Foreign key to the ComplianceComputerRole table.
ComplianceComputerInventorySourceTypeID	Type: integer. Key Whether this computer has ever been reported in inventory, or has been manually created and maintained. Foreign key to the ComplianceComputerInventorySourceType table.
AssetID	Type: integer. Key. Nullable When the computer is being managed as an asset, this is a foreign key to the Asset table; and is otherwise null.
OperatingSystem	Type: text (max 128 characters). Nullable The operating system of the computer.
ServicePack	Type: text (max 128 characters). Nullable The latest service pack reported as installed on the operating system.
NumberOfProcessors	Type: integer. Nullable The number of processors in the computer.
NumberOfProcessorsDefault	Type: integer. Nullable The inventoried number of processors in the computer.
ProcessorType	Type: text (max 256 characters). Nullable The type of processor in the computer.
ProcessorTypeDefault	Type: text (max 256 characters). Nullable The inventoried type of processor in the computer.
MaxClockSpeed	Type: integer. Nullable The maximum clock speed of the fastest processor in the computer in megahertz.

Database Column	Details
MaxClockSpeedDefault	<i>Type:</i> integer. Nullable The inventoried maximum clock speed of the fastest processor in the computer in megahertz.
TotalMemory	<i>Type:</i> big integer. Nullable The total RAM in the computer.
ChassisTypeID	<i>Type:</i> integer. Key The type of case for the computer, as reported in hardware inventory, defaulting to <code>Unknown</code> if no chassis type is reported. Foreign key to the <code>ComputerChassisType</code> table.
AssignedChassisTypeID	<i>Type:</i> integer. Nullable The type of case for the computer, as set by an operator. Foreign key to the <code>ComputerChassisType</code> table.
NumberOfHardDrives	<i>Type:</i> integer. Nullable The number of hard drives in the computer.
TotalDiskSpace	<i>Type:</i> big integer. Nullable The total size of all hard drives in the computer.
NumberOfNetworkCards	<i>Type:</i> integer. Nullable The number of network cards in the computer.
NumberOfDisplayAdapters	<i>Type:</i> integer. Nullable The number of graphics cards in the computer.
IPAddress	<i>Type:</i> text (max 256 characters). Nullable The IP address of the computer.
MACAddress	<i>Type:</i> text (max 256 characters). Nullable The MAC Addresses of the computer.
Manufacturer	<i>Type:</i> text (max 128 characters). Key. Nullable The manufacturer of the computer.
ModelNo	<i>Type:</i> text (max 128 characters). Nullable The model number of the computer.
ModelNoDefault	<i>Type:</i> text (max 128 characters). Nullable

Database Column	Details
	The inventoried model number of the computer.
SerialNo	<i>Type:</i> text (max 100 characters). Key. Nullable The serial number of the computer.
ComplianceUserID	<i>Type:</i> integer. Key. Nullable The end-user who last logged onto the computer. Foreign key to the ComplianceUser table.
AssignedUserID	<i>Type:</i> integer. Key. Nullable The end-user assigned to this computer by an operator. Foreign key to the ComplianceUser table.
CalculatedUserID	<i>Type:</i> integer. Key. Nullable An end-user of this computer, calculated by looking at usage. Foreign key to the ComplianceUser table.
LocationID	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise location associated with this computer. Foreign key to the GroupEx table.
BusinessUnitID	<i>Type:</i> text (max 128 characters). Key. Nullable Any corporate unit in the enterprise associated with this computer. Foreign key to the GroupEx table.
CostCenterID	<i>Type:</i> text (max 128 characters). Key. Nullable Any cost center in the enterprise associated with this computer. Foreign key to the GroupEx table.
CategoryID	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise category associated with this computer. Foreign key to the GroupEx table.
InventoryDate	<i>Type:</i> datetime. Nullable The date the computer last had inventory reported.
HardwareInventoryDate	<i>Type:</i> datetime. Nullable The date when the hardware was last reported.
ServicesInventoryDate	<i>Type:</i> datetime. Nullable The date when a service was last reported.

Database Column	Details
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The name of the operator who last updated the computer details.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the computer was created.
InventoryAgent	<i>Type:</i> text (max 64 characters). Nullable The name of the person or tool that performed the last inventory.
NumberOfCores	<i>Type:</i> integer. Nullable The number of cores in the computer.
NumberOfCoresDefault	<i>Type:</i> integer. Nullable The inventoried number of cores in the computer.
NumberOfSockets	<i>Type:</i> integer. Nullable The number of sockets in the computer.
NumberOfSocketsDefault	<i>Type:</i> integer. Nullable The inventoried number of sockets in the computer.
AssetComplianceStatusID	<i>Type:</i> integer. Nullable For computers managed as assets, the latest compliance status of the computer. Foreign key to the <code>AssetComplianceStatus</code> table.
PartialNumberOfProcessors	<i>Type:</i> decimal. Nullable The fractional processor count available to this computer.
PartialNumberOfProcessorsDefault	<i>Type:</i> decimal. Nullable The inventoried fractional processor count available to this computer.
UntrustedSerialNo	<i>Type:</i> boolean Is this computer known to have a serial number from a data source that should not be trusted.



Database Column	Details
ILMTAgentID	<i>Type:</i> big integer. Key. Nullable Store the unique ID used by the ILMT agent on this device, if the inventory source is aware of this value.
UUID	<i>Type:</i> unique identifier. Nullable The computer's UUID, in the byte order reported in inventory.
HostIdentifyingNumber	<i>Type:</i> text (max 128 characters). Key. Nullable Virtual hosts may have an identifier that is unique only across that hardware model. It is less unique than the true hardware serial number, for example.
HostType	<i>Type:</i> text (max 128 characters). Key. Nullable The type (similar to model number) of the host, used for matching.
NumberOfLogicalProcessors	<i>Type:</i> integer. Nullable The number of logical processors in the computer.
NumberOfLogicalProcessorsDefault	<i>Type:</i> integer. Nullable The inventoried number of logical processors in the computer.
PrimaryComplianceUserID	<i>Type:</i> integer. Key. Nullable Primary user of the computer based off the assigned user and calculated user.
MDScheduleGeneratedDate	<i>Type:</i> datetime. Nullable The last time the managed device schedule was regenerated.
MDScheduleContainsPVUScan	<i>Type:</i> boolean. Nullable Does this managed device include an event in its current schedule for running extra IBM PVU hardware scans.
HostID	<i>Type:</i> text (max 100 characters). Key. Nullable Numeric identifier of the current host
FirmwareSerialNumber	<i>Type:</i> text (max 100 characters). Key. Nullable Serial number in the system firmware such as BIOS, EEPROM etc.
MachineID	<i>Type:</i> text (max 100 characters). Key. Nullable For AIX, it is the System ID. For HP-UX, it is the Machine/Software ID. It is unset for other platforms.

## ComplianceComputerConnection Table

`ComplianceComputerConnection` stores a link between computers in `ComplianceComputer` which have been reported in inventory, and external IDs that can be used to identify them in their inventory sources. Computers reported in multiple inventory sources will appear multiple times in this table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 115: Database columns for `ComplianceComputerConnection` table**

Database Column	Details
<code>ComplianceComputerID</code>	<p>Type: integer. Key</p> <p>A unique identifier for the computer. Foreign key to the <code>ComplianceComputer</code> table.</p>
<code>ComplianceConnectionID</code>	<p>Type: integer. Key</p> <p>The inventory source where the computer was reported. Foreign key to the <code>ComplianceConnection</code> table.</p>
<code>ExternalID</code>	<p>Type: big integer</p> <p>The (hopefully unique) identifier for the computer in the external inventory source.</p>

## ComplianceComputerContract Table

`ComplianceComputerContract` stores links between computers and contracts, some of which may influence license compliance.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 116: Database columns for ComplianceComputerContract table**

Database Column	Details
ComplianceComputerContractID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this record.
ContractID	<i>Type:</i> integer. Key A unique identifier for a contract linked to a computer. Foreign key to the <code>Contract</code> table.
ComplianceComputerID	<i>Type:</i> integer. Key A unique identifier for a computer linked to a contract. Foreign key to the <code>ComplianceComputer</code> table.

## ComplianceComputerInventorySourceType Table

`ComplianceComputerInventorySourceType` is a static table used to define possible computer inventory source values (that is, whether the computer was created manually or reported by the compliance importer).

**Table 117: Database columns for ComplianceComputerInventorySourceType table**

Database Column	Details
ComplianceComputerInventorySourceTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>ComplianceComputerInventorySourceType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>1 = Automatic (computer was recently updated during an inventory import)</li> <li>2 = VM Host (a dummy or “light” computer created using the host inventory of a virtual machine)</li> <li>3 = Manual (computer was created manually by an operator, using FlexNet Manager Suite, and has never been updated by the compliance importer).</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a computer inventory source. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the inventory resource string has no translation.

# ComplianceComputerPropertyValue Table

For each computer, ComplianceComputerPropertyValue stores the values for the custom properties defined in ComplianceComputerTypeProperty.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 118: Database columns for ComplianceComputerPropertyValue table**

Database Column	Details
ComplianceComputerPropertyValueID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a property value.
ComplianceComputerID	<i>Type:</i> integer. Key The computer associated with this property value. Foreign key to the ComplianceComputer table
ComplianceComputerTypePropertyID	<i>Type:</i> integer. Key The property whose value is being stored. The type of the computer should match the type that the property is associated with. Foreign key to the ComplianceComputerTypeProperty table.
PropertyValue	<i>Type:</i> text (max 4000 characters) The value of the custom property.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.

## ComplianceComputerRole Table

`ComplianceComputerRole` is a static table listing all the different roles to which computers can be assigned, and which may impact licensing terms.

**Table 119: Database columns for `ComplianceComputerRole` table**

Database Column	Details
<code>ComplianceComputerRoleID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ComplianceComputerRole</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Production</li> <li>• 2 = Warm Standby / Passive Failover</li> <li>• 3 = Hot Standby / Active Failover</li> <li>• 4 = Backup / Archive</li> <li>• 5 = Test</li> <li>• 6 = Training</li> <li>• 7 = Cold Standby / Disaster recovery</li> <li>• 8 = Development.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a computer role. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the inventory resource string has no translation.</p>
<code>ManageLicenses</code>	<p><i>Type:</i> boolean</p> <p>Set this to <code>True</code> if computers in this role are to be included in compliance calculations, and to <code>False</code> if this role exempts a computer from the license management process. Of the computer roles listed above, only <code>Active</code> computers have their licenses managed.</p>

## ComplianceComputerStatus Table

`ComplianceComputerStatus` is a static table used to define possible values for the status of computers reported in FlexNet Manager Suite.

**Table 120: Database columns for ComplianceComputerStatus table**

Database Column	Details
ComplianceComputerStatusID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each ComplianceComputerStatus. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = New (this is the first appearance of this computer in inventory)</li> <li>• 2 = Ignored (an operator has marked this computer to be ignored)</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a computer status. Foreign key to the ComplianceResourceString table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the status resource string has no translation.</p>

## ComplianceComputerType Table

ComplianceComputerType is a static table listing all types of computers that can be created.

**Table 121: Database columns for ComplianceComputerType table**

Database Column	Details
ComplianceComputerTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each ComplianceComputerType. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Computer</li> <li>• 2 = VM Host</li> <li>• 3 = Virtual Machine</li> <li>• 4 = Remote Device.</li> <li>• 5 = Mobile Device.</li> <li>• 6 = VDI Template.</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a computer role. Foreign key to the ComplianceResourceString table.</p>

Database Column	Details
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.
XMLFile	<i>Type:</i> text. Nullable The layout of the property dialog for this type of computer, stored in XML format.
CanCreate	<i>Type:</i> boolean. Key Whether the end-user can manually create computers of this type.
CanEdit	<i>Type:</i> boolean. Key Whether the end-user can manually edit computers of this type.

## ComplianceComputerTypeProperty Table

`ComplianceComputerTypeProperty` defines extra custom properties for computers of the specified type.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 122: Database columns for `ComplianceComputerTypeProperty` table**

Database Column	Details
ComplianceComputerTypePropertyID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a property.
PropertyName	<i>Type:</i> text (max 256 characters). Key The name of the property.
ComplianceComputerTypeID	<i>Type:</i> integer. Key Computer type with which this property is associated. Foreign key to the <code>ComplianceComputerType</code> table.
CustomPropertyDisplayXMLID	<i>Type:</i> integer. Nullable Foreign key to a record in the <code>CustomPropertyDisplayXML</code> table, describing how to show the property on a property dialog.

## ComplianceComputerUsage Table

This table links user IDs with computer IDs, allowing ECM to determine who uses a computer most frequently; and this is one factor in determining the assigned user for a computer.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 123: Database columns for ComplianceComputerUsage table**

Database Column	Details
ComplianceComputerUsageID	Type: integer. Key. Generated ID Unique identifier for a ComplianceComputerUsage record.
ComplianceComputerID	Type: integer. Key Foreign key to the ComplianceComputer table.
ComplianceUserID	Type: integer. Key Foreign key to the ComplianceUser table.
DateRecorded	Type: datetime. Key The date and time that the record was inserted.

## ComplianceEvent Table

The ComplianceEvent table lists all the 'compliance events' that FlexNet Manager Suite has detected. These are any event, such as the arrival of a new application version or a change in primary application for a license, that should trigger recalculation of linked applications through upgrade and downgrade rights. Depending on license properties, some of these events trigger automatic recalculation, and others trigger a proposal to the operator for manual response. This table records the current state for each event, with a history of state changes available in the ComplianceEventHistory table. Where the compliance event results in changes to the applications linked to a license, further details are recorded in the SoftwareLicenseChangeEvent table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.



**Table 124: Database columns for ComplianceEvent table**

Database Column	Details
ComplianceEventID	<i>Type:</i> integer. Key. Generated ID A unique identifier for an event.
EventTypeID	<i>Type:</i> integer The type of event. Foreign key to the ComplianceEventType table.
Priority	<i>Type:</i> integer. Nullable The priority of the event.
Severity	<i>Type:</i> integer. Nullable The severity of the event.
EventActionID	<i>Type:</i> integer The proposed action for the event. Foreign key to the ComplianceEventAction table.
EventStateID	<i>Type:</i> integer The current state of the event. Foreign key to the ComplianceEventState table.
UpdatedBy	<i>Type:</i> text (max 200 characters) The last operator to update the event.
UpdatedDate	<i>Type:</i> datetime The date the event was last updated.

## ComplianceEventAction Table

The ComplianceEventAction table holds the list of possible actions in the handling of 'compliance events'. These are any event, such as the arrival of a new application version or a change in primary application for a license, that should trigger recalculation of linked applications through upgrade and downgrade rights.

**Table 125: Database columns for ComplianceEventAction table**

Database Column	Details
EventActionID	<i>Type:</i> integer. Key. Generated ID

Database Column	Details
	<p>A unique identifier for each <code>ComplianceEventAction</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>1 = Notification (the event is automatically managed, and the operator is to be advised of the result)</li> <li>2 = Request for Action (the license is not managed automatically, and the operator receives a suggested action).</li> </ul>
<code>EventActionResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an event type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>EventActionDefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## ComplianceEventHistory Table

`ComplianceEventHistory` stores a history of state changes for each compliance event.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 126: Database columns for `ComplianceEventHistory` table**

Database Column	Details
<code>ComplianceEventHistoryID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>Unique identifier for an event history record.</p>
<code>ComplianceEventID</code>	<p><i>Type:</i> integer. Key</p> <p>The event whose history is being recorded. Foreign key to the <code>ComplianceEvent</code> table.</p>
<code>UserName</code>	<p><i>Type:</i> text (max 60 characters)</p> <p>The operator who made the change.</p>
<code>HistoryDate</code>	<p><i>Type:</i> datetime</p> <p>The date of the change.</p>

Database Column	Details
FieldName	<i>Type:</i> text (max 256 characters). Nullable The field name that has been updated. Foreign key to the <code>ComplianceResourceString</code> table.
OldValue	<i>Type:</i> text (max 500 characters). Nullable The value before the change.
NewValue	<i>Type:</i> text (max 500 characters). Nullable The value after the change.

## ComplianceEventState Table

`ComplianceEventState` is a static table holding all possible event states.

**Table 127: Database columns for `ComplianceEventState` table**

Database Column	Details
EventStateID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>ComplianceEventState</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = New (action needs to be taken for this event)</li> <li>• 2 = Postponed (no action needs to be taken at this time)</li> <li>• 3 = Accepted (the proposed action has been taken for this event)</li> <li>• 4 = Rejected (the proposed action will not be taken).</li> </ul>
EventStateResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing an event state. Foreign key to the <code>ComplianceResourceString</code> table.
EventStateDefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the state resource string has no translation.

## ComplianceEventType Table

`ComplianceEventType` is a static table that holds all possible types of event.

**Table 128: Database columns for ComplianceEventType table**

Database Column	Details
EventTypeID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for each ComplianceEventType. Reserved for future expansion. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>1 = Software License Change.</li> </ul>
EventTypeResourceName	<p>Type: text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an event type. Foreign key to the ComplianceResourceString table.</p>
EventTypeDefaultValue	<p>Type: text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## ComplianceHistory Table

The ComplianceHistory table records changes to many entities used in FlexNet Manager Suite. This table has a series of ID columns, any one (or sometimes more) of which may be set to associate the history with a particular item. These ID columns no longer have foreign keys to other tables. This allows us to retain history of deleted objects in order to maintain an audit trail (as yet, there is no UI around this information), and also to improve performance when deleting objects.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database TenantID has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 129: Database columns for ComplianceHistory table**

Database Column	Details
ComplianceHistoryID	<p>Type: integer. Key. Generated ID</p> <p>Unique identifier for a history record.</p>
AssetID	<p>Type: integer. Key. Nullable</p> <p>ID from the Asset table.</p>
ComplianceComputerID	<p>Type: integer. Key. Nullable</p> <p>ID from the ComplianceComputer table.</p>

Database Column	Details
ContractID	Type: integer. Key. Nullable ID from the Contract table.
VendorID	Type: integer. Key. Nullable ID from the Vendor table.
VirtualMachineID	Type: integer. Key. Nullable ID from the VirtualMachine table.
PurchaseOrderID	Type: integer. Key. Nullable ID from the PurchaseOrder table.
PurchaseOrderDetailID	Type: integer. Key. Nullable ID from the PurchaseOrderDetail table.
SoftwareLicenseID	Type: integer. Key. Nullable ID from the SoftwareLicense table
SoftwareTitleID	Type: integer. Key. Nullable ID from the SoftwareTitle table
PaymentScheduleID	Type: integer. Key. Nullable ID from the PaymentSchedule table
InstanceID	Type: integer. Key. Nullable ID from the Instance table
ComplianceUserID	Type: integer. Key. Nullable ID from the ComplianceUser table
ComplianceOperatorID	Type: integer. Key. Nullable ID from the ComplianceOperator table
DocumentID	Type: integer. Key. Nullable ID from the Document table
DocumentNoteID	Type: integer. Key. Nullable ID from the DocumentNote table
ContractNoteID	Type: integer. Key. Nullable

Database Column	Details
	ID from the <code>ContractNote</code> table
<code>ProjectID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>ID from the <code>Project</code> table</p>
<code>FieldName</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The field name that has been updated. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>OldValue</code>	<p><i>Type:</i> text (max 4000 characters). Nullable</p> <p>Typically the value before the change, although at times, when multiple pieces of information are required to identify the action taking place, this field may store other supporting information. For example, when an operator is granted rights to access a contract, this field stores the type of access (such as "Normal" or "Administrator") while the <code>NewValue</code> field stores the name of the contract.</p>
<code>NewValue</code>	<p><i>Type:</i> text (max 4000 characters). Nullable</p> <p>Typically the value after the change, although refer to the above definition of the <code>OldValue</code> column for a description of extenuating circumstances.</p>
<code>NeedsApproval</code>	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if the change requires approval. Used usually to track changes to computer hardware.</p>
<code>ValuesAreResourceStrings</code>	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if the old and new values should be looked up as resource strings.</p>
<code>ComplianceHistoryTypeID</code>	<p><i>Type:</i> integer</p> <p>Foreign key to the <code>HistoryType</code> table.</p>
<code>UserName</code>	<p><i>Type:</i> text (max 60 characters)</p> <p>The operator who made the change.</p>
<code>HistoryDate</code>	<p><i>Type:</i> datetime. Key</p> <p>The date of the change.</p>
<code>Comments</code>	<p><i>Type:</i> text (max 2000 characters). Nullable</p> <p>Comments recorded about the change after it was made.</p>

## ComplianceHistoryColumn Table

The `ComplianceHistoryColumn` table lists the fields (columns) for which history details can be recorded.

**Table 130: Database columns for `ComplianceHistoryColumn` table**

Database Column	Details
<code>ComplianceHistoryColumnID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for a history column.</p>
<code>TableName</code>	<p><i>Type:</i> text (max 128 characters). Key</p> <p>The name of the database table to which the history-record settings apply. This may have a suffix of .1 or .2. These suffixes are used for grouping purposes. Do not edit this field.</p>
<code>ColumnName</code>	<p><i>Type:</i> text (max 128 characters). Key</p> <p>A description of the column in the specified <code>TableName</code> for which the history record settings apply. If this row relates to an entire table, the <code>ColumnName</code> will contain the word "History", for example, "Asset History" or "Contract History".</p>
<code>BitwiseValue</code>	<p><i>Type:</i> integer. Key</p> <p>The bitwise value uniquely identifies each row relating to a single <code>TableName</code>. Typically, a value of 1 indicates that this row relates to an entire table. A value greater than 1 indicates that this row relates to a single field in the table. Do not edit this field.</p>
<code>RecordHistory</code>	<p><i>Type:</i> boolean</p> <p>Boolean field to indicate if history should be recorded. Set this value to 1 (True) to record history details. Set this value to 0 (False) if no history details should be recorded.</p>

## ComplianceHistoryType Table

`ComplianceHistoryType` is a static table listing all valid types of history records.

**Table 131: Database columns for `ComplianceHistoryType` table**

Database Column	Details
<code>ComplianceHistoryTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for a history type.</p>

Database Column	Details
ComplianceHistoryType Description	<p><i>Type:</i> text (max 100 characters)</p> <p>A unique identifier for each <code>ComplianceHistoryType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Insert</li> <li>• 2 = Delete</li> <li>• 3 = Update</li> <li>• 4 = Link</li> <li>• 5 = Unlink</li> <li>• 6 = Allocated</li> <li>• 7 = Unallocated</li> <li>• 8 = Assigned</li> <li>• 9 = Unassigned</li> <li>• 10 = Operator unlinked from user due to duplicate login (operator history)</li> <li>• 11 = Operator unlinked from user due to duplicate login (user history)</li> <li>• 12 = Rights to contract granted</li> <li>• 13 = Rights to contract updated</li> <li>• 14 = Rights to contract removed</li> <li>• 15 = Rights to document granted</li> <li>• 16 = Rights to document updated</li> <li>• 17 = Rights to document removed</li> <li>• 18 = Receives (referring to escalations or alerts)</li> <li>• 19 = No longer receives (referring to escalations or alerts)</li> <li>• 20 = Assigned responsibility</li> <li>• 21 = Unassigned responsibility</li> <li>• 22 = Final state of entity when deleted</li> <li>• 23 = Rights to contract removed because contract was deleted</li> <li>• 24 = Rights to document removed because document was deleted</li> <li>• 25 = No longer receives (referring to escalations or alerts) because entity deleted</li> <li>• 26 = Unassigned responsibility because entity was deleted</li> </ul>



Database Column	Details
	<ul style="list-style-type: none"> <li>27 = Responsibility type changed.</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a history type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.

## ComplianceImage Table

The `ComplianceImage` table stores a collection of images to use on property display dialogs.

**Table 132: Database columns for `ComplianceImage` table**

Database Column	Details
ComplianceImageName	<i>Type:</i> text (max 50 characters). Key The name of the image.
ComplianceImageFile	<i>Type:</i> text The binary representation of the image.

## ComplianceLicenseUser Table

If external end-users, reported by systems such as SAP and stored in the `LicenseUser` table, can be matched to existing end-users in the enterprise (stored in the `ComplianceUser` table), the link between them is recorded in the `ComplianceLicenseUser` table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 133: Database columns for `ComplianceLicenseUser` table**

Database Column	Details
LicenseUserID	<i>Type:</i> integer. Key

Database Column	Details
	A unique identifier for the external end-user. Foreign key to the <code>LicenseUser</code> table.
<code>ComplianceUserID</code>	<p><i>Type:</i> integer. Key</p> <p>A unique identifier for the end-user in the enterprise. Foreign key to the <code>ComplianceUser</code> table.</p>

## CompliancePredefinedSearch Table

The `CompliancePredefinedSearch` holds a list of the predefined asset and licenses searches available to the operator. Each predefined search has its own grid in the FlexNet Manager Suite UI, and is accessed from a node which is a child of either `Licenses` or `Assets` nodes.

**Table 134: Database columns for `CompliancePredefinedSearch` table**

Database Column	Details
<code>CompliancePredefinedSearchID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>CompliancePredefinedSearch</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = New Inventory</li> <li>• 2 = Changed Assets</li> <li>• 3 = Lease Expiry</li> <li>• 4 = Warranty Expiry</li> <li>• 5 = Missing Computers</li> <li>• 6 = License Breach</li> <li>• 7 = License Expiry</li> <li>• 8 = License Contract Expiry</li> <li>• 9 = License Unused</li> <li>• 10 = UnLicensed Apps</li> <li>• 11 = UnLicensed Installs</li> <li>• 12 = License Group Breach</li> <li>• 13 = License Upgrade Downgrade.</li> </ul>
<code>SearchNameResource</code>	<i>Type:</i> text (max 128 characters). Key

Database Column	Details
	Resource string identifying the predefined search.
SearchNameDefault	Type: text (max 128 characters) The name of the predefined search.
AmberThreshold	Type: integer Indicates when the amber state should be shown in the related traffic light summary.
RedThreshold	Type: integer Indicates when the red state should be shown in the related traffic light summary.
DateSearch	Type: boolean. Key True indicates that the search is date based. False means count based.
ComplianceSearchType	Type: text (max 128 characters). Key. Nullable Type of search. Matches the name of a row in the ComplianceSearchType table.

## ComplianceResponsibility Table

ComplianceResponsibility links end-users to a contract with various responsibility types.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 135: Database columns for ComplianceResponsibility table**

Database Column	Details
ComplianceResponsibilityID	Type: integer. Key. Generated ID A unique identifier for a record.
ResponsibilityTypeID	Type: integer The particular type of responsibility. Foreign key to the ResponsibilityType table.

Database Column	Details
ContractID	<i>Type:</i> integer. Key The contract for which this end-user has some responsibility. Foreign key to the <code>Contract</code> table.
ComplianceUserID	<i>Type:</i> integer. Key The end-user who has this responsibility for (or relationship to) the contract. Foreign key to the <code>ComplianceUser</code> table.
Comment	<i>Type:</i> text (max 500 characters). Nullable Any operator comments related to the user responsibility.

## ComplianceSavedSearch Table

The `ComplianceSavedSearch` table holds the name of a custom view and any descriptive information about it.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 136: Database columns for `ComplianceSavedSearch` table**

Database Column	Details
ComplianceSavedSearchID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a custom view.
SearchName	<i>Type:</i> text (max 64 characters). Nullable The name of the custom view.
Description	<i>Type:</i> text (max 1000 characters). Nullable A description of the custom view.
SearchGridLayout	<i>Type:</i> text. Nullable The grid layout used in the custom view.
SearchSQL	<i>Type:</i> text. Nullable SQL statement that generates the data set for the custom view.
SearchSQLConnection	<i>Type:</i> text (max 500 characters)

Database Column	Details
	SQL connection to use to execute search SQL: 'Live', 'DataWarehouse', 'QuerySnapshot', 'ExternalFNMEA', or connection string.
SearchMapping	Type: XML. Nullable Search query XML to SQL mapping.
SearchXML	Type: XML. Nullable Search query XML.
CreatedBy	Type: text (max 128 characters) The operator who created the custom view.
CreationDate	Type: datetime The date the custom view was created.
ModifiedBy	Type: text (max 128 characters). Nullable The operator who last modified the custom view.
ModificationDate	Type: datetime. Nullable The date the custom view was last modified.
ComplianceSearchTypeID	Type: integer. Key The type of the custom view. Foreign key to the ComplianceSearchType table.
ComplianceSearchFolderID	Type: integer. Key The folder in which this custom view is stored. Foreign key to the ComplianceSearchFolder table.
CreatedByOperatorID	Type: integer. Key. Nullable ID of the operator who created the view. Foreign key to the ComplianceOperator table.
RestrictedAccessTypeID	Type: integer. Key Defined access type to the view. Foreign key to the RestrictedAccessType table.
CanDelete	Type: boolean Set this to False for predefined custom views which an operator is not allowed to delete.
CanChangeMasterObject	Type: boolean

Database Column	Details
	Set this to <code>False</code> if the this view has a fixed master object.
<code>ComplianceSavedSearchSystemID</code>	<i>Type:</i> integer. Key. Nullable An identifier for a system custom view.
<code>SearchNameResourceName</code>	<i>Type:</i> text (max 256 characters). Nullable The unique name of the localizable resource string representing a column name. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DescriptionResourceName</code>	<i>Type:</i> text (max 256 characters). Nullable The unique name of the localizable resource string representing a column name. Foreign key to the <code>ComplianceResourceString</code> table.
<code>SavedSearchLink</code>	<i>Type:</i> text. Nullable The saved built in report or view link.
<code>SavedSearchFilter</code>	<i>Type:</i> text. Nullable The saved filter for report or view

## ComplianceSchedule Table

`ComplianceSchedule` defines schedules that take place repeatedly at a specified interval.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 137: Database columns for `ComplianceSchedule` table**

Database Column	Details
<code>ComplianceScheduleID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for the schedule.
<code>TermAndConditionID</code>	<i>Type:</i> integer. Key. Nullable The term/condition that the schedule is associated with. Foreign key to the <code>TermAndCondition</code> table.
<code>StartDate</code>	<i>Type:</i> datetime

Database Column	Details
	The date on which this schedule first applies.
EndDate	<i>Type:</i> datetime The date on which this schedule ends.
RepeatIntervalTypeID	<i>Type:</i> integer. Key. Nullable The type of repeat interval. Foreign key to the <code>IntervalType</code> table.
RepeatInterval	<i>Type:</i> integer. Nullable The interval between repeats of this schedule.

## ComplianceSearchFolder Table

The `ComplianceSearchFolder` table identifies a folder for storing a custom search (or view), and tracks the parent-child relationships of folders to establish their hierarchy.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 138: Database columns for `ComplianceSearchFolder` table**

Database Column	Details
ComplianceSearchFolderID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a saved search folder.
Name	<i>Type:</i> text (max 128 characters). Key. Nullable The name of the folder.
ParentFolderID	<i>Type:</i> integer. Key. Nullable Identifies the parent that contains this folder. Foreign key to another folder in this <code>ComplianceSearchFolder</code> table.
ComplianceSearchTypeID	<i>Type:</i> integer. Key The kind of custom view stored in this folder. Foreign key to the <code>ComplianceSearchType</code> table.
Path	<i>Type:</i> text (max 128 characters). Key. Nullable

Database Column	Details
	The internal path to the folder.
PredefinedSearchesCreated	<i>Type:</i> boolean. Nullable Set this field to <code>True</code> to indicate that this folder holds generated searches.
CanDelete	<i>Type:</i> boolean. Nullable Set this field to <code>False</code> for predefined folders which operators are not allowed to delete.
CreatedByOperatorID	<i>Type:</i> integer. Key. Nullable ID of the operator who created the view. Foreign key to the <code>ComplianceOperator</code> table.
RestrictedAccessTypeID	<i>Type:</i> integer. Key Defined access type to the view. Foreign key to the <code>RestrictedAccessType</code> table.
ComplianceSearchFolder SystemID	<i>Type:</i> integer. Key. Nullable An identifier for a system custom view folder.
NameResourceName	<i>Type:</i> text (max 256 characters). Key. Nullable The unique name of the localizable resource string representing a folder name. Foreign key to the <code>ComplianceResourceString</code> table.

## ComplianceSearchType Table

`ComplianceSearchType` is a static table holding the name of the basic objects, such as an asset or license, for which custom views can be created.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 139: Database columns for `ComplianceSearchType` table**

Database Column	Details
ComplianceSearchTypeID	<i>Type:</i> integer. Key. Generated ID



Database Column	Details
	<p>A unique identifier for a type of compliance search. Possible values, together with the associated names of the object being searched, are:</p> <ul style="list-style-type: none"> <li>• -1 = Custom</li> <li>• 1 = Asset</li> <li>• 2 = License</li> <li>• 3 = Contract</li> <li>• 4 = Vendor</li> <li>• 5 = PurchaseOrder</li> <li>• 6 = SoftwareTitle</li> <li>• 7 = User</li> <li>• 8 = Computer</li> <li>• 13 = PurchaseOrderDetail</li> <li>• 14 = VirtualMachine</li> <li>• 15 = InstalledSoftware</li> <li>• 16 = SoftwareLicenseAllocation</li> <li>• 17 = PaymentSchedule</li> <li>• 18 = PaymentScheduleDetail</li> <li>• 19 = OracleInstance</li> <li>• 20 = OracleComponent</li> <li>• 21 = Suite</li> <li>• 22 = SuiteMember</li> <li>• 23 = TermAndCondition</li> <li>• 24 = ContractHistoryView</li> <li>• 25 = ContractDocumentView</li> <li>• 26 = DocumentNote</li> <li>• 27 = ComplianceResponsibility</li> <li>• 28 = ContractNote</li> <li>• 29 = Location</li> <li>• 30 = CostCenter</li> <li>• 31 = CorporateStructure</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>32 = Category</li> <li>33 = VendorContact</li> <li>34 = Cluster.</li> </ul>
TypeName	<i>Type:</i> text (max 64 characters). Key The name of the objects being searched.
TypeNameResourceName	<i>Type:</i> text (max 256 characters). Nullable The unique name of the localizable resource string representing a type name. Foreign key to the <code>ComplianceResourceString</code> table.
QuerySetup	<i>Type:</i> text. Nullable Query pre-calculation statement executed before custom view query.
QueryFilter	<i>Type:</i> text. Nullable Query filter template executed before custom view query.
QueryTemplate	<i>Type:</i> text. Nullable Query template for this search type.
IsCustom	<i>Type:</i> boolean False if the relation is out of the box, false otherwise.

## ComplianceSearchTypeColumn Table

The `ComplianceSearchTypeColumn` table identifies all columns that may be used in custom views.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 140: Database columns for `ComplianceSearchTypeColumn` table**

Database Column	Details
ComplianceSearchTypeColumnID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a custom view column.

Database Column	Details
ColumnName	<p><i>Type:</i> text (max 128 characters). Key</p> <p>The default value of the display column name.</p>
ColumnNameResourceName	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The unique name of the localizable resource string representing a column name. Foreign key to the <code>ComplianceResourceString</code> table.</p>
QuerySetup	<p><i>Type:</i> text. Nullable</p> <p>Query pre-calculation statement executed before the custom view query.</p>
FromTable	<p><i>Type:</i> text. Nullable</p> <p>The name of the database table where the column can be found.</p>
SelectName	<p><i>Type:</i> text. Nullable</p> <p>The name of the column in the database.</p>
JoinClause	<p><i>Type:</i> text. Nullable</p> <p>The SQL join that links other tables to provide the relevant data for this column.</p>
WhereClause	<p><i>Type:</i> text. Nullable</p> <p>The SQL “WHERE” statement that limits the information returned by the custom view.</p>
SelectOptionsSQL	<p><i>Type:</i> text. Nullable</p> <p>The SQL that selects the predefined list that the user can display when filtering on this column.</p>
FilterGroupType	<p><i>Type:</i> integer. Nullable</p> <p>An ID that indicates the kind of value expected in this column, which in turn determines what kinds of filter options (such as <code>Contains</code>, <code>Starts With</code>) will be offered for this column. Possible values (and their associated meanings) are:</p> <ul style="list-style-type: none"> <li>• 1 = string</li> <li>• 2 = number</li> <li>• 3 = list</li> <li>• 4 = date</li> <li>• 5 = group</li> <li>• 6 = money</li> <li>• 7 = boolean.</li> </ul>

Database Column	Details
DefaultFilterType	<p><i>Type:</i> integer. Nullable</p> <p>The type of field that should be used to search for information in this column. Possible values (and their associated meanings) are the same as for the previous field.</p>
ComplianceSearchTypeID	<p><i>Type:</i> integer. Key</p> <p>The type of that the column is related to. Foreign key to the <code>ComplianceSearchType</code> table.</p>
RequiresSearchTypeID	<p><i>Type:</i> integer. Nullable</p> <p>For special cases, a column may need data from another compliance object as well. Foreign key to the <code>ComplianceSearchType</code> table.</p>
Mandatory	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if this column must always be returned in the SQL “SELECT” statement.</p>
PrimaryKey	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if this column is the primary key of the SQL “SELECT” statement.</p>
SelectByDefault	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if this column should be included (checked) by default when the operator is creating a custom view. If <code>False</code>, the operator may include it manually.</p>
IsCustom	<p><i>Type:</i> boolean</p> <p>False if the relation is out of the box, false otherwise.</p>

## ComplianceSearchTypeRelation Table

The `ComplianceSearchTypeRelation` table tracks relationships between different objects for which operators can create custom views.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 141: Database columns for ComplianceSearchTypeRelation table**

Database Column	Details
ComplianceSearchTypeRelationID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a relationship.
RelationName	<i>Type:</i> text (max 256 characters). Key The unique internal name of this relation.
DescriptionResourceName	<i>Type:</i> text (max 256 characters). Nullable The unique name of the localizable resource string representing a relationship name. Foreign key to the ComplianceResourceString table.
DescriptionDefault	<i>Type:</i> text (max 256 characters) The default description of the relationship.
FromSearchTypeID	<i>Type:</i> integer. Key The ComplianceSearchType that represents the source of the relationship.
ToSearchTypeID	<i>Type:</i> integer. Key The ComplianceSearchType that represents the destination of the relationship.
ToMany	<i>Type:</i> boolean Set this field to <code>True</code> to allow more than one related row in the destination table for each row in the source table. If this field is <code>False</code> , rows have a one-to-one relationship.
JoinClause	<i>Type:</i> text The SQL join clause used to join the source object with a related object.
FilterClause	<i>Type:</i> text The SQL filter clause used to filter the source object with a related object.
IsCustom	<i>Type:</i> boolean False if the relation is out of the box, false otherwise.

## ComplianceTask Table

ComplianceTask holds a collection of tasks, which are audit responsibilities generated by settings on a TermAndCondition.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 142: Database columns for ComplianceTask table**

Database Column	Details
ComplianceTaskID	Type: integer. Key. Generated ID A unique identifier for the task.
ComplianceScheduleID	Type: integer. Key. Nullable The schedule the task is associated with. Foreign key to the ComplianceSchedule table.
TaskDate	Type: datetime. Nullable The date for the task.

## ComplianceUserPropertyValue Table

For each end-user, ComplianceUserPropertyValue stores the values for the custom properties defined in ComplianceUserTypeProperty.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 143: Database columns for ComplianceUserPropertyValue table**

Database Column	Details
ComplianceUserPropertyValueID	Type: integer. Key. Generated ID A unique identifier for the property value.
ComplianceUserTypePropertyID	Type: integer. Key The property whose value is being stored. Foreign key to the ComplianceUserTypeProperty table.
ComplianceUserID	Type: integer. Key

Database Column	Details
	The end-user associated with this property value. Foreign key to the <code>ComplianceUser</code> table.
<code>PropertyValue</code>	<i>Type:</i> text (max 4000 characters) The value of the property for the specified <code>ComplianceUser</code> .
<code>CreationUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
<code>CreationDate</code>	<i>Type:</i> datetime The date the record was created.
<code>UpdatedUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
<code>UpdatedDate</code>	<i>Type:</i> datetime. Nullable The date the record was last updated.

## ComplianceUserTypeProperty Table

`ComplianceUserTypeProperty` defines extra custom properties for all end-users.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 144: Database columns for `ComplianceUserTypeProperty` table**

Database Column	Details
<code>ComplianceUserTypePropertyID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for the property.
<code>PropertyName</code>	<i>Type:</i> text (max 256 characters). Key The name of the property.
<code>CustomPropertyDisplayXMLID</code>	<i>Type:</i> integer. Nullable Foreign key to a record in the <code>CustomPropertyDisplayXML</code> table, describing how to show the property on a property dialog.

# ComputerChassisType Table

`ComputerChassisType` is a static table listing all possible computer chassis (case) types.

**Table 145: Database columns for `ComputerChassisType` table**

Database Column	Details
<code>ChassisTypeID</code>	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for each <code>ComputerChassisType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Other</li> <li>• 2 = Unknown</li> <li>• 3 = Desktop</li> <li>• 4 = Low Profile Desktop</li> <li>• 5 = Pizza Box</li> <li>• 6 = Mini Tower</li> <li>• 7 = Tower</li> <li>• 8 = Portable</li> <li>• 9 = Laptop</li> <li>• 10 = Notebook</li> <li>• 11 = Other Hand Held</li> <li>• 12 = Docking Station</li> <li>• 13 = All in One</li> <li>• 14 = Sub Notebook</li> <li>• 15 = Space-Saving</li> <li>• 16 = Lunch Box</li> <li>• 17 = Main System Chassis</li> <li>• 18 = Expansion Chassis</li> <li>• 19 = Sub-Chassis</li> <li>• 20 = Bus Expansion Chassis</li> <li>• 21 = Peripheral Chassis</li> <li>• 22 = Storage Chassis</li> <li>• 23 = Rack Mount Chassis</li> </ul>



Database Column	Details
	<ul style="list-style-type: none"> <li>• 24 = Sealed-Case PC.</li> <li>• 25 = Smart Phone</li> <li>• 26 = Tablet</li> </ul>
WMICHassisTypeID	<p><i>Type:</i> integer. Nullable</p> <p>The identifier for the chassis type identified in WMI.</p>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a computer role. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 128 characters)</p> <p>The text to display if the chassis type resource string has no translation.</p>
IncludeInLicenseRec SecondUseDefault	<p><i>Type:</i> boolean</p> <p>Determines whether or not a second installation of an application on a computer of this chassis type (as well as on a primary computer assigned to the same end-user) may be counted as a legal second use under the Right of Second Use granted by some licenses. Currently, this field is used to group together chassis types that can be treated as “laptops” for this purpose.</p>
SecondUseBitwiseValue	<p><i>Type:</i> integer</p> <p>Reserved for future use. Do not edit.</p>

## ConsolidatedLicenseUser Table

This table stores the data specific to a consolidated license user.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 146: Database columns for ConsolidatedLicenseUser table**

Database Column	Details
ConsolidatedLicenseUserID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for the consolidated license user.</p>

Database Column	Details
LicenseUserID	<i>Type:</i> integer Foreign key to the <code>LicenseUser</code> table.
ConsolidatedGroupNumber	<i>Type:</i> integer The unique identifier showing which users are duplicates of one another.
ConsolidatedName	<i>Type:</i> text The name of the consolidated user. If consolidated by rules engine, this column stores the name of the user with the lowest <code>LicenseUserID</code>
ConsolidationTypeID	<i>Type:</i> integer Foreign key to the <code>ConsolidationType</code> table.

## ConsolidationType Table

This table stores consolidation type.

**Table 147: Database columns for ConsolidationType table**

Database Column	Details
ConsolidationTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the consolidation type.
ResourceName	<i>Type:</i> text (max 256 characters). Key. Nullable A localizable resource string representing a consolidation type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the consolidation type resource string has no translation.

## Contract Table

The `Contract` table contains a list of all the contracts in the system.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 148: Database columns for Contract table**

Database Column	Details
ContractID	Type: integer. Key. Generated ID A unique identifier for the contract.
ContractNo	Type: text (max 60 characters) The contract number assigned by the operator.
ContractName	Type: text (max 100 characters) A contract name assigned by the operator.
ContractTypeID	Type: integer. Key Identifies the type of contract. Foreign key to the <i>ContractType</i> table.
ContractStatusID	Type: integer Identifies the status of the contract. Foreign key to the <i>ContractStatus</i> table.
NeverExpires	Type: boolean If set to <i>True</i> , this contract never expires. If <i>False</i> , the contract expires at the date specified in the <i>EndDate</i> field.
StartDate	Type: datetime. Nullable The start date of the contract.
EndDate	Type: datetime. Nullable The end date of the contract.
PreExpiryDate	Type: datetime. Nullable The date at which a contract should be reviewed prior to its expiry date.
RenewalDate	Type: datetime. Nullable The date at which a contract is due to be renewed.
Price	Type: currency. Nullable The price of the contract.
PriceRateID	Type: integer. Nullable

Database Column	Details
	The currency rate to be applied to the above contract price. Foreign key to the <code>CurrencyRate</code> table.
<code>PeriodTypeID</code>	<i>Type:</i> integer. Nullable The frequency with which the period payments are applicable. Foreign key to the <code>PeriodType</code> table.
<code>BuyoutCost</code>	<i>Type:</i> currency. Nullable The buyout cost of the contract.
<code>BuyoutCostRateID</code>	<i>Type:</i> integer. Nullable The currency rate to be applied to the above buyout cost. Foreign key to the <code>CurrencyRate</code> table.
<code>ManagerID</code>	<i>Type:</i> integer. Key. Nullable The person who manages the contract. Foreign key to the <code>ComplianceUser</code> table.
<code>Comments</code>	<i>Type:</i> text. Nullable Comments recorded about the contract.
<code>PeriodicPayment</code>	<i>Type:</i> currency. Nullable The price of periodic payments associated with this contract.
<code>PeriodicPaymentRateID</code>	<i>Type:</i> integer. Nullable The currency rate to be applied to the periodic payments figure above. Foreign key to the <code>CurrencyRate</code> table.
<code>VendorID</code>	<i>Type:</i> integer. Key. Nullable The vendor with which the contract agreement has been made. Foreign key to the <code>Vendor</code> table.
<code>MasterContractID</code>	<i>Type:</i> integer. Key. Nullable The contract that is the master of this contract. Foreign key to another contract in this <code>Contract</code> table.
<code>LocationID</code>	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise location associated with this contract. Foreign key to the <code>GroupEx</code> table.
<code>BusinessUnitID</code>	<i>Type:</i> text (max 128 characters). Key. Nullable

Database Column	Details
	Any enterprise corporate unit associated with this contract. Foreign key to the GroupEx table.
CostCenterID	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise cost center associated with this contract. Foreign key to the GroupEx table.
CategoryID	<i>Type:</i> text (max 128 characters). Key. Nullable Any category used in this enterprise that is associated with this contract. Foreign key to the GroupEx table.
LicenseDowngradeEnabled	<i>Type:</i> boolean If this field is set to <b>True</b> , licenses can inherit downgrade rights from this contract. If <b>False</b> (the default), licenses cannot inherit downgrade rights.
LicenseDowngradeToVersion	<i>Type:</i> boolean If this field is set to <b>True</b> , any license inheriting downgrade rights from this contract can cover all previous releases (with the same edition) of the primary application. If <b>False</b> , licenses inheriting downgrade rights may not downgrade to earlier versions.
LicenseDowngradeToEdition	<i>Type:</i> boolean If this field is set to <b>True</b> , any license inheriting downgrade rights from this contract can cover all lower editions of this version of the primary application. If <b>False</b> , licenses inheriting downgrade rights may not downgrade to lower editions.
LicenseUpgradeEnabled	<i>Type:</i> boolean If this field is set to <b>True</b> , licenses can inherit upgrade rights from this contract. If <b>False</b> (the default), licenses cannot inherit upgrade rights.
LicenseUpgradeToVersion	<i>Type:</i> boolean If this field is set to <b>True</b> , any license inheriting upgrade rights from this contract can cover all later releases (with the same edition) of the primary application. If <b>False</b> , licenses inheriting upgrade rights may not upgrade to later versions.
LicenseUpgradeUntilContractExpiry	<i>Type:</i> boolean If this field is set to <b>True</b> , any license inheriting upgrade rights from this contract can cover all later releases (with the same edition) of the primary application, as long as they were released before the expiry date ( <b>EndDate</b> )

Database Column	Details
	of the contract. If <code>False</code> , licenses inheriting upgrade rights do not take the application release date into consideration.
<code>GrantSecondUseToLicense</code>	<p><i>Type:</i> boolean</p> <p>If this field is set to <code>True</code>, licenses can inherit the right of second use from this contract. If <code>False</code> (the default), licenses cannot inherit the right of second use.</p>
<code>SecondUsageWorkLaptop</code>	<p><i>Type:</i> boolean</p> <p>If this field is set to <code>True</code>, any license inheriting from this contract will confer the right of second use on a work laptop. If <code>False</code>, licenses inheriting from this contract will not confer the right of second use.</p>
<code>SecondUsageAtHome</code>	<p><i>Type:</i> boolean</p> <p>If this field is set to <code>True</code>, any license inheriting from this contract will confer the right of second use on a home computer, for the same end-user as the primary end-user of the license entitlement consumed at work. If <code>False</code>, licenses inheriting from this contract will not confer the right of second use on a home computer.</p>
<code>GrantVirtualInstallsTo License</code>	<p><i>Type:</i> boolean</p> <p>If this field is set to <code>True</code>, licenses can inherit the virtual machine licensing rights from this contract. If <code>False</code> (the default), licenses cannot inherit virtual machine licensing rights.</p>
<code>CoverInstallsOnVirtual Machines</code>	<p><i>Type:</i> boolean</p> <p>If this field is set to <code>True</code>, any license inheriting virtual machine rights from this contract may be used to account for installations on virtual machines. If <code>False</code>, licenses inheriting virtual machine rights may only account for installations on physical machines.</p>
<code>LimitNumberOfVirtual Installs</code>	<p><i>Type:</i> boolean</p> <p>If this field is set to <code>True</code>, there is a limit to the number of virtual machine installations that may be covered by any license inheriting virtual machine rights from this contract. If this field is <code>False</code>, one license entitlement may cover any use on virtual machines (typically within one host computer).</p>
<code>NumberOfAllowedVirtual Installs</code>	<p><i>Type:</i> integer. Nullable</p> <p>If this contracts confers the right for an inheriting license to cover installations on virtual machines, this field specifies how many installations per host are allowed before an additional license entitlement (or point) is consumed.</p>
<code>LimitVirtualInstalls IncludesHost</code>	<p><i>Type:</i> boolean</p>

Database Column	Details
	If this field is <code>True</code> , the host operating system installations are included in the overall count of operating systems on the host when there is a limit on the number of allowed virtual installs for each license. If <code>False</code> , the host operating system is not considered when determining virtual install limits.
UseHostProcessor Information	<i>Type:</i> boolean  If virtual installs are allowed, this field controls whether host information is used by an inheriting license when calculating the license points consumed.
GrantLimitPointsToLicense	<i>Type:</i> boolean  If this field is set to <code>True</code> , licenses can inherit the right of multiple use from this contract. If <code>False</code> (the default), licenses cannot inherit the right of multiple use.
LimitNumberOf ApplicationsEach LicensePointCovers	<i>Type:</i> boolean  If this field is set to <code>True</code> , there is a limit, for any inheriting license, to the number of application installations allowed per license entitlement (or point). If this bit is <code>False</code> (the default), an inheriting license entitles you to any number of installations of software linked to this license on the one computer.
NumberOfApplication InstallsAllowedPer LicensePoint	<i>Type:</i> integer. Nullable  Where the previous field is set to <code>True</code> , this column defines the limited number of application installations allowed per entitlement (or point).
LimitNumberOfComputers UserLicenseCanBe InstalledOn	<i>Type:</i> boolean  If this field is set to <code>True</code> , there is a limit, for an inheriting user-based license, to the number of computers that an end-user can use per entitlement (or point) consumed. If this field is <code>False</code> (the default), a single end-user is entitled to install related software for his/her own use on any number of computers.
NumberOfComputers AllowedPerUserLicense Point	<i>Type:</i> integer. Nullable  Where the previous field is set to <code>True</code> , this column defines the limited number of application installations an end-user is allowed per entitlement (or point).
InitialPlatformQuantity	<i>Type:</i> integer. Nullable  The number of desktops covered by the Microsoft Enterprise Agreement platform license at the start of the agreement.
PurchaseProgramID	<i>Type:</i> integer. Nullable  Identifies the purchase program of contract. Foreign key to the <code>PurchaseProgram</code> table.
MSSelectApplication LevelID	<i>Type:</i> integer. Nullable

Database Column	Details
	Identifies the Microsoft Select level for applications. Foreign key to the <code>MSSelectLevel</code> table.
<code>MSSelectSystemLevelID</code>	<i>Type:</i> integer. Nullable Identifies the Microsoft Select level for systems. Foreign key to the <code>MSSelectLevel</code> table.
<code>MSSelectServerLevelID</code>	<i>Type:</i> integer. Nullable Identifies the Microsoft Select level for servers. Foreign key to the <code>MSSelectLevel</code> table.
<code>CreationUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
<code>CreationDate</code>	<i>Type:</i> datetime The date the record was created.
<code>UpdatedUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
<code>UpdatedDate</code>	<i>Type:</i> datetime. Nullable The date the record was last updated.
<code>TotalValue</code>	<i>Type:</i> currency. Nullable The total value of the contract.
<code>TotalValueRateID</code>	<i>Type:</i> integer. Nullable The rate for the total value. Foreign key to the <code>CurrencyRate</code> table.
<code>MonthlyValue</code>	<i>Type:</i> currency. Nullable The cost of the contract per month.
<code>MonthlyValueRateID</code>	<i>Type:</i> integer. Nullable The rate for the monthly cost. Foreign key to the <code>CurrencyRate</code> table.
<code>ProjectID</code>	<i>Type:</i> integer. Key. Nullable A project for the <code>Contract</code> . Foreign key to the <code>Project</code> table.
<code>SecurityTypeID</code>	<i>Type:</i> integer. Nullable The type of security to use when determining which operators have access to the contract. Foreign key to the <code>SecurityType</code> table.



Database Column	Details
PreviousContractID	<i>Type:</i> integer. Key. Nullable A link to a contract that this contract has replaced. Foreign key to the <code>Contract</code> table.
ContractStateID	<i>Type:</i> integer. Nullable The state of the contract. Foreign key to the <code>ContractState</code> table.
LastRenewedDate	<i>Type:</i> datetime. Nullable The date when the contract was last renewed.
LicenseConsumptionEnabled	<i>Type:</i> boolean If this field is set to <code>True</code> , licenses can inherit consumption rules from this contract. If <code>False</code> (the default), licenses cannot inherit consumption rules.
LicenseMobilityEnabled	<i>Type:</i> boolean If this field is set to <code>True</code> , licenses can inherit mobility rights from this contract. If <code>False</code> (the default), licenses cannot inherit mobility rights.
ProcessorLimitsEnabled	<i>Type:</i> boolean If this field is set to <code>True</code> , licenses can inherit rights related to processor limits from this contract. If <code>False</code> (the default), licenses cannot inherit rights related to processor limits.

## ContractNote Table

`ContractNote` stores a list of notes attached to a contract.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 149: Database columns for `ContractNote` table**

Database Column	Details
ContractNoteID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the contract note.
ContractID	<i>Type:</i> integer. Key

Database Column	Details
	The contract that the note is for. Foreign key to the <code>Contract</code> table.
<code>ShortDescription</code>	<i>Type:</i> text (max 100 characters) In the user interface, this maps to the contract reference to which the note relates.
<code>LongDescription</code>	<i>Type:</i> text. Nullable The content of the note.
<code>CreationUser</code>	<i>Type:</i> text (max 128 characters) The operator who created the note.
<code>CreationDate</code>	<i>Type:</i> datetime The date of creation of the note.
<code>UpdatedUser</code>	<i>Type:</i> text (max 128 characters) The operator who last updated the note.
<code>UpdatedDate</code>	<i>Type:</i> datetime The date of the last update to the note.

## ContractNotification Table

`ContractNotification` lists the notifications that need to be sent for a contract.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 150: Database columns for `ContractNotification` table**

Database Column	Details
<code>ContractNotificationID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for the contract notification.
<code>ContractID</code>	<i>Type:</i> integer. Key The contract this record is associated with. Foreign key to the <code>Contract</code> table.

Database Column	Details
NotificationInterval	<i>Type:</i> integer Defines how long before the contract notification is sent.
NotificationIntervalTypeID	<i>Type:</i> integer Defines the interval type used to work out how long before a contract notification is sent. Foreign key to the <code>IntervalType</code> table.
NotificationTypeID	<i>Type:</i> integer. Key Defines the type of notification (contract renewal or contract expiry). Foreign key to the <code>NotificationType</code> table.

## ContractNotificationResponsibility Table

`ContractNotificationResponsibility` keeps track of which responsibility groups need to be notified for contract expiry or renewals.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 151: Database columns for `ContractNotificationResponsibility` table**

Database Column	Details
ContractNotificationResponsibilityID	<i>Type:</i> integer. Key. Generated ID Unique identifier for contract notification responsibility groups.
ContractID	<i>Type:</i> integer. Key The contract generating notifications. Foreign key to the <code>Contract</code> table.
ResponsibilityTypeID	<i>Type:</i> integer. Key The responsibility type of the end-users receiving notifications about the contract. Foreign key to the <code>ResponsibilityType</code> table.
NotificationTypeID	<i>Type:</i> integer. Key The type of notification (renewal or expiry) that these responsibility groups should receive notifications for. Foreign key to the <code>NotificationType</code> table.

## ContractProperty Table

`ContractProperty` defines extra custom properties for contracts of a specified type.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 152: Database columns for `ContractProperty` table**

Database Column	Details
<code>ContractPropertyID</code>	Type: integer. Key. Generated ID Unique identifier for a contract property.
<code>ContractTypeID</code>	Type: integer. Key The type of contract to which this property may apply. Foreign key to the <code>ContractType</code> table.
<code>PropertyName</code>	Type: text (max 256 characters). Key The name of the custom property. A unique identifier for a resource string. Foreign key to the <code>ComplianceResourceString</code> table.
<code>CustomPropertyDisplayXMLID</code>	Type: integer. Nullable Reference to a record in the <code>CustomPropertyDisplayXML</code> table, describing how to show the property on a property dialog.

## ContractPropertyValue Table

For each contract, `ContractPropertyValue` stores the values for the custom properties defined in `ContractProperty`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 153: Database columns for ContractPropertyValue table**

Database Column	Details
ContractPropertyValueID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a property value.
ContractID	<i>Type:</i> integer. Key The individual contract to which this value applies. Foreign key to the <code>Contract</code> table.
ContractPropertyID	<i>Type:</i> integer. Key The property that contains this value. The contract should have the same type as the type associated with this property. Foreign key to the <code>ContractProperty</code> table.
PropertyValue	<i>Type:</i> text (max 4000 characters) The property value.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.

## ContractScopingData Table

`ContractScoping` links contracts to the enterprise groups to which they apply. Exactly one of `GroupExID` and `CategoryID` must be non-NULL.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 154: Database columns for ContractScopingData table**

Database Column	Details
ContractID	<i>Type:</i> integer. Key The contract the scoping applies to. Foreign key to the <code>Contract</code> table.
GroupExID	<i>Type:</i> text (max 128 characters). Key. Nullable The enterprise group that the scoping applies to. Foreign key to the <code>GroupEx</code> table.
CategoryID	<i>Type:</i> text (max 128 characters). Key. Nullable The category that the scoping applies to. Foreign key to the <code>Category</code> table.

## ContractSecurityUser Table

`ContractSecurityUser` stores a list of permissions granted to an operator for a contract with `Restricted` security.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 155: Database columns for ContractSecurityUser table**

Database Column	Details
ContractID	<i>Type:</i> integer. Key The contract with <code>Restricted</code> security. Foreign key to the <code>Contract</code> table.
ActionClassID	<i>Type:</i> integer. Key The type of permission being granted to the operator. Foreign key to the <code>ActionClass</code> table.
ComplianceOperatorID	<i>Type:</i> integer. Key The operator that the permission is granted to. Foreign key to the <code>ComplianceOperator</code> table.

## ContractState Table

`ContractState` holds the different states a contract can be in.

**Table 156: Database columns for `ContractState` table**

Database Column	Details
<code>ContractStateID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ContractState</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Draft</li> <li>• 2 = Suspended</li> <li>• 3 = Active</li> <li>• 4 = Archived</li> <li>• 5 = Cancelled</li> <li>• 6 = Expired</li> <li>• 7 = Completed.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a contract state. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the state resource string has no translation.</p>

## ContractStatus Table

`ContractStatus` is a static table listing all contract status values in the system.

**Table 157: Database columns for `ContractStatus` table**

Database Column	Details
<code>ContractStatusID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ContractStatus</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Active</li> <li>• 2 = Archived</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 3 = Draft</li> <li>• 4 = Suspended</li> <li>• 5 = Cancelled</li> <li>• 6 = Expired</li> <li>• 7 = Completed.</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a contract status. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the status resource string has no translation.</p>

## ContractType Table

`ContractType` is a static table listing all contract types in the system.

**Table 158: Database columns for `ContractType` table**

Database Column	Details
ContractTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ContractType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = General</li> <li>• 2 = Lease</li> <li>• 3 = Hardware Maintenance and Support</li> <li>• 4 = Software License</li> <li>• 5 = Software Maintenance and Support</li> <li>• 6 = Blanket purchase</li> <li>• 7 = Consulting services</li> <li>• 8 = Insurance</li> <li>• 9 = Rent</li> <li>• 10 = Subscription</li> </ul>



Database Column	Details
	<ul style="list-style-type: none"> <li>• 11 = Microsoft Business and Services Agreement</li> <li>• 12 = Microsoft Select License Agreement</li> <li>• 13 = Microsoft Select Plus Agreement</li> <li>• 14 = Microsoft Select License Enrollment</li> <li>• 15 = Microsoft Select Plus Affiliate</li> <li>• 16 = Microsoft Enterprise Agreement</li> <li>• 17 = Microsoft Enterprise Subscription Agreement.</li> </ul>
ContractTypeResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a contract type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
ContractTypeDefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>
XMLFile	<p><i>Type:</i> text. Nullable</p> <p>The layout of the property dialog for this type of computer, stored in XML format.</p>
PathResourceName	<p><i>Type:</i> text (max 256 characters)</p> <p>The unique name of the localizable resource string representing the parent contract type under which this contract type should be displayed. Foreign key to the <code>ComplianceResourceString</code> table.</p>
PathDefaultValue	<p><i>Type:</i> text (max 256 characters)</p> <p>The default parent contract type text to display if the resource string has no translation.</p>
PurchaseProgramID	<p><i>Type:</i> integer. Nullable</p> <p>The default purchase program for this contract type.</p>
CanCreate	<p><i>Type:</i> boolean</p> <p>Whether the end-user can manually create contracts of this type.</p>

## ContractUseRight Table

`ContractUseRight` contains licensing rules most of which can be set by PURL.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 159: Database columns for ContractUseRight table**

Database Column	Details
ContractUseRightID	Type: integer. Key. Generated ID A unique identifier
ContractID	Type: integer. Key A unique identifier for a contract.
ReassignmentTimeLimitAppliesDevice	Type: boolean If 1 then the license cannot be reassigned for some period of time (example is Microsoft 90 day rule)
ReassignmentTimeLimitAppliesUser	Type: boolean If 1 then the license cannot be reassigned for some period of time (example is Microsoft 90 day rule)
ReassignmentTimeLimitDevice	Type: integer. Nullable The period (in days) within which the license cannot be reassigned
ReassignmentTimeLimitUser	Type: integer. Nullable The period (in days) within which the license cannot be reassigned
LicenseMobilityApplies	Type: boolean 1 if eligible for bringing your own license to cloud environment
NumberOfOSEPerLicense	Type: integer. Nullable Number of OSE per license
NumberOfProcessorsPerOSE	Type: integer. Nullable Number of processors per OSE
TotalNumberOfCoresPerVMPerLicense	Type: integer. Nullable Total number of cores per VM per license
NumberOfCoresPerSocket	Type: integer. Nullable

Database Column	Details
	Number of cores per socket
ThirdPartyAccessAllowed	<p><i>Type:</i> boolean</p> <p>Access to applications is allowed to third party users. This field is defaulted to <code>True</code></p>
AllowExternalRoamingUse	<p><i>Type:</i> boolean. Nullable</p> <p>Set this field to <code>True</code> if license allows external roaming use. This field is defaulted to <code>False</code>. This is applicable for both device and user licenses and is related to virtual application access. If 1, this license will consume 1 entitlement per each user. If 0, this license will consume 1 license per each user device. And, if NULL, ignore virtual application access. This can be used in conjunction with <code>VirtualApplicationAccessMaximumUsagePeriod</code>.</p>
MeasurementDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date of the license measurment.</p>
ConsumptionUnit	<p><i>Type:</i> text. Nullable</p> <p>Unit description to describe the consumption amount.</p>
TargetOperatingSystemTypeID	<p><i>Type:</i> integer</p> <p>Type of Operating Systems to target</p>
VirtualApplicationAccessMaximumUsagePeriodDevice	<p><i>Type:</i> integer. Nullable</p> <p>This is a rule for virtual application access. This is used in conjunction with the <code>AllowExternalRoamingUse</code>. For Device licenses, a license will consume 1 entitlement per each user device when used in period specified here.</p>
VirtualApplicationAccessMaximumUsagePeriodUser	<p><i>Type:</i> integer. Nullable</p> <p>This is a rule for virtual application access. This is used in conjunction with the <code>AllowExternalRoamingUse</code>. For user licenses, if 1, this license will consume only when used in period specified here.</p>
AlwaysInstalled	<p><i>Type:</i> boolean</p> <p>If this field is <code>True</code>, this license is considered in to be used whenever it is allocated. If <code>False</code>, software usage is considered separately, and allocation merely defines the corporation's modelling of who is expected to consume entitlements.</p>
MinimumNumberOfLicensesPerVM	<p><i>Type:</i> integer</p>

Database Column	Details
	When licensing a Virtual Hardware System with a <code>MSServerCore</code> license ( <code>LicenseTypeID = 33</code> ), consume license entitlements as though the virtual machine had at least this number of virtual threads.
<code>AllowIBMPVUSubCapacityFromNonILMT</code>	<i>Type:</i> boolean  If the license does not use host processor information (not full capacity), set this field to <code>True</code> to allow non-ILMT sub-capacity PVU consumption calculations to be used.
<code>NumberOfAllowedProcessorsPerHost</code>	<i>Type:</i> integer. Nullable  This field specifies how many processors per host are allowed before an additional license entitlement (or point) is consumed. Null provides the default of 1. Zero provides unlimited.
<code>MinimumNumberOfProcessors</code>	<i>Type:</i> integer  The minimum number of processors that this license is for. This field is only used where the <code>SoftwareLicenseType</code> is <code>MSServerProcessor</code> ( <code>LicenseTypeID = 22</code> ).

## ContractUseRightIBM Table

`ContractUseRightIBM` contains IBM licensing rules most of which can be set by PURL.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 160: Database columns for `ContractUseRightIBM` table**

Database Column	Details
<code>ContractUseRightIBMid</code>	<i>Type:</i> integer. Key. Generated ID  A unique identifier
<code>ContractID</code>	<i>Type:</i> integer. Key  A unique identifier for a contract.
<code>PVULimitApplies</code>	<i>Type:</i> boolean  If 1 then PVU limits apply

Database Column	Details
PVULimit	Type: integer. Nullable PVU limit

## ContractVendor Table

ContractVendor stores the links between vendors and contracts.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 161: Database columns for ContractVendor table**

Database Column	Details
ContractVendorID	Type: integer. Key. Generated ID A unique identifier for the link.
ContractID	Type: integer. Key The contract that the vendor is linked to. Foreign key to the <code>Contract</code> table.
VendorID	Type: integer. Key. Nullable The vendor that the contract is linked to. Foreign key to the <code>Vendor</code> table.
ThirdParty	Type: boolean Set this field to <code>True</code> if this vendor is third-party.

## CurrencyRate Table

CurrencyRate stores the exchange rates assigned to any currency.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 162: Database columns for CurrencyRate table**

Database Column	Details
CurrencyRateID	<i>Type:</i> integer. Key. Generated ID Unique identifier for each record.
SnapshotID	<i>Type:</i> integer. Key Snapshot associated with this exchange rate. Foreign key to the CurrencyRateSnapshot table.
CurrencyID	<i>Type:</i> integer. Key Currency associated with this exchange rate. Foreign key to the Currency table.
Rate	<i>Type:</i> decimal Exchange rate assigned to the currency for the selected snapshot.
UpdatedUser	<i>Type:</i> text (max 256 characters). Nullable Operator who last modified the record.
UpdatedDate	<i>Type:</i> datetime. Nullable Date that the record was last modified.

## CurrencyRateSnapshot Table

Each record in CurrencyRateSnapshot represents a single currency snapshot.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 163: Database columns for CurrencyRateSnapshot table**

Database Column	Details
CurrencyRateSnapshotID	<i>Type:</i> integer. Key. Generated ID Unique identifier for this record.
SnapshotName	<i>Type:</i> text (max 256 characters)

Database Column	Details
	Name of the currency snapshot.
SnapshotResourceID	<i>Type:</i> text (max 64 characters). Nullable The resource string containing the name of the snapshot to display on the user interface.
SnapshotDate	<i>Type:</i> datetime. Nullable Start date of the currency snapshot.
SnapshotReferenceCurrencyID	<i>Type:</i> integer. Nullable Reference currency used for this snapshot. Foreign key to the <code>Currency</code> table.
IsStandardRateSnapshot	<i>Type:</i> boolean. Key Set to <code>True</code> if this is the default standard rate snapshot, which is created for each FNMP installation.
UpdatedUser	<i>Type:</i> text (max 256 characters). Nullable Operator who last modified this record.
UpdatedDate	<i>Type:</i> datetime. Nullable Date this record was last modified.

## CustomPropertyDisplayXML Table

`CustomPropertyDisplayXML` stores XML snippets with layout information for custom properties. The XML snippets in this table will be inserted into the default XML layout for the appropriate property dialog. Storing snippets in this table, rather than manually updating the default XML layout, ensures that custom properties will continue to be applied even after upgrading the product (since during a product upgrade, we typically overwrite all property display XML layout with the new defaults for that version of the product).

**Table 164: Database columns for CustomPropertyDisplayXML table**

Database Column	Details
CustomPropertyDisplayXMLID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this XML snippet.
XMLSnippet	<i>Type:</i> text An XML snippet that describes how to show this property in the properties dialog.

Database Column	Details
InsertXPath	<i>Type:</i> text XPath which selects an XML node where the snippet will be inserted.
XMLInsertTypeID	<i>Type:</i> integer How to insert this property at the selected XPath node. Foreign key to the <code>XMLInsertType</code> table.
InsertOrder	<i>Type:</i> integer The order in which to insert the XML snippet for this property into the XML layout file. If this value is higher than another, it will be inserted after it. Useful when the XML snippet for this property is to be inserted inside another - for instance, if a property creates a tab or group.

## DisplayXML Table

The static `DisplayXML` table stores the default XML code representing the property dialog layout for non-type-specific objects such as purchase orders, vendors and evidence. The XML files for type-specific entities (such as assets) are stored in the static type tables (such as `AssetType`) for those objects.

**Table 165: Database columns for `DisplayXML` table**

Database Column	Details
XMLType	<i>Type:</i> text (max 30 characters). Key A unique identifier for the type of object associated with the XML. Possible values are: <ul style="list-style-type: none"> <li>• Contract (not in use any longer - the contract XML files are now stored in <code>ContractType</code>)</li> <li>• Vendor</li> <li>• VendorContact</li> <li>• PurchaseOrder</li> <li>• PurchaseOrderDetail</li> <li>• SoftwareTitle</li> <li>• FileEvidence</li> <li>• InstallerEvidence</li> <li>• User</li> </ul>



Database Column	Details
	<ul style="list-style-type: none"> <li>• TermAndCondition</li> <li>• Operator</li> <li>• LicensePointsRuleSet.</li> </ul>
XMLFile	<p>Type: text. Nullable</p> <p>The layout of the property dialog for this type of entity, stored in XML format.</p>

## Document Table

The `Document` table stores details of documents or files relating to assets, contracts, purchase orders, licenses and terms and conditions.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 166: Database columns for Document table**

Database Column	Details
DocumentID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for the document.</p>
DocumentTypeID	<p>Type: integer</p> <p>The way that the document is stored and referenced. Foreign key to the <code>DocumentType</code> table.</p>
DocumentName	<p>Type: text (max 500 characters)</p> <p>The name of the document.</p>
DocumentFile	<p>Type: image. Nullable</p> <p>The binary data for the document (if it is stored in the FlexNet Manager Suite database).</p>
OpenWith	<p>Type: text (max 500 characters). Nullable</p> <p>The program to attempt to open the document with.</p>
DocumentDescription	<p>Type: text (max 3000 characters)</p>

Database Column	Details
	A description of the document.
PhysicalLocation	<p><i>Type:</i> text (max 500 characters). Nullable</p> <p>Physical location of a (possibly hard) copy of this document. NOTE: for compatibility with the FlexNet Manager Suite console, when the document type is 3 (Reference), the <code>DocumentName</code> column should be used instead, and this field set to null.</p>
DocumentSize	<p><i>Type:</i> integer. Nullable</p> <p>Document size in bytes.</p>
ContentType	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The MIME-type of the document file.</p>
AssetID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The asset to which this document may be linked. Foreign key to the <code>Asset</code> table.</p>
PurchaseOrderID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The purchase order to which the document may be linked. Foreign key to the <code>PurchaseOrder</code> table.</p>
PurchaseOrderDetailID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The purchase order detail (or PO line) to which the document may be linked. Foreign key to the <code>PurchaseOrderDetail</code> table.</p>
ContractID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The contract to which the document may be linked. Foreign key to the <code>Contract</code> table.</p>
SoftwareLicenseID	<p><i>Type:</i> integer. Nullable</p> <p>The license to which the document may be linked. Foreign key to the <code>SoftwareLicense</code> table.</p>
ComplianceUserID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The end-user to which the document may be linked. Foreign key to the <code>ComplianceUser</code> table.</p>
AttachDate	<p><i>Type:</i> datetime</p> <p>The date and time this document was linked.</p>
UserName	<i>Type:</i> text (max 256 characters)

Database Column	Details
	Operator who created the link between this document and the other object.
DocumentNoteID	<i>Type:</i> integer. Key. Nullable The note to which this document may be linked. Foreign key to the <code>DocumentNote</code> table.
ContractNoteID	<i>Type:</i> integer. Key. Nullable The contract note to which this document may be linked. Foreign key to the <code>ContractNote</code> table.
TermAndConditionID	<i>Type:</i> integer. Key. Nullable The term/condition to which this document may be linked. Foreign key to the <code>TermAndCondition</code> table.
SecurityTypeID	<i>Type:</i> integer. Key. Nullable Security type for this document (role-based or individual access). Foreign key to the <code>SecurityType</code> table.
FileType	<i>Type:</i> text (max 20 characters). Nullable The type of the file that has been uploaded, if any. This is used to provide full-text indexing.

## DocumentHistory Table

The `DocumentHistory` table stores history of documents or files relating to assets, contracts, purchase orders, licenses, and terms and conditions.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 167: Database columns for `DocumentHistory` table**

Database Column	Details
DocumentHistoryID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the document history.
DocumentID	<i>Type:</i> integer. Key

Database Column	Details
	The corresponding document. Foreign key to the <code>Document</code> table.
<code>DocumentTypeID</code>	<p><i>Type:</i> integer</p> <p>The way that the document is stored and referenced. Foreign key to the <code>DocumentType</code> table.</p>
<code>DocumentName</code>	<p><i>Type:</i> text (max 500 characters)</p> <p>The name of the document.</p>
<code>DocumentFile</code>	<p><i>Type:</i> image. Nullable</p> <p>The binary data for the document (if it is stored in the FlexNet Manager Suite database).</p>
<code>OpenWith</code>	<p><i>Type:</i> text (max 500 characters). Nullable</p> <p>The program to attempt to open the document with.</p>
<code>DocumentDescription</code>	<p><i>Type:</i> text (max 3000 characters)</p> <p>A description of the document.</p>
<code>PhysicalLocation</code>	<p><i>Type:</i> text (max 500 characters). Nullable</p> <p>Physical location of a (possibly hard) copy of this document. NOTE: for compatibility with the FlexNet Manager Suite console, when the document type is 3 (Reference), the <code>DocumentName</code> column should be used instead, and this field set to null.</p>
<code>DocumentSize</code>	<p><i>Type:</i> integer. Nullable</p> <p>Document size in bytes.</p>
<code>ContentType</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The MIME-type of the document file.</p>
<code>UserName</code>	<p><i>Type:</i> text (max 256 characters)</p> <p>Operator who created the link between this document and the other object.</p>
<code>AttachDate</code>	<p><i>Type:</i> datetime</p> <p>The date and time this document was linked to the other object.</p>
<code>FileType</code>	<p><i>Type:</i> text (max 20 characters). Nullable</p> <p>The type of the file that has been uploaded, if any. This is used to provide full-text indexing.</p>

## DocumentNote Table

`DocumentNote` stores a list of notes attached to a document. The document itself is attached to a contract.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 168: Database columns for `DocumentNote` table**

Database Column	Details
<code>DocumentNoteID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for the document note.
<code>DocumentID</code>	<i>Type:</i> integer The document that the note is for. Foreign key to the <code>Document</code> table.
<code>ShortDescription</code>	<i>Type:</i> text (max 100 characters) In the user interface, this maps to the document reference to which the note relates.
<code>LongDescription</code>	<i>Type:</i> text. Nullable The content of the note.
<code>CreationUser</code>	<i>Type:</i> text (max 128 characters) The operator who created the note.
<code>CreationDate</code>	<i>Type:</i> datetime The date of creation of the note.
<code>UpdatedUser</code>	<i>Type:</i> text (max 128 characters) The operator who last updated the note.
<code>UpdatedDate</code>	<i>Type:</i> datetime The date of the last update to the note.

## DocumentType Table

`DocumentType` is a static value listing the alternative ways that a document can be saved in the database.

**Table 169: Database columns for `DocumentType` table**

Database Column	Details
<code>DocumentTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>DocumentType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Document upload</li> <li>• 2 = File location</li> <li>• 3 = Physical location</li> <li>• 4 = URL.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a document type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## Event Table

The `Event` table stores errors and events processed by the beacon, devices, rules etc.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 170: Database columns for `Event` table**

Database Column	Details
<code>EventID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>Synthetic key for this table.</p>
<code>ActivityID</code>	<p><i>Type:</i> integer. Key</p> <p>Foreign key to the <code>Activity</code> table.</p>
<code>EventUID</code>	<p><i>Type:</i> unique identifier. Key</p> <p>UID to uniquely identify the event.</p>

Database Column	Details
EventTypeID	Type: integer. Key Foreign key to the EventType table.
CreationDate	Type: datetime Date and time (UTC) when the Event was created.
SessionUID	Type: unique identifier. Key. Nullable UID to uniquely identify the the session.

## EventLogCategory Table

The EventLogCategory table holds the different categories of events created by the system.

**Table 171: Database columns for EventLogCategory table**

Database Column	Details
EventLogCategoryID	Type: integer. Key. Generated ID A unique identifier for each EventLogCategory. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>1 = Email Notification.</li> </ul>
ResourceName	Type: text (max 256 characters). Key The unique name of the localizable resource string representing a event category. Foreign key to the ComplianceResourceString table.
DefaultValue	Type: text (max 100 characters) A description of the event category.

## EventLogDetail Table

The EventLogDetail table holds details of the events created by the system.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 172: Database columns for EventLogDetail table**

Database Column	Details
EventLogDetailID	<i>Type:</i> integer. Key. Generated ID A unique identifier for an event detail.
EventLogSummaryID	<i>Type:</i> integer. Key The unique identifier for an event. Foreign key to the EventLogSummary table.
EventLogLevelID	<i>Type:</i> integer. Key The level of event. Foreign key to the EventLogLevel table.
MessageTime	<i>Type:</i> datetime. Key The time that the event was raised.
Message	<i>Type:</i> text (max 256 characters) The brief event message.
Details	<i>Type:</i> text. Nullable The full event message.
ParentEventLogDetailID	<i>Type:</i> integer. Key. Nullable The parent event log detail. Foreign key to another event log detail in this same EventLogDetail table.

## EventLogLevel Table

The EventLogLevel table holds the different levels of events created by the system.

**Table 173: Database columns for EventLogLevel table**

Database Column	Details
EventLogLevelID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each EventLogLevel. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Information</li> <li>• 2 = Warning</li> <li>• 3 = Error.</li> </ul>



Database Column	Details
	<ul style="list-style-type: none"> <li>4 = Performance.</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing an event level. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) A description of the event level.

## EventLogStatus Table

The `EventLogStatus` table holds the different statuses of events created by the system.

**Table 174: Database columns for EventLogStatus table**

Database Column	Details
EventLogStatusID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>EventLogStatus</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>1 = In Progress</li> <li>2 = Success</li> <li>3 = Failed.</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing an event status. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) A description of the event status.

## EventLogSummary Table

The `EventLogSummary` table holds the top level summary of events created by the system.

**Table 175: Database columns for EventLogSummary table**

Database Column	Details
EventLogSummaryID	Type: integer. Key. Generated ID A unique identifier for an event.
StartTime	Type: datetime. Key The time that the event started.
EndTime	Type: datetime. Key. Nullable The time that the event finished.
EventLogCategoryID	Type: integer. Key The category of event. Foreign key to the EventLogCategory table.
EventName	Type: text (max 128 characters) Brief description of the event.
EventLogStatusID	Type: integer. Key The status of the event. Foreign key to the EventLogStatus table.

## EventParameter Table

The EventParameter table stores the links between Activities and EventParameterTypes.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database TenantID has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 176: Database columns for EventParameter table**

Database Column	Details
EventParameterID	Type: integer. Key. Generated ID Primary key for the EventParameter table
EventID	Type: integer. Key A link to the Event table
EventParameterTypeID	Type: integer

Database Column	Details
	A link the the <code>EventParameterType</code> table. this value specifies which kind of object the <code>EventParameter</code> is linked to.
Value	<i>Type:</i> text stores the value of this parameter.

## EventParameterType Table

The `EventParameterType` table stores details about the different types of Event Parameters.

**Table 177: Database columns for EventParameterType table**

Database Column	Details
<code>EventParameterTypeID</code>	<i>Type:</i> integer. Key. Generated ID Synthetic key for this table.
<code>EventParameterTypeName</code>	<i>Type:</i> text (max 256 characters). Key A short piece of text representing the Event Parameter. Internal use only- not to be displayed to the operator.
<code>IsResourceString</code>	<i>Type:</i> boolean A short piece of text representing the Event Parameter. Internal use only- not to be displayed to the operator.

## EventSeverity Table

`EventSeverity` is a static table listing all of the severity levels that an event type can have.

**Table 178: Database columns for EventSeverity table**

Database Column	Details
<code>EventSeverityID</code>	<i>Type:</i> integer. Key. Generated ID
<code>ResourceName</code>	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing the <code>EventSeverity</code> record. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DefaultValue</code>	<i>Type:</i> text (max 256 characters)

Database Column	Details
	The text to display if the state resource string has no translation.

## EventTarget Table

The `EventTarget` table stores the links between Activities and other tables in the database.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 179: Database columns for EventTarget table**

Database Column	Details
EventID	Type: integer. Key Link to the <code>Event</code> table
TargetTypeID	Type: integer. Key A link to the <code>TargetType</code> table. this value specifies which kind of object the Event is linked to.
TargetUID	Type: unique identifier. Key. Nullable used to attach the <code>Event</code> to its target. The target table depends on the <code>TargetTypeID</code> of the linked <code>EventType</code> .
TargetID	Type: integer. Nullable ID of the target. Referenced if the UID is not available.
TargetName	Type: text (max 128 characters). Nullable TargetName used to record the name of the target. Can be used when the UID or ID is not available.

## EventType Table

The `EventType` table stores details about the different types of Events.

**Table 180: Database columns for EventType table**

Database Column	Details
EventTypeID	<i>Type:</i> integer. Key. Generated ID Synthetic key for this table.
EventTypeName	<i>Type:</i> text (max 256 characters). Key Short text representing the Event Type. Internal use only- not to be displayed to the operator.
EventSeverityID	<i>Type:</i> integer The severity of the Event. 1 = information, 2 = warning, 3 = error, 4 = critical.
EventMessageResource	<i>Type:</i> text (max 256 characters) A resource name used to look up a description for this Event
EventTypeStatusID	<i>Type:</i> integer. Key Foreign key to the EventTypeStatus table
ActivityTypeID	<i>Type:</i> integer. Key Foreign key to the ActivityType table

## EventTypeStatus Table

The EventTypeStatus table stores progress stages for different processes.

**Table 181: Database columns for EventTypeStatus table**

Database Column	Details
EventTypeStatusID	<i>Type:</i> integer. Key. Generated ID Auto-generated status ID
EventTypeStatus ResourceName	<i>Type:</i> text (max 255 characters). Key Status name resource name
EventTypeStatusDefault Value	<i>Type:</i> text (max 255 characters). Nullable Default value for status

# ILMTPVUCounts Table

This table allows the summarised PVU sub capacity numbers to be imported from ImportedILMTPVUCounts.”.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 182: Database columns for ILMTPVUCounts table**

Database Column	Details
ILMTPVUCountsTableID	<i>Type:</i> integer. Key. Generated ID The ID of the ILMTPVUCounts Table
ComplianceComputerID	<i>Type:</i> integer. Key ID from the ComplianceComputer table.
TitleName	<i>Type:</i> text (max 512 characters). Key The name of the title these points apply to.
Publisher	<i>Type:</i> text (max 254 characters). Key The name of the publisher of the title these points apply to.
SubCapacityCores	<i>Type:</i> integer The number of sub-capacity licensable cores for the license on the computer.
FullCapacityCores	<i>Type:</i> integer The number of full-capacity licensable cores for the license on the computer.
SubCapacityPVU	<i>Type:</i> integer The number of sub-capacity PVU counts consumed for the license on the computer.
FullCapacityPVU	<i>Type:</i> integer The number of full-capacity PVU counts consumed for the license on the computer.
PeakSubCapacityPVU	<i>Type:</i> integer The peak number of sub-capacity PVU counts consumed for the license on the computer.

Database Column	Details
PeakFullCapacityPVU	<p>Type: integer</p> <p>The peak number of full-capacity PVU counts consumed for the license on the computer.</p>

## ImportResolverErrorResult Table

The `ImportResolverErrorResult` table stores all resolver error message



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 183: Database columns for `ImportResolverErrorResult` table**

Database Column	Details
ImportResolverErrorResultID	<p>Type: integer. Key. Generated ID</p> <p>Auto-generated ID for <code>ImportResolverErrorResult</code> table</p>
FileName	<p>Type: text (max 255 characters)</p> <p>Name of the file</p>
DateCreated	<p>Type: datetime</p> <p>Date time where file was resolved.</p>
ErrorMessage	<p>Type: text. Nullable</p> <p>error message</p>
ImportResolverTypeID	<p>Type: integer. Key</p> <p>Foreign key to the <code>ImportResolverType</code> table</p>

## ImportResolverType Table

The `ImportResolverType` table stores all the resolver types.

**Table 184: Database columns for ImportResolverType table**

Database Column	Details
ImportResolverTypeID	<i>Type:</i> integer. Key. Generated ID Auto-generated ID for ImportResolverType table
ImportResolverTypeName	<i>Type:</i> text (max 255 characters). Key Name of the resolver
ImportResolverTypeResource	<i>Type:</i> text (max 256 characters) A resource name used to look up a description for this resolver type

## InstalledSoftwareAttribute Table

InstalledSoftwareAttribute stores the attribute values for each installation of an application. Reserved for future expansion.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 185: Database columns for InstalledSoftwareAttribute table**

Database Column	Details
InstalledSoftwareID	<i>Type:</i> integer. Key The installation whose attribute value is being stored. Foreign key to the InstalledSoftware table.
AttributeID	<i>Type:</i> integer. Key The attribute whose value is being stored. Foreign key to the Attribute table.
Value	<i>Type:</i> text (max 400 characters) The value of this attribute of the installed application.

## Instance Table

Instance stores information about database instances.





**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 186: Database columns for Instance table**

Database Column	Details
InstanceID	Type: integer. Key. Generated ID A unique identifier for an instance.
ParentInstanceID	Type: integer. Key. Nullable The parent of the instance. Foreign key to another instance in the <code>Instance</code> table.
InstalledSoftwareID	Type: integer. Key. Nullable The installation associated with the instance. Foreign key to the <code>InstalledSoftware</code> table.
InstanceTypeID	Type: integer The type of this database instance. Foreign key to the <code>InstanceType</code> table
ComplianceComputerID	Type: integer. Key The host server running this database instance. Foreign key to the <code>ComplianceComputer</code> table.
SoftwareTitleID	Type: integer. Key The instance's application. Foreign key to the <code>SoftwareTitle</code> table
InstanceName	Type: text (max 256 characters). Key. Nullable The name of the database instance.
SerialNo	Type: text (max 256 characters). Nullable The serial number of the database instance.
InstallationPath	Type: text (max 512 characters). Nullable The installation path of the database instance.
BusinessApplicationName	Type: text (max 512 characters). Nullable The business application that uses the database instance.
IsLicensable	Type: boolean

Database Column	Details
	Set this to <code>False</code> if this instance does not require a license. The default is <code>True</code> , which means a license is required.
<code>IsLicensableForLicenseReconciliation</code>	<p><i>Type:</i> boolean</p> <p>Set this to <code>True</code> if this instance should be included in license reconciliation. <code>False</code> means that this instance will not be accounted for in license reconciliation.</p>
<code>NeverDelete</code>	<p><i>Type:</i> boolean</p> <p>When a computer does not return any inventory for a specified period of time, it may be deleted. Set this field to <code>True</code> to ensure that the instance record does not get deleted when there is no inventory.</p>
<code>SoftwareLicenseID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The software license covering this instance. Foreign key to the <code>SoftwareLicense</code> table.</p>
<code>UsedInInventory</code>	<p><i>Type:</i> boolean</p> <p>If the inventory importer detects that this database instance is used, it will set this field to <code>True</code>.</p>
<code>UsedOverride</code>	<p><i>Type:</i> boolean. Nullable</p> <p>An operator may manually specify whether this database instance is to be considered used (set this field to <code>True</code>), or not (set this field to <code>False</code>). This overrides the importer result (<code>UsedInInventory</code>) described above.</p>
<code>InventorySourceTypeID</code>	<p><i>Type:</i> integer</p> <p>Whether this instance has ever been reported in inventory, or has been manually created and maintained. Foreign key to the <code>ComplianceComputerInventorySourceType</code> table.</p>
<code>AuditEvidenceDate</code>	<p><i>Type:</i> datetime. Nullable</p> <p>Date and time the Oracle LMS audit evidence was collected by Flexera Inventory Manager</p>
<code>CreationUser</code>	<p><i>Type:</i> text (max 256 characters)</p> <p>The operator who created the database instance record.</p>
<code>CreationDate</code>	<p><i>Type:</i> datetime</p> <p>The date and time when this instance record was created.</p>
<code>UpdatedUser</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p>

Database Column	Details
	The operator who most recently updated the database instance record.
UpdatedDate	Type: datetime The date and time when this instance record was last updated.

## InstanceAttribute Table

InstanceAttribute stores the attribute values for each installed database instance.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 187: Database columns for InstanceAttribute table**

Database Column	Details
InstanceID	Type: integer. Key The database instance whose attribute value is being stored. Foreign key to the Instance table.
AttributeID	Type: integer. Key The attribute whose value is being stored. Foreign key to the Attribute table.
Value	Type: text (max 400 characters) The value of this attribute of the database instance.

## InstanceEnvironment Table

InstanceEnvironment is a static table listing the possible environments in which database instances may be deployed. For some vendors, the environment affects the costs of licensing the database instance.

**Table 188: Database columns for InstanceEnvironment table**

Database Column	Details
InstanceEnvironmentID	Type: integer. Key. Generated ID

Database Column	Details
	<p>A unique identifier for an <code>InstanceEnvironment</code>. Possible values and the corresponding default names are:</p> <ul style="list-style-type: none"> <li>• 1 = Development</li> <li>• 2 = Test</li> <li>• 3 = Staging</li> <li>• 4 = Production</li> <li>• 5 = Other.</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an instance environment. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the environment resource string has no translation.</p>

## InstancePropertyValue Table

For each instance, `InstancePropertyValue` stores the values for the custom properties defined in `InstanceTypeProperty`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 189: Database columns for `InstancePropertyValue` table**

Database Column	Details
InstancePropertyValueID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for a property value.</p>
InstanceID	<p><i>Type:</i> integer. Key</p> <p>The instance associated with this property. Foreign key to the <code>Instance</code> table.</p>
InstanceTypePropertyID	<p><i>Type:</i> integer. Key</p>

Database Column	Details
	The property whose value is being stored. The type of the instance should match the type that the property is associated with. Foreign key to the <code>InstanceTypeProperty</code> table.
<code>PropertyValue</code>	<i>Type:</i> text (max 4000 characters) The value of the property.
<code>CreationUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
<code>CreationDate</code>	<i>Type:</i> datetime The date and time when the record was created.
<code>UpdatedUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
<code>UpdatedDate</code>	<i>Type:</i> datetime. Nullable The date and time when the record was last updated.

## InstanceRole Table

`InstanceRole` is a static table listing the possible roles of database instances. For some vendors, the role of the database instance affects the costs of licensing.

**Table 190: Database columns for `InstanceRole` table**

Database Column	Details
<code>InstanceRoleID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for an <code>InstanceRole</code> . Possible values and the corresponding default names are: <ul style="list-style-type: none"> <li>• 1 = None</li> <li>• 2 = Backup</li> <li>• 3 = Failover</li> <li>• 4 = Mirroring</li> <li>• 5 = Standby</li> <li>• 6 = Other</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>7 = Primary.</li> </ul>
ResourceName	<p>Type: text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an instance role. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p>Type: text (max 100 characters)</p> <p>The text to display if the role resource string has no translation.</p>

## InstanceType Table

`InstanceType` is a static table listing the possible types of database instance.

**Table 191: Database columns for `InstanceType` table**

Database Column	Details
InstanceTypeID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for an <code>InstanceType</code>. Possible values and the corresponding default names are:</p> <ul style="list-style-type: none"> <li>1 = General (for non-Oracle applications)</li> <li>2 = Oracle</li> <li>3 = Application (for instances created for non-Oracle applications manually flagged as Oracle).</li> <li>4 = Oracle EBS Server</li> <li>5 = Oracle EBS Module</li> </ul>
ResourceName	<p>Type: text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an instance type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p>Type: text (max 100 characters)</p> <p>The text to display if the instance type resource string has no translation.</p>
XMLFile	<p>Type: text. Nullable</p> <p>The layout of the property dialog for this type of instance, stored in XML format.</p>

# InstanceTypeProperty Table

`InstanceTypeProperty` defines extra custom properties for instances of the specified type.

**Table 192: Database columns for `InstanceTypeProperty` table**

Database Column	Details
<code>InstanceTypePropertyID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each property.
<code>PropertyName</code>	<i>Type:</i> text (max 256 characters). Key The name of the property.
<code>InstanceTypeID</code>	<i>Type:</i> integer. Key Foreign key to the <code>InstanceType</code> table.
<code>CustomPropertyDisplayXMLID</code>	<i>Type:</i> integer. Nullable Foreign key to a record in the <code>CustomPropertyDisplayXML</code> table, describing how to show the property on a property dialog.

# InstanceUser Table

`InstanceUser` links end-users in `LicenseUser` with a particular instance of a database for license counting purposes.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 193: Database columns for `InstanceUser` table**

Database Column	Details
<code>InstanceID</code>	<i>Type:</i> integer. Key The instance used by the end-user. Foreign key to a database instance in the <code>Instance</code> table.
<code>LicenseUserID</code>	<i>Type:</i> integer. Key The end-user using the instance. Foreign key to the account name in the <code>LicenseUser</code> table.

Database Column	Details
Quantity	<p><i>Type:</i> integer</p> <p>The number of actual end-users of the database instance logging in to the Oracle database through this account. For example, if there is one “Shop Floor” account for all fork lift drivers, this field stores the number of individual drivers that must be accounted for.</p>
AccountStatus	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The current status of the end-user account.</p>
CreationDate	<p><i>Type:</i> datetime. Nullable</p> <p>Date and time when the end-user was created.</p>
LastLogonDate	<p><i>Type:</i> datetime. Nullable</p> <p>Date and time when the end-user last logged on.</p>
DefaultTablespace	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The default tablespace for an Oracle user.</p>
TempTablespace	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The temporary tablespace for an Oracle user.</p>
IsManualUser	<p><i>Type:</i> boolean</p> <p>Whether or not the user was created manually (or through Oracle).</p>

## IntervalType Table

`IntervalType` stores the types of interval used by schedules and by terms and conditions.

**Table 194: Database columns for `IntervalType` table**

Database Column	Details
IntervalTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>IntervalType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Day</li> <li>• 2 = Week</li> <li>• 3 = Month.</li> </ul>



Database Column	Details
ResourceName	Type: text (max 256 characters). Key The unique name of the localizable resource string representing an interval type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	Type: text (max 100 characters) The text to display if the type resource string has no translation.

## LicenseUser Table

The `LicenseUser` table lists account names (for end-users and other resources) that have been extracted from other products (such as Oracle databases). These external accounts cannot be reconciled with the end-users listed in the `ComplianceUser` table. Nevertheless, these accounts can be very important for licensing costs.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 195: Database columns for `LicenseUser` table**

Database Column	Details
LicenseUserID	Type: integer. Key. Generated ID A unique identifier for an external end-user.
LicenseUserLogin	Type: text (max 400 characters). Key The user login extracted from the original listing (for example, from an Oracle database).
Description	Type: text (max 400 characters) The description is usually a group name.
EmployeeNumber	Type: text (max 256 characters). Nullable The employee number of the external end-user.
FirstName	Type: text (max 256 characters). Nullable The first name of the end-user extracted from the original listing.
LastName	Type: text (max 256 characters). Nullable

Database Column	Details
	The last name of the end-user extracted from the original listing.
Email	Type: text (max 400 characters). Nullable The email of the end-user extracted from the original listing.
SAPClientCode	Type: text (max 2 characters). Nullable The end-user's SAP client code, where applicable.
SAPInstallationNumber	Type: text (max 10 characters). Nullable The end-user's SAP installation number, where applicable.
CostCenter	Type: text (max 128 characters). Nullable The SAP cost center that the end-user belongs to
LicenseUserID	Type: integer The type of external end-user. Foreign key to the <code>LicenseUserType</code> table.

## LicenseUserConnection Table

`ComplianceUserConnection` stores a link between external end-users in `LicenseUser` which have been reported in inventory, and external IDs which can be used to identify them in their inventory sources. End-users reported in multiple inventory sources will appear multiple times in this table.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 196: Database columns for `LicenseUserConnection` table**

Database Column	Details
LicenseUserID	Type: integer. Key A unique identifier for the external end-user. Foreign key to the <code>LicenseUser</code> table.
ComplianceConnectionID	Type: integer. Key. Nullable The inventory source where the end-user was reported. Foreign key to the <code>ComplianceConnection</code> table.

Database Column	Details
ExternalID	<p>Type: big integer. Key</p> <p>A (hopefully unique) identifier for the end-user in the external inventory source.</p>

## LicenseUserExcluded Table

Similarly to the `LicenseUser` table, `LicenseUserExcluded` lists account names extracted from other products (such as Oracle databases); but these accounts are to be excluded from license counts. The accounts are listed in full here since it is possible that they do not already appear in the `LicenseUser` table. Any that do appear in both tables, matched on the login names, are excluded from license counts.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 197: Database columns for `LicenseUserExcluded` table**

Database Column	Details
LicenseUserExcludedID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for an excluded end-user.</p>
LicenseUserLogin	<p>Type: text (max 400 characters). Key</p> <p>The user login extracted from the original listing (for example, from an Oracle database). For the account to be excluded from license counts, this must exactly match a <code>LicenseUserLogin</code> from the <code>LicenseUser</code> table.</p>
DefaultQuantity	<p>Type: integer</p> <p>The number of actual users of the database instance logging in through this account. For example, a "SYSTEM" account may allow for a number of administrators to log in. In this table, the default quantity is zero. If this field is non-zero and the end-user matches a <code>LicenseUser</code> record, then in some cases, we may exclude this number of end-users from license counting, but include any further accounts covered by the <code>LicenseUser</code> record.</p>

## LicenseUserType Table

`LicenseUserType` is a static table listing possible types of external end-users (in the `LicenseUser` table).

**Table 198: Database columns for LicenseUserType table**

Database Column	Details
LicenseUserTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>LicenseUserType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Default</li> <li>• 2 = Developer.</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an external end-user type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## LogFile Table

The `LogFile` table stores all the log file



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 199: Database columns for LogFile table**

Database Column	Details
LogFileID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>Primary key of the table <code>LogFile</code></p>
SessionUID	<p><i>Type:</i> unique identifier. Key</p> <p>Identified of the file</p>
TaskStepID	<p><i>Type:</i> integer. Key. Nullable</p> <p>Foreign key to the <code>TaskStep</code> table</p>
FileContent	<p><i>Type:</i> image</p>

Database Column	Details
	holds the log file content
FileExtension	Type: text (max 10 characters) Extension of the file

## MSEAARLSoftwareTitleEdition Table

MSEAARLSoftwareTitleEdition contains a list of available product editions for a Microsoft Enterprise Agreement.

**Table 200: Database columns for MSEAARLSoftwareTitleEdition table**

Database Column	Details
SoftwareRecognitionID	Type: text (max 32 characters). Key The factory unique ID (an MD5 digest) for the product edition in the Application Recognition Library.
IsPlatform	Type: boolean Whether this edition should be covered by the platform license.

## MSSelectLevel Table

MSSelectLevel is a static table listing all Microsoft Select price levels.

**Table 201: Database columns for MSSelectLevel table**

Database Column	Details
MSSelectLevelID	Type: integer. Key. Generated ID A unique identifier for each MSSelectLevel. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = A</li> <li>• 2 = B</li> <li>• 3 = C</li> <li>• 4 = D</li> </ul>
ResourceName	Type: text (max 256 characters). Key

Database Column	Details
	The unique name of the localizable resource string representing a price level. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DefaultValue</code>	<i>Type:</i> text (max 16 characters) The text to display if the price level resource string has no translation.
<code>NumberOfPoints</code>	<i>Type:</i> integer The number of points that must be purchased to achieve the price level.

## MSSelectPool Table

`MSSelectPool` is a static table listing all Microsoft Select pools.

**Table 202: Database columns for MSSelectPool table**

Database Column	Details
<code>MSSelectPoolID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>MSSelectPool</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Applications</li> <li>• 2 = Systems</li> <li>• 3 = Servers</li> </ul>
<code>ResourceName</code>	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a pool. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DefaultValue</code>	<i>Type:</i> text (max 64 characters) The text to display if the pool resource string has no translation.

## MobileDevice Table

`MobileDevice` extends the `ComplianceComputer` table to store mobile device related property values.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 203: Database columns for `MobileDevice` table**

Database Column	Details
<code>MobileDeviceID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a <code>MobileDevice</code> Column use to specify clustered index.
<code>ComplianceComputerID</code>	<i>Type:</i> integer. Key A unique identifier for a <code>MobileDevice</code> . Foreign key to the <code>ComplianceComputer</code> table.
<code>IMEI</code>	<i>Type:</i> text (max 256 characters). Nullable IMEI value of the mobile device.
<code>PhoneNo</code>	<i>Type:</i> text (max 128 characters). Nullable Phone number of the mobile device.
<code>EmailAddress</code>	<i>Type:</i> text (max 256 characters). Nullable The stmp email account associated to a mobile device when the device is connected to ActiveSync.

## NotificationItem Table

`NotificationItem` lists notifications that were sent to end-users.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 204: Database columns for `NotificationItem` table**

Database Column	Details
<code>NotificationItemID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for this notification.
<code>NotificationTypeID</code>	<i>Type:</i> integer. Key

Database Column	Details
	The type of notification to be sent. Foreign key to the <code>NotificationType</code> table.
<code>NotificationDate</code>	<i>Type:</i> datetime. Key The date the notification should be sent.
<code>TaskID</code>	<i>Type:</i> integer. Key. Nullable The task the notification is for, if any. Foreign key to the <code>TermAndConditionTask</code> table.
<code>ContractID</code>	<i>Type:</i> integer. Key. Nullable The contract the notification is for, if any. Foreign key to the <code>Contract</code> table.
<code>ComplianceUserID</code>	<i>Type:</i> integer. Key The end-user that is receiving the notification. Foreign key to the <code>ComplianceUser</code> table.
<code>SentDate</code>	<i>Type:</i> datetime. Key. Nullable The date the notification was actually sent.

## NotificationTemplate Table

`NotificationTemplate` stores a list of email templates used to generate notification emails.

**Table 205: Database columns for `NotificationTemplate` table**

Database Column	Details
<code>NotificationTemplateID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>NotificationTemplate</code> . The default templates provided are: <ul style="list-style-type: none"> <li>• -1 = Contract expiry notification template</li> <li>• -2 = Contract renewal notification template</li> <li>• -3 = Task due notification template</li> <li>• -4 = Task reminder notification template</li> <li>• -5 = Task escalation notification template.</li> </ul>
<code>FileName</code>	<i>Type:</i> text (max 255 characters). Key



Database Column	Details
	The template's file name.
Content	Type: text. Nullable The template content.

## NotificationType Table

`NotificationType` stores a list of notification types that can be sent to end-users.

**Table 206: Database columns for `NotificationType` table**

Database Column	Details
<code>NotificationTypeID</code>	Type: integer. Key. Generated ID A unique identifier for each <code>NotificationType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>1 = Contract Expiry (a notification sent to end-users responsible for a contract when it is due to expire)</li> <li>2 = Contract Renewal (a notification sent to end-users responsible for a contract when it is due for renewal)</li> <li>3 = Task Due (a notification sent to the end-user assigned to a task when it is due for completion)</li> <li>4 = Task Reminder (a notification sent to the end-user assigned to a task as a reminder that the task is nearing completion)</li> <li>5 = Task Escalation (a notification sent to the end-user assigned to receive escalations, typically when a task is not completed on time).</li> </ul>
<code>ResourceName</code>	Type: text (max 256 characters). Key The unique name of the localizable resource string representing a notification type. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DefaultValue</code>	Type: text (max 100 characters) The text to display if the type resource string has no translation.
<code>NotificationTemplateID</code>	Type: integer. Key. Nullable The template to use when sending notifications of this type. Foreign key to the <code>NotificationTemplate</code> table.

## OperatorManageState Table

The `OperatorManageState` table lists the possible states for managing who has responsibility for maintaining certain business data. This is for internal use.

**Table 207: Database columns for `OperatorManageState` table**

Database Column	Details
<code>OperatorManageStateID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for the operator management state of business data.
<code>Name</code>	<i>Type:</i> text (max 64 characters). Key A unique name for the state
<code>DescriptionResourceName</code>	<i>Type:</i> text (max 256 characters). Nullable The unique name of the localizable resource string representing the description of the state. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DescriptionDefaultValue</code>	<i>Type:</i> text (max 256 characters) A string representing the default name of the state. Foreign key to the <code>ComplianceResourceString</code> table.
<code>IsLocked</code>	<i>Type:</i> boolean Is the data locked from edits by an operator.
<code>IsModified</code>	<i>Type:</i> boolean Is the data modified by an operator.
<code>IsFactory</code>	<i>Type:</i> boolean Is the data from the Reference ARL factory.
<code>AutoUpdate</code>	<i>Type:</i> boolean Is the data to be updated automatically.
<code>Priority</code>	<i>Type:</i> integer Is the data locked from edits by an operator.

## OperatorTaskTypeSetting Table

The `OperatorTaskTypeSetting` table stores data related to background task type.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 208: Database columns for `OperatorTaskTypeSetting` table**

Database Column	Details
<code>OperatorTaskTypeSettingID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated operator task type setting ID
<code>ComplianceOperatorID</code>	<i>Type:</i> integer. Key Foreign key to the <code>ComplianceOperator</code> table
<code>ActivityTypeID</code>	<i>Type:</i> integer. Key Foreign key to the <code>ActivityType</code> table
<code>Enabled</code>	<i>Type:</i> boolean Enabled flag for a setting

## OracleInstance Table

`OracleInstance` stores key characteristics specific to instances of Oracle databases which may impact the cost of licensing.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 209: Database columns for `OracleInstance` table**

Database Column	Details
<code>InstanceID</code>	<i>Type:</i> integer. Key The database instance whose attributes are being stored. Foreign key to the <code>Instance</code> table.
<code>InstanceEnvironmentID</code>	<i>Type:</i> integer

Database Column	Details
	The environment of the database instance. Foreign key to the <code>InstanceEnvironment</code> table.
<code>InstanceRoleID</code>	<i>Type:</i> integer The role of the database instance. Foreign key to the <code>InstanceRole</code> table.

## PaymentSchedule Table

`PaymentSchedule` contains details of the payment schedules managed by FlexNet Manager Suite.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 210: Database columns for `PaymentSchedule` table**

Database Column	Details
<code>PaymentScheduleID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a payment schedule.
<code>ContractID</code>	<i>Type:</i> integer. Key Identifies a contract to which this payment schedule applies. Foreign key to the <code>Contract</code> table.
<code>PaymentScheduleTypeID</code>	<i>Type:</i> integer Identifies the type of this payment schedule. Foreign key to the <code>PaymentScheduleType</code> table.
<code>PaymentScheduleTermID</code>	<i>Type:</i> integer Identifies the term of payment for this payment schedule. Foreign key to the <code>PaymentScheduleTerm</code> table.
<code>PaymentScheduleCategoryID</code>	<i>Type:</i> integer Identifies the category of this payment schedule. Foreign key to the <code>PaymentScheduleCategory</code> table.
<code>Description</code>	<i>Type:</i> text (max 100 characters) Name of this payment schedule.

Database Column	Details
StartDate	<i>Type:</i> datetime The date on which this payment schedule starts.
EndDate	<i>Type:</i> datetime. Nullable The date on which this payment schedule ends.
PeriodTypeID	<i>Type:</i> integer Identifies the period type of this payment schedule. Foreign key to the <code>PeriodType</code> table.
IncludeNewAssetsAndLicenses	<i>Type:</i> boolean If this field is set to <code>True</code> , then when a new asset or license is linked to the contract associated with this payment schedule, the item will also be linked to this payment schedule. If <code>False</code> , new items linked to the related contract are not automatically linked to the payment schedule (although a manual link can still be made).
LeaseTerminationDate	<i>Type:</i> datetime. Nullable The termination date of this payment schedule's lease. Only applicable if the payment schedule type is <code>Lease</code> .
LeaseTerminationReason	<i>Type:</i> text (max 100 characters). Nullable The reason this payment schedule's lease was terminated. Only applicable if the payment schedule type is <code>Lease</code> .
LeaseNumber	<i>Type:</i> text (max 150 characters). Nullable The number of this payment schedule's lease. Only applicable if the payment schedule type is <code>Lease</code> .
BuyoutCost	<i>Type:</i> currency. Nullable The buyout cost for this payment schedule's lease. Only applicable if the payment schedule type is <code>Lease</code> .
BuyoutCostRateID	<i>Type:</i> integer. Nullable Identifies the currency rate to be applied to this payment schedule's lease buyout cost. Only applicable if the payment schedule type is <code>Lease</code> . Foreign key to the <code>CurrencyRate</code> table.
PreviousPurchases	<i>Type:</i> integer. Nullable

Database Column	Details
	In the case of a Microsoft Enterprise Agreement renewal, the number of desktops covered by the associated platform license at the end of the previous Microsoft EA.
Comment	<i>Type:</i> text. Nullable Operator's comments about this payment schedule.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the payment schedule was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator to make the last change to this record.
UpdatedDate	<i>Type:</i> datetime The date the last change was made to this payment schedule record.

## PaymentScheduleCategory Table

`PaymentScheduleCategory` is a static table listing categories that can be assigned to a payment schedule.

**Table 211: Database columns for `PaymentScheduleCategory` table**

Database Column	Details
<code>PaymentScheduleCategoryId</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>PaymentScheduleCategory</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Fixed</li> <li>• 2 = License true up</li> <li>• 3 = Per hardware item</li> <li>• 4 = Per license quantity.</li> </ul>
<code>ResourceString</code>	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a payment schedule category. Foreign key to the <code>ComplianceResourceString</code> table.

Database Column	Details
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the category resource string has no translation.

## PaymentScheduleDetail Table

PaymentScheduleDetail lists all individual periods of a payment schedule.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 212: Database columns for PaymentScheduleDetail table**

Database Column	Details
PaymentScheduleDetailID	<i>Type:</i> integer. Key. Generated ID Uniquely identifies this payment schedule period.
PaymentScheduleID	<i>Type:</i> integer. Key Identifies the payment schedule to which this period applies. Foreign key to the PaymentSchedule table.
PeriodCovered	<i>Type:</i> text (max 50 characters) A string describing the period to which this payment schedule period is applicable. This is a calculated field.
PeriodStartDate	<i>Type:</i> datetime. Key The date on which this payment schedule period starts.
PeriodEndDate	<i>Type:</i> datetime The date on which this payment schedule period ends.
DueDate	<i>Type:</i> datetime. Key. Nullable The date on which this payment is due.
PaymentScheduleDetail PaymentStatusID	<i>Type:</i> integer. Key

Database Column	Details
	Identifies the state type of this payment schedule. The default value 2 corresponds to an <code>Incomplete</code> status. Foreign key to the <code>PaymentScheduleDetailPaymentStatus</code> table.
<code>PaymentDate</code>	<i>Type:</i> datetime. Nullable Records the date the payment was made.
<code>ActualAmount</code>	<i>Type:</i> currency. Nullable The actual amount paid in this payment schedule period.
<code>ActualAmountRateID</code>	<i>Type:</i> integer. Nullable Identifies the currency rate to be applied to the amount paid in this payment schedule period. Foreign key to the <code>CurrencyRate</code> table.
<code>EstimatedAmount</code>	<i>Type:</i> currency. Nullable The estimated amount for this payment schedule period.
<code>EstimatedAmountRateID</code>	<i>Type:</i> integer. Nullable Identifies the currency rate to be applied to the estimated amount for this payment schedule period. Foreign key to the <code>CurrencyRate</code> table.
<code>BudgetedAmount</code>	<i>Type:</i> currency. Nullable The budgeted amount for this payment schedule period.
<code>BudgetedAmountRateID</code>	<i>Type:</i> integer. Nullable Identifies the currency rate to be applied to the budgeted amount for this payment schedule period. Foreign key to the <code>CurrencyRate</code> table.
<code>Obligated</code>	<i>Type:</i> boolean If this field is set to <code>True</code> , the payee is obligated to pay during this payment schedule period. If this bit is <code>False</code> (the default), payment can presumably be deferred.
<code>Quantity</code>	<i>Type:</i> integer. Nullable The quantity for this payment schedule period.
<code>UnitPrice</code>	<i>Type:</i> currency. Nullable The unit price for this payment schedule period.
<code>UnitPriceRateID</code>	<i>Type:</i> integer. Nullable



Database Column	Details
	Identifies the currency rate to be applied to the unit price for this payment schedule period. Foreign key to the <code>CurrencyRate</code> table.
<code>SoftwareAssuranceUnitPrice</code>	<i>Type:</i> currency. Nullable The unit price for support (Software Assurance) for this payment schedule period.
<code>SoftwareAssuranceUnitPriceRateID</code>	<i>Type:</i> integer. Nullable Identifies the currency rate to be applied to the unit price for support in this payment schedule period. Foreign key to the <code>CurrencyRate</code> table.
<code>Notes</code>	<i>Type:</i> text. Nullable The notes field.
<code>PeriodCoveredResourceName</code>	<i>Type:</i> text (max 256 characters). Nullable The resource name used to describe the period to which this payment schedule period is applicable.
<code>PeriodCoveredResourceParameters</code>	<i>Type:</i> text (max 256 characters). Nullable The parameters used by the resource name used to describe the period to which this payment schedule period is applicable.

## PaymentScheduleDetailPaymentStatus Table

`PaymentScheduleDetailPaymentStatus` is a static table listing the possible status values for payment schedules.

**Table 213: Database columns for `PaymentScheduleDetailPaymentStatus` table**

Database Column	Details
<code>PaymentScheduleDetailPaymentStatusID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>PaymentScheduleDetailPaymentStatus</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Complete</li> <li>• 2 = Incomplete</li> <li>• 3 = Not going to pay.</li> </ul>
<code>ResourceName</code>	<i>Type:</i> text (max 256 characters). Key

Database Column	Details
	The unique name of the localizable resource string representing a payment schedule status. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 256 characters) The text to display if the status resource string has no translation.

## PaymentScheduleTerm Table

`PaymentScheduleTerm` is a static table listing possible payment schedule terms (the timing of payments in relation to each payment period).

**Table 214: Database columns for `PaymentScheduleTerm` table**

Database Column	Details
<code>PaymentScheduleTermID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>PaymentScheduleTerm</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Pre-paid</li> <li>• 2 = At the end of each period</li> <li>• 3 = At the beginning of each period.</li> </ul>
<code>ResourceString</code>	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a payment schedule term. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the term resource string has no translation.

## PaymentScheduleType Table

`PaymentScheduleType` is a static table listing possible payment schedule types.

**Table 215: Database columns for `PaymentScheduleType` table**

Database Column	Details
<code>PaymentScheduleTypeID</code>	<i>Type:</i> integer. Key. Generated ID

Database Column	Details
	<p>A unique identifier for each <code>PaymentScheduleType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = General</li> <li>• 2 = Lease</li> <li>• 3 = Hardware maintenance and support</li> <li>• 4 = Software license</li> <li>• 5 = Software maintenance and support</li> <li>• 6 = Consulting services</li> <li>• 7 = Insurance</li> <li>• 8 = Rent</li> <li>• 9 = Subscription</li> <li>• 10 = EA professional platform</li> <li>• 11 = EA other application.</li> </ul>
<code>ResourceString</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a payment schedule type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>
<code>XMLFile</code>	<p><i>Type:</i> text. Nullable</p> <p>The layout of the property dialog for this type of payment schedule, stored in XML format.</p>

## Project Table

Details about each `Project`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 216: Database columns for Project table**

Database Column	Details
ProjectID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the project.
ProjectName	<i>Type:</i> text (max 100 characters). Key The name of the project.
Comments	<i>Type:</i> text. Nullable Comments recorded about the project.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.

## PurchaseOrder Table

The `PurchaseOrder` table contains a list of all the purchase orders in the system.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 217: Database columns for PurchaseOrder table**

Database Column	Details
PurchaseOrderID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the purchase order.
PurchaseOrderNo	<i>Type:</i> text (max 50 characters). Key

Database Column	Details
	The purchase order number.
ShortDescription	<i>Type:</i> text (max 250 characters). Nullable A short description of the purchase order.
PurchaseOrderDate	<i>Type:</i> datetime The date recorded for the purchase order.
PurchaseOrderStatusID	<i>Type:</i> integer. Nullable The current state of the purchase order. Foreign key to the <code>PurchaseOrderStatus</code> table. The default value of 1 links to a “New” status.
PurchaseOrderTypeID	<i>Type:</i> integer. Nullable The type of the purchase order. Foreign key to the <code>PurchaseOrderType</code> table.
InvoiceNo	<i>Type:</i> text (max 50 characters). Nullable The invoice number that relates to the purchase order.
InvoiceDate	<i>Type:</i> datetime. Nullable The date on the invoice that relates to the purchase order.
TotalPrice	<i>Type:</i> currency. Nullable The total price of the purchase order.
TotalPriceRateID	<i>Type:</i> integer. Nullable The currency rate to be applied to this purchase order. Foreign key to the <code>CurrencyRate</code> table.
ShippingAndHandling	<i>Type:</i> currency. Nullable The amount of money spent on shipping and handling.
ShippingAndHandlingRateID	<i>Type:</i> integer. Nullable The currency rate to be applied to the shipping and handling costs related to this purchase order. Foreign key to the <code>CurrencyRate</code> table.
SalesTax	<i>Type:</i> currency. Nullable The amount of sales tax paid as part of this purchase order.
SalesTaxRateID	<i>Type:</i> integer. Nullable The currency rate to be applied to the sales tax related to this purchase order. Foreign key to the <code>CurrencyRate</code> table.

Database Column	Details
AutoCalculateCostFromChildren	<p><i>Type:</i> boolean</p> <p>The default value of <code>True</code> indicates that the total price, shipping, and sales tax values should be calculated from the purchase order lines that are children of this purchase order. A value of <code>False</code> means that these values are manually inserted into this purchase order header.</p>
ShippingMethodID	<p><i>Type:</i> integer. Nullable</p> <p>The type shipping used to deliver the product. Foreign key to the <code>ShippingMethod</code> table.</p>
ShippingLocationID	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>The location to which the ordered material is shipped. Foreign key to the <code>GroupEx</code> table.</p>
ShippingDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date the ordered material was shipped.</p>
RequestNo	<p><i>Type:</i> text (max 60 characters). Nullable</p> <p>The request number for the purchase order.</p>
RequestDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date the purchase order was requested.</p>
RequestedByID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The person who requested the purchase order. Foreign key to the <code>ComplianceUser</code> table.</p>
AuthorizedByID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The person who authorized the purchase order. Foreign key to the <code>ComplianceUser</code> table.</p>
ProcessedByID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The person who processed the purchase order. Foreign key to the <code>ComplianceUser</code> table.</p>
Comments	<p><i>Type:</i> text. Nullable</p> <p>Comments recorded about the purchase order.</p>
VendorID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The vendor fulfilling this purchase order. Foreign key to the <code>Vendor</code> table.</p>

Database Column	Details
ContractID	<i>Type:</i> integer. Key. Nullable Foreign key to the <code>Contract</code> table, identifying any existing contract related to this purchase order.
LocationID	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise location associated with this purchase order. Foreign key to the <code>GroupEx</code> table.
BusinessUnitID	<i>Type:</i> text (max 128 characters). Key. Nullable Any corporate unit in the enterprise associated with this purchase order. Foreign key to the <code>GroupEx</code> table.
CostCenterID	<i>Type:</i> text (max 128 characters). Key. Nullable Any cost center in the enterprise associated with this purchase order. Foreign key to the <code>GroupEx</code> table.
CategoryID	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise category associated with this purchase order. Foreign key to the <code>GroupEx</code> table.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.

## PurchaseOrderDetail Table

The `PurchaseOrderDetail` table contains a list of all the individual purchase order lines in the system.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 218: Database columns for PurchaseOrderDetail table**

Database Column	Details
PurchaseOrderDetailID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the purchase order line.
PurchaseOrderID	<i>Type:</i> integer. Key The parent purchase order to which this line belongs. Foreign key to the <i>PurchaseOrder</i> table.
PurchaseOrderDetailParentID	<i>Type:</i> integer. Nullable When a purchase order line is nested as a child of another, this link identifies the parent. Foreign key to another purchase order line in this <i>PurchaseOrderDetail</i> table.
ItemDescription	<i>Type:</i> text (max 250 characters) A description of the item ordered in this PO line.
SequenceNumber	<i>Type:</i> integer. Key The sequence number of the PO line in the overall purchase order.
PartNo	<i>Type:</i> text (max 100 characters). Nullable Deprecated, use LicensePartNo.
Quantity	<i>Type:</i> integer. Nullable The quantity of items purchased in this PO line.
QuantityPerUnit	<i>Type:</i> integer. Nullable Where the purchase order refers to software licenses, this is the quantity of license included in per unit of this purchase order.
EffectiveQuantity	<i>Type:</i> integer. Nullable The license entitlements brought in by this purchase.
LicenseQuantity	<i>Type:</i> integer. Nullable Where the purchase order refers to software licenses, this is the number of license entitlements conferred by the item ordered in this line. This is distinct from the purchase quantity on the line item. For example, it would be possible



Database Column	Details
	to order “Qty 50 of XYZ license 10-pack”, which would mean a <code>Quantity</code> field of 50 and a <code>LicenseQuantity</code> of 500.
<code>LicensePartNo</code>	<i>Type:</i> text (max 100 characters). Key. Nullable The part number or SKU of the item ordered in this PO line.
<code>UnitPrice</code>	<i>Type:</i> currency. Nullable The unit price of items ordered on this PO line.
<code>UnitPriceRateID</code>	<i>Type:</i> integer. Nullable The currency rate to be applied to the above unit price. Foreign key to the <code>CurrencyRate</code> table.
<code>SalesTax</code>	<i>Type:</i> currency. Nullable The amount of sales tax paid on this PO line item. May be left null if sales tax is only entered on the purchase order header.
<code>SalesTaxRateID</code>	<i>Type:</i> integer. Nullable The currency rate to be applied to the above sales tax. Foreign key to the <code>CurrencyRate</code> table.
<code>TotalPrice</code>	<i>Type:</i> currency. Nullable The total price of items in this PO line.
<code>TotalPriceRateID</code>	<i>Type:</i> integer. Nullable The currency rate to be applied to the above total price. Foreign key to the <code>CurrencyRate</code> table.
<code>AutoCalculateTotal</code>	<i>Type:</i> boolean Set this field to <code>True</code> (the default) for the total price to be calculated automatically as $(UnitPrice * Quantity) + ShippingAndHandling + SalesTax$ . If <code>False</code> , the operator must enter the total manually.
<code>ShippingAndHandling</code>	<i>Type:</i> currency. Nullable The amount of money spent on shipping and handling.
<code>ShippingAndHandlingRateID</code>	<i>Type:</i> integer. Nullable The currency rate to be applied to the above shipping and handling costs. Foreign key to the <code>CurrencyRate</code> table.
<code>InheritPOContractID</code>	<i>Type:</i> boolean. Key

Database Column	Details
	A bit which, if set to 1 (the default), means that the following contract ID is inherited from the parent purchase order.
ContractID	<i>Type:</i> integer. Key. Nullable A link to a contract related to this PO line. Foreign key to the <code>Contract</code> table.
InheritPOShippingDetails	<i>Type:</i> boolean Set this field to <code>True</code> (the default) for the following shipping details to be inherited from the parent purchase order. If <code>False</code> , an operator has to complete the following details manually.
ShippingDate	<i>Type:</i> datetime. Nullable The date the product was shipped.
ShippingMethodID	<i>Type:</i> integer. Nullable The delivery method used to deliver the item ordered in this PO line. Foreign key to the <code>ShippingMethod</code> table.
ShippingLocationID	<i>Type:</i> text (max 128 characters). Key. Nullable The location to which the item is shipped. Foreign key to the <code>GroupEx</code> table.
MaintenanceOrService Agreement	<i>Type:</i> boolean Set this field to <code>True</code> when this PO line includes maintenance or another type of service agreement. If <code>False</code> (the default), there is no maintenance or other service agreement associated with this PO line.
EffectiveDate	<i>Type:</i> datetime. Nullable The effective date for the Purchase Order Line.
ExpiryDate	<i>Type:</i> datetime. Nullable The expiry date for the Purchase Order Line.
InheritPOEnterpriseGroup	<i>Type:</i> boolean Set this field to <code>True</code> (the default) for the following enterprise groups to be inherited from the parent purchase order. If <code>False</code> , an operator has to complete the following details manually.
LocationID	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise location associated with this PO line. Foreign key to the <code>GroupEx</code> table.
BusinessUnitID	<i>Type:</i> text (max 128 characters). Key. Nullable

Database Column	Details
	Any corporate unit within the enterprise associated with this PO line. Foreign key to the <code>GroupEx</code> table.
<code>CostCenterID</code>	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise cost center associated with this PO line. Foreign key to the <code>GroupEx</code> table.
<code>CategoryID</code>	<i>Type:</i> text (max 128 characters). Key. Nullable Any category used within the enterprise associated with this PO line. Foreign key to the <code>GroupEx</code> table.
<code>InheritPOProcessDetails</code>	<i>Type:</i> boolean Set this field to <code>True</code> (the default) for the following process details to be inherited from the parent purchase order. If <code>False</code> , an operator has to complete the following details manually.
<code>RequestNo</code>	<i>Type:</i> text (max 60 characters). Nullable The request number for the PO line.
<code>RequestDate</code>	<i>Type:</i> datetime. Nullable The date the related product was requested.
<code>RequestedByID</code>	<i>Type:</i> integer. Key. Nullable The person who requested the purchase order line. Foreign key to the <code>ComplianceUser</code> table.
<code>AuthorizedByID</code>	<i>Type:</i> integer. Key. Nullable The person who authorized the purchase order line. Foreign key to the <code>ComplianceUser</code> table.
<code>ProcessedByID</code>	<i>Type:</i> integer. Key. Nullable The person who processed the purchase order line. Foreign key to the <code>ComplianceUser</code> table.
<code>Comments</code>	<i>Type:</i> text. Nullable Comments recorded about the purchase order line.
<code>InheritPOInvoiceDetails</code>	<i>Type:</i> boolean Set this field to <code>True</code> (the default) for the following invoicing details to be inherited from the parent purchase order. If <code>False</code> , an operator has to complete the following details manually.

Database Column	Details
InvoiceNo	<i>Type:</i> text (max 50 characters). Nullable The invoice number relating to this PO line.
InvoiceDate	<i>Type:</i> datetime. Nullable The invoice date for the purchase order line.
OrderedProduct	<i>Type:</i> text (max 256 characters). Nullable A description of the item ordered in this PO line.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.
ExternalID	<i>Type:</i> text (max 32 characters). Nullable A text field where an operator may record the ID of the PO line in any external system it was imported from.
PurchaseOrderDetailTypeID	<i>Type:</i> integer. Key The type of the PO line. Foreign key to the <code>PurchaseOrderDetailType</code> table.
MSSelectPoolID	<i>Type:</i> integer. Nullable Identifies the Microsoft Select pool. Foreign key to the <code>MSSelectPool</code> table.
MSSelectPoints	<i>Type:</i> decimal. Nullable The number of points consumed by this purchase.
AutoAcceptRecommendation	<i>Type:</i> boolean Set this field to <code>True</code> to automatically accept recommendation calculated for this purchase order line in Link Licenses node.
SoftwareSkuID	<i>Type:</i> integer. Key. Nullable

Database Column	Details
	The SKU that was recognized. This value is optional. Foreign key to the <code>SoftwareSku</code> table.
<code>PurchaseOrderDetailStatusID</code>	<i>Type:</i> integer The current state of the purchase order details. Foreign key to the <code>PurchaseOrderDetailStatus</code> table. The default value of 1 links to a “New” status.
<code>PublisherID</code>	<i>Type:</i> integer. Nullable The publisher of this line item. This value is optional. Foreign key to the <code>Vendor</code> table.

## PurchaseOrderDetailProperty Table

`PurchaseOrderDetailProperty` defines extra custom properties for all purchase order lines.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 219: Database columns for `PurchaseOrderDetailProperty` table**

Database Column	Details
<code>PurchaseOrderDetailPropertyID</code>	<i>Type:</i> integer. Key. Generated ID Unique identifier for a purchase order line property.
<code>PropertyName</code>	<i>Type:</i> text (max 256 characters). Key The name of the custom property. Foreign key to the <code>ComplianceResourceString</code> table.
<code>CustomPropertyDisplayXMLID</code>	<i>Type:</i> integer. Nullable Reference to a record in the <code>CustomPropertyDisplayXML</code> table, describing how to show the property on a property dialog.

## PurchaseOrderDetailPropertyValue Table

For each purchase order line, PurchaseOrderDetailPropertyValue stores the values for the custom properties defined in PurchaseOrderDetailProperty.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 220: Database columns for PurchaseOrderDetailPropertyValue table**

Database Column	Details
PurchaseOrderDetailPropertyValueID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a property value.
PurchaseOrderDetailID	<i>Type:</i> integer. Key The purchase order line associated with the property. Foreign key to the PurchaseOrderDetail table
PurchaseOrderDetailPropertyID	<i>Type:</i> integer. Key the property whose value is being stored. Foreign key to the PurchaseOrderDetailProperty table
PropertyValue	<i>Type:</i> text (max 4000 characters) The property value.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.

## PurchaseOrderDetailStatus Table

`PurchaseOrderDetailStatus` is a static table listing the possible states for purchase order details, broadly tracking the associated business processes.

**Table 221: Database columns for `PurchaseOrderDetailStatus` table**

Database Column	Details
<code>PurchaseOrderDetailStatusID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>PurchaseOrderDetailStatus</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = New</li> <li>• 2 = Pending</li> <li>• 3 = Completed</li> <li>• 4 = Cancelled</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a purchase order status. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the status resource string has no translation.</p>

## PurchaseOrderDetailType Table

`PurchaseOrderDetailType` is a static table listing the possible types of purchase order line item.

**Table 222: Database columns for `PurchaseOrderDetailType` table**

Database Column	Details
<code>PurchaseOrderDetailTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>PurchaseOrderDetailType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Not set</li> <li>• 2 = Software</li> <li>• 3 = Hardware</li> <li>• 4 = Service</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 5 = Other</li> <li>• 6 = Software upgrade</li> <li>• 7 = Software maintenance</li> <li>• 8 = Disk kit</li> <li>• 9 = Hardware maintenance</li> <li>• 10 = Software Baseline.</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a purchase order line item type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## PurchaseOrderProperty Table

`PurchaseOrderProperty` defines extra custom properties for all purchase orders.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 223: Database columns for `PurchaseOrderProperty` table**

Database Column	Details
PurchaseOrderPropertyID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>Unique identifier for a purchase order property.</p>
PropertyName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The name of the property.</p>
CustomPropertyDisplayXMLID	<p><i>Type:</i> integer. Nullable</p> <p>Foreign key to a record in the <code>CustomPropertyDisplayXML</code> table, describing how to show the property on a property dialog.</p>



# PurchaseOrderPropertyValue Table

For each purchase order, `PurchaseOrderPropertyValue` stores the values for the custom properties defined in `PurchaseOrderProperty`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 224: Database columns for `PurchaseOrderPropertyValue` table**

Database Column	Details
<code>PurchaseOrderPropertyValueID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a property value.
<code>PurchaseOrderID</code>	<i>Type:</i> integer. Key The purchase order associated with this property. Foreign key to the <code>PurchaseOrder</code> table.
<code>PurchaseOrderPropertyID</code>	<i>Type:</i> integer. Key The property whose value is being stored. Foreign key to the <code>PurchaseOrderProperty</code> table.
<code>PropertyValue</code>	<i>Type:</i> text (max 4000 characters) The property value.
<code>CreationUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
<code>CreationDate</code>	<i>Type:</i> datetime The date the record was created.
<code>UpdatedUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
<code>UpdatedDate</code>	<i>Type:</i> datetime. Nullable The date the record was last updated.

## PurchaseOrderStatus Table

`PurchaseOrderStatus` is a static table listing the possible states for purchase orders, broadly tracking the associated business processes.

**Table 225: Database columns for `PurchaseOrderStatus` table**

Database Column	Details
<code>PurchaseOrderStatusID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>PurchaseOrderStatus</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = New</li> <li>• 2 = Completed</li> <li>• 3 = Cancelled</li> <li>• 4 = Sent to approver</li> <li>• 5 = Sent to vendor</li> <li>• 6 = Item received.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a purchase order status. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the status resource string has no translation.</p>

## PurchaseOrderType Table

`PurchaseOrderType` is a static table listing the possible types of purchase order. Reserved for future expansion.

**Table 226: Database columns for `PurchaseOrderType` table**

Database Column	Details
<code>PurchaseOrderTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>PurchaseOrderType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = None.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p>

Database Column	Details
	The unique name of the localizable resource string representing a purchase order type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	Type: text (max 100 characters) The text to display if the type resource string has no translation.

## PurchaseProgram Table

`PurchaseProgram` is a static table listing all known contract purchase programs.

**Table 227: Database columns for `PurchaseProgram` table**

Database Column	Details
PurchaseProgramID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for each <code>PurchaseProgram</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Microsoft Select Agreement</li> <li>• 2 = Microsoft Enterprise Agreement</li> <li>• 3 = Microsoft Open Agreement</li> <li>• 4 = Adobe Cumulative Licensing Program</li> <li>• 5 = Adobe Transactional Licensing Program</li> <li>• 6 = Adobe Site License Program</li> <li>• 7 = Acronis Licensing Program</li> <li>• 8 = Attachmate Volume Purchase Account</li> <li>• 9 = Business Objects Open Licensing Program</li> <li>• 10 = CA Master License Program</li> <li>• 11 = CA Open License Program</li> <li>• 12 = Citrix Easy Licensing Program</li> <li>• 13 = Citrix Enterprise License Program</li> <li>• 14 = Citrix Open Licensing Program</li> <li>• 15 = Citrix Premium Licensing Program</li> <li>• 16 = Corel Contractual License</li> <li>• 17 = Corel Transactional Licensing</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 18 = IBM Passport Advantage</li> <li>• 19 = McAfee TSP Licensing Program</li> <li>• 20 = Novell Corporate License Agreement</li> <li>• 21 = Novell Master License Agreement</li> <li>• 22 = Novell Volume License Agreement</li> <li>• 23 = Symantec Elite</li> <li>• 24 = Symantec Express</li> <li>• 25 = Symantec Open Licensing Program</li> <li>• 26 = Symantec Rewards</li> <li>• 27 = Symantec Volume Licensing Program</li> <li>• 28 = VMware Purchasing Program</li> <li>• 29 = Macromedia Volume License Program</li> <li>• 30 = Symantec Enterprise Option</li> <li>• 31 = Symantec Enterprise VPA.</li> <li>• 32 = Oracle Master Agreement</li> <li>• 33 = Oracle Unlimited Agreement</li> <li>• 34 = Oracle License and Services Agreement</li> <li>• 35 = Adobe Enterprise Term Licensing Agreement</li> <li>• 36 = Microsoft Products and Services Agreement</li> <li>• 37 = IBM Passport Advantage Express</li> <li>• 38 = IBM Enterprise License Agreement</li> <li>• 39 = IBM Enterprise Software and Services Option</li> </ul>
Name	<p><i>Type:</i> text (max 100 characters). Key</p> <p>The display name of the purchase program.</p>
PublisherName	<p><i>Type:</i> text (max 64 characters). Key</p> <p>The name of publisher under which this purchase program applies.</p>
Code	<p><i>Type:</i> text (max 16 characters). Key</p> <p>A short code used to represent this purchase program.</p>

# QuerySnapshot Table

QuerySnapshot holds the snapshot of data for a report



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 228: Database columns for QuerySnapshot table**

Database Column	Details
QuerySnapshotID	Type: integer. Key. Generated ID A unique identifier for a query snapshot.
QueryContext	Type: text (max 200 characters). Key The query context to partition different queries.
ComplianceSavedSearchID	Type: integer. Key. Nullable The query definition this snapshot is for. Foreign key to the ComplianceSavedSearch table.
ComplianceOperatorID	Type: integer. Key The operator who ran the report. Foreign key to the ComplianceOperator table.
SnapshotName	Type: text (max 200 characters) Name of snapshot.
SnapshotSchema	Type: XML Schema of snapshot.
SnapshotDate	Type: datetime Date and time of snapshot (UTC)
SnapshotBuildTime	Type: big integer Number of milliseconds taken to build the snapshot.
SnapshotRows	Type: big integer Number of rows in the snapshot.

## RelationType Table

RelationType is a static table containing types of relationship between objects

**Table 229: Database columns for RelationType table**

Database Column	Details
RelationTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each RelationType. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>1 = VMware ESX host managed by vCenter</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a relation type. Foreign key to the ComplianceResourceString table.</p>
DefaultValue	<p><i>Type:</i> text (max 256 characters)</p> <p>The text to display if the type resource string has no translation.</p>
ImporterString	<p><i>Type:</i> text (max 100 characters). Key</p> <p>The text value provided by adapters when importing relation type.</p>

## ResponsibilityType Table

ResponsibilityType is a static table listing possible end-user responsibilities.

**Table 230: Database columns for ResponsibilityType table**

Database Column	Details
ResponsibilityTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for an end-user's title or responsibility. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>1 = Blank</li> <li>2 = Owner</li> <li>3 = Signatory</li> <li>4 = Contract Manager</li> <li>5 = Point of Contact</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 6 = Negotiator</li> <li>• 7 = Interested Party.</li> </ul>
ResourceString	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a user responsibility. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the responsibility resource string has no translation.

## RestrictedAccessType Table

`RestrictedAccessType` is a static table holding access types

**Table 231: Database columns for `RestrictedAccessType` table**

Database Column	Details
RestrictedAccessTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a type of access. Values are: <ul style="list-style-type: none"> <li>• 1 = All users</li> <li>• 2 = Accessible only to creator</li> </ul>
RestrictedAccessTypeName	<i>Type:</i> text (max 512 characters). Key Access type name.

## RulesEngineRuleDefinition Table

This table stores rule definitions used for consolidating users.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 232: Database columns for RulesEngineRuleDefinition table**

Database Column	Details
RuleDefinitionID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the rule definition.
RuleDefinitionName	<i>Type:</i> text (max 128 characters) Name of the rule.
RuleTypeID	<i>Type:</i> integer Foreign key to the rule type.
RuleDefinition	<i>Type:</i> text The rule definition XML used to build the rule statement used by the rules engine.
IsActive	<i>Type:</i> boolean Whether or not this rule is active for execution.
CreationUser	<i>Type:</i> text (max 256 characters) The user who created the system landscape.
CreationDate	<i>Type:</i> datetime The data and time the system landscape was created.
UpdatedUser	<i>Type:</i> text (max 256 characters) The last user who update the system landscape.
UpdatedDate	<i>Type:</i> datetime The date and time the system landscape was last updated.

## RulesEngineRuleType Table

This table stores the available rule types used for rulesengine.

**Table 233: Database columns for RulesEngineRuleType table**

Database Column	Details
RuleTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the rule type.



Database Column	Details
TypeName	<i>Type:</i> text (max 100 characters). Key A unique name for the rule type.
TitleResourceName	<i>Type:</i> text (max 256 characters). Nullable A localizable resource string representing a rule type. Foreign key to the <code>ComplianceResourceString</code> table.
TitleDefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the rule type resource string has no translation.
RuleTemplate	<i>Type:</i> text The template used to build a rule for the rules engine.
DefaultRuleDefinition	<i>Type:</i> text. Nullable Default rule definition for newly created rule

## SecurityType Table

`SecurityType` lists the types of security model that can be used to determine access to a contract or document.

**Table 234: Database columns for `SecurityType` table**

Database Column	Details
SecurityTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SecurityType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>1 = Public (security is controlled by the operator's roles)</li> <li>2 = Restricted (security is controlled by an access control list of account names).</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a security type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.

## SerialNumberBlackList Table

`SerialNumberBlackList` stores a blacklist of invalid serial numbers.

**Table 235: Database columns for `SerialNumberBlackList` table**

Database Column	Details
<code>SerialNumberBlackListID</code>	<i>Type:</i> integer. Key. Generated ID The unique identifier for a blacklisted serial number.
<code>SerialNo</code>	<i>Type:</i> text (max 100 characters). Key The blacklisted serial number.

## SessionUIDBeacon Table

The `SessionUIDBeacon` table stores the task's SessionUID and the beacon where the task is running .



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 236: Database columns for `SessionUIDBeacon` table**

Database Column	Details
<code>SessionUID</code>	<i>Type:</i> unique identifier. Key Unique task run identifier
<code>BeaconID</code>	<i>Type:</i> integer. Key Beacon where the task's session ran

## ShippingMethod Table

`ShippingMethod` is a static table listing possible delivery methods. Reserved for future expansion.

**Table 237: Database columns for ShippingMethod table**

Database Column	Details
ShippingMethodID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ShippingMethod</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>1 = None.</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a shipping method. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the shipping method resource string has no translation.</p>

## SoftwareLicenseContractPaymentSchedule Table

`SoftwareLicenseContractPaymentSchedule` links a payment schedule to a software license, via a link from that software license to a contract.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 238: Database columns for SoftwareLicenseContractPaymentSchedule table**

Database Column	Details
SoftwareLicenseContractID	<p><i>Type:</i> integer. Key</p> <p>Identifies a link between a software license and a contract. Foreign key to the <code>SoftwareLicenseContract</code> table.</p>
PaymentScheduleID	<p><i>Type:</i> integer. Key</p> <p>Identifies a payment schedule. Foreign key to the <code>PaymentSchedule</code> table.</p>

## SystemShutdown Table

A row in this table indicates that the system is being taken down, and is used to show a warning to users.

**Table 239: Database columns for SystemShutdown table**

Database Column	Details
SystemShutdownID	Type: integer. Key. Generated ID Synthetic key for this table.
MessageResourceName	Type: text (max 256 characters). Nullable A resource name used to look up a message to show to the operator
StartTime	Type: datetime The time the shutdown is scheduled to begin
EndTime	Type: datetime The estimated time that the shutdown will end

## TaskExecutionStatus Table

The TaskExecutionStatus table stores progress data for rules and background tasks.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 240: Database columns for TaskExecutionStatus table**

Database Column	Details
TaskExecutionStatusID	Type: integer. Key. Generated ID Auto-generated task execution status ID
SessionUID	Type: unique identifier. Key Unique task run identifier
TaskName	Type: text (max 255 characters). Key The name of task.
ActivityTypeID	Type: integer. Key Foreign key to the ActivityType table
DateStarted	Type: datetime. Nullable

Database Column	Details
	Start date and time for a task.
DateCompleted	Type: datetime. Nullable Completion date and time for a task.
EventTypeStatusID	Type: integer. Key Foreign key to the EventTypeStatus table
BeaconRuleID	Type: integer. Key. Nullable Foreign key to the BeaconRule table
ScheduledTriggerDateTick	Type: big integer. Key. Nullable Executed date time in Tick.
BeaconID	Type: integer. Key. Nullable Beacon where the task is executing.
BeaconPolicyRevision Number	Type: integer. Nullable Beacon policy revision number
OperatorLogin	Type: text (max 255 characters). Nullable Login of the operator who started task.

## TaskExecutionStatusStep Table

The TaskExecutionStatusStep table stores progress data for rule or background task steps.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 241: Database columns for TaskExecutionStatusStep table**

Database Column	Details
TaskExecutionStatusStepID	Type: integer. Key. Generated ID Auto-generated task step execution status ID
TaskExecutionStatusID	Type: integer. Key

Database Column	Details
	Foreign key to <code>TaskExecutionStatus</code> table.
<code>TaskStepID</code>	<i>Type:</i> integer. Key The ID of task step.
<code>BeaconRuleActionPropertyID</code>	<i>Type:</i> integer. Key. Nullable The ID of rule action subtask.
<code>DateStarted</code>	<i>Type:</i> datetime. Nullable Start date and time for a step.
<code>DateCompleted</code>	<i>Type:</i> datetime. Nullable Completion date and time for a step.
<code>EventTypeStatusID</code>	<i>Type:</i> integer. Key Foreign key to the <code>EventTypeStatus</code> table
<code>BeaconUID</code>	<i>Type:</i> unique identifier. Key. Nullable Beacon ID.
<code>EventTypeID</code>	<i>Type:</i> integer. Key. Nullable Foreign key to the <code>EventType</code> table
<code>EventID</code>	<i>Type:</i> integer. Key. Nullable Foreign key to the <code>Event</code> table
<code>Location</code>	<i>Type:</i> text (max 255 characters). Nullable Server name where operation was performed.
<code>TaskParameters</code>	<i>Type:</i> XML. Nullable parameters for the task step.

## TaskStep Table

The `TaskStep` table stores task steps.

**Table 242: Database columns for TaskStep table**

Database Column	Details
TaskStepID	<i>Type:</i> integer. Key. Generated ID Auto-generated task step ID
ActivityTypeID	<i>Type:</i> integer. Key Foreign key to the ActivityType table
TaskStepResourceName	<i>Type:</i> text (max 255 characters). Key Task step name resource name
TaskStepDefaultValue	<i>Type:</i> text (max 255 characters) Task step name default value
TaskStepOrder	<i>Type:</i> integer Task step order index

## TaskStepEventType Table

The TaskStepEventType table stores eventType related to the taskStep.

**Table 243: Database columns for TaskStepEventType table**

Database Column	Details
TaskStepID	<i>Type:</i> integer. Key Foreign key to the TaskStep table
EventTypeID	<i>Type:</i> integer. Key Foreign key to the EventType table

## TermAndCondition Table

TermAndCondition stores a list of terms and conditions related to a contract.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 244: Database columns for *TermAndCondition* table**

Database Column	Details
TermAndConditionID	Type: integer. Key. Generated ID A unique identifier for the term/condition.
TermAndConditionTypeID	Type: integer. Key The type of term/condition. Foreign key to <i>TermAndConditionType</i> table.
Description	Type: text (max 100 characters). Key A description assigned by the operator.
DocReference	Type: text (max 100 characters). Nullable A text reference to a document for this term/condition.
Comments	Type: text. Nullable Comments about this term/condition.
BeginDate	Type: datetime. Nullable The start date for this term or condition.
EndDate	Type: datetime. Nullable The end date for this term or condition.
ContractID	Type: integer. Key The contract to which this term/condition applies. Foreign key to the <i>Contract</i> table.
CreationUser	Type: text (max 128 characters). Nullable The operator who created the record.
CreationDate	Type: datetime The date the term/condition was created.
UpdatedUser	Type: text (max 128 characters). Nullable The name of the operator who last updated the term/condition.
UpdatedDate	Type: datetime. Nullable



Database Column	Details
	The date the record was last updated.
EmailComplianceUserID	<i>Type:</i> integer. Key. Nullable A user who may be emailed according to conditions on this term/condition. Foreign key to the <code>ComplianceUser</code> table.
EmailIntervalTypeID	<i>Type:</i> integer. Key. Nullable The interval type for <code>EmailInterval</code> . Foreign key to the <code>IntervalType</code> table.
EmailInterval	<i>Type:</i> integer. Nullable The interval used when sending emails.
ReminderIntervalTypeID	<i>Type:</i> integer. Key. Nullable The interval type for <code>ReminderInterval</code> . Foreign key to the <code>IntervalType</code> table.
ReminderInterval	<i>Type:</i> integer. Nullable The interval used when sending reminders.
EscalationComplianceUserID	<i>Type:</i> integer. Key. Nullable A user who may be emailed if the term/condition needs to be escalated. Foreign key to the <code>ComplianceUser</code> table.
EscalationIntervalTypeID	<i>Type:</i> integer. Key. Nullable The interval type for <code>EscalationInterval</code> . Foreign key to the <code>IntervalType</code> table.
EscalationInterval	<i>Type:</i> integer. Nullable The interval used when sending escalation messages.
Auditable	<i>Type:</i> boolean Boolean to indicate whether the term/condition is auditable.

## TermAndConditionTask Table

`TermAndConditionTask` holds extra information about a task.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 245: Database columns for `TermAndConditionTask` table**

Database Column	Details
<code>ComplianceTaskID</code>	<p>Type: integer. Key</p> <p>The task this extra information applies to. Foreign key to the <code>ComplianceTask</code> table.</p>
<code>Completed</code>	<p>Type: boolean</p> <p>Set this field to <code>True</code> if this task has been completed.</p>
<code>CompletionDate</code>	<p>Type: datetime. Nullable</p> <p>The date of completion of the task.</p>
<code>ComplianceUserID</code>	<p>Type: integer. Key. Nullable</p> <p>The end-user this task is assigned to. Foreign key to the <code>ComplianceUser</code> table.</p>
<code>Notes</code>	<p>Type: text. Nullable</p> <p>Notes or comments related to the task.</p>

## TermAndConditionType Table

`TermAndConditionType` stores a list of types of different terms/conditions that may be associated with contracts.

**Table 246: Database columns for `TermAndConditionType` table**

Database Column	Details
<code>TermAndConditionTypeID</code>	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for each <code>TermAndConditionType</code>. The default values and corresponding default strings are:</p> <ul style="list-style-type: none"> <li>1 = Acceptance Period</li> <li>2 = Price Change</li> <li>3 = Cancellation</li> <li>4 = Renewal</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 5 = Expiry</li> <li>• 6 = Review</li> <li>• 7 = Limitation.</li> </ul>
TermAndConditionType ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a term/condition type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
TermAndConditionType DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>
ManageSoftType	<p><i>Type:</i> boolean</p> <p>If set to <code>True</code>, this field indicates that this term and condition type was created by FlexNet Manager Suite and should not be deleted or edited. If <code>False</code>, the type has been created by an operator, and may be modified.</p>

## UserNameBlacklist Table

`UserNameBlacklist` stores a list of excluded accounts that will not be imported into FlexNet Manager Suite. If an end-user with account name matching a record in `UserNameBlacklist` already exists in FlexNet Manager Suite, that end-user will not be included in compliance calculations and will not appear in many of the end-user lists.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 247: Database columns for `UserNameBlacklist` table**

Database Column	Details
UserNameBlacklistID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for the blacklisted account.</p>
UserName	<p><i>Type:</i> text (max 64 characters). Key</p> <p>A blacklisted account name. May contain wildcards (% , _). End-users whose domain\SAM account name match this value will be excluded from compliance calculations.</p>

## VMEnabledState Table

VMEnabledState is a static table listing the possible operational states of a virtual machine.

**Table 248: Database columns for VMEnabledState table**

Database Column	Details
VMEnabledStateID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for each VMEnabledState. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Started</li> <li>• 2 = Stopped</li> <li>• 3 = Suspended</li> <li>• 4 = Unknown.</li> </ul>
ResourceName	<p>Type: text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a virtual machine operational state. Foreign key to the ComplianceResourceString table.</p>
DefaultValue	<p>Type: text (max 100 characters)</p> <p>The text to display if the operational state resource string has no translation.</p>

## VMHostManagedBySoftware Table

VMHostManagedBySoftware stores relationships between management software and VM hosts it manages. The RelationTypeID specifies the context of these relationships



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 249: Database columns for VMHostManagedBySoftware table**

Database Column	Details
VMHostManagedBySoftwareID	<p>Type: integer. Key. Generated ID</p> <p>The primary key of VMHostManagedBySoftware.</p>

Database Column	Details
InstalledSoftwareID	<i>Type:</i> integer. Key A unique identifier of an <code>InstalledSoftware</code> .
RelationTypeID	<i>Type:</i> integer. Key The type of relationship between management software and the VM hosts. Foreign key to the <code>RelationType</code> table.
ComplianceComputerID	<i>Type:</i> integer. Key A unique identifier of a <code>ComplianceComputer</code> .

## VMPool Table

VMPool contains information about virtual machine pools (logical groups of VMs or partitions).



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 250: Database columns for VMPool table**

Database Column	Details
VMPoolID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a virtual machine pool.
PoolName	<i>Type:</i> text (max 100 characters). Key The name of the pool.
PoolFriendlyName	<i>Type:</i> text (max 256 characters) The friendly name of the pool.
Path	<i>Type:</i> text (max 1000 characters) The full path of the pool (including parent pool names).
VCOBJECTID	<i>Type:</i> text (max 256 characters). Nullable The ID of the virtual machine folder (pool) in Virtual Center.
NextChild	<i>Type:</i> integer

Database Column	Details
	One more than the number of children this pool has.
PoolPathID	<i>Type:</i> text (max 128 characters) A numerical representation of the path of this pool, constructed from <code>VMPoolID</code> values (something like: "1.2.").
HostComplianceComputerID	<i>Type:</i> integer. Key. Nullable A link to the host computer that this pool exists on. This is a foreign key to the <code>ComplianceComputer</code> table.
VMPoolTypeID	<i>Type:</i> integer. Key The type of pool. Foreign key to the <code>VMPoolType</code> table.
VirtualMachineID	<i>Type:</i> integer. Nullable If this pool is a virtual machine or partition itself, this is a link to that virtual machine or partition. Foreign key to the <code>VirtualMachine</code> table.
NumberOfProcessors	<i>Type:</i> decimal. Nullable The number of processors in this pool.
NumberOfCores	<i>Type:</i> decimal. Nullable The number of cores in this pool.

## VMPoolType Table

`VMPoolType` is a static table listing the possible types of a virtual machine pool.

**Table 251: Database columns for `VMPoolType` table**

Database Column	Details
VMPoolTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a <code>VMPoolType</code> . Possible values and the corresponding default names are: <ul style="list-style-type: none"> <li>• 1 = Folder</li> <li>• 2 = Data Center</li> <li>• 3 = Compute Resource</li> <li>• 4 = Host System</li> <li>• 5 = Resource Pool</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 6 = Virtual Machine</li> <li>• 7 = Physical Shared Pool</li> <li>• 8 = Virtual Shared Pool</li> <li>• 9 = LPAR</li> <li>• 10 = RSET</li> <li>• 11 = Cluster Compute Resource.</li> </ul>
VCTypeID	<i>Type:</i> text (max 32 characters) The type of the virtual machine folder in Virtual Center.
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a pool type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the pool type resource string has no translation.

## VMSourceType Table

`VMSourceType` is a static table used to define possible virtual machine inventory source values (that is, whether the properties were created manually or reported by the compliance importer).

**Table 252: Database columns for `VMSourceType` table**

Database Column	Details
VMSourceTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>VMSourceType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Manual (the virtual machine properties were manually created and have not been updated by the compliance importer)</li> <li>• 2 = VM Host (the virtual machine's host recently reported inventory and updated these virtual machine properties).</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a virtual machine source type. Foreign key to the <code>ComplianceResourceString</code> table.

Database Column	Details
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the source type resource string has no translation.

## VMState Table

`VMState` is a static table listing the possible relationships between a virtual machine and a physical (inventoried) computer.

**Table 253: Database columns for `VMState` table**

Database Column	Details
VMStateID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>VMState</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>1 = Linked (the virtual machine is linked to an inventoried or manually created computer)</li> <li>2 = Unlinked (the virtual machine is only linked to a “light” computer, automatically created from the host computer’s inventory)</li> <li>3 = Duplicated (the virtual machine has a duplicate UUID and is not linked to an inventoried or manually created computer).</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a virtual machine state. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the state resource string has no translation.

## VMType Table

`VMType` is a static table listing the possible types of virtual machine or partition.

**Table 254: Database columns for `VMType` table**

Database Column	Details
VMTypeID	<i>Type:</i> integer. Key. Generated ID



Database Column	Details
	<p>A unique identifier for a <code>VMType</code>. Possible values and the corresponding default names are:</p> <ul style="list-style-type: none"> <li>• 1 = VMware</li> <li>• 2 = Hyper-V</li> <li>• 3 = LPAR</li> <li>• 4 = WPAR</li> <li>• 5 = nPar</li> <li>• 6 = vPar</li> <li>• 7 = SRP</li> <li>• 8 = Zone</li> <li>• 9 = Unknown.</li> <li>• 10 = Oracle VM</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a virtual machine or partition type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## Vendor Table

The `Vendor` table contains a list of all the vendors in the system.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 255: Database columns for `Vendor` table**

Database Column	Details
VendorID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for the vendor.</p>

Database Column	Details
VendorName	<i>Type:</i> text (max 64 characters). Key The name of the vendor.
VendorPreviousName	<i>Type:</i> text (max 64 characters). Nullable Any earlier name that the vendor was previously known as.
BusinessPhoneNumber	<i>Type:</i> text (max 30 characters). Nullable The business phone number of the vendor.
FaxPhoneNumber	<i>Type:</i> text (max 30 characters). Nullable The fax number of the vendor.
Address_Street	<i>Type:</i> text (max 200 characters). Nullable The street address of the vendor.
Address_City	<i>Type:</i> text (max 200 characters). Nullable The city of the vendor.
Address_State	<i>Type:</i> text (max 200 characters). Nullable The state or province of the vendor.
Address_ZIP	<i>Type:</i> text (max 20 characters). Nullable The ZIP or postal code of the vendor.
Address_Country	<i>Type:</i> text (max 100 characters). Nullable The country of the vendor.
Address2_Street	<i>Type:</i> text (max 200 characters). Nullable The second street address of the vendor, if applicable.
Address2_City	<i>Type:</i> text (max 200 characters). Nullable The second city of the vendor.
Address2_State	<i>Type:</i> text (max 200 characters). Nullable The second state or province of the vendor.
Address2_ZIP	<i>Type:</i> text (max 20 characters). Nullable The second ZIP or postal code of the vendor.
Address2_Country	<i>Type:</i> text (max 100 characters). Nullable

Database Column	Details
	The second country of the vendor.
WebSite	<i>Type:</i> text (max 200 characters). Nullable The web site of the vendor.
Email	<i>Type:</i> text (max 200 characters). Nullable The email address of the vendor.
ParentVendorID	<i>Type:</i> integer. Nullable A link to a vendor's parent vendor. Foreign key to another vendor record in this <code>Vendor</code> table. Vendor hierarchies are not currently implemented.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.
AutomaticallyAcceptPurchases	<i>Type:</i> boolean Whether purchases from this vendor should have their license linking recommendations in the <code>EntitlementRecommendation</code> table automatically accepted.

## VendorContact Table

`VendorContact` contains a list of all the vendor contacts, or individuals employed by the vendor with whom this enterprise has contact.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 256: Database columns for VendorContact table**

Database Column	Details
VendorContactID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the contact.
UserTitleID	<i>Type:</i> integer. Nullable The title of the contact's name. Foreign key to the <code>UserTitle</code> table.
FirstName	<i>Type:</i> text (max 128 characters) The first name of the contact.
MiddleName	<i>Type:</i> text (max 128 characters). Nullable The middle name(s) of the contact.
LastName	<i>Type:</i> text (max 128 characters). Nullable The last name name of the contact.
UserSuffixID	<i>Type:</i> integer. Nullable The suffix to the name of the contact.
JobTitle	<i>Type:</i> text (max 128 characters). Nullable The job title of the contact.
VendorID	<i>Type:</i> integer. Key A link to the contact's parent vendor. Foreign key to the <code>Vendor</code> table.
BusinessPhoneNumber	<i>Type:</i> text (max 30 characters). Nullable The business phone number of the contact.
MobilePhoneNumber	<i>Type:</i> text (max 30 characters). Nullable The mobile phone number of the contact.
FaxPhoneNumber	<i>Type:</i> text (max 30 characters). Nullable The fax number of the contact.
Address_Street	<i>Type:</i> text (max 200 characters). Nullable The street address of the contact.
Address_City	<i>Type:</i> text (max 200 characters). Nullable The city of the contact.

Database Column	Details
Address_State	<i>Type:</i> text (max 200 characters). Nullable The state or province of the contact.
Address_ZIP	<i>Type:</i> text (max 20 characters). Nullable The ZIP or postal code of the contact.
Address_Country	<i>Type:</i> text (max 100 characters). Nullable The country of the contact.
Address2_Street	<i>Type:</i> text (max 200 characters). Nullable The second street address of the contact, if applicable.
Address2_City	<i>Type:</i> text (max 200 characters). Nullable The second city of the contact.
Address2_State	<i>Type:</i> text (max 200 characters). Nullable The second state or province of the contact.
Address2_ZIP	<i>Type:</i> text (max 20 characters). Nullable The second ZIP or postal code of the contact.
Address2_Country	<i>Type:</i> text (max 100 characters). Nullable The second country of the contact.
Email	<i>Type:</i> text (max 200 characters). Nullable The email address of the contact.
Messenger	<i>Type:</i> text (max 200 characters). Nullable The instant messenger address of the contact.
Comments	<i>Type:</i> text. Nullable Comments recorded about the contact.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable

Database Column	Details
	The operator who last updated the record.
UpdatedDate	Type: datetime. Nullable The date the record was last updated.

## VendorProperty Table

`VendorProperty` defines extra custom properties for all vendors.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 257: Database columns for `VendorProperty` table**

Database Column	Details
VendorPropertyID	Type: integer. Key. Generated ID Unique identifier for a vendor property.
PropertyName	Type: text (max 256 characters). Key The name of the custom property. Foreign key to the <code>ComplianceResourceString</code> table.
CustomPropertyDisplayXMLID	Type: integer. Nullable Reference to a record in the <code>CustomPropertyDisplayXML</code> table, describing how to show the property on a property dialog.

## VendorPropertyValue Table

For each vendor, `VendorPropertyValue` stores the values for the custom properties defined in `VendorProperty`.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 258: Database columns for VendorPropertyValue table**

Database Column	Details
VendorPropertyValueID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a property value.
VendorID	<i>Type:</i> integer. Key The vendor associated with this property. Foreign key to the <code>Vendor</code> table.
VendorPropertyID	<i>Type:</i> integer. Key The property whose value is being stored. Foreign key to the <code>VendorProperty</code> table.
PropertyValue	<i>Type:</i> text (max 4000 characters) The property value.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.

## VirtualMachine Table

`VirtualMachine` stores extra information for computers identified as virtual machines or hardware partitions.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 259: Database columns for VirtualMachine table**

Database Column	Details
VirtualMachineID	<i>Type:</i> integer. Key. Generated ID A unique identifier for virtual machine or partition properties associated with a computer.
HostComplianceComputerID	<i>Type:</i> integer. Key. Nullable The virtual machines or partition's host computer. Foreign key to the <code>ComplianceComputer</code> table.
ComplianceComputerID	<i>Type:</i> integer. Key The computer associated with these virtual machine or partition properties. Computer in the <code>ComplianceComputer</code> table.
VMTypeID	<i>Type:</i> integer The type of virtual machine or partition. Foreign key to the <code>VMType</code> table.
UUID	<i>Type:</i> text (max 256 characters). Nullable The UUID (Universally Unique Identifier) of the virtual machine. Used to match virtual machine properties to their associated <code>ComplianceComputer</code> .
VMName	<i>Type:</i> text (max 256 characters). Nullable The name of the virtual machine or partition.
VMLocation	<i>Type:</i> text (max 256 characters). Nullable The location of the virtual machine on the file system.
GuestFullName	<i>Type:</i> text (max 256 characters). Nullable The configured operating system for the guest.
FriendlyName	<i>Type:</i> text (max 256 characters). Nullable The friendly name of the virtual machine or partition.
VCOBJECTID	<i>Type:</i> text (max 256 characters). Nullable The ID of the virtual machine in Virtual Center.
TotalMemory	<i>Type:</i> big integer. Nullable The total memory of the virtual machine (in bytes).
VMStateID	<i>Type:</i> integer



Database Column	Details
	The state of the virtual machine, related to whether it is linked to a computer or not. Foreign key to the <code>VMState</code> table.
<code>VMPoolID</code>	<i>Type:</i> integer. Nullable The resource pool that the virtual machine belongs to. Foreign key to the <code>VMPool</code> table.
<code>CPUUsage</code>	<i>Type:</i> integer. Nullable The maximum CPU usage of the Virtual Machine (measured in MHz).
<code>MemoryUsage</code>	<i>Type:</i> big integer. Nullable The maximum memory usage of the Virtual Machine (in bytes).
<code>VMAEnabledStateID</code>	<i>Type:</i> integer The operational state of the virtual machine (powered on, off, and so on). Foreign key to the <code>VMAEnabledState</code> table.
<code>VMSourceTypeID</code>	<i>Type:</i> integer Whether the virtual machine properties are manually entered or created from inventory. Foreign key to the <code>VMSourceType</code> table.
<code>CreationUser</code>	<i>Type:</i> text (max 256 characters) The operator who created this record.
<code>CreationDate</code>	<i>Type:</i> datetime The date/time when this record was created.
<code>UpdatedUser</code>	<i>Type:</i> text (max 256 characters). Nullable The operator who last updated this record.
<code>UpdatedDate</code>	<i>Type:</i> datetime The date/time when this record was last updated.
<code>AffinityEnabled</code>	<i>Type:</i> boolean Set this to <code>True</code> if this VM is unable to move to different host computers.
<code>LocatedInCloud</code>	<i>Type:</i> boolean 1 - if the virtual machine is hosted in a cloud environment
<code>ServiceProvider</code>	<i>Type:</i> text (max 250 characters). Nullable Cloud provider (data center)

Database Column	Details
CPUAffinity	<i>Type:</i> text (max 256 characters). Nullable Contains the CPU Affinity value for virtual machines (Host Logical processors)
CoreAffinity	<i>Type:</i> text (max 256 characters). Nullable Contains the Core Affinity value for virtual machine
PartitionID	<i>Type:</i> text (max 100 characters). Nullable Partition ID generated and used by the managing virtualization platform
PartitionNumber	<i>Type:</i> integer. Nullable Number of this partition
IsHostAssignedManually	<i>Type:</i> boolean Was the virtual machine assigned to its host manually? This prevents unlinking of the virtual machine.

## XMLInsertType Table

`XMLInsertType` is a static table storing how custom property XML snippets will be inserted into the default property display layout XML file.

**Table 260: Database columns for `XMLInsertType` table**

Database Column	Details
XMLInsertTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>XMLInsertType</code> . Possible values are: <ul style="list-style-type: none"> <li>• 1 = Before (the new snippet needs to go before the existing XML element)</li> <li>• 2 = After (the new snippet needs to go after the existing XML element)</li> <li>• 3 = Replace (the new snippet needs to replace the existing XML element)</li> <li>• 4 = First child (the new snippet needs to be added as the first child of the existing XML element)</li> <li>• 5 = Last child (the new snippet needs to be added as the last child of the existing XML element).</li> </ul>
TypeDescription	<i>Type:</i> text (max 50 characters). Key A description of the insert type.

# Compliance.Logic.Discovery Tables

The complete set of database tables documented here includes:

- ASN1Object table (see *ASN1Object Table* on page 259)
- DeviceRole table (see *DeviceRole Table* on page 260)
- DiscoveredDevice table (see *DiscoveredDevice Table* on page 260)
- DiscoveredDeviceCalculatedMember table (see *DiscoveredDeviceCalculatedMember Table* on page 263)
- DiscoveredDeviceParent table (see *DiscoveredDeviceParent Table* on page 264)
- DiscoveredDeviceSNMPInfo table (see *DiscoveredDeviceSNMPInfo Table* on page 265)
- DiscoveredDeviceVDIBrokerInfo table (see *DiscoveredDeviceVDIBrokerInfo Table* on page 266)
- DiscoveredDeviceVDIInfo table (see *DiscoveredDeviceVDIInfo Table* on page 267)
- DiscoveredDeviceVirtualizationInfo table (see *DiscoveredDeviceVirtualizationInfo Table* on page 268)
- KnownOracleListener table (see *KnownOracleListener Table* on page 269)
- KnownOracleService table (see *KnownOracleService Table* on page 270)
- Site table (see *Site Table* on page 271)
- SiteSubnet table (see *SiteSubnet Table* on page 271)
- VirtualizationProductName table (see *VirtualizationProductName Table* on page 272)

## ASN1Object Table

Stores a mapping from a ASN ObjectID (OID) to a type of device.

**Table 261: Database columns for ASN1Object table**

Database Column	Details
OID	Type: text (max 128 characters). Key ASN object identifier.
Description	Type: text (max 512 characters) The fully expanded text version of the object identifier.
ObjectRole	Type: integer. Nullable What role does the device perform?

## DeviceRole Table

The role of a network device.

**Table 262: Database columns for DeviceRole table**

Database Column	Details
DeviceRoleID	<p>Type: integer. Key. Generated ID</p> <p>The id of the device role.</p>
Description	<p>Type: text (max 64 characters). Key</p> <p>The name of the device role. Possible id and name pairs are:</p> <ul style="list-style-type: none"> <li>• 0 = Computer</li> <li>• 1 = Workstation</li> <li>• 2 = Server</li> <li>• 3 = Printer</li> <li>• 4 = Switch</li> <li>• 5 = Router</li> <li>• 6 = Hub</li> <li>• 7 = NetworkDevice</li> <li>• 8 = Vendor.</li> </ul>

## DiscoveredDevice Table

A `DiscoveredDevice` is a loose record of the discovery of a device on a network, using any of a number of discovery methods. As such, the same device may be found in more than one way (see `DuplicateID` which may be able to track this fact if known), or by more than one distinguishing feature. Accordingly this table has a somewhat unsatisfactory primary key!



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 263: Database columns for DiscoveredDevice table**

Database Column	Details
DeviceID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number.
DuplicateID	<i>Type:</i> integer. Key. Nullable Reference to another discovery record for this device, if known.
ComputerID	<i>Type:</i> integer. Key. Nullable FlexNet Manager Suite computer id, if known
DeviceName	<i>Type:</i> text (max 64 characters). Key. Nullable NetBIOS name for computers or any name for other devices, if known.
DNSFullName	<i>Type:</i> text (max 256 characters). Key. Nullable Fully qualified DNS name, if known.
NTDomainName	<i>Type:</i> text (max 256 characters). Key. Nullable NT domain name, if known.
IPAddress	<i>Type:</i> text (max 64 characters). Key. Nullable IP address of the device.
IPSubnet	<i>Type:</i> text (max 64 characters). Nullable IP subnet that contains the node.
IPSubnetMask	<i>Type:</i> text (max 64 characters). Nullable IP subnet mask for the subnet contains the device.
PhysicalAddress	<i>Type:</i> text (max 64 characters). Key. Nullable Network adapter physical address of the node. Can be a MAC address or token ring address.
DeviceRole	<i>Type:</i> integer. Nullable What role does the device perform? <ul style="list-style-type: none"> <li>• NULL = unknown</li> <li>• 0 = Computer (don't know if server or workstation)</li> <li>• 1 = Workstation</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 2 = Server</li> <li>• 3 = Printer</li> <li>• 4 = Switch</li> <li>• 5 = Router</li> <li>• 6 = Hub</li> </ul>
OperatingSystem	<i>Type:</i> text (max 128 characters). Nullable Operating system of the node, if it is a computer.
IsManaged	<i>Type:</i> integer. Key. Nullable Is the device to be managed by FlexNet Manager Suite? 0 = no, 1 = yes, NULL = unknown.
Description	<i>Type:</i> text (max 256 characters). Nullable Customer-entered description of the device.
SystemDescription	<i>Type:</i> text (max 256 characters). Nullable This field is currently unused.
SystemLocation	<i>Type:</i> text (max 256 characters). Nullable This field is currently unused.
SystemContact	<i>Type:</i> text (max 256 characters). Nullable This field is currently unused.
FirstDiscovered	<i>Type:</i> datetime The date and time that the node was first discovered.
LastUpdate	<i>Type:</i> datetime The last time the node was checked or updated.
LastDataSourceName	<i>Type:</i> text (max 128 characters). Key. Nullable A name that identifies where the discovery information came from (for example: physical location, server, and so on).
LastDataSourceType	<i>Type:</i> text (max 32 characters). Key. Nullable The type of data source (for example: Excel, Fluke, NM, Text).
OpenPortsTCP	<i>Type:</i> text (max 512 characters). Nullable

Database Column	Details
	The comma-delimited list of TCP ports which were found to be open on scan.
OpenPortsUDP	<i>Type:</i> text (max 512 characters). Nullable The comma-delimited list of UDP ports which were found to be open on scan.
ScannedOperatingSystem	<i>Type:</i> text (max 512 characters). Nullable The IP scan tool's best guess at the operating system. This is based on corner cases in the behavior of the network protocol stack.
ScannedOsType	<i>Type:</i> text (max 512 characters). Nullable OS Type, as reported by scan tool.
ScannedOsVendor	<i>Type:</i> text (max 512 characters). Nullable OS Vendor, as reported by scan tool.
ScannedOsFamily	<i>Type:</i> text (max 512 characters). Nullable OS family, as reported by scan tool.
ScannedOsGen	<i>Type:</i> text (max 512 characters). Nullable OS Generation(Versions), as reported by scan tool.
ScannedMacAddress	<i>Type:</i> text (max 64 characters). Nullable MAC Address, as reported by scan tool.
ScannedMacVendor	<i>Type:</i> text (max 512 characters). Nullable MAC Vendor, as reported by scan tool.
SQLDiscoveredBy	<i>Type:</i> text (max 128 characters). Nullable The discovery tool used to discover SQL Server.
SQLPorts	<i>Type:</i> text (max 128 characters). Nullable The ports where SQL Server has been discovered.
IPAddressInt	<i>Type:</i> big integer. Key. Nullable Integer representation of IPAddress column.

## DiscoveredDeviceCalculatedMember Table

Stores summary strings of `DiscoveredDevice` details that are expensive to calculate on demand.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 264: Database columns for DiscoveredDeviceCalculatedMember table**

Database Column	Details
DeviceID	Type: integer. Key Device identity number.
IsOracle	Type: boolean. Nullable Have we discovered Oracle on this machine?
OracleListeners	Type: text (max 512 characters). Nullable A summary string representing any known Oracle Listeners, and the port they can be contacted on.
OracleServices	Type: text (max 512 characters). Nullable A summary string representing any known Oracle Services.
IsSQL	Type: boolean. Nullable Have we discovered SQL Server on this machine?
IsVDI	Type: boolean. Nullable Is this machine a virtual desktop?
IsVDIBroker	Type: boolean. Nullable Have we discovered a VDI broker on this machine?

## DiscoveredDeviceParent Table

Records any parent child relationships between DiscoveredDevice records.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.



**Table 265: Database columns for DiscoveredDeviceParent table**

Database Column	Details
DeviceID	Type: integer. Key The child DiscoveredDevice ID
ParentDeviceID	Type: integer. Key The parent DiscoveredDevice ID

## DiscoveredDeviceSNMPInfo Table

Records any SNMP information discovered for a `DiscoveredDevice`.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 266: Database columns for DiscoveredDeviceSNMPInfo table**

Database Column	Details
DeviceID	Type: integer. Key Device identity number.
SNMP_snmpCommunityString	Type: text (max 256 characters). Nullable The SNMP Community String that was used for obtaining SNMP MIBs. This and all following SNMP attributes are defined in RFC1907 and others available from the IETF websites.
SNMP_sysDescr	Type: text (max 256 characters). Nullable A textual description of the device. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software.
SNMP_sysObjectID	Type: text (max 256 characters). Nullable The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining 'what kind of device' is being managed. For example, if vendor

Database Column	Details
	'Flintstones, Inc.' was assigned the subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its 'Fred Router'.
SNMP_sysObjectIDSymbolic	<i>Type:</i> text (max 256 characters). Nullable The symbolic representation of the same value as sysObjectID.
SNMP_sysUpTime	<i>Type:</i> big integer. Nullable The time (in hundredths of a second) since the network management portion of the system was last re-initialized.
SNMP_sysContact	<i>Type:</i> text (max 256 characters). Nullable The textual identification of the contact person for this managed node, together with information on how to contact this person.
SNMP_sysName	<i>Type:</i> text (max 256 characters). Nullable An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name.
SNMP_sysLocation	<i>Type:</i> text (max 256 characters). Nullable The physical location of this node (for example, 'telephone closet, 3rd floor').
SNMP_sysServices	<i>Type:</i> integer. Nullable a bitmask indicating at which of the seven OSI protocol levels the system provides services (physical=1, TCP = 8, applications = 64, etc)
SNMP_ipForwarding	<i>Type:</i> integer. Nullable Set to 1 if the device forwards IP packets, 2 otherwise.

## DiscoveredDeviceVDIBrokerInfo Table

Records any VDI information discovered for a `DiscoveredDevice`.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 267: Database columns for DiscoveredDeviceVDIBrokerInfo table**

Database Column	Details
DeviceID	Type: integer. Key Device identity number.
VDISiteName	Type: text (max 256 characters). Key. Nullable The Site to which this VDI Broker belongs.
BrokerType	Type: text (max 256 characters). Key The type of VDI broker found.

## DiscoveredDeviceVDIInfo Table

Records any VDI information discovered for a `DiscoveredDevice`.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 268: Database columns for DiscoveredDeviceVDIInfo table**

Database Column	Details
DeviceID	Type: integer. Key Device identity number.
VDIGroupName	Type: text (max 256 characters). Nullable The Desktop Group to which this VDI belongs.
VDITemplateName	Type: text (max 256 characters). Nullable The template from which this VDI device was cloned.
VDISiteName	Type: text (max 256 characters). Key. Nullable The Site to which this VDI belongs.
BrokerType	Type: text (max 256 characters). Key. Nullable The type of broker that serves up this VDI belongs.
BrokerMachineName	Type: text (max 64 characters). Nullable

Database Column	Details
	NetBIOS name for the VDI broker.
BrokerDomainName	Type: text (max 256 characters). Nullable NT domain name of the broker.
BrokerIPAddress	Type: text (max 256 characters). Nullable The IP of the broker.
IsPersistent	Type: boolean Whether or not the VDI device is a persistent one.

## DiscoveredDeviceVirtualizationInfo Table

Records any virtualization server information discovered for a `DiscoveredDevice`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 269: Database columns for `DiscoveredDeviceVirtualizationInfo` table**

Database Column	Details
DeviceID	Type: integer. Key Device identity number.
Protocol	Type: text (max 16 characters). Nullable The protocol by which the virtualization API is accessed on the device.
Port	Type: integer The TCP port used by the protocol.
APIType	Type: text (max 32 characters). Nullable The reported API type.
APIVersion	Type: text (max 16 characters). Nullable The supported version of the API.
ProductNameID	Type: integer

Database Column	Details
	The reported product name.
ProductVersion	Type: text (max 16 characters). Nullable The reported product version.

## KnownOracleListener Table

Records any discovered Oracle listeners a `DiscoveredDevice` is providing.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 270: Database columns for KnownOracleListener table**

Database Column	Details
KnownOracleListenerID	Type: integer. Key. Generated ID Unique id for the known listener.
DeviceID	Type: integer. Key Device identity number.
Port	Type: integer. Key Port for this listener.
Name	Type: text (max 128 characters) The name of the service provided by the device.
Version	Type: text (max 32 characters) The version of the service provided by the device.
ManuallyAdded	Type: boolean Boolean field specifying whether the KnownService record has been manually added by the user.
DiscoveredRemotely	Type: boolean True means this listener is discovered using remote discovery, false otherwise

Database Column	Details
DiscoveredLocally	<i>Type:</i> boolean True means this listener is discovered using local discovery, false otherwise
DiscoveredViaTNSNames	<i>Type:</i> boolean True means this listener is discovered from a TNSNames file on a beacon

## KnownOracleService Table

Records any discovered Oracle services (databases) on a `DiscoveredDevice`



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 271: Database columns for KnownOracleService table**

Database Column	Details
KnownOracleServiceID	<i>Type:</i> integer. Key. Generated ID Unique id for the known Oracle service.
KnownOracleListenerID	<i>Type:</i> integer. Key. Nullable Listener identity number.
DeviceID	<i>Type:</i> integer. Key Network device identity number.
Name	<i>Type:</i> text (max 128 characters). Key The name of the service provided by the device.
ManuallyAdded	<i>Type:</i> boolean Boolean field specifying whether the KnownService record has been manually added by the user.
DiscoveredRemotely	<i>Type:</i> boolean True means this service is discovered using remote discovery, false otherwise
DiscoveredLocally	<i>Type:</i> boolean

Database Column	Details
	True means this service is discovered using local discovery, false otherwise
DiscoveredViaTNSNames	<i>Type:</i> boolean True means this service is discovered from a TNSNames file on a beacon

## Site Table

The Site table contains data about network locations (sites)



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 272: Database columns for Site table**

Database Column	Details
SiteID	<i>Type:</i> integer. Key. Generated ID The ID for the Site
Name	<i>Type:</i> text (max 256 characters). Key The name of the Site
AutoPopulated	<i>Type:</i> boolean Specifies whether the row was populated automatically(1) or manually(0).
Enabled	<i>Type:</i> boolean Specifies whether the row will be used when mapping domains and devices to Locations

## SiteSubnet Table

The Subnet table contains data about subnets in a location.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 273: Database columns for SiteSubnet table**

Database Column	Details
SubnetID	Type: integer. Key. Generated ID The ID for the Subnet
IPSubnet	Type: text (max 64 characters). Key The IPSubnet of the Subnet
IPSubnetBits	Type: tiny integer. Key The number of bits in the IPSubnet
SiteID	Type: integer. Key SiteID of the Site in which the Subnet resides
AutoPopulated	Type: boolean Specifies whether the row was populated automatically(1) or manually(0).
Enabled	Type: boolean Specifies whether the row will be used when mapping domains and devices to Locations
IPAddressRangeFrom	Type: big integer. Key Specifies whether the row will be used when mapping domains and devices to Locations
IPAddressRangeTo	Type: big integer. Key Specifies whether the row will be used when mapping domains and devices to Locations

## VirtualizationProductName Table

Stores unique virtualization server software names for a *DiscoveredDevice*.



**Table 274: Database columns for VirtualizationProductName table**

Database Column	Details
VirtualizationProduct NameID	Type: integer. Key. Generated ID Device identity number.
ProductName	Type: text (max 256 characters). Key The reported product name.

## Compliance.Logic.Licensing Tables

The complete set of database tables documented here includes:

- AccessMode table (see *AccessMode Table* on page 280)
- Cluster table (see *Cluster Table* on page 280)
- ClusterComputer table (see *ClusterComputer Table* on page 282)
- ClusterHostAffinityRule table (see *ClusterHostAffinityRule Table* on page 282)
- ClusterHostAffinityRuleType table (see *ClusterHostAffinityRuleType Table* on page 283)
- ClusterNodeType table (see *ClusterNodeType Table* on page 284)
- ClusterType table (see *ClusterType Table* on page 285)
- ComplianceComputerSnapshot table (see *ComplianceComputerSnapshot Table* on page 285)
- ComplianceComputerTag table (see *ComplianceComputerTag Table* on page 286)
- ComplianceUserSnapshot table (see *ComplianceUserSnapshot Table* on page 287)
- ComplianceUserTag table (see *ComplianceUserTag Table* on page 288)
- DatabaseMutex table (see *DatabaseMutex Table* on page 288)
- EntitlementRecommendation table (see *EntitlementRecommendation Table* on page 289)
- EntitlementRecommendationState table (see *EntitlementRecommendationState Table* on page 290)
- EntitlementTransaction table (see *EntitlementTransaction Table* on page 291)
- EntitlementTransactionOtherCandidate table (see *EntitlementTransactionOtherCandidate Table* on page 293)
- EntitlementTransactionState table (see *EntitlementTransactionState Table* on page 295)
- EntitlementTransactionType table (see *EntitlementTransactionType Table* on page 295)
- EvidenceExistenceRule table (see *EvidenceExistenceRule Table* on page 296)
- EvidenceStatus table (see *EvidenceStatus Table* on page 297)

- FNMEAFeature table (see *FNMEAFeature Table* on page 297)
- FNMEALicensedFeature table (see *FNMEALicensedFeature Table* on page 298)
- FileEvidenceCompany table (see *FileEvidenceCompany Table* on page 299)
- FileEvidenceEx table (see *FileEvidenceEx Table* on page 299)
- FileEvidenceFile table (see *FileEvidenceFile Table* on page 300)
- FileEvidenceLanguage table (see *FileEvidenceLanguage Table* on page 300)
- FileEvidenceMatchCount table (see *FileEvidenceMatchCount Table* on page 301)
- FileEvidencePath table (see *FileEvidencePath Table* on page 301)
- GroupSnapshot table (see *GroupSnapshot Table* on page 302)
- ImporterRun table (see *ImporterRun Table* on page 302)
- ImporterStepValidationIssue table (see *ImporterStepValidationIssue Table* on page 304)
- ImporterStepValidationIssueType table (see *ImporterStepValidationIssueType Table* on page 305)
- InstalledFileEvidence table (see *InstalledFileEvidence Table* on page 305)
- InstalledInstallerAttribute table (see *InstalledInstallerAttribute Table* on page 306)
- InstalledInstallerEvidence table (see *InstalledInstallerEvidence Table* on page 307)
- InstalledInstanceReplacement table (see *InstalledInstanceReplacement Table* on page 308)
- InstalledSoftwareData table (see *InstalledSoftwareData Table* on page 308)
- InstalledSoftwareRemoval table (see *InstalledSoftwareRemoval Table* on page 309)
- InstalledSoftwareReplacement table (see *InstalledSoftwareReplacement Table* on page 310)
- InstalledSoftwareUsageData table (see *InstalledSoftwareUsageData Table* on page 310)
- InstalledWMIEvidence table (see *InstalledWMIEvidence Table* on page 312)
- InstallerEvidence table (see *InstallerEvidence Table* on page 312)
- InstallerEvidenceEx table (see *InstallerEvidenceEx Table* on page 313)
- InstallerEvidenceMatchCount table (see *InstallerEvidenceMatchCount Table* on page 314)
- InstallerEvidenceType table (see *InstallerEvidenceType Table* on page 315)
- LicenseBreachReason table (see *LicenseBreachReason Table* on page 316)
- LicenseDefinitionTitle table (see *LicenseDefinitionTitle Table* on page 317)
- LicenseDefinitionType table (see *LicenseDefinitionType Table* on page 317)
- LicenseDefinitionUsageRight table (see *LicenseDefinitionUsageRight Table* on page 318)
- LicenseMeasurement table (see *LicenseMeasurement Table* on page 318)
- LicenseSimulation table (see *LicenseSimulation Table* on page 319)

- LicenseSimulationBreachStatus table (see *LicenseSimulationBreachStatus Table* on page 320)
- LicenseSimulationChangeType table (see *LicenseSimulationChangeType Table* on page 321)
- LicenseSimulationHWDetails table (see *LicenseSimulationHWDetails Table* on page 321)
- LicenseSimulationLicenseDetails table (see *LicenseSimulationLicenseDetails Table* on page 324)
- LicenseSimulationResults table (see *LicenseSimulationResults Table* on page 325)
- LicenseSimulationRowType table (see *LicenseSimulationRowType Table* on page 326)
- LicenseSimulationSWDetails table (see *LicenseSimulationSWDetails Table* on page 327)
- LicenseSimulationScenario table (see *LicenseSimulationScenario Table* on page 328)
- LicenseStatus table (see *LicenseStatus Table* on page 329)
- NewFileEvidence table (see *NewFileEvidence Table* on page 329)
- OracleLegacyLicenseType table (see *OracleLegacyLicenseType Table* on page 331)
- PODetailProcess table (see *PODetailProcess Table* on page 332)
- PeriodType table (see *PeriodType Table* on page 332)
- ProcessAction table (see *ProcessAction Table* on page 333)
- ProcessState table (see *ProcessState Table* on page 336)
- ReconcileInstalledSoftwareData table (see *ReconcileInstalledSoftwareData Table* on page 337)
- ReconcileInstalledSoftwareUsageData table (see *ReconcileInstalledSoftwareUsageData Table* on page 338)
- ReconcileInterestingLicenses table (see *ReconcileInterestingLicenses Table* on page 339)
- ReconcileInterestingTitles table (see *ReconcileInterestingTitles Table* on page 339)
- ReconcileSoftwareLicenseComputerProblem table (see *ReconcileSoftwareLicenseComputerProblem Table* on page 340)
- ReconcileSoftwareLicenseCoresConsumedData table (see *ReconcileSoftwareLicenseCoresConsumedData Table* on page 341)
- ReconcileSoftwareLicenseGroupPointsConsumedData table (see *ReconcileSoftwareLicenseGroupPointsConsumedData Table* on page 341)
- ReconcileSoftwareLicenseILMTPointsConsumedData table (see *ReconcileSoftwareLicenseILMTPointsConsumedData Table* on page 342)
- ReconcileSoftwareLicensePointsConsumedData table (see *ReconcileSoftwareLicensePointsConsumedData Table* on page 343)
- ReconcileSoftwareLicensePointsConsumedReason table (see *ReconcileSoftwareLicensePointsConsumedReason Table* on page 344)
- ReconcileSoftwareLicenseProcessorData table (see *ReconcileSoftwareLicenseProcessorData Table* on page 345)

- ReconcileSoftwareLicenseSecondUseMappingData table (see *ReconcileSoftwareLicenseSecondUseMappingData Table* on page 346)
- ReconcileSoftwareUserLicensePointsConsumedData table (see *ReconcileSoftwareUserLicensePointsConsumedData Table* on page 347)
- ReconcileVirtualMachineLayer table (see *ReconcileVirtualMachineLayer Table* on page 348)
- RegistryEvidence table (see *RegistryEvidence Table* on page 349)
- RegistryEvidenceHive table (see *RegistryEvidenceHive Table* on page 350)
- RegistryEvidenceKey table (see *RegistryEvidenceKey Table* on page 350)
- RegistryEvidenceValue table (see *RegistryEvidenceValue Table* on page 351)
- RelatedInstalledInstallerEvidence table (see *RelatedInstalledInstallerEvidence Table* on page 351)
- RelatedInstalledInstallerEvidenceSourceMap table (see *RelatedInstalledInstallerEvidenceSourceMap Table* on page 352)
- RelatedInstalledSoftwareData table (see *RelatedInstalledSoftwareData Table* on page 353)
- SAPSoftwareLicense table (see *SAPSoftwareLicense Table* on page 354)
- SAPSoftwareLicenseType table (see *SAPSoftwareLicenseType Table* on page 355)
- SAPSpecialVersion table (see *SAPSpecialVersion Table* on page 356)
- SoftwareAccessMode table (see *SoftwareAccessMode Table* on page 356)
- SoftwareLicense table (see *SoftwareLicense Table* on page 357)
- SoftwareLicenseAllocation table (see *SoftwareLicenseAllocation Table* on page 370)
- SoftwareLicenseAllocationStatus table (see *SoftwareLicenseAllocationStatus Table* on page 372)
- SoftwareLicenseAllocationUserType table (see *SoftwareLicenseAllocationUserType Table* on page 373)
- SoftwareLicenseBreachReasonData table (see *SoftwareLicenseBreachReasonData Table* on page 373)
- SoftwareLicenseChangeEvent table (see *SoftwareLicenseChangeEvent Table* on page 374)
- SoftwareLicenseChangeEventReason table (see *SoftwareLicenseChangeEventReason Table* on page 375)
- SoftwareLicenseChangeEventSource table (see *SoftwareLicenseChangeEventSource Table* on page 375)
- SoftwareLicenseComplianceStatus table (see *SoftwareLicenseComplianceStatus Table* on page 376)
- SoftwareLicenseComputerProblemData table (see *SoftwareLicenseComputerProblemData Table* on page 377)
- SoftwareLicenseComputerProblemType table (see *SoftwareLicenseComputerProblemType Table* on page 378)
- SoftwareLicenseContract table (see *SoftwareLicenseContract Table* on page 378)
- SoftwareLicenseCoresConsumedData table (see *SoftwareLicenseCoresConsumedData Table* on page 379)
- SoftwareLicenseCreation table (see *SoftwareLicenseCreation Table* on page 380)

- SoftwareLicenseDefinition table (see *SoftwareLicenseDefinition Table* on page 380)
- SoftwareLicenseDuration table (see *SoftwareLicenseDuration Table* on page 382)
- SoftwareLicenseExemptionReason table (see *SoftwareLicenseExemptionReason Table* on page 382)
- SoftwareLicenseExemptionRole table (see *SoftwareLicenseExemptionRole Table* on page 383)
- SoftwareLicenseGroupAllocationReportingType table (see *SoftwareLicenseGroupAllocationReportingType Table* on page 384)
- SoftwareLicenseGroupAssignmentHistory table (see *SoftwareLicenseGroupAssignmentHistory Table* on page 385)
- SoftwareLicenseGroupAssignmentHistoryType table (see *SoftwareLicenseGroupAssignmentHistoryType Table* on page 386)
- SoftwareLicenseGroupBreachStatus table (see *SoftwareLicenseGroupBreachStatus Table* on page 387)
- SoftwareLicenseGroupPointsConsumedData table (see *SoftwareLicenseGroupPointsConsumedData Table* on page 387)
- SoftwareLicenseLMTPointsConsumedData table (see *SoftwareLicenseLMTPointsConsumedData Table* on page 389)
- SoftwareLicenseKey table (see *SoftwareLicenseKey Table* on page 389)
- SoftwareLicenseKeyType table (see *SoftwareLicenseKeyType Table* on page 390)
- SoftwareLicenseMetric table (see *SoftwareLicenseMetric Table* on page 391)
- SoftwareLicensePartitioningDefault table (see *SoftwareLicensePartitioningDefault Table* on page 395)
- SoftwareLicensePoints table (see *SoftwareLicensePoints Table* on page 395)
- SoftwareLicensePointsConsumedData table (see *SoftwareLicensePointsConsumedData Table* on page 397)
- SoftwareLicensePointsConsumedReasonData table (see *SoftwareLicensePointsConsumedReasonData Table* on page 397)
- SoftwareLicensePointsConsumedReasonType table (see *SoftwareLicensePointsConsumedReasonType Table* on page 398)
- SoftwareLicensePointsDefault table (see *SoftwareLicensePointsDefault Table* on page 399)
- SoftwareLicensePointsRule table (see *SoftwareLicensePointsRule Table* on page 400)
- SoftwareLicensePointsRuleSet table (see *SoftwareLicensePointsRuleSet Table* on page 403)
- SoftwareLicenseProcessorPointsData table (see *SoftwareLicenseProcessorPointsData Table* on page 403)
- SoftwareLicensePropertyValue table (see *SoftwareLicensePropertyValue Table* on page 405)
- SoftwareLicenseProposalStatus table (see *SoftwareLicenseProposalStatus Table* on page 406)
- SoftwareLicensePurchaseType table (see *SoftwareLicensePurchaseType Table* on page 406)
- SoftwareLicenseReservation table (see *SoftwareLicenseReservation Table* on page 407)

- SoftwareLicenseReservationStatus table (see *SoftwareLicenseReservationStatus Table* on page 408)
- SoftwareLicenseReservationType table (see *SoftwareLicenseReservationType Table* on page 408)
- SoftwareLicenseScopeTag table (see *SoftwareLicenseScopeTag Table* on page 409)
- SoftwareLicenseScopeTagType table (see *SoftwareLicenseScopeTagType Table* on page 409)
- SoftwareLicenseScoping table (see *SoftwareLicenseScoping Table* on page 410)
- SoftwareLicenseSecondUseMappingData table (see *SoftwareLicenseSecondUseMappingData Table* on page 410)
- SoftwareLicenseSnapshot table (see *SoftwareLicenseSnapshot Table* on page 411)
- SoftwareLicenseTierType table (see *SoftwareLicenseTierType Table* on page 412)
- SoftwareLicenseType table (see *SoftwareLicenseType Table* on page 413)
- SoftwareLicenseTypeChangeProposal table (see *SoftwareLicenseTypeChangeProposal Table* on page 416)
- SoftwareLicenseTypePriority table (see *SoftwareLicenseTypePriority Table* on page 417)
- SoftwareLicenseTypeProperty table (see *SoftwareLicenseTypeProperty Table* on page 418)
- SoftwareLicenseUseRight table (see *SoftwareLicenseUseRight Table* on page 419)
- SoftwareLicenseUseRightIBM table (see *SoftwareLicenseUseRightIBM Table* on page 420)
- SoftwareLicenseUseRightName table (see *SoftwareLicenseUseRightName Table* on page 421)
- SoftwareLicenseUseRightProposal table (see *SoftwareLicenseUseRightProposal Table* on page 423)
- SoftwareRecognition table (see *SoftwareRecognition Table* on page 425)
- SoftwareSKULookup table (see *SoftwareSKULookup Table* on page 427)
- SoftwareSku table (see *SoftwareSku Table* on page 427)
- SoftwareTitle table (see *SoftwareTitle Table* on page 428)
- SoftwareTitleAction table (see *SoftwareTitleAction Table* on page 431)
- SoftwareTitleClassification table (see *SoftwareTitleClassification Table* on page 432)
- SoftwareTitleEdition table (see *SoftwareTitleEdition Table* on page 433)
- SoftwareTitleEx table (see *SoftwareTitleEx Table* on page 433)
- SoftwareTitleFileEvidence table (see *SoftwareTitleFileEvidence Table* on page 435)
- SoftwareTitleHierarchy table (see *SoftwareTitleHierarchy Table* on page 435)
- SoftwareTitleHierarchyEx table (see *SoftwareTitleHierarchyEx Table* on page 437)
- SoftwareTitleInstallerEvidence table (see *SoftwareTitleInstallerEvidence Table* on page 437)
- SoftwareTitleLicense table (see *SoftwareTitleLicense Table* on page 438)
- SoftwareTitleLicenseProposal table (see *SoftwareTitleLicenseProposal Table* on page 439)

- SoftwareTitleLicenseProposalAction table (see *SoftwareTitleLicenseProposalAction Table* on page 440)
- SoftwareTitleLicenseReason table (see *SoftwareTitleLicenseReason Table* on page 441)
- SoftwareTitleOracle table (see *SoftwareTitleOracle Table* on page 441)
- SoftwareTitleProduct table (see *SoftwareTitleProduct Table* on page 442)
- SoftwareTitleProperty table (see *SoftwareTitleProperty Table* on page 443)
- SoftwareTitlePropertyValue table (see *SoftwareTitlePropertyValue Table* on page 444)
- SoftwareTitlePublisher table (see *SoftwareTitlePublisher Table* on page 445)
- SoftwareTitleRegistryEvidence table (see *SoftwareTitleRegistryEvidence Table* on page 445)
- SoftwareTitleSuite table (see *SoftwareTitleSuite Table* on page 446)
- SoftwareTitleSuiteEx table (see *SoftwareTitleSuiteEx Table* on page 446)
- SoftwareTitleType table (see *SoftwareTitleType Table* on page 447)
- SoftwareTitleVersion table (see *SoftwareTitleVersion Table* on page 448)
- SoftwareTitleWMIEvidence table (see *SoftwareTitleWMIEvidence Table* on page 449)
- SoftwareUserLicensePointsConsumedData table (see *SoftwareUserLicensePointsConsumedData Table* on page 449)
- SoftwareUserLicensePointsConsumedSuggested table (see *SoftwareUserLicensePointsConsumedSuggested Table* on page 450)
- SoftwareUserLicensePointsConsumedSuggestedHistory table (see *SoftwareUserLicensePointsConsumedSuggestedHistory Table* on page 451)
- SoftwareUserLicensePointsHistory table (see *SoftwareUserLicensePointsHistory Table* on page 452)
- Tag table (see *Tag Table* on page 452)
- TargetOperatingSystemType table (see *TargetOperatingSystemType Table* on page 453)
- VDI table (see *VDI Table* on page 454)
- VDIEndPointAccess table (see *VDIEndPointAccess Table* on page 455)
- VDIGroup table (see *VDIGroup Table* on page 455)
- VDISite table (see *VDISite Table* on page 456)
- VDITemplate table (see *VDITemplate Table* on page 457)
- VDIUser table (see *VDIUser Table* on page 457)
- WMIEvidence table (see *WMIEvidence Table* on page 458)
- WMIEvidenceMatchCount table (see *WMIEvidenceMatchCount Table* on page 459)



## AccessMode Table

The `AccessMode` table holds the available states an application can be considered accessed.

**Table 275: Database columns for `AccessMode` table**

Database Column	Details
<code>AccessModeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>AccessMode</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Local</li> <li>• 2 = App-V</li> <li>• 3 = XenApp</li> <li>• 4 = XenDesktop</li> <li>• 5 = VMware View</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an access mode. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the access mode resource string has no translation.</p>

## Cluster Table

The `Cluster` table stores information about a logical group of computers which form a cluster.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 276: Database columns for `Cluster` table**

Database Column	Details
<code>ClusterID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for the cluster.</p>



Database Column	Details
ParentClusterID	<i>Type:</i> integer. Key. Nullable An optional link back to a parent cluster.
ExternalName	<i>Type:</i> text (max 256 characters). Nullable The identifier of the cluster in the external cluster management system.
Name	<i>Type:</i> text (max 256 characters). Key The user-visible name of the cluster.
Namespace	<i>Type:</i> text (max 256 characters). Key. Nullable The name of the domain or datacenter containing the cluster.
ClusterTypeID	<i>Type:</i> integer. Key Foreign key to the <code>ClusterType</code> table.
ComplianceComputerInventorySourceTypeID	<i>Type:</i> integer Whether this cluster has ever been reported in inventory, or has been manually created and maintained. Foreign key to the <code>ComplianceComputerInventorySourceType</code> table.
InventoryDate	<i>Type:</i> datetime. Nullable The date the computer last had inventory reported.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The name of the operator who last updated the computer details.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the cluster was created.
InventoryAgent	<i>Type:</i> text (max 64 characters). Nullable The name of the person or tool that performed the last inventory.
DRS	<i>Type:</i> boolean. Nullable Whether Distributed Resource Scheduler (DRS) is enabled

Database Column	Details
DPM	Type: boolean. Nullable Whether Distributed Power Management (DPM) is enabled

## ClusterComputer Table

The `ClusterComputer` table stores information about the relationship of computers to a cluster.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 277: Database columns for `ClusterComputer` table**

Database Column	Details
<code>ClusterComputerID</code>	Type: integer. Key. Generated ID A unique identifier for the cluster computer.
<code>ClusterID</code>	Type: integer. Key Foreign key to the <code>Cluster</code> table.
<code>ComplianceComputerID</code>	Type: integer. Key Foreign key to the <code>ComplianceComputer</code> table.
<code>ClusterNodeTypeID</code>	Type: integer Foreign key to the <code>ClusterNodeType</code> table.
<code>ComplianceComputerInventorySourceTypeID</code>	Type: integer Whether this cluster computer relationship has ever been reported in inventory, or has been manually created and maintained. Foreign key to the <code>ComplianceComputerInventorySourceType</code> table.

## ClusterHostAffinityRule Table

The `ClusterHostAffinityRule` table stores rules that define whether there is affinity between different VM groups and host groups within a cluster.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 278: Database columns for ClusterHostAffinityRule table**

Database Column	Details
ClusterHostAffinityRuleID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each ClusterHostAffinityRule.
ClusterHostAffinityRuleTypeID	<i>Type:</i> integer A unique identifier indicating a type of Cluster Host Affinity Rule.
Name	<i>Type:</i> text (max 256 characters). Key The name assigned to an affinity rule.
HostGroupClusterID	<i>Type:</i> integer The unique identifier of the host group to which the affinity rule applies. Foreign key to the Cluster table.
VMGroupClusterID	<i>Type:</i> integer The unique identifier of the VM group to which the affinity rule applies. Foreign key to the Cluster table.
ClusterID	<i>Type:</i> integer. Key Foreign key to the Cluster table.
ComplianceComputerInventorySourceTypeID	<i>Type:</i> integer Whether this cluster host affinity rule has ever been reported in inventory, or has been manually created and maintained. Foreign key to the ComplianceComputerInventorySourceType table.

## ClusterHostAffinityRuleType Table

ClusterHostAffinityRuleType is a static table listing all of the types of cluster host affinity rules.

**Table 279: Database columns for ClusterHostAffinityRuleType table**

Database Column	Details
ClusterHostAffinityRuleTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ClusterHostAffinityRuleType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = must run on (VMs in the LHS group MUST run on hosts specified in the RHS group )</li> <li>• 2 = must not run on (VMs in the LHS group MUST NOT run on any of the hosts specified in the RHS group )</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing the type of a cluster host affinity rule. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## ClusterNodeType Table

`ClusterNodeType` is a static table listing all of the roles a computer can have in a cluster.

**Table 280: Database columns for ClusterNodeType table**

Database Column	Details
ClusterNodeTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ClusterNodeType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Active (a node that is powered on and in use.)</li> <li>• 2 = Passive (a node that is powered on but not in use unless an active node fails over to it)</li> <li>• 3 = Hot (an active node—IBM nomenclature)</li> <li>• 4 = Warm (a passive node—IBM nomenclature)</li> <li>• 5 = Cold (a node that is powered off—IBM nomenclature)</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p>

Database Column	Details
	The unique name of the localizable resource string representing a cluster node type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.

## ClusterType Table

`ClusterType` is a static table listing all of the types of a cluster.

**Table 281: Database columns for `ClusterType` table**

Database Column	Details
ClusterTypeID	<i>Type:</i> integer. Key. Generated ID  A unique identifier for each <code>ClusterType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = vMotion (a mobility cluster based on VMWare ESX technology)</li> <li>• 2 = Hyper-V (a mobility cluster based on Microsoft's Hyper-V virtualization technology )</li> <li>• 5 = Oracle VM (a cluster based on Oracle VM virtualization technology )</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key  The unique name of the localizable resource string representing a cluster type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.
XMLFile	<i>Type:</i> text. Nullable  The layout of the property dialog for this type of cluster, stored in XML format.

## ComplianceComputerSnapshot Table

The `ComplianceComputerSnapshot` table lists all the snapshotted computers.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 282: Database columns for ComplianceComputerSnapshot table**

Database Column	Details
ComplianceComputerID	Type: integer. Key The snapshotted ComplianceComputerID.
ComputerName	Type: text (max 256 characters). Nullable The snapshotted computer name.
Domain	Type: text (max 256 characters). Nullable The snapshotted computer domain name.
LocationID	Type: text (max 128 characters). Key. Nullable The snapshotted LocationID.
BusinessUnitID	Type: text (max 128 characters). Key. Nullable The snapshotted BusinessUnitID.
CostCenterID	Type: text (max 128 characters). Key. Nullable The snapshotted CostCenterID.
CategoryID	Type: text (max 128 characters). Key. Nullable The snapshotted CategoryID.
LicenseMeasurementID	Type: integer. Key The snapshot ID. Foreign key to the LicenseMeasurement table.

## ComplianceComputerTag Table

Reserved for future development.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 283: Database columns for ComplianceComputerTag table**

Database Column	Details
ComplianceComputerID	Type: integer. Key Foreign key to the ComplianceComputer table
TagID	Type: integer. Key Foreign key to the Tag table.

## ComplianceUserSnapshot Table

The ComplianceUserSnapshot table lists all the users for each snapshot.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 284: Database columns for ComplianceUserSnapshot table**

Database Column	Details
ComplianceUserID	Type: integer. Key The snapshotted ComplianceUserID.
UserName	Type: text (max 256 characters). Nullable The snapshotted user name.
Domain	Type: text (max 256 characters). Nullable The snapshotted user domain name.
LocationID	Type: text (max 128 characters). Key. Nullable The snapshotted LocationID.
BusinessUnitID	Type: text (max 128 characters). Key. Nullable The snapshotted BusinessUnitID.
CostCenterID	Type: text (max 128 characters). Key. Nullable The snapshotted CostCenterID.
CategoryID	Type: text (max 128 characters). Key. Nullable

Database Column	Details
	The snapshotted CategoryID.
LicenseMeasurementID	Type: integer. Key The snapshot ID. Foreign key to the LicenseMeasurement table.

## ComplianceUserTag Table

Reserved for future use.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 285: Database columns for ComplianceUserTag table**

Database Column	Details
ComplianceUserID	Type: integer. Key Foreign key to the ComplianceUser table.
TagID	Type: integer. Key Foreign key to the Tag table.

## DatabaseMutex Table

The DatabaseMutex table lists all current database mutexes.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 286: Database columns for DatabaseMutex table**

Database Column	Details
DatabaseMutexID	Type: integer. Key. Generated ID



Database Column	Details
	A unique identifier for the database mutex.
Name	Type: text (max 256 characters). Key The name of the mutex.

## EntitlementRecommendation Table

EntitlementRecommendation is a table listing all of the recommendations that have been made to link entitlements to licenses.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 287: Database columns for EntitlementRecommendation table**

Database Column	Details
EntitlementRecommendationID	Type: integer. Key. Generated ID A unique identifier for this recommendation.
SoftwareLicenseID	Type: integer. Key. Nullable The license affected by this recommendation, null if a new license is being created. Foreign key to the SoftwareLicense table.
SoftwareLicenseDefinitionID	Type: integer. Key. Nullable The license definition of the new license being created. Foreign key to the SoftwareLicenseDefinition table.
SoftwareLicenseDefinition	Type: text. Nullable Encrypted XML definition of the customised license being created if any.
MaintenanceDefinition	Type: text. Nullable Encrypted XML definition of the maintenance being applied to the license associated with this recommendation.
ContractID	Type: integer. Key. Nullable The contract affected by this recommendation, if any. Foreign key to the Contract table.

Database Column	Details
MaintenanceContractID	<i>Type:</i> integer. Nullable The contract providing maintenance for this recommendation, if any. Foreign key to the <code>Contract</code> table.
ProcessActionID	<i>Type:</i> integer. Key. Nullable The action that is recommended by this recommendation. Foreign key to the <code>ProcessAction</code> table.
EntitlementRecommendationStateID	<i>Type:</i> integer. Nullable The state that the recommendation is in. Foreign key to the <code>EntitlementRecommendationState</code> table.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.
DoTransferSoftwareLicenseAllocations	<i>Type:</i> boolean. Nullable Indicates whether to transfer Group Assignments and Allocations when performing an upgrade and all the entitlements are transferred to the new license.

## EntitlementRecommendationState Table

`EntitlementRecommendationState` is a static table listing all of the states a entitlement recommendation or transaction can be in.

**Table 288: Database columns for `EntitlementRecommendationState` table**

Database Column	Details
EntitlementRecommendationStateID	<i>Type:</i> integer. Key. Generated ID

Database Column	Details
	<p>A unique identifier for each <code>EntitlementRecommendationState</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Automatically recommended</li> <li>• 2 = Manually created</li> <li>• 3 = Edited by an operator</li> <li>• 4 = Accepted by an operator or automatically</li> <li>• 5 = Rolled back by an operator</li> <li>• 6 = Deferred by an operator</li> <li>• 7 = Failed to be accepted.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing the entitlement recommendation's state. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 256 characters)</p> <p>The text to display if the state resource string has no translation.</p>

## EntitlementTransaction Table

`EntitlementTransaction` is a table listing all of the recommendations that have been made to link entitlements to licenses.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 289: Database columns for `EntitlementTransaction` table**

Database Column	Details
<code>EntitlementTransactionID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for this transaction.</p>
<code>EntitlementRecommendationID</code>	<p><i>Type:</i> integer. Key. Nullable</p>

Database Column	Details
	The recommendation this transaction is related to if any. Foreign key to the <code>EntitlementRecommendation</code> table.
<code>SoftwareLicenseID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The license affected by this recommendation. If a new license is being created from a recommendation but the recommendation is pending, the value of this field is null. The license identified depends on the <code>EntitlementTransactionType</code>. For a recommendation, this could be the license being updated (the “from” license) or it could be the new license (the “to” license). Foreign key to the <code>SoftwareLicense</code> table.</p>
<code>PurchaseOrderDetailID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The purchase order line associated with this transaction. Foreign key to the <code>PurchaseOrderDetail</code> table.</p>
<code>Adjustment</code>	<p><i>Type:</i> integer. Nullable</p> <p>The (potentially partial) amount of the purchased license quantity that is being applied to the license.</p>
<code>OtherCandidates</code>	<p><i>Type:</i> boolean. Nullable</p> <p>Whether there were other licenses which could have been recommended.</p>
<code>EntitlementTransactionTypeID</code>	<p><i>Type:</i> integer. Nullable</p> <p>The type of the transaction. Foreign key to the <code>EntitlementTransactionType</code> table.</p>
<code>EntitlementRecommendationStateID</code>	<p><i>Type:</i> integer. Nullable</p> <p>The state that the transaction is in. Foreign key to the <code>EntitlementRecommendationState</code> table.</p>
<code>IsDeferred</code>	<p><i>Type:</i> boolean</p> <p>Flags the entitlement transaction whether it is deferred for later processing.</p>
<code>TransactionUser</code>	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The operator who last updated the record.</p>
<code>TransactionDate</code>	<p><i>Type:</i> datetime. Nullable</p> <p>The date the record was last updated.</p>
<code>PreviousMaintenanceDefinition</code>	<p><i>Type:</i> text. Nullable</p> <p>Encrypted XML definition of the maintenance previously applied to the license associated with this transaction.</p>

Database Column	Details
PreviousMaintenanceContractID	<p><i>Type:</i> integer. Nullable</p> <p>The ID of the contract previously giving maintenance to the license associated with this transaction. Foreign key to the <code>Contract</code> table.</p>
LicenseNameMatched	<p><i>Type:</i> boolean</p> <p>Indicates whether or not there was a license name match.</p>
PrimaryApplicationMatched	<p><i>Type:</i> boolean</p> <p>Indicates whether or not there was a primary application match.</p>
AnyApplicationMatched	<p><i>Type:</i> boolean</p> <p>Indicates whether or not there was a match on any application.</p>
MaintenanceSettingsMatched	<p><i>Type:</i> boolean</p> <p>Indicates whether or not there was a match based on maintenance settings.</p>
EnterpriseGroupMatched	<p><i>Type:</i> boolean</p> <p>Indicates whether or not there was a match based on enterprise groups.</p>
NumberOfVersionsDifferent	<p><i>Type:</i> integer</p> <p>Indicated the number of versions between the version being upgraded to from the version being upgraded from.</p>
EntitlementTransactionStateID	<p><i>Type:</i> integer</p> <p>The state of the transaction. Foreign key to the <code>EntitlementTransactionState</code> table.</p>
AdjustmentDefault	<p><i>Type:</i> integer. Nullable</p> <p>The default amount of the purchased license quantity that is being applied to the license.</p>

## EntitlementTransactionOtherCandidate Table

`EntitlementTransactionOtherCandidate` is a table listing all of the other possible license recommendations that have been made to for entitlements.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 290: Database columns for EntitlementTransactionOtherCandidate table**

Database Column	Details
EntitlementTransactionOtherCandidateID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this possible candidate.
EntitlementTransactionID	<i>Type:</i> integer. Key The entitlement the recommendation belongs to.
SoftwareLicenseID	<i>Type:</i> integer. Key The license affected by this recommendation.
UpgradeFrom	<i>Type:</i> boolean Indicates whether this license was a candidate to upgrade from or not.
LicenseNameMatched	<i>Type:</i> boolean Indicates whether or not there was a license name match.
PrimaryApplicationMatched	<i>Type:</i> boolean Indicates whether or not there was a primary application match.
AnyApplicationMatched	<i>Type:</i> boolean Indicates whether or not there was a match on any application.
MaintenanceSettingsMatched	<i>Type:</i> boolean Indicates whether or not there was a match based on maintenance settings.
EnterpriseGroupMatched	<i>Type:</i> boolean Indicates whether or not there was a match based on enterprise groups.
NumberOfVersionsDifferent	<i>Type:</i> integer Indicated the number of versions between the version being upgraded to from the version being upgraded from.

## EntitlementTransactionState Table

`EntitlementTransactionState` is a static table listing all of the states that can be associated with purchased entitlements.

**Table 291: Database columns for `EntitlementTransactionState` table**

Database Column	Details
<code>EntitlementTransactionStateID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>EntitlementTransactionState</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Enabled</li> <li>• 2 = Disabled</li> <li>• 3 = Always enabled.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing the enabled state of the transaction. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 256 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## EntitlementTransactionType Table

`EntitlementTransactionType` is a static table listing all of the types of transactions that can be performed associating purchased entitlements to a license.

**Table 292: Database columns for `EntitlementTransactionType` table**

Database Column	Details
<code>EntitlementTransactionTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>EntitlementTransactionType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Purchased entitlements added to license</li> <li>• 2 = Purchased entitlements removed from license</li> <li>• 3 = Purchased entitlements taken from this license for upgrade purposes</li> <li>• 4 = Entitlements adjusted manually on the license by an operator</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 5 = Maintenance entitlements adjusted on the license.</li> <li>• 6 = Maintenance entitlements adjusted manually on the license.</li> <li>• 7 = Upgrade entitlements adjusted manually on the license.</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing the type of transaction. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 256 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## EvidenceExistenceRule Table

`EvidenceExistenceRule` is a static table listing the rules to be applied to file evidence and its relationship to a software (application) title.

**Table 293: Database columns for `EvidenceExistenceRule` table**

Database Column	Details
EvidenceExistenceRuleID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>Unique identifier for each <code>EvidenceExistenceRule</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Required (the file evidence must be present for the title to be considered installed)</li> <li>• 2 = Not for recognition (not used for recognizing application installations - the presence of this file evidence does not guarantee installation of the title)</li> <li>• 3 = Not allowed (if the file evidence is present, the title is not installed).</li> <li>• 4 = At least one (the presence of any of the file evidence identified this way is enough for the title to be considered installed).</li> </ul>
RuleResourceString	<p><i>Type:</i> text (max 50 characters). Key</p> <p>The unique name of the localizable resource string representing an evidence rule. Foreign key to the <code>ComplianceResourceString</code> table.</p>
RuleDefaultString	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the rule resource string has no translation.</p>



## EvidenceStatus Table

The collection of status values for installation evidence.

**Table 294: Database columns for EvidenceStatus table**

Database Column	Details
EvidenceStatusID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for an evidence status. Possible values (and associated default names) are:</p> <ul style="list-style-type: none"> <li>• 1 = Active</li> <li>• 2 = Inactive</li> <li>• 3 = Unassigned</li> <li>• 4 = Ignored</li> <li>• 5 = Assigned.</li> </ul>
StatusResourceString	<p>Type: text (max 50 characters). Key</p> <p>The name of the resource string containing the text to display on the user interface.</p>
StatusDefaultString	<p>Type: text (max 100 characters)</p> <p>The value to display if there is no resource string available for this status.</p>

## FNMEAFeature Table

FNMEAFeature records additional license features, associated with a specific license, that have been imported from FlexNet Manager for Engineering Applications.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 295: Database columns for FNMEAFeature table**

Database Column	Details
FNMEAFeatureID	Type: integer. Key. Generated ID

Database Column	Details
	A unique identifier for the FNM-EA feature record.
Name	<i>Type:</i> text (max 256 characters) Name of the feature.
Version	<i>Type:</i> text (max 60 characters). Nullable Version of the feature.
PublisherID	<i>Type:</i> integer. Nullable The publisher of the license associated with this feature. Foreign key to the <code>Vendor</code> table.
NumberPurchased	<i>Type:</i> integer The quantity of purchased feature entities.
NumberInstalled	<i>Type:</i> integer The quantity of software installations accounted for by this feature.
SoftwareLicense ComplianceStatusID	<i>Type:</i> integer The compliance status of the license associated with this feature. Defaults to <code>Compliant</code> . Foreign key to the <code>SoftwareLicenseComplianceStatus</code> table.

## FNMEALicensedFeature Table

`FNMEALicensedFeature` associated imported FlexNet Manager for Engineering Applications features with software licenses.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 296: Database columns for `FNMEALicensedFeature` table**

Database Column	Details
FNMEAFeatureID	<i>Type:</i> integer. Key The feature associated with a license. Foreign key to the <code>FNMEAFeature</code> table.

Database Column	Details
SoftwareLicenseID	<i>Type:</i> integer. Key The license associated with a feature. Foreign key to the <code>SoftwareLicense</code> table.
QuantityPerLicense	<i>Type:</i> integer The quantity of feature entitlements per associated license purchased.
ProductID	<i>Type:</i> text (max 256 characters). Key The external identifier of the product the linked feature is a part of.
ComplianceConnectionID	<i>Type:</i> integer. Key An identifier for the data source the product has been imported from.

## FileEvidenceCompany Table

`FileEvidenceCompany` contains the company names appearing in the headers of files used as evidence that an application is installed.

**Table 297: Database columns for `FileEvidenceCompany` table**

Database Column	Details
FileEvidenceCompanyID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this company.
Company	<i>Type:</i> text (max 100 characters). Key The name of the company.

## FileEvidenceEx Table

The `FileEvidenceEx` table contains additional information on the file evidence managed by FlexNet Manager Suite.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 298: Database columns for FileEvidenceEx table**

Database Column	Details
FileEvidenceID	<i>Type:</i> integer. Key A unique identifier for an file evidence record.
OperatorManageStateID	<i>Type:</i> integer. Nullable The management responsibility for this information. Foreign key to the OperatorManageState table.
Ignored	<i>Type:</i> boolean. Nullable Set this field to <code>True</code> if the file evidence is not used for application recognition.

## FileEvidenceFile Table

FileEvidenceFile contains the names of the files used as evidence that an application is installed.

**Table 299: Database columns for FileEvidenceFile table**

Database Column	Details
FileEvidenceFileID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the file.
FileName	<i>Type:</i> text (max 256 characters). Key The name of the file.

## FileEvidenceLanguage Table

FileEvidenceLanguage contains the language names appearing in headers of files used as evidence that an application is installed.

**Table 300: Database columns for FileEvidenceLanguage table**

Database Column	Details
FileEvidenceLanguageID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this language.
Language	<i>Type:</i> text (max 200 characters). Key

Database Column	Details
	The name of the language.

## FileEvidenceMatchCount Table

`FileEvidenceMatchCount` tracks the number of times that each file evidence (rule) has been detected as installed and recorded in the data source. A separate count is kept for each file evidence rule, and for each data source.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 301: Database columns for FileEvidenceMatchCount table**

Database Column	Details
<code>FileEvidenceMatchCountID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A synthetic unique identifier is required, since <code>ComplianceConnectionID</code>, being nullable, cannot be included in the primary key.</p>
<code>FileEvidenceID</code>	<p><i>Type:</i> integer. Key</p> <p>The file evidence rule being matched. Foreign key to the <code>NewFileEvidence</code> table.</p>
<code>ComplianceConnectionID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The data source where the match is occurring. Foreign key to the <code>ComplianceConnection</code> table.</p>
<code>MatchedCount</code>	<p><i>Type:</i> integer</p> <p>The number of installed files in this data source matching this file evidence rule.</p>
<code>InstallCount</code>	<p><i>Type:</i> integer</p> <p>The number of physical application installations recognized in this data source using this file evidence rule.</p>

## FileEvidencePath Table

`FileEvidencePath` contains the file paths to files used as evidence that an application is installed.

**Table 302: Database columns for FileEvidencePath table**

Database Column	Details
FileEvidencePathID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this path.
FilePath	<i>Type:</i> text (max 400 characters). Key The content of the file path.

## GroupSnapshot Table

The `GroupSnapshot` table lists all the snapshotted groups.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 303: Database columns for GroupSnapshot table**

Database Column	Details
GroupID	<i>Type:</i> integer. Key The snapshotted GroupID.
GroupExID	<i>Type:</i> text (max 128 characters). Key The snapshotted GroupExID.
Path	<i>Type:</i> text (max 500 characters) The snapshotted Path.
LicenseMeasurementID	<i>Type:</i> integer. Key The snapshot ID. Foreign key to the <code>LicenseMeasurement</code> table.

## ImporterRun Table

The `ImporterRun` table lists all previously run imports.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 304: Database columns for `ImporterRun` table**

Database Column	Details
<code>ImporterRunID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for the import run.
<code>LicenseMeasurementID</code>	<i>Type:</i> integer. Key. Nullable The <code>LicenseMeasurementID</code> if a license reconcile was performed. Foreign key to the <code>LicenseMeasurement</code> table.
<code>StartDate</code>	<i>Type:</i> datetime. Nullable The time the import was started.
<code>EndDate</code>	<i>Type:</i> datetime. Nullable The time the import was completed.
<code>ImportSourcesAppliedDate</code>	<i>Type:</i> datetime. Nullable If non-licensing writers ran and completed successfully, this field will be set to the date/time of their completion. In effect, it records the application of data from the importer staging tables in to the core tables. This is the case even if the record as a whole is marked as a failure, as the writers processing will have already completed.
<code>Arguments</code>	<i>Type:</i> text (max 1024 characters) The command line arguments to the import.
<code>RunAs</code>	<i>Type:</i> text (max 1024 characters) The user who performed the import.
<code>Comment</code>	<i>Type:</i> text (max 1024 characters). Nullable Comments related to the import.
<code>EventLogSummaryID</code>	<i>Type:</i> integer. Key. Nullable The <code>EventLogSummaryID</code> for the import. Foreign key to the <code>EventLogSummary</code> table.
<code>Success</code>	<i>Type:</i> boolean. Key. Nullable Determines whether the import completed successfully.

# ImporterStepValidationIssue Table

The `ImporterStepValidationIssue` table lists any validation issues that occurred during an import, that the user may need to review.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 305: Database columns for `ImporterStepValidationIssue` table**

Database Column	Details
<code>ImporterStepValidationIssueID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for the import validation.
<code>ImporterRunID</code>	<i>Type:</i> integer. Key Foreign key to the <code>ImporterRun</code> table.
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key Foreign key to the <code>ComplianceConnection</code> table.
<code>ProcedureName</code>	<i>Type:</i> text (max 256 characters). Nullable The procedure that contains the issue.
<code>StepName</code>	<i>Type:</i> text (max 512 characters). Nullable The step that contains the issue.
<code>RowSkipped</code>	<i>Type:</i> boolean Source to object validation issue specifying if row skipped.
<code>ColErrorReason</code>	<i>Type:</i> integer. Nullable Source to object validation issue specifying reason for error on particular row.
<code>ColumnName</code>	<i>Type:</i> text (max 128 characters). Nullable Column name of the failed source to object validation issue.
<code>RowNumber</code>	<i>Type:</i> big integer. Nullable Row number of the failed source to object validation issue.
<code>AffectedItem</code>	<i>Type:</i> text (max 512 characters). Nullable



Database Column	Details
	An optional description for any further related item.
ImporterStepValidationIssueTypeID	Type: integer. Nullable Foreign key to the ImporterStepValidationIssueType table.
OccurrenceDate	Type: datetime. Nullable The time the issue was raised.

## ImporterStepValidationIssueType Table

ImporterStepValidationIssueType is a static table listing all of the validation issues that can occur on a ComplianceConnection.

**Table 306: Database columns for ImporterStepValidationIssueType table**

Database Column	Details
ImporterStepValidationIssueTypeID	Type: integer. Key. Generated ID
ResourceName	Type: text (max 256 characters). Key The unique name of the localizable resource string representing the ImporterStepValidationIssueType record. Foreign key to the ComplianceResourceString table.
DefaultValue	Type: text (max 256 characters) The text to display if the state resource string has no translation.

## InstalledFileEvidence Table

InstalledFileEvidence lists file evidence that has been installed on a computer.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 307: Database columns for InstalledFileEvidence table**

Database Column	Details
FileEvidenceID	<i>Type:</i> integer. Key An identifier for a file evidence record. Foreign key to the <code>NewFileEvidence</code> table.
ComplianceComputerID	<i>Type:</i> integer. Key The managed computer on which this evidence was found. Foreign key to the <code>ComplianceComputer</code> table.
AccessModeID	<i>Type:</i> integer. Key The state an application was considered accessed. Foreign key to the <code>AccessMode</code> table.

## InstalledInstallerAttribute Table

InstalledInstallerAttribute installer evidence attributes that exist on a computer.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 308: Database columns for InstalledInstallerAttribute table**

Database Column	Details
InstallerEvidenceID	<i>Type:</i> integer. Key An identifier for an installer evidence record. Foreign key to the <code>InstallerEvidence</code> table.
ComplianceComputerID	<i>Type:</i> integer. Key An identifier for a computer record. Foreign key to the <code>ComplianceComputer</code> table.
InstanceName	<i>Type:</i> text (max 256 characters). Key. Nullable The name of the instance on the computer where this installer evidence was found.
AttributeID	<i>Type:</i> integer. Key

Database Column	Details
	The installer evidence attribute. Foreign key to the <code>Attribute</code> table.
Value	Type: text The value of the attribute.

## InstalledInstallerEvidence Table

`InstalledInstallerEvidence` lists installer evidence that has been installed on a computer.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 309: Database columns for `InstalledInstallerEvidence` table**

Database Column	Details
<code>InstallerEvidenceID</code>	Type: integer. Key An identifier for an installer evidence record. Foreign key to the <code>InstallerEvidence</code> table.
<code>ComplianceComputerID</code>	Type: integer. Key An identifier for a computer record. Foreign key to the <code>ComplianceComputer</code> table.
<code>InstanceName</code>	Type: text (max 256 characters). Key. Nullable The name of the instance on the computer where this installer evidence was found.
<code>InstallDate</code>	Type: datetime. Nullable The install date of the installer evidence.
<code>DiscoveryDate</code>	Type: datetime. Nullable The date that the installer evidence was first seen.
<code>AccessModeID</code>	Type: integer. Key The state an application was considered accessed. Foreign key to the <code>AccessMode</code> table.

## InstalledInstanceReplacement Table

`InstalledInstanceReplacement` tracks the particular installations instances where a software suite replaced the installation record of its member application.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 310: Database columns for `InstalledInstanceReplacement` table**

Database Column	Details
<code>InstanceID</code>	<p>Type: integer. Key</p> <p>The installation instance of the software suite. Foreign key to the <code>Instance</code> table.</p>
<code>ReplacedSoftwareTitleID</code>	<p>Type: integer. Key</p> <p>Software title that has been replaced by its parent suite. Foreign key to the <code>SoftwareTitle</code> table.</p>

## InstalledSoftwareData Table

`InstalledSoftware` lists all the installations of an application (as defined in the `SoftwareTitle` table).



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 311: Database columns for `InstalledSoftwareData` table**

Database Column	Details
<code>InstalledSoftwareID</code>	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for an installed software record.</p>
<code>ComplianceComputerID</code>	<p>Type: integer. Key</p> <p>The computer on which the software is installed. Foreign key to the <code>ComplianceComputer</code> table.</p>

Database Column	Details
SoftwareTitleID	<i>Type:</i> integer. Key The software that is installed. Foreign key to the <code>SoftwareTitle</code> table.
IsUsed	<i>Type:</i> boolean Set this field to <code>True</code> if the software title is installed according to usage thresholds in the <code>SoftwareTitle</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key. Nullable The link to the license this install has been counted against. Foreign key to the <code>SoftwareLicense</code> table.
SoftwareLicenseAllocationID	<i>Type:</i> integer. Key. Nullable The link to the license allocation this installation has consumed. Foreign key to the <code>SoftwareLicenseAllocation</code> table.
IsLicensed	<i>Type:</i> boolean Set this field to <code>True</code> when this installation is licensed.
PointsUsed	<i>Type:</i> integer. Nullable The number of this installation consumes on a points-based license.
InstallDate	<i>Type:</i> datetime. Nullable The install date of the software.
DiscoveryDate	<i>Type:</i> datetime. Nullable The date that the software was first seen.
LastUsedDate	<i>Type:</i> datetime. Nullable The date that the software was last used.

## InstalledSoftwareRemoval Table

`InstalledSoftwareRemoval` table keeps track of software titles that have been recognised, but then removed due to precedence. This is typically because a higher quality (more specific) title has been found.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 312: Database columns for InstalledSoftwareRemoval table**

Database Column	Details
InstalledSoftwareID	<i>Type:</i> integer. Key Installation record for lower quality title. Foreign key to the <code>InstalledSoftware</code> table.
RemovedSoftwareTitleID	<i>Type:</i> integer. Key Software title whose installation is now being ignored due to the presence of a higher quality title. Foreign key to the <code>SoftwareTitle</code> table.

## InstalledSoftwareReplacement Table

`InstalledSoftwareReplacement` tracks which individual application installation records have (ever) been subsumed by recognition of their parent software suite installed on the same computer. Only the suite and its member application are linked here.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 313: Database columns for InstalledSoftwareReplacement table**

Database Column	Details
InstalledSoftwareID	<i>Type:</i> integer. Key The suite's installation record. Foreign key to the <code>InstalledSoftware</code> table.
ReplacedSoftwareTitleID	<i>Type:</i> integer. Key The software title that has been replaced by its parent suite. Foreign key to the <code>SoftwareTitle</code> table.

## InstalledSoftwareUsageData Table

`InstalledSoftwareUsage` records the end-users who are using a piece of software installed on a computer.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 314: Database columns for InstalledSoftwareUsageData table**

Database Column	Details
InstalledSoftwareUsageID	Type: integer. Key. Generated ID The unique identifier for this record.
ComplianceUserID	Type: integer. Key. Nullable The end-user using the application. Foreign key to the ComplianceUser table.
SoftwareLicenseID	Type: integer. Nullable The license that covers this installation. Foreign key to the SoftwareLicense table.
SoftwareLicenseAllocationID	Type: integer. Key. Nullable A link to any individual allocation that this installation consumes. Foreign key to the SoftwareLicenseAllocation table.
IsLicensed	Type: boolean Set this field to True if this usage is licensed.
UsageSessions	Type: integer The number of sessions for (or times that the application was used by) this end-user on this computer.
UsageActiveTime	Type: integer The amount of time this application was in active use (in the foreground) for this end-user on this computer.
ComplianceComputerID	Type: integer. Key The application. Foreign key to the ComplianceComputer table.
SoftwareTitleID	Type: integer. Key The application. Foreign key to the SoftwareTitle table.
LastUsedDate	Type: datetime. Nullable The date that the installed software was last used.
AccessModeID	Type: integer. Key

Database Column	Details
	The date that the installed software was last used.

## InstalledWMIEvidence Table

InstalledWMIEvidence lists WMI evidence that has been installed on a computer.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 315: Database columns for InstalledWMIEvidence table**

Database Column	Details
WMIEvidenceID	Type: integer. Key An identifier for a WMI evidence record. Foreign key to the WMIEvidence table.
ComplianceComputerID	Type: integer. Key An identifier for a computer record. Foreign key to the ComplianceComputer table.
AccessModeID	Type: integer. Key The state an application was considered accessed. Foreign key to the AccessMode table.
InstanceName	Type: text (max 256 characters). Key The name of the WMI class instance used in the source connection for the WMI evidence

## InstallerEvidence Table

InstallerEvidence lists installer evidence that is used to identify that a particular item of software (defined in the SoftwareTitle table) has been installed on a computer.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.



**Table 316: Database columns for InstallerEvidence table**

Database Column	Details
InstallerEvidenceID	<i>Type:</i> integer. Key. Generated ID A unique identifier for an installer evidence record.
InstallerEvidenceTypeID	<i>Type:</i> integer. Key Identifies the type of installer evidence. Defaults to MSI. Foreign key to the InstallerEvidenceType table.
DisplayName	<i>Type:</i> text (max 256 characters). Key The display name of the software as reported by the installer evidence.
Version	<i>Type:</i> text (max 72 characters). Key The version of the software as reported by the installer evidence.
Publisher	<i>Type:</i> text (max 200 characters). Key The publisher of the software as reported by the installer evidence.
OperatorManageStateID	<i>Type:</i> integer. Key The management responsibility for this information. Foreign key to the OperatorManageState table.
Ignored	<i>Type:</i> boolean Set this field to <code>True</code> if the installer evidence is not used for application recognition.
IsShared	<i>Type:</i> boolean

## InstallerEvidenceEx Table

The `InstallerEvidenceEx` table contains additional information on the installer evidence managed by FlexNet Manager Suite.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 317: Database columns for `InstallerEvidenceEx` table**

Database Column	Details
<code>InstallerEvidenceID</code>	<i>Type:</i> integer. Key A unique identifier for an installer evidence record.
<code>OperatorManageStateID</code>	<i>Type:</i> integer. Nullable The management responsibility for this information. Foreign key to the <code>OperatorManageState</code> table.
<code>Ignored</code>	<i>Type:</i> boolean. Nullable Set this field to <code>True</code> if the installer evidence is not used for application recognition.

## InstallerEvidenceMatchCount Table

`InstallerEvidenceMatchCount` tracks the number of times that each installer evidence (rule) has been detected as installed and recorded in the data source. A separate count is kept for each installer evidence rule, and for each data source.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 318: Database columns for `InstallerEvidenceMatchCount` table**

Database Column	Details
<code>InstallerEvidenceMatchCountID</code>	<i>Type:</i> integer. Key. Generated ID A synthetic unique identifier is required, since <code>ComplianceConnectionID</code> , being nullable, cannot be included in the primary key.
<code>InstallerEvidenceID</code>	<i>Type:</i> integer. Key The installer evidence which is being matched. Foreign key to the <code>InstallerEvidence</code> table.
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key. Nullable The data source where the match is occurring. Foreign key to the <code>ComplianceConnection</code> table.
<code>MatchedCount</code>	<i>Type:</i> integer

Database Column	Details
	The number of installed installer evidence records in this data source matching this installer evidence rule.
InstallCount	<i>Type:</i> integer The number of physical application installations recognized in this data source using this installer evidence rule.

## InstallerEvidenceType Table

`InstallerEvidenceType` is a static table listing the types of installer evidence that can be used to determine whether an item of software has been installed.

**Table 319: Database columns for `InstallerEvidenceType` table**

Database Column	Details
InstallerEvidenceTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>InstallerEvidenceType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Any</li> <li>• 2 = Add/Remove Programs</li> <li>• 3 = Software ID Tag</li> <li>• 4 = MSI</li> <li>• 5 = Unknown</li> <li>• 6 = ILMT</li> <li>• 7 = RPM</li> <li>• 8 = OS X App</li> <li>• 9 = LPP</li> <li>• 10 = SDUX</li> <li>• 11 = SUNPKG</li> <li>• 12 = IA</li> <li>• 13 = BEA</li> <li>• 14 = ISMP</li> <li>• 15 = IPS</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 16 = ADDM</li> <li>• 17 = OracleEBSModule</li> <li>• 18 = BDNA</li> <li>• 19 = FlexeralID</li> <li>• 20 = DPKG</li> <li>• 21 = App-V</li> <li>• 22 = OUI</li> <li>• 23 = IIM</li> <li>• 24 = DSPMQ</li> <li>• 25 = VMware</li> <li>• 26 = HPUD</li> </ul>
TypeResourceString	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an installer evidence type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
TypeDefaultString	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>
ImporterString	<p><i>Type:</i> text (max 100 characters)</p> <p>The text value provided by adapters when importing installer evidence.</p>

## LicenseBreachReason Table

`LicenseBreachReason` is a static table holding the collection of reasons why a license can be in breach.

**Table 320: Database columns for `LicenseBreachReason` table**

Database Column	Details
LicenseBreachReasonID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>LicenseBreachReason</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Installed Greater Than Purchased</li> <li>• 2 = Child License In Breach</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 3 = Install Linked to License has Invalid Sockets</li> <li>• 4 = Software License Does Not Meet Minimums</li> <li>• 5 = Software License Has Expired</li> <li>• 6 = Unlicensed Component Installed</li> <li>• 7 = Peak Consumed Quantity Greater Than Purchased.</li> <li>• 8 = Nested License In Breach.</li> </ul>
BreachResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a breach reason. Foreign key to the <code>ComplianceResourceString</code> table.</p>
BreachDefaultValue	<p><i>Type:</i> text (max 512 characters)</p> <p>The text to display if the reason resource string has no translation.</p>

## LicenseDefinitionTitle Table

`LicenseDefinitionTitle` associates software license definitions with their related applications.

**Table 321: Database columns for `LicenseDefinitionTitle` table**

Database Column	Details
SoftwareLicenseDefinitionID	<p><i>Type:</i> integer. Key</p> <p>The license definition. Foreign key to the <code>SoftwareLicenseDefinition</code> table.</p>
SoftwareRecognitionID	<p><i>Type:</i> text (max 100 characters). Key</p> <p>The encrypted FlexNet Manager Suite factory unique ID for the linked application in the Application Recognition Library.</p>

## LicenseDefinitionType Table

`LicenseDefinitionType` is a static table listing supported software license definition types, which are used to distinguish records downloaded from the Product Use Rights Library.

**Table 322: Database columns for LicenseDefinitionType table**

Database Column	Details
LicenseDefinitionTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for a definition type. Possible values (and associated default names) are:</p> <ul style="list-style-type: none"> <li>• 1 = License</li> <li>• 2 = Product</li> <li>• 3 = Usage Right.</li> </ul>
TypeName	<p><i>Type:</i> text (max 100 characters). Key</p> <p>Unique internal name for this definition type.</p>

## LicenseDefinitionUsageRight Table

`LicenseDefinitionUsageRight` associates software license definitions and Application Recognition Library software applications to recommended usage rights.

**Table 323: Database columns for LicenseDefinitionUsageRight table**

Database Column	Details
LicenseDefinitionFactoryUID	<p><i>Type:</i> text (max 100 characters). Key</p> <p>The encrypted factory unique ID for a license definition or ARL application.</p>
UsageRightFactoryUID	<p><i>Type:</i> text (max 100 characters). Key</p> <p>The encrypted factory unique ID for a usage right template.</p>

## LicenseMeasurement Table

The `LicenseMeasurement` table is used to store license measurement snapshots.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 324: Database columns for LicenseMeasurement table**

Database Column	Details
LicenseMeasurementID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the license measurement.
MeasurementCode	<i>Type:</i> text (max 128 characters) The unique code for this measurement.
MeasurementTime	<i>Type:</i> datetime. Key The date and time this measurement was started.
MeasurementEndTime	<i>Type:</i> datetime. Nullable The date and time this measurement was completed.
Success	<i>Type:</i> boolean Determines whether the measurement completed successfully.
Description	<i>Type:</i> text (max 50 characters) The description of this measurement.
IsPartial	<i>Type:</i> boolean Indicate whether this licence run was a partial run or not.

## LicenseSimulation Table

A `LicenseSimulation` is made up of an initial scenario, and a cloned version of this scenario. The user can modify the rows in this cloned scenario.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 325: Database columns for LicenseSimulation table**

Database Column	Details
LicenseSimulationID	<i>Type:</i> integer. Key. Generated ID Unique ID for the <code>LicenseSimulation</code> table.

Database Column	Details
LicenseSimulationScenarioID	<i>Type:</i> integer. Key Foreign key to the <code>LicenseSimulationScenario</code> table.
LastModified	<i>Type:</i> datetime The last time this simulation was modified.
ComplianceOperatorID	<i>Type:</i> integer. Key The compliance operator responsible for this scenario
DisplayName	<i>Type:</i> text (max 256 characters). Nullable The name given to this simulation by the owner/operator.
DisplayRateID	<i>Type:</i> integer. Nullable The rate to be used to display all price values in this simulation. Foreign key to the <code>CurrencyRate</code> table. If null, then the user's default can be used.

## LicenseSimulationBreachStatus Table

`LicenseSimulationBreachStatus` is a static table listing all of the breach states a license can be in, once it is modelled in a Simulation.

**Table 326: Database columns for `LicenseSimulationBreachStatus` table**

Database Column	Details
LicenseSimulationBreachStatusID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>LicenseSimulationBreachStatus</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Still compliant</li> <li>• 2 = Still in breach</li> <li>• 3 = Now compliant</li> <li>• 4 = Now in breach.</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing breach status in a license simulation. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters)



Database Column	Details
	The text to display if the type resource string has no translation.

## LicenseSimulationChangeType Table

`LicenseSimulationChangeType` is a static table listing all the types of operations that can be applied as changes to simulation data

**Table 327: Database columns for `LicenseSimulationChangeType` table**

Database Column	Details
<code>LicenseSimulationChangeTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>LicenseSimulationChangeType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Unchanged</li> <li>• 2 = Added</li> <li>• 3 = Deleted</li> <li>• 4 = Modified</li> <li>• 5 = Moved.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a computer role. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## LicenseSimulationHWDDetails Table

`LicenseSimulationHWDDetails` stores a complete snapshot of hardware data for simulations. The `LicenseSimulationScenario` associated with each record could be an original snapshot of data, or a user modifiable scenario.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 328: Database columns for LicenseSimulationHWDetails table**

Database Column	Details
LicenseSimulationHWDetailsID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a hardware item that is part of a simulation scenario.
LicenseSimulationScenarioID	<i>Type:</i> integer. Key The simulation scenario this hardware item is part of. Foreign key to the LicenseSimulationScenario table.
Name	<i>Type:</i> text (max 256 characters). Nullable The friendly name for this hardware item.
LicenseSimulationRowTypeID	<i>Type:</i> integer The type of hardware for this item.
Manufacturer	<i>Type:</i> text (max 128 characters). Nullable The manufacturer of this hardware item. Typically applies to a virtualisation server.
ModelNo	<i>Type:</i> text (max 128 characters). Nullable The model number of this hardware item. Typically applies to a virtualisation server.
ChassisNumber	<i>Type:</i> text (max 128 characters). Nullable The chassis number of this hardware item. Typically applies to a virtualisation server.
SerialNo	<i>Type:</i> text (max 100 characters). Nullable The serial number of this hardware item. Typically applies to a virtualisation server or physical machine.
ProcessorType	<i>Type:</i> text (max 256 characters). Nullable The processor type of this hardware item.
MaxClockSpeed	<i>Type:</i> integer. Nullable The maximum clock speed of this hardware item.
PurchaseDate	<i>Type:</i> datetime. Nullable The date this hardware item was purchased on, if it has an associated Asset.
NumSockets	<i>Type:</i> integer. Nullable

Database Column	Details
	The number of physical CPU sockets of this hardware item.
PoolTypeID	<i>Type:</i> integer. Nullable The type of pool technology of this hardware item. Typically applies to resource pools. Foreign key to the <code>VMPoolType</code> table.
VMTypeID	<i>Type:</i> integer. Nullable The type of virtual machine technology of this hardware item. Typically applies to virtual machines. Foreign key to the <code>VMType</code> table.
OperatingSystem	<i>Type:</i> text (max 128 characters). Nullable The operating system running on this hardware item.
NumProcessors	<i>Type:</i> decimal. Nullable The number of processors available to this hardware item.
NumCores	<i>Type:</i> decimal. Nullable The number of cores available to this hardware item.
NumThreads	<i>Type:</i> integer. Nullable The number of threads available to this hardware item.
ParentLicense SimulationHWDetailsID	<i>Type:</i> integer. Key. Nullable The parent hardware item of this item.
HostLicenseSimulationH WDetailsID	<i>Type:</i> integer. Nullable The host hardware item of this item.
ComplianceComputerID	<i>Type:</i> integer. Key. Nullable The actual computer record for this hardware item. Foreign key to the <code>ComplianceComputer</code> table.
VMLayerID	<i>Type:</i> integer. Key. Nullable Internal unique identifier used when populating hardware items to create a new simulation.
LicenseSimulation ChangeTypeID	<i>Type:</i> integer Tracks the state of the hardware item, as it gets modified by the simulation user. Foreign key to the <code>LicenseSimulationChangeType</code> table.
ClusterID	<i>Type:</i> integer. Nullable

Database Column	Details
	The hardware cluster to which this computer belongs, if any. Foreign key to the <code>Cluster</code> table.
<code>AffinityEnabled</code>	<i>Type:</i> boolean Whether this VM is locked to its current host computer.
<code>CoreAffinity</code>	<i>Type:</i> text (max 256 characters). Nullable Contains the Core Affinity value for virtual machine

## LicenseSimulationLicenseDetails Table

`LicenseSimulationLicenseDetails` stores properties associated with each license included in a simulation scenario. The `LicenseSimulationScenario` associated with each record could be an original snapshot of data, or a user modifiable scenario.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 329: Database columns for `LicenseSimulationLicenseDetails` table**

Database Column	Details
<code>LicenseSimulationLicenseDetailsID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a license item that is part of a simulation scenario.
<code>OriginalLicenseSimulationLicenseDetailsID</code>	<i>Type:</i> integer. Nullable The original version of this license, that has not been modified by a simulation user.
<code>LicenseSimulationScenarioID</code>	<i>Type:</i> integer. Key The simulation scenario this software license is part of. Foreign key to the <code>LicenseSimulationScenario</code> table.
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key The software license for this simulation license. Foreign key to the <code>SoftwareLicense</code> table.
<code>UnitPrice</code>	<i>Type:</i> currency. Nullable

Database Column	Details
	The unit price associated with this license.
UnitPriceRateID	<i>Type:</i> integer. Nullable The rate for the total value. Foreign key to the <code>CurrencyRate</code> table.
LicenseSimulationChangeTypeID	<i>Type:</i> integer Tracks the state of the software license, as it gets modified by the simulation user. Foreign key to the <code>LicenseSimulationChangeType</code> table.

## LicenseSimulationResults Table

`LicenseSimulationResults` stores points consumed by each item in a simulation scenario against each license included in the simulation scenario.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 330: Database columns for `LicenseSimulationResults` table**

Database Column	Details
LicenseSimulationHWDetailsID	<i>Type:</i> integer. Key The hardware item for this license simulation result. Foreign key to the <code>LicenseSimulationHWDetails</code> table.
LicenseSimulationScenarioID	<i>Type:</i> integer. Key The scenario for this license simulation result. Foreign key to the <code>LicenseSimulationScenario</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key The software license for this license simulation result. Foreign key to the <code>SoftwareLicense</code> table.
InstalledCount	<i>Type:</i> decimal The number of processors/cores on which a software title licensed by the license is installed.
UsedCount	<i>Type:</i> decimal

Database Column	Details
	The number of processors/cores on which a software title licensed by the license is used.
CapacityCount	<i>Type:</i> decimal The number of processors/cores that apply to a software license under full capacity counting rules.
IsCapped	<i>Type:</i> boolean Does this layer implement hard partitioning for this license?
PointsFactor	<i>Type:</i> decimal The number of points consumed per processor/core on this computer for this license.
PointsConsumed	<i>Type:</i> decimal. Nullable The number of processor/core points required to cover the above InstalledCount.
PointsUsed	<i>Type:</i> decimal. Nullable The number of processor/core points required to cover the above UsedCount.
CapacityPointsConsumed	<i>Type:</i> decimal. Nullable The number of processor/core points required to cover the above CapacityCount.

## LicenseSimulationRowType Table

LicenseSimulationRowType is a static table listing all types of rows that can be displayed in the Simulation UI. Entries in the LicenseSimulationSWDetails table are assumed to be type 4 (Software installation)

**Table 331: Database columns for LicenseSimulationRowType table**

Database Column	Details
LicenseSimulationRowTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each LicenseSimulationRowType. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Host</li> <li>• 2 = Shared pool</li> <li>• 3 = Virtual Machine</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>4 = Software installation</li> <li>5 = Physical machine.</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing the type of a row in a license simulation. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## LicenseSimulationSWDetails Table

`LicenseSimulationSWDetails` stores a complete snapshot of software data for simulations. The `LicenseSimulationHWDetails` record associated with each `LicenseSimulationSWDetails` record could be part of an original snapshot of data, or a user modifiable scenario.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 332: Database columns for `LicenseSimulationSWDetails` table**

Database Column	Details
LicenseSimulationSWDetailsID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for a software installation that is part of a simulation scenario.</p>
LicenseSimulationHWDetailsID	<p><i>Type:</i> integer. Key</p> <p>The hardware item that this software title is installed on. Foreign key to the <code>LicenseSimulationHWDetails</code> table.</p>
LicenseSimulationScenarioID	<p><i>Type:</i> integer. Key</p> <p>The simulation scenario this software installation is part of. Foreign key to the <code>LicenseSimulationScenario</code> table.</p>
OriginalLicenseSimulationSWDetailsID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The original version of this software installation, that has not been modified by a simulation user.</p>

Database Column	Details
Name	<i>Type:</i> text (max 512 characters) The friendly name of this software installation.
SoftwareTitleID	<i>Type:</i> integer. Key The software title that is installed here. Foreign key to the <code>SoftwareTitle</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key The software license that this install is assigned to. Foreign key to the <code>SoftwareLicense</code> table.
LicenseSimulation ChangeTypeID	<i>Type:</i> integer Tracks the state of the software installation, as it gets modified by the simulation user. Foreign key to the <code>LicenseSimulationChangeType</code> table.
IsUsed	<i>Type:</i> boolean Set this field to <code>True</code> if the software title is installed according to usage thresholds in the <code>SoftwareTitle</code> table.

## LicenseSimulationScenario Table

A `LicenseSimulationScenario` is a set of hardware and software inventory details that are recorded at a particular point in time. A scenario can be modified by the user for the purposes of simulation.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 333: Database columns for `LicenseSimulationScenario` table**

Database Column	Details
LicenseSimulation ScenarioID	<i>Type:</i> integer. Key. Generated ID Unique ID for the <code>LicenseSimulationScenario</code> table.
OriginalLicense SimulationScenarioID	<i>Type:</i> integer. Nullable The original (unmodified) scenario that a user-modifiable scenario was based on



## LicenseStatus Table

`LicenseStatus` is a static table storing the collection of possible license states.

**Table 334: Database columns for `LicenseStatus` table**

Database Column	Details
<code>LicenseStatusID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>LicenseStatus</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Active</li> <li>• 2 = Retired</li> <li>• 3 = In Stock</li> <li>• 4 = Purchased</li> <li>• 5 = Received.</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a license status. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the status resource string has no translation.</p>

## NewFileEvidence Table

`NewFileEvidence` identifies files used as evidence that an application (defined in the `SoftwareTitle` table) has been installed on a computer. File evidence may have wildcards, so each record in this table should be considered a rule, which one or more physical files on a computer may match.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 335: Database columns for `NewFileEvidence` table**

Database Column	Details
<code>FileEvidenceID</code>	<p><i>Type:</i> integer. Key. Generated ID</p>

Database Column	Details
	A unique identifier for a file evidence record.
FileEvidenceFileID	<i>Type:</i> integer. Key The file name. Foreign key to the FileEvidenceFile table.
FileEvidenceCompanyID	<i>Type:</i> integer. Key The company publishing the software. Foreign key to the FileEvidenceCompany table.
FileEvidencePathID	<i>Type:</i> integer. Key. Nullable The file path where the file was located. Foreign key to the FileEvidencePath table.
FileEvidenceLanguageID	<i>Type:</i> integer. Key. Nullable The language identified in the file header. Foreign key to the FileEvidenceLanguage table.
FileVersion	<i>Type:</i> text (max 100 characters). Key The version number of the file used as evidence of software installation.
ProductName	<i>Type:</i> text (max 200 characters). Nullable The product name in the file header.
ProductVersion	<i>Type:</i> text (max 200 characters). Nullable The product version number in the file header.
Description	<i>Type:</i> text (max 200 characters). Key The description in the file header.
FileSize	<i>Type:</i> integer. Key. Nullable The size of the file.
OperatorManageStateID	<i>Type:</i> integer. Key The management responsibility for this information. Foreign key to the OperatorManageState table.
Ignored	<i>Type:</i> boolean Set this field to <code>True</code> to indicate that this file evidence is ignored for application recognition.
IsShared	<i>Type:</i> boolean

# OracleLegacyLicenseType Table

OracleLegacyLicenseType lists some of the legacy Oracle license types.

**Table 336: Database columns for OracleLegacyLicenseType table**

Database Column	Details
OracleLegacyLicenseTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each OracleLegacyLicenseType. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Named User</li> <li>• 2 = Named User Network license</li> <li>• 3 = Named User Single Server</li> <li>• 4 = Named User Multi Server</li> <li>• 5 = Concurrent Device</li> <li>• 6 = Concurrent Device Network License</li> <li>• 7 = UPU</li> <li>• 8 = Developer</li> <li>• 9 = Developer Network License</li> <li>• 10 = Concurrent User</li> <li>• 11 = Concurrent User Network License</li> <li>• 12 = Application Specific Full User Licensing</li> <li>• 13 = Embedded Software License</li> <li>• 14 = Site.</li> </ul>
OracleLegacyLicenseTypeResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an Oracle legacy license type. Foreign key to the ComplianceResourceString table.</p>
OracleLegacyLicenseTypeDefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## PODetailProcess Table

PODetailProcess records the processing steps taken when applying upgrades to software installations. The newly-purchased upgrade license is linked here to the original license being upgraded.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 337: Database columns for PODetailProcess table**

Database Column	Details
PurchaseOrderDetailID	Type: integer. Key The purchase order line that defines this upgrade. Foreign key to the PurchaseOrderDetail table.
FromSoftwareLicenseID	Type: integer. Key. Nullable The original software license to which an upgrade is being applied. Foreign key to the SoftwareLicense table.
ToSoftwareLicenseID	Type: integer. Key. Nullable The upgrade license referenced in the PO line and permitting the installation of the software upgrade. Foreign key to the SoftwareLicense table.
ProcessActionID	Type: integer The processing action taken with respect to this upgrade. Defaults to <i>Defer</i> . Foreign key to the ProcessAction table.
ProcessStateID	Type: integer. Key The resulting process state of the upgrade. Foreign key to the ProcessState table.
CreationDate	Type: datetime The date this record was created.

## PeriodType Table

PeriodType is a static table holding a collection of supported time periods to indicate the frequency of license charge-backs.

**Table 338: Database columns for `PeriodType` table**

Database Column	Details
<code>PeriodTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>PeriodType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = None</li> <li>• 2 = Weekly</li> <li>• 3 = Monthly</li> <li>• 4 = Quarterly</li> <li>• 5 = Yearly</li> <li>• 6 = Lump Sum.</li> </ul>
<code>PeriodTypeResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a period type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>PeriodTypeDefaultValue</code>	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## ProcessAction Table

`ProcessAction` is a static table holding a collection of possible actions that can be applied while processing a SKU, with a special focus on processing software license upgrades.

**Table 339: Database columns for `ProcessAction` table**

Database Column	Details
<code>ProcessActionID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ProcessAction</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Link to existing license</li> <li>• 2 = Create new from SKU</li> <li>• 3 = Create new from PO line</li> <li>• 4 = Upgrade license: Link to existing license and upgrade from existing license</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 5 = Upgrade license: Link to existing license and select upgrade from license</li> <li>• 6 = Upgrade license: Link to new license created from SKU and select upgrade from license</li> <li>• 7 = Remove from list</li> <li>• 8 = Upgrade license: Link to new license created from PO line and select upgrade from license</li> <li>• 9 = Create new from SKU with fixed maintenance</li> <li>• 10 = Create new from PO line with fixed maintenance</li> <li>• 11 = Create new from SKU with unlimited maintenance</li> <li>• 12 = Create new from PO line with unlimited maintenance</li> <li>• 13 = Create new from SKU with maintenance from contract</li> <li>• 14 = Create new from PO line with maintenance from contract</li> <li>• 15 = Apply contract maintenance to an existing license</li> <li>• 16 = Apply fixed maintenance to an existing license</li> <li>• 17 = Apply unlimited maintenance to an existing license</li> <li>• 18 = Apply contract maintenance to an existing license by SKU</li> <li>• 19 = Apply fixed maintenance to an existing license by SKU</li> <li>• 20 = Apply unlimited maintenance to an existing license by SKU</li> <li>• 21 = Apply contract maintenance to a non-existent license for SKU</li> <li>• 22 = Apply fixed maintenance to a non-existent license for SKU</li> <li>• 23 = Apply unlimited maintenance to a non-existent license for SKU</li> <li>• 24 = Upgrade license: Link to existing license and upgrade from existing license with contract maintenance</li> <li>• 25 = Upgrade license: Link to existing license and upgrade from existing license with fixed maintenance</li> <li>• 26 = Upgrade license: Link to existing license and upgrade from existing license with unlimited maintenance</li> <li>• 27 = Upgrade license: Link to existing license and select upgrade from license with contract maintenance</li> <li>• 28 = Upgrade license: Link to existing license and select upgrade from license with fixed maintenance</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 29 = Upgrade license: Link to existing license and select upgrade from license with unlimited maintenance</li> <li>• 30 = Upgrade license: Link to new license created from SKU and select upgrade from license with contract maintenance</li> <li>• 31 = Upgrade license: Link to new license created from SKU and select upgrade from license with fixed maintenance</li> <li>• 32 = Upgrade license: Link to new license created from SKU and select upgrade from license with unlimited maintenance</li> <li>• 33 = Upgrade license: Link to new license created from PO line and select upgrade from license with contract maintenance</li> <li>• 34 = Upgrade license: Link to new license created from PO line and select upgrade from license with fixed maintenance</li> <li>• 35 = Upgrade license: Link to new license created from PO line and select upgrade from license with unlimited maintenance</li> <li>• 36 = Apply maintenance to a contract</li> <li>• 37 = No recommendation</li> <li>• 38 = Create a new license</li> <li>• 39 = Create a new license with a maintenance contract</li> <li>• 40 = Create a new license with fixed maintenance</li> <li>• 41 = Create a new license with unlimited maintenance</li> <li>• 42 = Add entitlements to a license</li> <li>• 43 = Add entitlements to a license with a maintenance contract</li> <li>• 44 = Add entitlements to a license with fixed maintenance</li> <li>• 45 = Add entitlements to a license with unlimited maintenance</li> <li>• 46 = Upgrade to a new license</li> <li>• 47 = Upgrade to a new license with a maintenance contract</li> <li>• 48 = Upgrade to a new license with fixed maintenance</li> <li>• 49 = Upgrade to a new license with unlimited maintenance</li> <li>• 50 = Upgrade to an existing license</li> <li>• 51 = Upgrade to an existing license with a maintenance contract</li> <li>• 52 = Upgrade to an existing license with fixed maintenance</li> <li>• 53 = Upgrade to an existing license with unlimited maintenance</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 54 = Apply maintenance from a contract to an existing license</li> <li>• 55 = Apply fixed maintenance to an existing license</li> <li>• 56 = Apply unlimited maintenance to an existing license</li> </ul>
ProcessActionResourceName	<p>Type: text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an action. Foreign key to the <code>ComplianceResourceString</code> table.</p>
ProcessActionDefaultValue	<p>Type: text (max 256 characters)</p> <p>The text to display if the action resource string has no translation.</p>

## ProcessState Table

`ProcessState` is a static table holding the collection of processing states that a purchase order line containing a SKU can be left in.

**Table 340: Database columns for `ProcessState` table**

Database Column	Details
ProcessStateID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for each <code>ProcessState</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Unprocessed</li> <li>• 2 = Processed</li> <li>• 3 = Deferred</li> <li>• 4 = Discarded.</li> </ul>
ProcessStateResourceName	<p>Type: text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a processing state. Foreign key to the <code>ComplianceResourceString</code> table.</p>
ProcessStateDefaultValue	<p>Type: text (max 256 characters)</p> <p>The text to display if the state resource string has no translation.</p>



# ReconcileInstalledSoftwareData Table

A list of all the installations of an application, or item of software (as defined in the `SoftwareTitle` table).



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 341: Database columns for ReconcileInstalledSoftwareData table**

Database Column	Details
InstalledSoftwareID	<i>Type:</i> integer. Key A unique identifier for an installed software record.
ComplianceComputerID	<i>Type:</i> integer. Key The computer on which the software is installed. Foreign key to the <code>ComplianceComputer</code> table.
SoftwareTitleID	<i>Type:</i> integer. Key The software that is installed. Foreign key to the <code>SoftwareTitle</code> table.
IsUsed	<i>Type:</i> boolean. Key Set this field to <code>True</code> if the software title is installed according to usage thresholds in the <code>SoftwareTitle</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key. Nullable The link to the license this install has been counted against. Foreign key to the <code>SoftwareLicense</code> table.
SoftwareLicenseAllocationID	<i>Type:</i> integer. Key. Nullable The link to the license allocation this installation has consumed. Foreign key to the <code>SoftwareLicenseAllocation</code> table.
IsLicensed	<i>Type:</i> boolean Set this field to <code>True</code> when this installation is licensed.
PointsUsed	<i>Type:</i> integer. Nullable The number of this installation consumes on a points-based license.
AccessModeID	<i>Type:</i> integer. Key

Database Column	Details
	The access mode that indicates why this computer was associated with this software title.
LastUsedDate	Type: datetime. Nullable The date of the installed software was last used.

## ReconcileInstalledSoftwareUsageData Table

This is a staging table for `InstalledSoftwareUsage` that is used during license reconciliation process, to store calculated values, and then bulk update the main table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 342: Database columns for ReconcileInstalledSoftwareUsageData table**

Database Column	Details
ComplianceUserID	Type: integer. Key. Nullable The end-user using the application. Foreign key to the <code>ComplianceUser</code> table.
SoftwareLicenseID	Type: integer. Nullable The license that covers this installation. Foreign key to the <code>SoftwareLicense</code> table.
SoftwareLicenseAllocationID	Type: integer. Key. Nullable A link to any individual allocation that this installation consumes. Foreign key to the <code>SoftwareLicenseAllocation</code> table.
IsLicensed	Type: boolean Set this field to <code>True</code> if this usage is licensed.
UsageSessions	Type: integer The number of sessions for (or times that the application was used by) this end-user on this computer.
UsageActiveTime	Type: integer

Database Column	Details
	The amount of time this application was in active use (in the foreground) for this end-user on this computer.
ComplianceComputerID	<i>Type:</i> integer. Key The application. Foreign key to the <code>ComplianceComputer</code> table.
SoftwareTitleID	<i>Type:</i> integer. Key The application. Foreign key to the <code>SoftwareTitle</code> table.
LastUsedDate	<i>Type:</i> datetime. Nullable The last used date of the application.
AccessModeID	<i>Type:</i> integer. Key The date that the installed software was last used.

## ReconcileInterestingLicenses Table

A list of all licenses that are interesting to the current execution of license reconcile.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 343: Database columns for `ReconcileInterestingLicenses` table**

Database Column	Details
SoftwareLicenseID	<i>Type:</i> integer. Key The unique identifier for a software license that is interesting to an execution of reconcile.

## ReconcileInterestingTitles Table

A list of all titles that are interesting to the current execution of license reconcile.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 344: Database columns for ReconcileInterestingTitles table**

Database Column	Details
SoftwareTitleID	<p>Type: integer. Key</p> <p>The unique identifier for a software title that is interesting to an execution of reconcile.</p>

## ReconcileSoftwareLicenseComputerProblem Table

ReconcileSoftwareLicenseComputerProblem is a license reconciliation staging table for the SoftwareLicenseComputerProblemData table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 345: Database columns for ReconcileSoftwareLicenseComputerProblem table**

Database Column	Details
SoftwareLicenseID	<p>Type: integer. Key</p> <p>The software license. Foreign key to the SoftwareLicense table.</p>
ComplianceComputerID	<p>Type: integer. Key</p> <p>The computer consuming license entitlements. Foreign key to the ComplianceComputer table.</p>
SoftwareLicense ComputerProblemTypeID	<p>Type: integer</p> <p>The type of problem this computer's inventory causes for a given license. For example, core-based licenses require accurate inventory of processor core counts to determine their compliance status.</p> <p>Foreign key to the SoftwareLicenseComputerProblemType table.</p>

## ReconcileSoftwareLicenseCoresConsumedData Table

This is a staging table for `SoftwareLicenseCoresConsumedData` that stores values calculated by license reconciliation. The main table is populated at the end of license reconciliation by a single bulk update.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 346: Database columns for ReconcileSoftwareLicenseCoresConsumedData table**

Database Column	Details
<code>ComplianceComputerID</code>	<i>Type:</i> integer. Key The computer under examination. Foreign key to the <code>ComplianceComputer</code> table.
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key The license being assessed. Foreign key to the <code>SoftwareLicense</code> table.
<code>CoresConsumed</code>	<i>Type:</i> integer The number of cores that have contributed to license point consumption for the license on the computer.

## ReconcileSoftwareLicenseGroupPointsConsumedData Table

This serves as a staging table for `SoftwareLicenseGroupPointsConsumed` during reconciliation process.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 347: Database columns for ReconcileSoftwareLicenseGroupPointsConsumedData table**

Database Column	Details
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key

Database Column	Details
	The license that owns the pre-calculated totals for a group. Foreign key to the <code>SoftwareLicense</code> table.
<code>GroupTypeID</code>	<i>Type:</i> integer. Key Type of the group(Location, Cost center, etc)
<code>GroupExID</code>	<i>Type:</i> text (max 128 characters). Key. Nullable The group where the local and rolledup values are calculated. Foreign key to the <code>GroupEx</code> table.
<code>RolledUpNumberConsumed</code>	<i>Type:</i> integer The sum of points consumed of the current group and of all its child groups.
<code>LocalNumberConsumed</code>	<i>Type:</i> integer The sum of points consumed of the current group
<code>RolledUpNumberUsed</code>	<i>Type:</i> integer The sum of used points f the current group and of all its child groups.
<code>LocalNumberUsed</code>	<i>Type:</i> integer The sum of used points of the current group
<code>RolledUpNumberPurchased</code>	<i>Type:</i> integer The rolled up purchase counts of the license.
<code>LocalNumberPurchased</code>	<i>Type:</i> integer The local purchase counts of the license

## ReconcileSoftwareLicenseILMTPointsConsumedData Table

This is a staging table for `SoftwareLicenseILMTPointsConsumed` that is used during license reconciliation process, to store calculated values, and then bulk update the main table.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 348: Database columns for ReconcileSoftwareLicenseILMTPointsConsumedData table**

Database Column	Details
ComplianceComputerID	<i>Type:</i> integer. Key The computer under examination. Foreign key to the <code>ComplianceComputer</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key The license being assessed. Foreign key to the <code>SoftwareLicense</code> table.
CoreCount	<i>Type:</i> integer The number of licensable cores for the license on the computer.
PVUCount	<i>Type:</i> integer The number of PVU counts consumed for the license on the computer.
PeakPVUCount	<i>Type:</i> integer The number of PVU counts consumed for the license on the computer at the time where the peak for this license occurred.
ProductCount	<i>Type:</i> integer The number of products that are consuming same license.

## ReconcileSoftwareLicensePointsConsumedData Table

This is a staging table for `SoftwareLicensePointsConsumed` that is used during license reconciliation process, to store calculated values, and then bulk update the main table.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 349: Database columns for ReconcileSoftwareLicensePointsConsumedData table**

Database Column	Details
ComplianceComputerID	<i>Type:</i> integer. Key The computer under examination. Foreign key to the <code>ComplianceComputer</code> table.

Database Column	Details
SoftwareLicenseID	<i>Type:</i> integer. Key The license being assessed. Foreign key to the <code>SoftwareLicense</code> table.
LicensesConsumed	<i>Type:</i> integer The number of entitlements (or points) consumed for the license on the computer.
LicensesUsed	<i>Type:</i> integer How many of the points consumed are for installations actually being used.

## ReconcileSoftwareLicensePointsConsumedReason Table

This is a staging table for `SoftwareLicensePointsConsumedReasonData` that is used during license reconciliation process, to store calculated values, and then bulk update the main table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 350: Database columns for ReconcileSoftwareLicensePointsConsumedReason table**

Database Column	Details
ComplianceComputerID	<i>Type:</i> integer. Key The computer under examination. Foreign key to the <code>ComplianceComputer</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key The license being assessed. Foreign key to the <code>SoftwareLicense</code> table.
ReasonTypeID	<i>Type:</i> integer. Key The reason for the points to be consumed here. Foreign key to the <code>SoftwareLicensePointsConsumedReasonType</code> table.



## ReconcileSoftwareLicenseProcessorData Table

This serves as an intermediate table during reconciliation process to store the number of processors (or cores) on which licensed software is installed and used for each computer, and the calculated points.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 351: Database columns for ReconcileSoftwareLicenseProcessorData table**

Database Column	Details
ComplianceComputerID	<i>Type:</i> integer. Key The host computer under examination. Foreign key to the ComplianceComputer table.
SoftwareLicenseID	<i>Type:</i> integer. Key The license being assessed. Foreign key to the SoftwareLicense table.
VMLayerID	<i>Type:</i> integer. Key. Nullable The virtual machine layer under examination. Foreign key to the ReconcileVirtualMachineLayer table.
IsHost	<i>Type:</i> boolean. Key Does this refer to the top layer for this host?
IsCapped	<i>Type:</i> boolean Does this layer implement hard partitioning for this license?
InstalledCount	<i>Type:</i> decimal The number of processors/cores on which a software title licensed by the license is installed.
UsedCount	<i>Type:</i> decimal The number of processors/cores on which a software title licensed by the license is used.
CapacityCount	<i>Type:</i> decimal The number of processors/cores that apply to a software title licensed by the license under full capacity counting rules.

Database Column	Details
PointsFactor	<i>Type:</i> decimal The number of points consumed per processor/core on this computer.
InstalledPoints	<i>Type:</i> integer The number of processor/core points required to cover the above InstalledCount.
UsedPoints	<i>Type:</i> integer The number of processor/core points required to cover the above UsedCount.
CapacityPoints	<i>Type:</i> integer The number of processor/core points required to cover the above CapacityCount.

## ReconcileSoftwareLicenseSecondUseMappingData Table

This is a staging table for `SoftwareLicenseSecondUseMapping` that is used during license reconciliation process, to store calculated values, and then bulk update the main table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 352: Database columns for ReconcileSoftwareLicenseSecondUseMappingData table**

Database Column	Details
SoftwareLicenseID	<i>Type:</i> integer. Key The license conferring the right of second use. Foreign key to the <code>SoftwareLicense</code> table.
DesktopComputerID	<i>Type:</i> integer. Key The desktop or primary computer on which the related software is installed. Foreign key to the <code>ComplianceComputer</code> table.
SecondUseComputerID	<i>Type:</i> integer. Key

Database Column	Details
	The laptop or second computer covered by this license's right of second use, relative to the installation on the primary computer tracked in the previous field. Foreign key to the <code>ComplianceComputer</code> table.
<code>TotalLicenseGrabs</code>	<i>Type:</i> integer  For internal use only. Temporary storage for calculations of overlapping second use and multiple install rights.
<code>IsExternalRoamingLink</code>	<i>Type:</i> boolean  Is this a second use link or is it actually an 'external roaming' right?

## ReconcileSoftwareUserLicensePointsConsumedData Table

This is a staging table for `SoftwareUserLicensePointsConsumed` that is used during license reconciliation process, to store calculated values, and then bulk update the main table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 353: Database columns for ReconcileSoftwareUserLicensePointsConsumedData table**

Database Column	Details
<code>ComplianceUserID</code>	<i>Type:</i> integer. Key  The end-user. Foreign key to the <code>ComplianceUser</code> table.
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key  The license. Foreign key to the <code>SoftwareLicense</code> table.
<code>LicensesConsumed</code>	<i>Type:</i> integer  The number of points (or entitlements) consumed for the license by the end-user.
<code>LicensesUsed</code>	<i>Type:</i> integer  How many of the points consumed are for installations that are actually being used.

Database Column	Details
LicenseMeasurementID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The associated SAP license measurement snapshot, where appropriate. Foreign key to the <code>LicenseMeasurement</code> table.</p>

## ReconcileVirtualMachineLayer Table

This serves as an intermediate table during reconciliation process to store virtual machines, pools and hosts in a generalized tree structure.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 354: Database columns for ReconcileVirtualMachineLayer table**

Database Column	Details
VMLayerID	<p><i>Type:</i> integer. Key</p> <p>A unique identifier for a ReconcileVirtualMachineLayer.</p>
HostComplianceComputerID	<p><i>Type:</i> integer. Key</p> <p>The host computer on which the layer resides, or the computer itself. Foreign key to the <code>ComplianceComputer</code> table.</p>
VMPoolID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The identifier of the virtual pool containing this VM, or the pool itself. Foreign key to the <code>VMPool</code> table.</p>
VMPoolTypeID	<p><i>Type:</i> integer. Nullable</p> <p>The type of this VM pool. Foreign key to the <code>VMPoolType</code> table.</p>
VirtualMachineID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The identifier of this virtual machine. Foreign key to the <code>VirtualMachine</code> table.</p>
VMTypeID	<p><i>Type:</i> integer. Nullable</p> <p>The type of this virtual machine. Foreign key to the <code>VMType</code> table.</p>
ParentVMPoolID	<p><i>Type:</i> integer. Nullable</p>

Database Column	Details
	The identifier of the parent VM pool of this pool. Foreign key to the <code>VMPool</code> table.
<code>ParentVMLayerID</code>	<i>Type:</i> integer. Key. Nullable The parent layer. Foreign key to the <code>ReconcileVirtualMachineLayer</code> table.
<code>ComplianceComputerID</code>	<i>Type:</i> integer. Key. Nullable The identifier of the computer running inside this virtual machine. Foreign key to the <code>ComplianceComputer</code> table.
<code>Name</code>	<i>Type:</i> text (max 256 characters). Nullable The name of the layer (host/pool/VM).
<code>PartialNumberOfProcessors</code>	<i>Type:</i> decimal. Nullable The fractional processor count available to this computer.
<code>NumberOfProcessors</code>	<i>Type:</i> decimal. Nullable The processor count for this computer.
<code>NumberOfCores</code>	<i>Type:</i> decimal. Nullable The core count for this computer.
<code>Depth</code>	<i>Type:</i> integer. Key The number of layers between this and the host computer.

## RegistryEvidence Table

Reserved for future expansion.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 355: Database columns for RegistryEvidence table**

Database Column	Details
<code>RegistryEvidenceID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a software registry evidence record.

Database Column	Details
RegistryEvidenceHiveID	<i>Type:</i> integer. Key The registry hive for the registry evidence.
RegistryEvidenceKeyID	<i>Type:</i> integer. Key The registry key for the registry evidence.
RegistryEvidenceValueID	<i>Type:</i> integer. Key The value of the registry evidence.
RegistryData	<i>Type:</i> text (max 400 characters). Key The data contained in the registry value for the registry evidence.
Ignored	<i>Type:</i> boolean If <code>True</code> this registry evidence is ignored for application recognition.
IsShared	<i>Type:</i> boolean

## RegistryEvidenceHive Table

Reserved for future use.

**Table 356: Database columns for RegistryEvidenceHive table**

Database Column	Details
RegistryEvidenceHiveID	<i>Type:</i> integer. Key. Generated ID Unique identifier for a registry hive.
RegistryHive	<i>Type:</i> text (max 50 characters). Key The registry hive for the registry evidence.

## RegistryEvidenceKey Table

Reserved for future use.

**Table 357: Database columns for RegistryEvidenceKey table**

Database Column	Details
RegistryEvidenceKeyID	Type: integer. Key. Generated ID Unique identifier for a registry key.
RegistryKey	Type: text (max 200 characters). Key The registry key for the registry evidence.

## RegistryEvidenceValue Table

Reserved for future use.

**Table 358: Database columns for RegistryEvidenceValue table**

Database Column	Details
RegistryEvidenceValueID	Type: integer. Key. Generated ID Unique identifier for a registry value
RegistryValue	Type: text (max 50 characters). Key The registry value for the registry evidence.

## RelatedInstalledInstallerEvidence Table

RelatedInstalledInstallerEvidence table holds parent-child relationship between installer evidence.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 359: Database columns for RelatedInstalledInstallerEvidence table**

Database Column	Details
RelatedInstalled InstallerEvidenceID	Type: integer. Key. Generated ID A synthetic unique identifier
ParentInstallerEvidenceID	Type: integer. Key

Database Column	Details
	An parent identifier for an installer evidence record. Foreign key to the <code>InstallerEvidence</code> table.
<code>ParentComplianceComputerID</code>	<i>Type:</i> integer. Key An parent identifier for a computer record. Foreign key to the <code>ComplianceComputer</code> table.
<code>ParentAccessModeID</code>	<i>Type:</i> integer. Key The state an application was considered accessed. Foreign key to the <code>AccessMode</code> table.
<code>ChildInstallerEvidenceID</code>	<i>Type:</i> integer. Key An child identifier for an installer evidence record. Foreign key to the <code>InstallerEvidence</code> table.
<code>ChildComplianceComputerID</code>	<i>Type:</i> integer. Key An child identifier for a computer record. Foreign key to the <code>ComplianceComputer</code> table.
<code>ChildAccessModeID</code>	<i>Type:</i> integer. Key The state an application was considered accessed. Foreign key to the <code>AccessMode</code> table.
<code>IsCharged</code>	<i>Type:</i> boolean. Key The identifier used in the source connection to determine the pricing relation between parent and child installer evidence (specifies if it is charged = 1 or free = 0).
<code>ConfidenceLevel</code>	<i>Type:</i> integer. Nullable Confidence level for each bundled installer evidence (as a percentage).

## RelatedInstalledInstallerEvidenceSourceMap Table

`RelatedInstalledInstallerEvidenceSourceMap` Maps related installed installer evidence to the evidence source type.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.



**Table 360: Database columns for RelatedInstalledInstallerEvidenceSourceMap table**

Database Column	Details
RelatedInstalledInstallerEvidenceSourceMapID	<i>Type:</i> integer. Key. Generated ID A synthetic unique identifier
RelatedInstalledInstallerEvidenceID	<i>Type:</i> integer. Key An identifier for an related installer evidence record. Foreign key to the <code>RelatedInstalledInstallerEvidence</code> table.
ComplianceConnectionID	<i>Type:</i> integer. Key The inventory source where the end-user was reported. Foreign key to the <code>ComplianceConnection</code> table.

## RelatedInstalledSoftwareData Table

`RelatedInstalledSoftware` stores parent-child relationship among application installations. This is used for modelling application bundling.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 361: Database columns for RelatedInstalledSoftwareData table**

Database Column	Details
RelatedInstalledSoftwareID	<i>Type:</i> integer. Key. Generated ID Unique identifier for this record.
ParentInstalledSoftwareID	<i>Type:</i> integer. Key The parent installed application. Foreign key to the <code>InstalledSoftware</code> table.
ChildInstalledSoftwareID	<i>Type:</i> integer. Key The child installed application. Foreign key to the <code>InstalledSoftware</code> table.
IsCharged	<i>Type:</i> boolean. Key

Database Column	Details
	The identifier used in the source connection to determine the pricing relation between parent and child installer evidence (specifies if it is charged = 1 or free = 0).
ConfidenceLevel	Type: integer. Nullable Confidence level for each bundled installer evidence (as a percentage).

## SAPSoftwareLicense Table

SAPSoftwareLicense stores additional SAP-specific licensing information for SAP licenses.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 362: Database columns for SAPSoftwareLicense table**

Database Column	Details
SoftwareLicenseID	Type: integer. Key The SAP license. Foreign key to the SoftwareLicense table.
SAPServerName	Type: text (max 256 characters). Nullable The name of the SAP server. Should match the ComputerName of the record in the ComplianceComputer table which corresponds to the computer on which SAP is installed.
SAPBaseLicenseTypeID	Type: integer The SAP base license type, coming from the first pair of symbols in the “xx-xx-xx” license code. Foreign key to the SAPSoftwareLicenseType table.
SAPSpecialVersionID	Type: integer The SAP special version (language, country, etc.), coming from the second pair of symbols in the “xx-xx-xx” license type code. Foreign key to the SAPSoftwareLicenseType table.
SAPSurchargeID	Type: integer

Database Column	Details
	The SAP surcharge special version, coming from the third pair of symbols in the “xx-xx-xx” license code. Foreign key to the SAPSoftwareLicenseType table
SAPLicenseCode	<i>Type:</i> text (max 32 characters) The SAP license code, consisting of the license type, special version and surcharge.
HasUsage	<i>Type:</i> boolean Set this field to <code>True</code> if this license contains SAP usage/optimization information.
Description	<i>Type:</i> text (max 512 characters). Nullable A description of the SAP license.

## SAPSoftwareLicenseType Table

SAPSoftwareLicenseType lists the SAP base license types and special versions, and is part of the full “xx-xx-xx” code.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 363: Database columns for SAPSoftwareLicenseType table**

Database Column	Details
SAPSoftwareLicenseTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this SAP base license type.
LicenseCode	<i>Type:</i> text (max 32 characters). Key The unique code for this license type - one of the “xx” parts of the full “xx-xx-xx” code.
SAPSpecialVersionID	<i>Type:</i> integer. Key. Nullable If this is a base license type, this field is NULL (and the <code>LicenseCode</code> comes from the first “xx” part of the full “xx-xx-xx” code). Otherwise, it is a special SAP

Database Column	Details
	version (the <code>LicenseCode</code> comes from the second or third “xx” part), and is foreign key to the <code>SAPSpecialVersion</code> table.
<code>DescriptionResourceName</code>	<i>Type:</i> text (max 256 characters). Nullable The unique name of the localizable resource string representing the license code description. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DescriptionDefaultValue</code>	<i>Type:</i> text (max 256 characters) The text to display if the license code resource string has no translation.

## SAPSpecialVersion Table

`SAPSpecialVersion` lists the types of special versions, indicating which part of the “xx-xx-xx” code the SAP software license type comes from.

**Table 364: Database columns for `SAPSpecialVersion` table**

Database Column	Details
<code>SAPSpecialVersionID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SAPSpecialVersion</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Generic special version</li> <li>• 2 = Surcharge special version.</li> </ul>
<code>InternalDescription</code>	<i>Type:</i> text (max 50 characters) Internal description for developers.

## SoftwareAccessMode Table

The `SoftwareAccessMode` table holds the states an application has been accessed.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 365: Database columns for SoftwareAccessMode table**

Database Column	Details
SoftwareAccessModeID	<i>Type:</i> integer. Key. Generated ID The primary key of the SoftwareAccessMode table.
AccessModeID	<i>Type:</i> integer. Key The access mode for the application. Foreign key to the AccessMode table.
InstalledSoftwareID	<i>Type:</i> integer. Key The installed software title to which the access mode applies. Foreign key to the InstalledSoftware table
IsACL	<i>Type:</i> boolean. Key Determines whether the software access mode record came from ACL data.

## SoftwareLicense Table

SoftwareLicense contains details of the software licenses managed by FlexNet Manager Suite.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 366: Database columns for SoftwareLicense table**

Database Column	Details
SoftwareLicenseID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a software license.
ParentLicenseID	<i>Type:</i> integer. Key. Nullable The id of any bundle that this license is a part of.
Name	<i>Type:</i> text (max 256 characters). Key Name of the license.
Version	<i>Type:</i> text (max 60 characters). Key. Nullable Version of the license.

Database Column	Details
Edition	<i>Type:</i> text (max 60 characters). Nullable Edition of the license.
LicenseTypeID	<i>Type:</i> integer. Key The license type. Foreign key to the <code>SoftwareLicenseType</code> table.
SoftwareLicenseMetricID	<i>Type:</i> integer. Nullable Custom licensing metric for this license. Foreign key to the <code>SoftwareLicenseMetric</code> table.
DurationID	<i>Type:</i> integer The duration of this license. Foreign key to the <code>SoftwareLicenseDuration</code> table.
SoftwareLicense ComplianceStatusID	<i>Type:</i> integer The compliance status of this license. Foreign key to the <code>SoftwareLicenseComplianceStatus</code> table. Defaults to "Compliant".
LicenseStatusID	<i>Type:</i> integer The status of this license. Foreign key to the <code>LicenseStatus</code> table.
SoftwareLicense PurchaseTypeID	<i>Type:</i> integer. Nullable The kind of purchase. Foreign key to the <code>SoftwareLicensePurchaseType</code> table.
VendorID	<i>Type:</i> integer. Key. Nullable The vendor from whom the license was purchased. Foreign key to the <code>Vendor</code> table.
PublisherID	<i>Type:</i> integer. Key. Nullable The software publisher associated with this license. Foreign key to the <code>Vendor</code> table.
ManagerID	<i>Type:</i> integer. Key. Nullable The manager of this license. Foreign key to the <code>ComplianceUser</code> table.
PartNo	<i>Type:</i> text (max 100 characters). Nullable The publisher's part number for this license.
SerialNumber	<i>Type:</i> text (max 256 characters). Nullable

Database Column	Details
	The serial number of the license.
LicenseKeyTypeID	<i>Type:</i> integer The type of license keys managed on this license. Foreign key to the <code>SoftwareLicenseKeyType</code> table. Defaults to "No key".
LicenseKey	<i>Type:</i> text (max 256 characters). Nullable The multiple-use license key of the license. Only used when the license key type is a multi-use key (for example, an Enterprise key used to cover multiple installs).
RequestNo	<i>Type:</i> text (max 60 characters). Nullable The request number for the license.
AcquisitionModeID	<i>Type:</i> integer The method of acquisition used for the asset this license covers. Defaults to <code>Purchased</code> . Foreign key to the <code>AcquisitionMode</code> table.
PurchaseOrderNumber	<i>Type:</i> text (max 50 characters). Nullable The purchase order number which was used to purchase the license.
PurchaseOrderDate	<i>Type:</i> datetime. Nullable The original purchase order date for the license.
PurchasePrice	<i>Type:</i> currency. Nullable The initial purchase price of the license.
PurchasePriceRateID	<i>Type:</i> integer. Nullable The currency rate applied to the purchase price of the license. Foreign key to the <code>CurrencyRate</code> table.
ChargeBackPrice	<i>Type:</i> currency. Nullable Amount to be charged for each computer on which the license is installed.
ChargeBackPriceRateID	<i>Type:</i> integer. Nullable The currency rate applied to the charge-back price. Foreign key to the <code>CurrencyRate</code> table.
ChargeBackPeriodTypeID	<i>Type:</i> integer The frequency with which the charge back price is charged. Defaults to <code>None</code> . Foreign key to the <code>PeriodType</code> table.

Database Column	Details
ExpiryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date this license expires. A NULL value means the license does not expire.</p>
DeliveryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date this license became active. A NULL value means the license is inactive.</p>
RetirementDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date this license was retired. A NULL value means the license is active.</p>
WarrantyExpiryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date the warranty on this license expires. This refers to a warranty <code>Contract</code> associated with the license.</p>
NumberOfProcessors	<p><i>Type:</i> integer</p> <p>The number of processors that this license is for. This field is only used where the <code>SoftwareLicenseType</code> is <code>Device (Processor-Limited)</code> (<code>LicenseTypeID = 11</code>).</p>
NumberOfCores	<p><i>Type:</i> integer</p> <p>The number of cores per processor that this license is for. This field is only used where the <code>SoftwareLicenseType</code> is <code>Device (Core-Limited)</code> (<code>LicenseTypeID = 14</code>).</p>
NumberOfSockets	<p><i>Type:</i> integer</p> <p>The number of sockets that this license is for. The value zero is reserved to mean unlimited. This field is only used where the <code>SoftwareLicenseType</code> is <code>Oracle Processor (LicenseTypeID = 16)</code> or <code>Oracle Named User Plus (LicenseTypeID = 17)</code>.</p>
MinimumNumberOfProcessors	<p><i>Type:</i> integer</p> <p>The minimum number of processors that this license is for. This field is only used where the <code>SoftwareLicenseType</code> is <code>Microsoft Server Processor (LicenseTypeID = 22)</code>.</p>
MinimumNumberOfLicensesPerVM	<p><i>Type:</i> integer</p> <p>When licensing a Virtual Hardware System with a <code>Microsoft Server Core</code> license (<code>LicenseTypeID = 33</code>), consume license entitlements as though the virtual machine had at least this number of virtual threads.</p>
MSPool	<p><i>Type:</i> text (max 120 characters). Nullable</p>



Database Column	Details
	The name of the Microsoft license pool to which the license belongs.
MSPoints	<p><i>Type:</i> integer</p> <p>The points value of each installed version of this license, for use when calculating Microsoft licensing reports. This field is only valid when the <code>MSPool</code> field is set.</p>
WarrantyTypeID	<p><i>Type:</i> integer</p> <p>The type of warranty for the license. Defaults to <code>None</code>. Foreign key to the <code>AssetWarrantyType</code> table.</p>
EndOfLifeRecipient	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The person or organization who received the asset associated with this license when it was disposed of.</p>
EndOfLifeReasonID	<p><i>Type:</i> integer</p> <p>The reason the asset was associated with this license was disposed of. Foreign key to the <code>EndOfLifeReason</code> table.</p>
ResalePrice	<p><i>Type:</i> currency. Nullable</p> <p>The amount the asset associated with this license was sold for.</p>
ResalePriceRateID	<p><i>Type:</i> integer. Nullable</p> <p>The currency rate to be applied to the sale price of the asset associated with this license.</p>
CreationUser	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The operator who created this license.</p>
CreationDate	<p><i>Type:</i> datetime</p> <p>The date and time the license was created.</p>
UpdatedUser	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The operator who last updated this license.</p>
UpdatedDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date and time the license was last updated.</p>
Comments	<p><i>Type:</i> text. Nullable</p> <p>Comments about the license recorded by an operator. This field may also be used for storing license keys.</p>

Database Column	Details
NumberPurchased	<p><i>Type:</i> integer</p> <p>The quantity of purchased license entitlements.</p>
NumberInstalled	<p><i>Type:</i> integer</p> <p>The quantity of software installations accounted for by this license. This value is calculated and updated during the data import process, based on the software inventory details imported.</p>
ResourceUnitsConsumed	<p><i>Type:</i> decimal</p> <p>The quantity consumed of a resource relevant to this license. The type of resource is identified by the associated <code>SoftwareLicenseMetric</code>. On the IBM Resource Value Unit license type this will have a points rule set applied to it to calculate the final license consumption value.</p>
PeakConsumed	<p><i>Type:</i> integer</p> <p>The peak quantity of software installations accounted for by this license. This value is a high-water mark of the Consumed entitlements for the license.</p>
AdditionalBulkUsers Regular	<p><i>Type:</i> integer</p> <p>A number of regular users associated with this license in addition to those specified individually in <code>SoftwareLicenseAllocation</code>. For IBM User Value Unit licenses this will have a points rule set applied to it to calculate the final license consumption value.</p>
AdditionalBulkUsers Infrequent	<p><i>Type:</i> integer</p> <p>A number of infrequent users associated with this license in addition to those specified individually in <code>SoftwareLicenseAllocation</code>. For IBM User Value Unit licenses this will have an infrequent user multiplier and points rule set applied to it to calculate the final license consumption value.</p>
AdditionalBulkUsers External	<p><i>Type:</i> integer</p> <p>A number of external users associated with this license in addition to those specified individually in <code>SoftwareLicenseAllocation</code>. For IBM User Value Unit licenses this will have an external user multiplier and points rule set applied to it to calculate the final license consumption value.</p>
UserMultiplierInfrequent	<p><i>Type:</i> decimal</p> <p>The fraction of a regular user's consumption to use for infrequent users.</p>
UserMultiplierExternal	<p><i>Type:</i> decimal</p> <p>The fraction of a regular user's consumption to use for external users.</p>

Database Column	Details
NumberUsed	<p><i>Type:</i> integer</p> <p>The number of software installations covered by this license that are actually being used.</p>
NumberAllocated	<p><i>Type:</i> integer</p> <p>The quantity of license entitlements allocated to individual end-users or computers.</p>
NumberAssigned	<p><i>Type:</i> integer</p> <p>The quantity of license entitlements that have been assigned to enterprise groups.</p>
LastCalculatedNUPMinimum	<p><i>Type:</i> integer. Nullable</p> <p>The last calculated minimum for Oracle Named User Plus licenses.</p>
AlwaysInstalled	<p><i>Type:</i> boolean</p> <p>If this field is <code>True</code>, this license is considered in to be used whenever it is allocated. If <code>False</code>, software usage is considered separately, and allocation merely defines the corporation's modelling of who is expected to consume entitlements.</p>
LocationID	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>Any enterprise location linked to this license. Foreign key to the <code>GroupEx</code> table.</p>
BusinessUnitID	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>Any enterprise corporate unit linked to this license. Foreign key to the <code>GroupEx</code> table.</p>
CostCenterID	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>Any cost center in the enterprise that is linked to this license. Foreign key to the <code>GroupEx</code> table.</p>
CategoryID	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>Any enterprise category associated with this license. Foreign key to the <code>GroupEx</code> table.</p>
CoverInstallsOnVirtualMachines	<p><i>Type:</i> boolean</p> <p>This is known in the UI as "Enable special handling for virtual machines".</p> <p>Its effect usually includes enabling sub-capacity licensing of virtual installs and/or capping of license consumption at the host level, but its exact effect depends on the specific license type.</p>

Database Column	Details
	For license types that expose additional virtualization properties, this property must be set for the other properties to be used.
LimitNumberOfVirtualInstalls	<p><i>Type:</i> boolean</p> <p>If this field is <code>True</code>, there is a limit to the number of installations on virtual machines that can be covered by each license entitlement. If <code>False</code>, one license entitlement may cover use on any number of virtual machines (typically on one host computer).</p>
NumberOfAllowedVirtualInstalls	<p><i>Type:</i> integer. Nullable</p> <p>If the license covers installations on virtual machines, this field specifies how many installations per host are allowed before an additional license entitlement (or point) is consumed.</p>
LimitVirtualInstallsIncludesHost	<p><i>Type:</i> boolean</p> <p>If this field is <code>True</code>, the host operating system installations are included in the overall count of operating systems on the host when there is a limit on the number of allowed virtual installs for each license. If <code>False</code>, the host operating system is not considered when determining virtual install limits.</p>
NumberOfAllowedProcessorsPerHost	<p><i>Type:</i> integer. Nullable</p> <p>This field specifies how many processors per host are allowed before an additional license entitlement (or point) is consumed. Null provides the default of 1. Zero provides unlimited.</p>
UseHostProcessorInformation	<p><i>Type:</i> boolean</p> <p>If virtual installs are allowed, set this field to <code>True</code> if host information should be used when calculating license points consumed.</p>
AllowIBMPVUSubCapacityFromNonILMT	<p><i>Type:</i> boolean</p> <p>If the license does not use host processor information (not full capacity), set this field to <code>True</code> to allow non-ILMT sub-capacity PVU consumption calculations to be used.</p>
LimitNumberOfApplicationsEachLicensePointCovers	<p><i>Type:</i> boolean</p> <p>If this field is <code>True</code>, there is a limit on the number of application installations allowed per license entitlement (or point). If <code>False</code> (the default), then a license entitles you to any number of installations of software linked to this license on the one computer.</p>
NumberOfApplicationInstallsAllowedPerLicensePoint	<p><i>Type:</i> integer. Nullable</p>

Database Column	Details
	Where the previous field is set to <code>True</code> , this field defines the limited number of application installations allowed per entitlement (or point).
<code>LimitNumberOfComputers UserLicenseCanBe InstalledOn</code>	<i>Type:</i> boolean  If this field is <code>True</code> , there is a limit to the number of computers that a user-based license can be linked to per entitlement (or point) consumed. If <code>False</code> (the default), a single end-user is entitled to install related software for his/her own use on any number of computers.
<code>NumberOfComputers AllowedPerUserLicense Point</code>	<i>Type:</i> integer. Nullable  Where the previous field is set to <code>True</code> , this field defines the limited number of application installations an end-user is allowed per entitlement (or point).
<code>MinimumNumberOfUsers</code>	<i>Type:</i> integer  The minimum number of users allowed for the license. This is used for <code>Oracle Named User Plus</code> licenses.
<code>MinimumNumberOfUsers MultipliedByProcessors</code>	<i>Type:</i> boolean  Whether the previous field a fixed value for the license or it is a multiple of the number of processor points consumed by the license. This is used for <code>Oracle Named User Plus</code> licenses.
<code>SecondUsageWorkLaptop</code>	<i>Type:</i> boolean  If this field is <code>True</code> , the license confers the right of second use on a work laptop. If <code>False</code> , there is no right of second use allowed on a work laptop.
<code>SecondUsageAtHome</code>	<i>Type:</i> boolean  If this field is <code>True</code> , the license confers the right of second use on a home computer by the same end-user as the primary end-user of the license entitlement consumed at work. If <code>False</code> (the default), there is no right of second use allowed on a home computer.
<code>MultiUseInheritFrom Contract</code>	<i>Type:</i> boolean  Set this field to <code>True</code> if the license should inherit the values for right of multiple use from a contract.
<code>MultiUseInheritFrom ContractID</code>	<i>Type:</i> integer. Nullable  If the previous field is <code>True</code> , this is the contract that right of multiple use is inherited from. Foreign key to <code>Contract</code> table.
<code>SecondUsageInheritFrom Contract</code>	<i>Type:</i> boolean

Database Column	Details
	Set this field to <code>True</code> if the license should inherit the values for right of second use from a contract.
SecondUsageInheritFromContractID	<p><i>Type:</i> integer. Nullable</p> <p>If the previous field is <code>True</code>, this is the contract that right of second use is inherited from. Foreign key to <code>Contract</code> table.</p>
CoverInstallsOnVMInheritFromContract	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if the license should inherit virtual machine rights from a contract.</p>
CoverInstallsOnVMInheritFromContractID	<p><i>Type:</i> integer. Nullable</p> <p>If the previous field is <code>True</code>, this is the contract that virtual machine rights are inherited from. Foreign key to <code>Contract</code> table.</p>
CurrentSoftwareTitleID	<p><i>Type:</i> integer. Key. Nullable</p> <p>Identifies the primary application for the license (which may change over time as upgrade rights are applied). Foreign key to the <code>SoftwareTitle</code> table.</p>
InheritDowngradeFromContract	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if this license inherits its downgrade rights from a contract. If <code>False</code> (the default), downgrade rights must be configured directly on the license properties.</p>
InheritDowngradeFromContractID	<p><i>Type:</i> integer. Nullable</p> <p>If the previous field is <code>True</code>, this is the contract that downgrade rights are inherited from. Foreign key to the <code>Contract</code> table.</p>
InheritUpgradeFromContract	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if this license inherits its upgrade rights from a contract. If <code>False</code> (the default), upgrade rights must be configured directly on the license properties.</p>
InheritUpgradeFromContractID	<p><i>Type:</i> integer. Nullable</p> <p>If the previous field is <code>True</code>, this is the contract that downgrade rights are inherited from. Foreign key to the <code>Contract</code> table.</p>
AutoManageTitles	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if the license should have its application links automatically managed for upgrade and downgrade rights. When this field is <code>False</code>, the operator must manually manage links between this license and any applications.</p>

Database Column	Details
DowngradeEnabled	<p><i>Type:</i> boolean</p> <p>If this field is <code>True</code>, this license can cover previous releases, or lower editions, of applications linked to this license. If this field is <code>False</code> (the default), there is no downgrade right conferred by this license.</p>
DowngradeToVersion	<p><i>Type:</i> boolean</p> <p>If this field is <code>True</code>, the license covers previous releases (with the same edition) of the primary application. If this field is <code>False</code> (the default), earlier versions of the primary application are not covered by downgrade rights.</p>
DowngradeToVersionID	<p><i>Type:</i> integer. Nullable</p> <p>If the previous field is <code>True</code> and the value of this field is <code>NULL</code>, downgrade rights cover all earlier releases (with the same edition) of the primary application. If not <code>NULL</code>, downgrade rights cover all versions of the primary application down to and including this version. Foreign key to the <code>SoftwareTitleVersion</code> table.</p>
DowngradeToEdition	<p><i>Type:</i> boolean</p> <p>If this field is <code>True</code>, the license covers lower editions (with the same version) of the primary application. If this field is <code>False</code> (the default), lower editions of the primary application are not covered by downgrade rights.</p>
DowngradeToEditionID	<p><i>Type:</i> integer. Nullable</p> <p>If the previous field is <code>True</code> and the value of this field is <code>NULL</code>, downgrade rights cover all lower editions (with the same version) of the primary application. If not <code>NULL</code>, downgrade rights cover all editions of the primary application down to and including this edition. Foreign key to the <code>SoftwareTitleEdition</code> table.</p>
DowngradeOnlyToVersionLegacy	<p><i>Type:</i> text (max 60 characters). Nullable</p> <p>A repository for backward-compatible custom data.</p>
UpgradeEnabled	<p><i>Type:</i> boolean</p> <p>If this field is <code>True</code>, the license can cover future releases (with the same edition) of the primary application. If this bit is <code>False</code> (the default), there is no upgrade right conferred by this license.</p>
UpgradeToVersion	<p><i>Type:</i> boolean</p> <p>If this field is <code>True</code>, the license covers later releases (with the same edition) of the primary application. If this field is <code>False</code> (the default), later versions of the primary application are not covered by upgrade rights.</p>

Database Column	Details
UpgradeToVersionID	<p><i>Type:</i> integer. Nullable</p> <p>If the previous field is <code>True</code> and the value of this field is <code>NULL</code>, upgrade rights cover all later version (with the same edition) of the primary application. If not <code>NULL</code>, upgrade rights cover all versions of the primary application up to and including this version. Foreign key to the <code>SoftwareTitleEdition</code> table.</p>
UpgradeUntil	<p><i>Type:</i> boolean</p> <p>If this bit is 1, the upgrade right covers future releases of applications that get linked to this license, provided that the release date of each version is before (or on) a specified date. If this bit is zero (the default), the upgrade right is not date limited.</p>
UpgradeUntilDate	<p><i>Type:</i> datetime. Nullable</p> <p>If this field is set, only applications released before this date are covered by upgrade rights.</p>
UpgradeOnlyToVersion Legacy	<p><i>Type:</i> text (max 60 characters). Nullable</p> <p>A repository for backward-compatible custom data.</p>
TrueUp	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if the license is a true-up license (and so never goes into breach).</p>
OracleLegacyLicenseTypeID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The specific Oracle legacy license type, where appropriate. Foreign key to the <code>OracleLegacyLicenseType</code> table.</p>
GroupAllocationTypeID	<p><i>Type:</i> integer. Key</p> <p>Allocations of entitlements under any license can be made to only one type of enterprise group, specified here. Foreign key to the <code>GroupType</code> table.</p>
GroupAllocation ReportingTypeID	<p><i>Type:</i> integer. Key</p> <p>Determines when enterprise groups will be considered to have breached their allocations of entitlements under this license. Foreign key to the <code>SoftwareLicenseGroupAllocationReportingType</code> table.</p>
GroupAllocation ComplianceLevel	<p><i>Type:</i> integer. Nullable</p> <p>Determines the depth level of groups to be used for calculating the breach status for a license.</p>
CannotManuallyUpdate GroupAssignments	<p><i>Type:</i> boolean</p>



Database Column	Details
	Set this field to <code>True</code> if the operator must make group assignments through a <code>Assign License Entitlements</code> dialog box. If <code>False</code> , changes can be made directly in the license properties pages.
<code>CalculateCompliance</code>	<p><i>Type:</i> boolean</p> <p>When this field is <code>True</code> (the default), and the associated <code>SoftwareLicenseType</code> also has its <code>CalculateCompliance</code> field set to <code>True</code> (true for most license types), license consumption must be calculated from imported inventory. When <code>False</code>, the compliance state must be imported, not calculated.</p>
<code>IsSharableToLibrary</code>	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> (the default) if the license is sharable to the downloadable FlexNet Manager Suite ARL library.</p>
<code>CopyEditionAndVersion</code>	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> (the default) if edition and version should be automatically copied to the license from the primary application.</p>
<code>SoftwareLicenseTierTypeID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>Type of the tier, for <code>Tiered Device</code> license type only. Foreign key to the <code>SoftwareLicenseTierType</code> table.</p>
<code>SoftwareLicenseTierCode</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The actual tier of the license, corresponding to the tier type. For <code>Tiered Device</code> license type only.</p>
<code>ImportedFromFNMEA</code>	<p><i>Type:</i> boolean</p> <p>Set this to <code>True</code> if this license was imported from FlexNet Manager for Engineering Applications.</p>
<code>SoftwareLicensePointsRuleSetID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The points rule set used to calculate compliance for this license. Foreign key to the <code>SoftwareLicensePointsRuleSet</code> table.</p>
<code>BaselineQuantity</code>	<p><i>Type:</i> integer. Nullable</p> <p>The baseline value for this license</p>
<code>BaselineDate</code>	<p><i>Type:</i> datetime. Nullable</p> <p>The date at which the baseline applies.</p>

Database Column	Details
AlternateNonInventoriedUsers	<p><i>Type:</i> integer. Nullable</p> <p>Number of non-inventoried users who are consuming this license. For <code>Oracle Named User Plus</code> and <code>Oracle Application User</code> licenses, this acts as an alternate mean to specify user consumption in the case where no instance users are available from inventory. The number of non-inventoried users are added to the number of unique users found from inventory when number installed and number used are calculated in license reconcile.</p>
InheritLicenseMobilityFromContract	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if this license inherits its license mobility rights from a contract. If <code>False</code> (the default), license mobility rights must be configured directly on the license properties.</p>
InheritLicenseMobilityFromContractID	<p><i>Type:</i> integer. Nullable</p> <p>If the previous field is <code>True</code>, this is the contract that mobility rights are inherited from. Foreign key to the <code>Contract</code> table.</p>
InheritLicenseConsumptionFromContract	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if this license inherits its license consumption rules from a contract. If <code>False</code> (the default), license consumption rules must be configured directly on the license properties.</p>
InheritLicenseConsumptionFromContractID	<p><i>Type:</i> integer. Nullable</p> <p>If the previous field is <code>True</code>, this is the contract that license consumption rules are inherited from. Foreign key to the <code>Contract</code> table.</p>
InheritProcessorLimitsFromContract	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if this license inherits its processor limits rights from a contract. If <code>False</code> (the default), license processor limits rights must be configured directly on the license properties.</p>
InheritProcessorLimitsFromContractID	<p><i>Type:</i> integer. Nullable</p> <p>If the previous field is <code>True</code>, this is the contract that processor limits rights are inherited from. Foreign key to the <code>Contract</code> table.</p>

## SoftwareLicenseAllocation Table

`SoftwareLicenseAllocation` records the allocations of individual computers, end-users, enterprise groups or instances to licenses.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 367: Database columns for SoftwareLicenseAllocation table**

Database Column	Details
SoftwareLicenseAllocationID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the license allocation record.
SoftwareLicenseID	<i>Type:</i> integer. Key The license that has been allocated. Foreign key to the <i>SoftwareLicense</i> table.
ComplianceComputerID	<i>Type:</i> integer. Key. Nullable The computer to which the license is allocated. Foreign key to the <i>ComplianceComputer</i> table.
ComplianceUserID	<i>Type:</i> integer. Key. Nullable The end-user to which the license is allocated. Foreign key to the <i>ComplianceUser</i> table.
InstanceID	<i>Type:</i> integer. Key. Nullable The instance to which the license is allocated. Foreign key to the <i>Instance</i> table.
GroupExID	<i>Type:</i> text (max 128 characters). Key. Nullable The enterprise group to which the license is assigned. Foreign key to the <i>GroupEx</i> table.
LicenseUserID	<i>Type:</i> integer. Key. Nullable The external end-user to whom the license is allocated. Foreign key to the <i>LicenseUser</i> table.
SoftwareLicenseAllocationUserTypeID	<i>Type:</i> integer. Key. Nullable Indicates for user allocations whether they are a regular user or some special type of user for this license. Foreign key to the <i>SoftwareLicenseAllocationUserType</i> table.
NumberAllocated	<i>Type:</i> integer. Nullable The number of license entitlements assigned. This is used for group assignments.

Database Column	Details
NumberUsed	<i>Type:</i> integer. Nullable The number of license entitlements where the application is recorded as being used.
SoftwareLicenseAllocationStatusID	<i>Type:</i> integer. Nullable Indicates the status of an allocation. Foreign key to the <code>SoftwareLicenseAllocationStatus</code> table.
SoftwareLicenseKeyID	<i>Type:</i> integer. Key. Nullable The software license key that is allocated to this end-user/computer. Foreign key to the <code>SoftwareLicenseKey</code> table.
SoftwareLicenseExemptionReasonID	<i>Type:</i> integer. Key. Nullable The reason why this allocation is exempted from consuming a license entitlement. Foreign key to the <code>SoftwareLicenseExemptionReason</code> table.

## SoftwareLicenseAllocationStatus Table

`SoftwareLicenseAllocationStatus` is a static table storing a collection of status values for a license allocation.

**Table 368: Database columns for `SoftwareLicenseAllocationStatus` table**

Database Column	Details
SoftwareLicenseAllocationStatusID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SoftwareLicenseAllocationStatus</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Allocated</li> <li>• 2 = Awaiting Inventory</li> <li>• 3 = Permanent</li> <li>• 4 = Unallocated.</li> </ul>
StatusResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing an allocation status. Foreign key to the <code>ComplianceResourceString</code> table.
StatusDefaultValue	<i>Type:</i> text (max 50 characters)

Database Column	Details
	The text to display if the status resource string has no translation.

## SoftwareLicenseAllocationUserType Table

`SoftwareLicenseAllocationUserType` is a static table storing a collection of user type values for a license allocation.

**Table 369: Database columns for `SoftwareLicenseAllocationUserType` table**

Database Column	Details
<code>SoftwareLicenseAllocationUserTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>SoftwareLicenseAllocationUserType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Normal</li> <li>• 2 = Infrequent</li> <li>• 3 = External.</li> </ul>
<code>UserTypeResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a user allocation type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>UserTypeDefaultValue</code>	<p><i>Type:</i> text (max 50 characters)</p> <p>The text to display if the user type resource string has no translation.</p>

## SoftwareLicenseBreachReasonData Table

`SoftwareLicenseBreachReasonData` identifies the reasons why non-compliant software licenses are in this state.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 370: Database columns for SoftwareLicenseBreachReasonData table**

Database Column	Details
SoftwareLicenseID	<i>Type:</i> integer. Key The software license. Foreign key to the <code>SoftwareLicense</code> table.
LicenseBreachReasonID	<i>Type:</i> integer. Key The license breach reason. Foreign key to the <code>LicenseBreachReason</code> table.
LicenseMeasurementID	<i>Type:</i> integer. Key The license measurement ID. Foreign key to the <code>LicenseMeasurement</code> table.

## SoftwareLicenseChangeEvent Table

The `SoftwareLicenseChangeEvent` table holds the details of all license change events.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 371: Database columns for SoftwareLicenseChangeEvent table**

Database Column	Details
ComplianceEventID	<i>Type:</i> integer. Key A unique identifier and foreign key to the <code>ComplianceEvent</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key The license involved in the change event. Foreign key to the <code>SoftwareLicense</code> table.
SoftwareTitleID	<i>Type:</i> integer. Key The software title that needs to be added or removed. Foreign key to the <code>SoftwareTitle</code> table.
SoftwareLicenseChangeEventSourceID	<i>Type:</i> integer What caused the event. Foreign key to the <code>SoftwareLicenseChangeEventSource</code> table.

Database Column	Details
SoftwareLicenseChangeEventReasonID	<i>Type:</i> integer The type of event. Foreign key to the <code>SoftwareLicenseChangeEventReason</code> table.
SoftwareTitleLicenseReasonID	<i>Type:</i> integer. Nullable When a software title has been added to a license, the reason it has been added (ie because upgrade rights allow it, for example). Foreign key to the <code>SoftwareTitleLicenseReason</code> table.

## SoftwareLicenseChangeEventReason Table

`SoftwareLicenseChangeEventReason` is a static table holding all the valid reasons why a license change event was generated.

**Table 372: Database columns for `SoftwareLicenseChangeEventReason` table**

Database Column	Details
SoftwareLicenseChangeEventReasonID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SoftwareLicenseChangeEventReason</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Add Application</li> <li>• 2 = Remove Application.</li> </ul>
ChangeEventReasonResourceString	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a change event reason. Foreign key to the <code>ComplianceResourceString</code> table.
ChangeEventReasonDefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the reason resource string has no translation.

## SoftwareLicenseChangeEventSource Table

`SoftwareLicenseChangeEventSource` is a static table holding all the valid sources of license change events.

**Table 373: Database columns for SoftwareLicenseChangeEventSource table**

Database Column	Details
SoftwareLicenseChangeEventSourceID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each SoftwareLicenseChangeEventSource. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = ARL</li> <li>• 2 = Software License</li> <li>• 3 = Software Title (the application properties)</li> <li>• 4 = Contract</li> <li>• 5 = Version (changing the relative priorities or weights of application versions linked to a license)</li> <li>• 6 = Edition (changing the relative priorities or weights of application editions linked to a license).</li> </ul>
ChangeEventSourceResourceString	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a change event source. Foreign key to the ComplianceResourceString table.</p>
ChangeEventSourceDefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>Default value for a license change event source if the source resource has no translation.</p>

## SoftwareLicenseComplianceStatus Table

SoftwareLicenseComplianceStatus is a static table listing valid compliance states for a license.

**Table 374: Database columns for SoftwareLicenseComplianceStatus table**

Database Column	Details
SoftwareLicenseComplianceStatusID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each SoftwareLicenseComplianceStatus. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Compliant</li> <li>• 2 = In Breach</li> <li>• 3 = Unknown</li> </ul>



Database Column	Details
	<ul style="list-style-type: none"> <li>4 = Not Tracked.</li> </ul>
StatusResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a compliance status. Foreign key to the <code>ComplianceResourceString</code> table.</p>
StatusDefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the status resource string has no translation.</p>

## SoftwareLicenseComputerProblemData Table

`SoftwareLicenseComputerProblemData` identifies the problems with individual `ComplianceComputers` that contributed to an associated license having an unknown compliance status. For example, some license types calculate entitlement consumption based on the number of processor cores present in a computer, but that detail is not available from Microsoft SCCM before version 2012, so computers from this inventory source will cause associated licenses to have unknown compliance status.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 375: Database columns for `SoftwareLicenseComputerProblemData` table**

Database Column	Details
SoftwareLicenseID	<p><i>Type:</i> integer. Key</p> <p>The software license. Foreign key to the <code>SoftwareLicense</code> table.</p>
ComplianceComputerID	<p><i>Type:</i> integer. Key</p> <p>The computer consuming license entitlements. Foreign key to the <code>ComplianceComputer</code> table.</p>
SoftwareLicenseComputerProblemTypeID	<p><i>Type:</i> integer</p> <p>The type of problem this computer's inventory causes for a given license. For example, core-based licenses require accurate inventory of processor core counts to determine their compliance status.</p> <p>Foreign key to the <code>SoftwareLicenseComputerProblemType</code> table.</p>
LicenseMeasurementID	<p><i>Type:</i> integer. Key</p>

Database Column	Details
	The license measurement ID. Foreign key to the <code>LicenseMeasurement</code> table.

## SoftwareLicenseComputerProblemType Table

`SoftwareLicenseComputerProblemType` is a static table holding the collection of problems that a computer can have which might cause a particular type of license to have an unknown compliance status.

**Table 376: Database columns for `SoftwareLicenseComputerProblemType` table**

Database Column	Details
<code>SoftwareLicenseComputerProblemTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>SoftwareLicenseComputerProblemType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Core count missing from inventory</li> <li>• 2 = Processor count missing from inventory</li> <li>• 3 = Socket count missing from inventory</li> <li>• 4 = Thread count missing from inventory.</li> </ul>
<code>ProblemTypeResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a problem type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>ProblemTypeDefaultValue</code>	<p><i>Type:</i> text (max 512 characters)</p> <p>The text to display if the problem type resource string has no translation.</p>

## SoftwareLicenseContract Table

`SoftwareLicenseContract` links licenses to related contracts.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 377: Database columns for SoftwareLicenseContract table**

Database Column	Details
SoftwareLicenseContractID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this record.
SoftwareLicenseID	<i>Type:</i> integer. Key The license to which the contract is related. Foreign key to the <code>SoftwareLicense</code> table.
ContractID	<i>Type:</i> integer. Key The contract related to the license. Foreign key to the <code>Contract</code> table.

## SoftwareLicenseCoresConsumedData Table

`SoftwareLicenseCoresConsumedData` records how many cores have contributed to license point consumption for a given license by a given computer.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 378: Database columns for SoftwareLicenseCoresConsumedData table**

Database Column	Details
ComplianceComputerID	<i>Type:</i> integer. Key The computer under examination. Foreign key to the <code>ComplianceComputer</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key The license being assessed. Foreign key to the <code>SoftwareLicense</code> table.
CoresConsumed	<i>Type:</i> integer The number of cores that have contributed to license point consumption for the license on the computer.
LicenseMeasurementID	<i>Type:</i> integer. Key The license measurement ID. Foreign key to the <code>LicenseMeasurement</code> table.

## SoftwareLicenseCreation Table

`SoftwareLicenseCreation` records which SKU definition was used to create a software license.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 379: Database columns for `SoftwareLicenseCreation` table**

Database Column	Details
<code>SoftwareLicenseCreationID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for this record.
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key The software license created. Foreign key to the <code>SoftwareLicense</code> table.
<code>SoftwareSkuID</code>	<i>Type:</i> integer. Key. Nullable The SKU that was recognized. This value is optional, as the software license could have been created directly using a definition selected by the operator, without a SKU being used as the link. Foreign key to the <code>SoftwareSku</code> table.
<code>SoftwareLicenseDefinitionID</code>	<i>Type:</i> integer. Key The license definition used to create the software license. Foreign key to the <code>SoftwareLicenseDefinition</code> table.
<code>LicenseDefinitionVersion</code>	<i>Type:</i> integer. Key Which version of the license definition was used to create the software license.

## SoftwareLicenseDefinition Table

`SoftwareLicenseDefinition` maps SKUs to the license definitions and applications that it relates to.

**Table 380: Database columns for `SoftwareLicenseDefinition` table**

Database Column	Details
<code>SoftwareLicenseDefinitionID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for this record.

Database Column	Details
LicenseDefinitionFactoryUID	<i>Type:</i> text (max 30 characters). Key The FlexNet Manager Suite factory unique identifier for this record.
LicenseDefinitionTypeID	<i>Type:</i> integer. Key The license definition type. Foreign key to the <code>LicenseDefinitionType</code> table.
LicenseDefinition	<i>Type:</i> text The license definition. Contains information relevant to license creation and application links.
ProductName	<i>Type:</i> text (max 2000 characters) When a license is created using this definition, this will be its license name.
ProductVersion	<i>Type:</i> text (max 2000 characters) When a license is created using this definition, this will be its license version.
ProductPublisher	<i>Type:</i> text (max 2000 characters) When a license is created using this definition, this will be its license publisher.
LicenseTypeID	<i>Type:</i> integer. Key This definition will create a license of this type. Foreign key to the <code>LicenseType</code> table.
IsUpgrade	<i>Type:</i> boolean Set this field to <code>True</code> if this definition will create an upgrade license. If this field is <code>False</code> , this definition creates a standard license.
Version	<i>Type:</i> integer. Key The current version of this SKU definition.
PreviousVersion	<i>Type:</i> integer. Key. Nullable The version of the SKU definition prior to the current version.
CreationDate	<i>Type:</i> datetime The date that this record was created.
UpdatedDate	<i>Type:</i> datetime. Nullable The date that this record was last updated.

## SoftwareLicenseDuration Table

The collection of durations for which a license can be active.

**Table 381: Database columns for SoftwareLicenseDuration table**

Database Column	Details
SoftwareLicenseDurationID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for a license duration. Possible values (and associated default names) are:</p> <ul style="list-style-type: none"> <li>• 1 = Perpetual</li> <li>• 2 = TimeLimited</li> <li>• 3 = Subscription.</li> </ul>
DurationResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The name of the resource string containing the text to display on the user interface.</p>
DurationDefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The value to display if there is no resource string available for this status</p>

## SoftwareLicenseExemptionReason Table

The collection of types exemption reasons that may be associated with software license allocations.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 382: Database columns for SoftwareLicenseExemptionReason table**

Database Column	Details
SoftwareLicenseExemptionReasonID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each SoftwareLicenseExemptionReason. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Alpha, beta, early support program</li> <li>• 2 = Backup, disaster recovery</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 3 = Component of a non-PVU licensed offering</li> <li>• 4 = Component is not compatible with the server or agent system</li> <li>• 5 = Development</li> <li>• 6 = Evaluation, trial</li> <li>• 7 = Fail-over</li> <li>• 8 = Not eligible for PVU licensing</li> <li>• 9 = Other</li> <li>• 10 = Second use</li> <li>• 11 = Test</li> <li>• 12 = Covered by related product</li> <li>• 13 = Covered by virtual application access</li> <li>• 14 = No usage for virtual application within specified time limit</li> </ul>
ResourceName	<p>Type: text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an exemption reason. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p>Type: text (max 100 characters)</p> <p>The text to display if the type resource string has no translation.</p>

## SoftwareLicenseExemptionRole Table

`SoftwareLicenseExemptionRole` table holds information on role exemption rule for licenses. Contains many to many relationship between licenses and device roles.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 383: Database columns for `SoftwareLicenseExemptionRole` table**

Database Column	Details
SoftwareLicenseExemptionRoleID	Type: integer. Key. Generated ID

Database Column	Details
	A unique identifier for the license exemption role record.
SoftwareLicenseID	<i>Type:</i> integer. Key The license that has role exemption rule. Foreign key to the <code>SoftwareLicense</code> table.
ExemptionRoleID	<i>Type:</i> integer. Key The device role that is exempted from license consumption. Foreign key to the <code>ComplianceComputerRole</code> table.
ExemptionLimit	<i>Type:</i> integer. Nullable The number of devices that can be exempted, having an exempted role.

## SoftwareLicenseGroupAllocationReportingType Table

`SoftwareLicenseGroupAllocationReportingType` stores the set of tests that can be used to determine whether a license is in “group breach” for one or more of its associated enterprise groups.

**Table 384: Database columns for `SoftwareLicenseGroupAllocationReportingType` table**

Database Column	Details
SoftwareLicenseGroupAllocationReportingTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SoftwareLicenseGroupAllocationReportingType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 0 = None</li> <li>• 1 = Consumed Exceeds Purchased</li> <li>• 2 = Consumed Exceeds Assigned.</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a group breach test type. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 256 characters) The text to display if the type resource string has no translation.



# SoftwareLicenseGroupAssignmentHistory Table

SoftwareLicenseGroupAssignmentHistory is used to keep track of changes made to assignments of software license entitlements to enterprise groups.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 385: Database columns for SoftwareLicenseGroupAssignmentHistory table**

Database Column	Details
SoftwareLicenseGroupAssignmentHistoryID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the history record.
GroupExID	<i>Type:</i> text (max 128 characters). Key This is the primary group that had a change of assignments. Foreign key to the GroupEx table.
FromGroupExID	<i>Type:</i> text (max 128 characters). Key. Nullable If assignments were transferred, this is the source group who had assignments taken away. Foreign key to the GroupEx table.
FromGroupExPath	<i>Type:</i> text (max 500 characters). Nullable The path of the group that assignments were transferred from. This field is used to display the group name (at the time that the transfer took place) when showing history after the group has been deleted.
HistoryDate	<i>Type:</i> datetime The date of the change.
SoftwareLicenseID	<i>Type:</i> integer. Key The license for which entitlements are being assigned. Foreign key to the SoftwareLicense table.
UserName	<i>Type:</i> text (max 60 characters) The operator who made the change.
Comments	<i>Type:</i> text (max 2000 characters). Nullable Comments recorded about the change.

Database Column	Details
NumberAdded	<i>Type:</i> integer The number of assignments added to or removed from the group.
Total	<i>Type:</i> integer The progressive total of assignments to the group following this change.
SoftwareLicenseGroupAssignmentHistoryTypeID	<i>Type:</i> integer. Key The type of history record. This records the kind of change that was made (eg, a flat increase/decrease of the assignment count, a transfer, and so on). Foreign key to the <code>SoftwareLicenseGroupAssignmentHistoryType</code> table.

## SoftwareLicenseGroupAssignmentHistoryType Table

`SoftwareLicenseGroupAssignmentHistoryType` stores a collection of the types of history record that can be stored in the `SoftwareLicenseGroupAssignmentHistory` table.

**Table 386: Database columns for `SoftwareLicenseGroupAssignmentHistoryType` table**

Database Column	Details
SoftwareLicenseGroupAssignmentHistoryTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SoftwareLicenseGroupAssignmentHistoryType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Manual (manual increase/decrease of the group assignment quantity)</li> <li>• 2 = ManualDialog (manual increase/decrease of the group assignment quantity, using the <code>Assign License Entitlements dialog_</code></li> <li>• 3 = Transfer (a transfer of entitlements from one enterprise group to another)</li> <li>• 4 = ClearAssignments (the <code>Clear Assignments</code> button has been used to remove all entitlements from a group)</li> <li>• 5 = AssignPurchased (the <code>Assign Purchases</code> button has been used to copy purchases within the group to the group assignment total)</li> <li>• 6 = ChangeGroupType (assignments have been cleared because the group assignment type has been changed).</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Nullable

Database Column	Details
	The unique name of the localizable resource string representing a history type. Foreign key to the <code>ComplianceResourceString</code> table.
<code>SoftwareLicenseGroupAssignmentHistoryTypeName</code>	<i>Type:</i> text (max 64 characters). Key A description of the history type.
<code>DefaultValue</code>	<i>Type:</i> text (max 50 characters) The text to display if the type resource string has no translation.

## SoftwareLicenseGroupBreachStatus Table

`SoftwareLicenseGroupBreachStatus` stores the collection of possible outcomes of group breach testing.

**Table 387: Database columns for `SoftwareLicenseGroupBreachStatus` table**

Database Column	Details
<code>SoftwareLicenseGroupBreachStatusID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SoftwareLicenseGroupBreachStatus</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 0 = Ignored</li> <li>• 1 = Not In Group Breach</li> <li>• 2 = In Group Breach.</li> </ul>
<code>ResourceName</code>	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a group breach status. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DefaultValue</code>	<i>Type:</i> text (max 256 characters) The text to display if the status resource string has no translation.

## SoftwareLicenseGroupPointsConsumedData Table

`SoftwareLicenseGroupPointsConsumed` records the licenses pre-calculated local and rolledup totals for groups.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 388: Database columns for `SoftwareLicenseGroupPointsConsumedData` table**

Database Column	Details
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key The license that owns the pre-calculated totals for a group. Foreign key to the <code>SoftwareLicense</code> table.
<code>GroupTypeID</code>	<i>Type:</i> integer. Key Type of the group(Location, Cost center, etc)
<code>GroupExID</code>	<i>Type:</i> text (max 128 characters). Key. Nullable The group where the local and rolledup values are calculated. Foreign key to the <code>GroupEx</code> table.
<code>RolledUpNumberConsumed</code>	<i>Type:</i> integer The sum of points consumed of the current group and of all its child groups.
<code>LocalNumberConsumed</code>	<i>Type:</i> integer The sum of points consumed of the current group
<code>RolledUpNumberUsed</code>	<i>Type:</i> integer The sum of used points of the current group and of all its child groups.
<code>LocalNumberUsed</code>	<i>Type:</i> integer The sum of used points of the current group
<code>RolledUpNumberPurchased</code>	<i>Type:</i> integer The rolled up purchase counts of the license.
<code>LocalNumberPurchased</code>	<i>Type:</i> integer The local purchase counts of the license
<code>LicenseMeasurementID</code>	<i>Type:</i> integer. Key The license measurement ID. Foreign key to the <code>LicenseMeasurement</code> table.

## SoftwareLicenseILMTPointsConsumedData Table

SoftwareLicenseILMTPointsConsumed records how many PVU counts and their corresponding core counts have been consumed for a given license by a given computer.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 389: Database columns for SoftwareLicenseILMTPointsConsumedData table**

Database Column	Details
ComplianceComputerID	<i>Type:</i> integer. Key The computer under examination. Foreign key to the ComplianceComputer table.
SoftwareLicenseID	<i>Type:</i> integer. Key The license being assessed. Foreign key to the SoftwareLicense table.
CoreCount	<i>Type:</i> integer The number of licensable cores for the license on the computer.
PVUCount	<i>Type:</i> integer The number of PVU counts consumed for the license on the computer.
PeakPVUCount	<i>Type:</i> integer The number of PVU counts consumed for the license on the computer at the time where the peak for this license occurred.
LicenseMeasurementID	<i>Type:</i> integer. Key The license measurement ID. Foreign key to the LicenseMeasurement table.

## SoftwareLicenseKey Table

The SoftwareLicenseKey table contains installation keys that are linked to software licenses.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 390: Database columns for `SoftwareLicenseKey` table**

Database Column	Details
<code>SoftwareLicenseKeyID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for this license key.
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key The software license that this installation key belongs to. Foreign key to the <code>SoftwareLicense</code> table.
<code>KeyValue</code>	<i>Type:</i> text (max 400 characters). Key The installation key value.

## SoftwareLicenseKeyType Table

The collection of types of installation keys that may be associated with software licenses.

**Table 391: Database columns for `SoftwareLicenseKeyType` table**

Database Column	Details
<code>SoftwareLicenseKeyTypeID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SoftwareLicenseKeyType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = No keys</li> <li>• 2 = One (multi-install) key per license</li> <li>• 3 = One (multi-install) key per application</li> <li>• 4 = One (single-install) key per installation.</li> <li>• 5 = One (multi-install) key per installation.</li> </ul>
<code>KeyTypeResourceName</code>	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a license key type. Foreign key to the <code>ComplianceResourceString</code> table.
<code>KeyTypeDefaultValue</code>	<i>Type:</i> text (max 100 characters)

Database Column	Details
	The text to display if the type resource string has no translation.

## SoftwareLicenseMetric Table

`SoftwareLicenseMetric` holds the pre-defined list of licensing custom metrics.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 392: Database columns for `SoftwareLicenseMetric` table**

Database Column	Details
<code>SoftwareLicenseMetricID</code>	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for each <code>SoftwareLicenseMetric</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = \$M cost of goods sold</li> <li>• 2 = \$M freight under management</li> <li>• 3 = \$M in revenue</li> <li>• 4 = \$M revenue under management</li> <li>• 5 = Active Oracle node</li> <li>• 6 = Cluster</li> <li>• 7 = Compensated individual</li> <li>• 8 = Connector</li> <li>• 9 = Developer</li> <li>• 10 = Drive</li> <li>• 11 = Electronic order line</li> <li>• 12 = Employees</li> <li>• 13 = Expense report</li> <li>• 14 = External connector</li> <li>• 15 = External recipient</li> <li>• 16 = Field technician</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 17 = Floating user</li> <li>• 18 = Form</li> <li>• 19 = Front end GB</li> <li>• 20 = Front end TB</li> <li>• 21 = FTE student</li> <li>• 22 = Gateway</li> <li>• 23 = Gigabyte</li> <li>• 24 = Guest</li> <li>• 25 = Host</li> <li>• 26 = Internet connector</li> <li>• 27 = IP</li> <li>• 28 = Mailbox</li> <li>• 29 = OSE</li> <li>• 30 = Partner organization</li> <li>• 31 = Person</li> <li>• 32 = Per 1000 invoice lines</li> <li>• 33 = Per 1000 records</li> <li>• 34 = Per rule set</li> <li>• 35 = Per tape drive</li> <li>• 36 = Port</li> <li>• 37 = Record</li> <li>• 38 = Server bundle</li> <li>• 39 = Service order line</li> <li>• 40 = Storage domain</li> <li>• 41 = Terabyte</li> <li>• 42 = Tiered NAS device</li> <li>• 43 = Tivoli management point</li> <li>• 44 = Trainee</li> <li>• 45 = Transaction</li> <li>• 46 = UPK module</li> </ul>



Database Column	Details
	<ul style="list-style-type: none"> <li>• 47 = Folio download</li> <li>• 48 = Document</li> <li>• 49 = Per 1000 minutes</li> <li>• 50 = Exam</li> <li>• 51 = Support incidents</li> <li>• 52 = Time</li> <li>• 53 = Recipient</li> <li>• 54 = Employees + non employees</li> <li>• 100 (Oracle Processor) = Cores</li> <li>• 101 (Oracle NUP) = Cores</li> <li>• 102 (Oracle Processor) = Sockets</li> <li>• 103 (Oracle NUP) = Sockets</li> <li>• 150 (IBM RVU) = Million Service Units</li> <li>• 151 (IBM RVU) = Messages</li> <li>• 152 (IBM RVU) = Engines</li> <li>• 153 (IBM RVU) = Terabytes</li> <li>• 154 (IBM RVU) = Tape Drives</li> <li>• 155 (IBM RVU) = Gigabytes</li> <li>• 156 (IBM RVU) = Premium Income \$US Billions (1 Resource Per US\$500M, rounded up to nearest US\$500M)</li> <li>• 157 (IBM RVU) = Capital Asset Value (\$US Billions)</li> <li>• 158 (IBM RVU) = Activated Processor Cores</li> <li>• 159 (IBM RVU) = Pages Per Month</li> <li>• 160 (IBM RVU) = Soft Goods &amp; Services Entities</li> <li>• 161 (IBM RVU) = Manufactured Goods Entities</li> <li>• 162 (IBM RVU) = Assets &amp; Commodities Entities</li> <li>• 163 (IBM RVU) = Locations Entities and Trading Partners &amp; Parties Entities</li> <li>• 164 (IBM RVU) = Client Devices</li> <li>• 165 (IBM RVU) = Server Devices</li> <li>• 166 (IBM RVU) = Annual Web Sessions</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 167 (IBM RVU) = 1,000 Web Interactions</li> <li>• 168 (IBM RVU) = 1,000,000 Data Source Records</li> <li>• 169 (IBM RVU) = 1,000,000 Monthly Server Calls</li> <li>• 170 (IBM RVU) = 1,000,000 Subscribers</li> <li>• 171 (IBM RVU) = 10,000 Records</li> <li>• 172 (IBM RVU) = 100 Records</li> <li>• 173 (IBM RVU) = 100,000 Records</li> <li>• 174 (IBM RVU) = Assets</li> <li>• 175 (IBM RVU) = Authorized Retail, Host, and Mobile Sites</li> <li>• 176 (IBM RVU) = Conversion Units</li> <li>• 177 (IBM RVU) = Enterprise Identifiers</li> <li>• 178 (IBM RVU) = Managed Devices</li> <li>• 179 (IBM RVU) = Records</li> <li>• 180 (IBM RVU) = Resources</li> <li>• 181 (IBM RVU) = Revenue \$US Billions</li> <li>• 182 (IBM RVU) = Secondary Sites</li> <li>• 183 (IBM RVU) = Servers</li> <li>• 184 (IBM RVU) = Transportation Events Per Calendar Month</li> <li>• 185 (IBM RVU) = Value Units</li> <li>• 186 (IBM RVU) = Virtual Servers</li> <li>• 187 (IBM RVU) = Web Pages.</li> </ul>
SoftwareLicenseTypeID	<p><i>Type:</i> integer. Key</p> <p>The software license type to which this metric applies. Foreign key to the <code>SoftwareLicenseType</code> table.</p>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing a licensing metric. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the ResourceName has no translation.</p>

## SoftwareLicensePartitioningDefault Table

`SoftwareLicensePartitioningDefault` contains the sub-capacity licensing rules: the types of virtual machines/partitions and pools which each license type treats as “hard” (able to put a hard limit on processor usage).

**Table 393: Database columns for `SoftwareLicensePartitioningDefault` table**

Database Column	Details
<code>SoftwareLicenseTypeID</code>	<p><i>Type:</i> integer. Key</p> <p>The software license type to which this rule applies. Foreign key to the <code>SoftwareLicenseType</code> table.</p>
<code>VMTypeID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>A virtual machine/partition type that is “hard” for the purposes of this license type. Foreign key to the <code>VMType</code> table.</p>
<code>VMPoolTypeID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>A virtual machine/partition pool type that is “hard” for the purposes of this license type. Foreign key to the <code>VMPoolType</code> table.</p>

## SoftwareLicensePoints Table

The `SoftwareLicensePoints` table holds the criteria for points-based licenses.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 394: Database columns for `SoftwareLicensePoints` table**

Database Column	Details
<code>SoftwareLicensePointsID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for a software license criterion.</p>
<code>SoftwareLicenseID</code>	<p><i>Type:</i> integer. Key</p> <p>The license to which this information applies. Foreign key to the <code>SoftwareLicense</code> table.</p>

Database Column	Details
ProcessorType	<i>Type:</i> text (max 256 characters). Key The type of processor a computer must have for this criterion to apply, such as "AMD" or "Intel".
ComputerModelNo	<i>Type:</i> text (max 128 characters). Key The model number a computer must have for this criterion to apply, such as "IBM PS701" or "IBM JS12".
MinCores	<i>Type:</i> integer. Key The minimum number of processor cores a computer must have for this criterion to apply.
MaxCores	<i>Type:</i> integer. Key The maximum number of processor cores a computer must have for this criterion to apply.
MinProcessors	<i>Type:</i> integer. Key The minimum number of processors a computer must have for this criterion to apply.
MaxProcessors	<i>Type:</i> integer. Key The maximum number of processors a computer must have for this criterion to apply.
MinSockets	<i>Type:</i> integer. Key The minimum number of processor sockets a computer must have for this criterion to apply.
MaxSockets	<i>Type:</i> integer. Key The maximum number of processor sockets a computer must have for this criterion to apply.
MinCoresPerSocket	<i>Type:</i> integer. Key The minimum number of processor cores per socket a computer must have for this criterion to apply.
MaxCoresPerSocket	<i>Type:</i> integer. Key The maximum number of processor cores per socket a computer must have for this criterion to apply.
Points	<i>Type:</i> decimal

Database Column	Details
	The points value per core or processor.

## SoftwareLicensePointsConsumedData Table

`SoftwareLicensePointsConsumed` records how many license entitlements have been consumed for a given license by a given computer.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 395: Database columns for `SoftwareLicensePointsConsumedData` table**

Database Column	Details
<code>ComplianceComputerID</code>	<i>Type:</i> integer. Key The computer under examination. Foreign key to the <code>ComplianceComputer</code> table.
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key The license being assessed. Foreign key to the <code>SoftwareLicense</code> table.
<code>LicensesConsumed</code>	<i>Type:</i> integer The number of entitlements (or points) consumed for the license on the computer.
<code>LicensesUsed</code>	<i>Type:</i> integer How many of the points consumed are for installations actually being used.
<code>LicenseMeasurementID</code>	<i>Type:</i> integer. Key The license measurement ID. Foreign key to the <code>LicenseMeasurement</code> table.

## SoftwareLicensePointsConsumedReasonData Table

This table stores information about why an entry in `SoftwareLicensePointsConsumed` exists.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 396: Database columns for SoftwareLicensePointsConsumedReasonData table**

Database Column	Details
ComplianceComputerID	Type: integer. Key The computer under examination. Foreign key to the ComplianceComputer table.
SoftwareLicenseID	Type: integer. Key The license being assessed. Foreign key to the SoftwareLicense table.
ReasonTypeID	Type: integer The reason for the points to be consumed here. Foreign key to the SoftwareLicensePointsConsumedReasonType table.
LicenseMeasurementID	Type: integer. Key The license measurement ID. Foreign key to the LicenseMeasurement table.

## SoftwareLicensePointsConsumedReasonType Table

SoftwareLicensePointsConsumedReasonType stores all the different important attributes that can be stored against a SoftwareLicensePointsConsumed record.

**Table 397: Database columns for SoftwareLicensePointsConsumedReasonType table**

Database Column	Details
ReasonTypeID	Type: integer. Key. Generated ID A unique identifier for the SoftwareLicensePointsConsumedReasonType table.
ReasonResourceName	Type: text (max 256 characters). Key The unique name of the localizable resource string representing the reason a license was linked to a title. Foreign key to the ComplianceResourceString table.
ReasonDefaultValue	Type: text (max 100 characters) The text to display if the reason resource string has no translation.

## SoftwareLicensePointsDefault Table

The `SoftwareLicensePointsDefault` table stores a collection of default license points associated with a particular license type.

**Table 398: Database columns for `SoftwareLicensePointsDefault` table**

Database Column	Details
<code>SoftwareLicensePointsDefaultID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a default points record.
<code>SoftwareLicenseTypeID</code>	<i>Type:</i> integer. Key The software license type to which this points record applies. Foreign key to the <code>SoftwareLicenseType</code> table.
<code>ProcessorType</code>	<i>Type:</i> text (max 256 characters) The type of processor a computer must have for this criterion to apply, such as "AMD" or "Intel".
<code>ComputerModelNo</code>	<i>Type:</i> text (max 128 characters) The model number a computer must have for this criterion to apply, such as "IBM PS701" or "IBM JS12".
<code>MinCores</code>	<i>Type:</i> integer The minimum number of processor cores a computer must have for this criterion to apply.
<code>MaxCores</code>	<i>Type:</i> integer The maximum number of processor cores a computer must have for this criterion to apply.
<code>MinProcessors</code>	<i>Type:</i> integer The minimum number of processors a computer must have for this criterion to apply.
<code>MaxProcessors</code>	<i>Type:</i> integer The maximum number of processors a computer must have for this criterion to apply.
<code>MinSockets</code>	<i>Type:</i> integer The minimum number of processor sockets a computer must have for this criterion to apply.

Database Column	Details
MaxSockets	<i>Type:</i> integer The maximum number of processor sockets a computer must have for this criterion to apply.
MinCoresPerSocket	<i>Type:</i> integer The minimum number of processor cores per socket a computer must have for this criterion to apply.
MaxCoresPerSocket	<i>Type:</i> integer The maximum number of processor cores per socket a computer must have for this criterion to apply.
Points	<i>Type:</i> decimal The points value per core or processor.
DateEffective	<i>Type:</i> datetime. Nullable The date from which these default values are effective. This is used to group sets of rows into sets.
Description	<i>Type:</i> text (max 1024 characters). Nullable A description of the points rules.

## SoftwareLicensePointsRule Table

The `SoftwareLicensePointsRule` table stores individual license points rules (mapping of criteria to point value) belonging to a given points rule set.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 399: Database columns for `SoftwareLicensePointsRule` table**

Database Column	Details
SoftwareLicensePointsRuleID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a points rule record.



Database Column	Details
SoftwareLicensePointsRuleSetID	<i>Type:</i> integer. Key The set to which this rule applies. Foreign key to the SoftwareLicensePointsRuleSet table.
IsCustom	<i>Type:</i> boolean. Key True if the rule is managed by the customer (versus by the ARL).
Description	<i>Type:</i> text (max 1024 characters). Nullable A human-readable description or identifier for the rule.
Points	<i>Type:</i> decimal. Key The points value per core, processor, user, or other resource metric.
ProcessorType	<i>Type:</i> text (max 256 characters) The type of processor a computer must have for this criterion to apply, such as "AMD" or "Intel".
ComputerModelNo	<i>Type:</i> text (max 128 characters) The model number a computer must have for this criterion to apply, such as "IBM PS701" or "IBM JS12".
MinCores	<i>Type:</i> integer The minimum number of processor cores a computer must have for this criterion to apply.
MaxCores	<i>Type:</i> integer The maximum number of processor cores a computer must have for this criterion to apply.
MinProcessors	<i>Type:</i> integer The minimum number of processors a computer must have for this criterion to apply.
MaxProcessors	<i>Type:</i> integer The maximum number of processors a computer must have for this criterion to apply.
MinSockets	<i>Type:</i> integer The minimum number of processor sockets a computer must have for this criterion to apply.

Database Column	Details
MaxSockets	<i>Type:</i> integer The maximum number of processor sockets a computer must have for this criterion to apply.
MinCoresPerSocket	<i>Type:</i> integer The minimum number of processor cores per socket a computer must have for this criterion to apply.
MaxCoresPerSocket	<i>Type:</i> integer The maximum number of processor cores per socket a computer must have for this criterion to apply.
MinResource	<i>Type:</i> decimal. Key The minimum resource value for an IBM RVU license for this criterion to apply.
MaxResource	<i>Type:</i> decimal. Key The maximum resource value for an IBM RVU license for this criterion to apply.
MinUsers	<i>Type:</i> integer. Key The minimum number of users relevant to an IBM UVU license for this criterion to apply.
MaxUsers	<i>Type:</i> integer. Key The maximum number of users relevant to an IBM UVU license for this criterion to apply.
MinClockSpeed	<i>Type:</i> integer The minimum value of the highest frequency of fastest processor, measured in megahertz, for this criterion to apply.
MaxClockSpeed	<i>Type:</i> integer The maximum value of the highest frequency of fastest processor, measured in megahertz, for this criterion to apply.
MinPurchaseDate	<i>Type:</i> datetime. Nullable The earliest date on which the asset must have been purchased for this criterion to apply.
MaxPurchaseDate	<i>Type:</i> datetime. Nullable The latest date on which the asset must have been purchased for this criterion to apply.

Database Column	Details
IsShared	Type: boolean

## SoftwareLicensePointsRuleSet Table

The `SoftwareLicensePointsRuleSet` table stores named sets of points rules associated with a particular license type.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 400: Database columns for `SoftwareLicensePointsRuleSet` table**

Database Column	Details
SoftwareLicensePointsRuleSetID	Type: integer. Key. Generated ID A unique identifier for a points rule set record.
SoftwareLicenseTypeID	Type: integer. Key The software license type to which this set applies. Foreign key to the <code>SoftwareLicenseType</code> table.
Description	Type: text (max 256 characters). Key. Nullable A human-readable description or identifier for the set.
IsShared	Type: boolean

## SoftwareLicenseProcessorPointsData Table

Stores the number of processors/cores on which points-based licensed software is installed and used, and the corresponding points and factors.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 401: Database columns for SoftwareLicenseProcessorPointsData table**

Database Column	Details
ComplianceComputerID	<i>Type:</i> integer. Key The computer under examination. Foreign key to the ComplianceComputerSnapshot table.
SoftwareLicenseID	<i>Type:</i> integer. Key The license being assessed. Foreign key to the SoftwareLicenseSnapshot table.
InstalledCount	<i>Type:</i> decimal The number of processors/cores on which a software title licensed by the license is installed.
UsedCount	<i>Type:</i> decimal The number of processors/cores on which a software title licensed by the license is used.
CapacityCount	<i>Type:</i> decimal The number of processors/cores that apply to a software title licensed by the license under full capacity counting rules.
PointsFactor	<i>Type:</i> decimal The number of points consumed per processor/core on this computer for this license.
InstalledPoints	<i>Type:</i> integer The number of processor/core points required to cover the above InstalledCount.
UsedPoints	<i>Type:</i> integer The number of processor/core points required to cover the above UsedCount.
CapacityPoints	<i>Type:</i> integer The number of processor/core points required to cover the above CapacityCount.
LicenseMeasurementID	<i>Type:</i> integer. Key The license measurement ID. Foreign key to the LicenseMeasurement table.

## SoftwareLicensePropertyValue Table

For each end-user, SoftwareLicensePropertyValue stores the values for the custom properties defined in SoftwareLicenseTypeProperty.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 402: Database columns for SoftwareLicensePropertyValue table**

Database Column	Details
SoftwareLicensePropertyValueID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a property value.
SoftwareLicenseTypePropertyID	<i>Type:</i> integer. Key The property whose value is being stored. The type of the license should match the type that the property is associated with. Foreign key to the SoftwareLicenseTypeProperty table.
SoftwareLicenseID	<i>Type:</i> integer. Key License whose property value is being stored. Foreign key to the SoftwareLicense table
PropertyValue	<i>Type:</i> text (max 4000 characters) The property value.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was last updated.

## SoftwareLicenseProposalStatus Table

`SoftwareLicenseProposalStatus` is a static table listing all of the states that a license change proposal can be in.

**Table 403: Database columns for `SoftwareLicenseProposalStatus` table**

Database Column	Details
<code>SoftwareLicenseProposalStatusID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>SoftwareLicenseProposalStatus</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Pending</li> <li>• 2 = Accepted</li> <li>• 3 = Ignored</li> </ul>
<code>ResourceName</code>	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing the license change proposal state. Foreign key to the <code>ComplianceResourceString</code> table.</p>
<code>DefaultValue</code>	<p><i>Type:</i> text (max 256 characters)</p> <p>The text to display if the state resource string has no translation.</p>

## SoftwareLicensePurchaseType Table

`SoftwareLicensePurchaseType` holds a list of purchase types for licenses.

**Table 404: Database columns for `SoftwareLicensePurchaseType` table**

Database Column	Details
<code>SoftwareLicensePurchaseTypeID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>SoftwareLicensePurchaseType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Volume</li> <li>• 2 = Shrink Wrap</li> <li>• 3 = OEM</li> <li>• 4 = Subscription.</li> </ul>

Database Column	Details
SoftwareLicense PurchaseTypeResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing a purchase type. Foreign key to the <code>ComplianceResourceString</code> table.
SoftwareLicense PurchaseTypeDefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.

## SoftwareLicenseReservation Table

The `SoftwareLicenseReservation` table lists all reservations for a license entitlement for an application.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 405: Database columns for `SoftwareLicenseReservation` table**

Database Column	Details
SoftwareLicense ReservationID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this reservation.
SoftwareTitleID	<i>Type:</i> integer. Key The application being reserved. Foreign key to the <code>SoftwareTitle</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key. Nullable The license affected by this reservation, null if any license for the application can be consumed. Foreign key to the <code>SoftwareLicense</code> table.
ComplianceComputerID	<i>Type:</i> integer. Key The computer making the reservation. Foreign key to the <code>ComplianceComputer</code> table.
ComplianceUserID	<i>Type:</i> integer. Key. Nullable The user making the reservation. Foreign key to the <code>ComplianceUser</code> table.
PointsReserved	<i>Type:</i> integer The number of points this reservation will ultimately consume.

Database Column	Details
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
SoftwareLicense ReservationTypeID	<i>Type:</i> integer The type of reservation.
SoftwareLicense ReservationStatusID	<i>Type:</i> integer Stores the status of the reservation

## SoftwareLicenseReservationStatus Table

The collection of status values for reservation.

**Table 406: Database columns for SoftwareLicenseReservationStatus table**

Database Column	Details
SoftwareLicense ReservationStatusID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the reservation status id
SoftwareLicense ReservationStatusName	<i>Type:</i> text (max 128 characters) The name of the reservation status.

## SoftwareLicenseReservationType Table

The collection of status values for reservation types.

**Table 407: Database columns for SoftwareLicenseReservationType table**

Database Column	Details
SoftwareLicense ReservationTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the reservation type id
SoftwareLicense ReservationTypeName	<i>Type:</i> text (max 128 characters)



Database Column	Details
	The name of the reservation type.

## SoftwareLicenseScopeTag Table

Reserved for future development.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 408: Database columns for SoftwareLicenseScopeTag table**

Database Column	Details
SoftwareLicenseID	Type: integer. Key Foreign key to the SoftwareLicense table.
TagID	Type: integer. Key Foreign key to the Tag table.
ScopeTagTypeID	Type: integer. Key Foreign key to the SoftwareLicenseScopeTagType table.

## SoftwareLicenseScopeTagType Table

Reserved for future development.

**Table 409: Database columns for SoftwareLicenseScopeTagType table**

Database Column	Details
ScopeTagTypeID	Type: integer. Key. Generated ID A unique ID for this record.
TypeDescription	Type: text (max 50 characters). Key The text value for this type.

## SoftwareLicenseScoping Table

SoftwareLicenseScoping links software licenses to enterprise groups, to restrict the rights granted by the licenses to the selected group and its descendents (license scoping).



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 410: Database columns for SoftwareLicenseScoping table**

Database Column	Details
SoftwareLicenseID	Type: integer. Key The scoped license. Foreign key to the SoftwareLicense table.
GroupExID	Type: text (max 128 characters). Key The enterprise group that this license is restricted to. Any children of this enterprise group are also included in the scope of the license. Foreign key to the GroupEx table.

## SoftwareLicenseSecondUseMappingData Table

SoftwareLicenseSecondUseMapping maps pairs of desktop computers and laptop computers against each license conferring the right of second use and covering installations on these computers.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 411: Database columns for SoftwareLicenseSecondUseMappingData table**

Database Column	Details
SoftwareLicenseID	Type: integer. Key The license conferring the right of second use. Foreign key to the SoftwareLicense table.
DesktopComputerID	Type: integer. Key

Database Column	Details
	The desktop or primary computer on which the related software is installed. Foreign key to the <code>ComplianceComputer</code> table.
<code>SecondUseComputerID</code>	<i>Type:</i> integer. Key  The laptop or second computer covered by this license's right of second use, relative to the installation on the primary computer tracked in the previous field. Foreign key to the <code>ComplianceComputer</code> table.
<code>TotalLicenseGrabs</code>	<i>Type:</i> integer  For internal use only. Temporary storage for calculations of overlapping second use and multiple install rights.
<code>IsExternalRoamingLink</code>	<i>Type:</i> boolean  Is this a second use link or is it actually an 'external roaming' right?

## SoftwareLicenseSnapshot Table

The `SoftwareLicenseSnapshot` table lists all the snapshotted software licenses.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 412: Database columns for `SoftwareLicenseSnapshot` table**

Database Column	Details
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key  The snapshotted <code>SoftwareLicenseID</code> .
<code>Name</code>	<i>Type:</i> text (max 256 characters)  The snapshotted license name.
<code>LicenseTypeID</code>	<i>Type:</i> integer. Key  The license type. Foreign key to the <code>SoftwareLicenseType</code> table.
<code>SoftwareLicenseComplianceStatusID</code>	<i>Type:</i> integer. Nullable  The compliance status of this license. Foreign key to the <code>SoftwareLicenseComplianceStatus</code> table. Defaults to "Compliant".

Database Column	Details
Consumed	<i>Type:</i> integer. Nullable The snapshotted license consumed count.
PurchaseQuantity	<i>Type:</i> integer. Nullable The snapshotted license purchase quantity.
PurchasePrice	<i>Type:</i> currency. Nullable The initial purchase price of the license.
PurchasePriceRateID	<i>Type:</i> integer. Nullable The currency rate applied to the purchase price of the license. Foreign key to the <code>CurrencyRate</code> table.
LicenseMeasurementID	<i>Type:</i> integer. Key The snapshot ID. Foreign key to the <code>LicenseMeasurement</code> table.

## SoftwareLicenseTierType Table

`SoftwareLicenseTierType` is a static table listing the tier types that a software license can have. Used for Tiered Device license type.

**Table 413: Database columns for `SoftwareLicenseTierType` table**

Database Column	Details
SoftwareLicenseTierTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SoftwareLicenseTierType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Generic</li> <li>• 2 = Per Processor</li> <li>• 3 = Symantec Server</li> <li>• 4 = Symantec Processor Type</li> <li>• 5 = Symantec Installed Operating System.</li> </ul>
TierTypeResourceName	<i>Type:</i> text (max 256 characters). Key. Nullable The unique name of the localizable resource string representing a tier type. Foreign key to the <code>ComplianceResourceString</code> table.

Database Column	Details
TierTypeDefaultValue	<p><i>Type:</i> text (max 256 characters)</p> <p>The text to display if the type resource string has no translation.</p>
TierCodeValidationRegEx	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The regular expression used to validate the tier code.</p>
TierCodeValidationMsg ResourceName	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The unique name of the localizable resource string representing the message shown when tier code validation fails. Foreign key to the <code>ComplianceResourceString</code> table.</p>
TierCodeValidationMsg DefaultValue	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The text to display if the resource string (for the message shown when tier code validation fails) has no translation.</p>

## SoftwareLicenseType Table

`SoftwareLicenseType` holds the collection of all valid license types.

**Table 414: Database columns for `SoftwareLicenseType` table**

Database Column	Details
SoftwareLicenseTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>SoftwareLicenseType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Enterprise</li> <li>• 2 = Device</li> <li>• 3 = Node-Locked</li> <li>• 4 = User</li> <li>• 5 = Concurrent User</li> <li>• 6 = Appliance</li> <li>• 7 = Client Server</li> <li>• 8 = OEM</li> <li>• 9 = Evaluation</li> <li>• 10 = Run-Time</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 11 = Device (Processor-Limited)</li> <li>• 12 = Site</li> <li>• 13 = Named User</li> <li>• 14 = Device (Core-Limited)</li> <li>• 15 = Core Points</li> <li>• 16 = Oracle Processor</li> <li>• 17 = Oracle Named User Plus</li> <li>• 18 = Processor Points</li> <li>• 19 = Oracle Legacy</li> <li>• 20 = Enterprise Agreement</li> <li>• 21 = SAP Named User</li> <li>• 22 = Microsoft Server Processor</li> <li>• 23 = CAL</li> <li>• 24 = Tiered Device</li> <li>• 25 = IBM Processor Value Unit</li> <li>• 26 = IBM Authorized User</li> <li>• 27 = IBM Concurrent User</li> <li>• 28 = IBM Floating User</li> <li>• 29 = Custom Metric</li> <li>• 30 = Processor</li> <li>• 31 = IBM Resource Value Unit</li> <li>• 32 = IBM User Value Unit</li> <li>• 33 = Microsoft Server Core</li> <li>• 34 = Oracle User</li> <li>• 35 = SAP Package</li> <li>• 36 = Microsoft SCCM Client Device</li> <li>• 37 = Microsoft SCCM Client User</li> <li>• 38 = Microsoft Developer Network</li> </ul>
TypeResourceName	Type: text (max 256 characters). Key

Database Column	Details
	The unique name of the localizable resource string representing a license type. Foreign key to the <code>ComplianceResourceString</code> table.
TypeDefaultValue	Type: text (max 100 characters) The text to display if the type resource string has no translation.
XMLFile	Type: text. Nullable The layout of the property dialog for this type of computer, stored in XML format.
CustomProcedureName	Type: text (max 256 characters). Nullable The stored procedure used to assign licenses for this license type.
DoesLicenseAllowUserAllocations	Type: boolean Set this field to <code>True</code> if the license supports allocations to individual end-users. When <code>False</code> , it cannot be allocated to end-users.
DoesLicenseAllowComputerAllocations	Type: boolean Set this field to <code>True</code> if the license supports allocations to individual computers. When it is <code>False</code> , it cannot be allocated to computers. (Note that for a custom license type, both this and the previous field may be set at the same time.)
DoesLicenseAllowVirtualApplications	Type: boolean Set this field to <code>True</code> if the license supports virtual applications. When it is <code>False</code> , it cannot be consumed by virtual applications. (Note that virtual applications have <code>AccessModelID &gt; 1</code> .)
CanConvertToAndFromType	Type: boolean Set this field to <code>True</code> if an operator is allowed to change the type of this license after it has been created. This field also determines whether this license type is included in the list of types that can be converted to. Oracle licenses, for example, cannot be converted to or from.
ExclusionReasonName	Type: text (max 256 characters). Nullable The unique name of the localizable resource string representing the reason why an installation linked to a license of this type may appear in the <code>Unlicensed Installs</code> node. Foreign key to the <code>ComplianceResourceString</code> table.
ExclusionReasonDefault	Type: text (max 500 characters) The text to display if the reason resource string has no translation.

Database Column	Details
IncludeInSQLAssignment	<p><i>Type:</i> boolean</p> <p>Set this field to <code>True</code> if licenses of this type should be processed during the SQL part of the license reconciliation process.</p>
CalculateCompliance	<p><i>Type:</i> boolean</p> <p>When this field is <code>True</code> (the default), and a <code>SoftwareLicense</code> of this type also has its <code>CalculateCompliance</code> field set to <code>True</code> (the default), that license must have its consumption calculated from imported inventory. When <code>False</code>, the compliance state of licenses with this type must be imported or otherwise set manually, not calculated.</p>
ReconcileAsSoftwareLicenseTypeID	<p><i>Type:</i> integer. Nullable</p> <p>If specified, treat this license type as if it were another for license reconciliation purposes. Foreign key to another type in this <code>SoftwareLicenseType</code> table.</p>
Enabled	<p><i>Type:</i> boolean</p> <p>Indicates whether this license type is enabled</p>

## SoftwareLicenseTypeChangeProposal Table

The `SoftwareLicenseTypeChangeProposal` table is used to store a proposed change of type for a particular software license. The changes have been inferred from changes to the license definition used to create the software license.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 415: Database columns for `SoftwareLicenseTypeChangeProposal` table**

Database Column	Details
SoftwareLicenseTypeChangeProposalID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>Primary key for the <code>SoftwareLicenseTypeChangeProposal</code> table.</p>
SoftwareLicenseID	<p><i>Type:</i> integer. Key</p> <p>Foreign key to the <code>SoftwareLicense</code> table.</p>



Database Column	Details
SoftwareLicenseDefinitionID	<i>Type:</i> integer Foreign key to the <code>SoftwareLicenseDefinition</code> table.
LicenseDefinitionVersion	<i>Type:</i> integer The version of the license definition that has been used for these proposed changes.
SoftwareLicenseUseRightNameID	<i>Type:</i> integer The proposed use right being changed on the software license.
SoftwareLicenseTypeID	<i>Type:</i> integer. Key The proposed license type for the software license.
OldSoftwareLicenseTypeID	<i>Type:</i> integer The existing license type for the software license.
SoftwareLicenseProposalStatusID	<i>Type:</i> integer The state of this software license change proposal.
Conflicted	<i>Type:</i> boolean Whether this license type change proposal conflicts with another type proposed for the same license.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was updated.

## SoftwareLicenseTypePriority Table

`SoftwareLicenseTypePriority` holds the priority order of license types.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 416: Database columns for SoftwareLicenseTypePriority table**

Database Column	Details
SoftwareLicenseTypeID	<p>Type: integer. Key</p> <p>The software license type to which this priority applies. Foreign key to the SoftwareLicenseType table.</p>
CompliancePriority	<p>Type: integer</p> <p>The priority order of the license type when calculating compliance. Licenses with higher priority will be consumed first.</p>

## SoftwareLicenseTypeProperty Table

SoftwareLicenseTypeProperty defines extra custom properties for all end-users.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 417: Database columns for SoftwareLicenseTypeProperty table**

Database Column	Details
SoftwareLicenseTypePropertyID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for an individual property.</p>
PropertyName	<p>Type: text (max 256 characters). Key</p> <p>The name of the property.</p>
SoftwareLicenseTypeID	<p>Type: integer. Key</p> <p>License type with which this property is associated. Foreign key to the LicenseType table.</p>
CustomPropertyDisplayXMLID	<p>Type: integer. Nullable</p>

Database Column	Details
	Foreign key to a record in the <code>CustomPropertyDisplayXML</code> table, describing how to show the property on a property dialog.

## SoftwareLicenseUseRight Table

`SoftwareLicenseUseRight` contains licensing rules most of which can be set by PURL.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 418: Database columns for `SoftwareLicenseUseRight` table**

Database Column	Details
<code>SoftwareLicenseUseRightID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key A unique identifier for a software license.
<code>ReassignmentTimeLimitApplies</code>	<i>Type:</i> boolean If 1 then the license cannot be reassigned for some period of time (example is Microsoft 90 day rule)
<code>ReassignmentTimeLimit</code>	<i>Type:</i> integer. Nullable The period (in days) within which the license cannot be reassigned
<code>LicenseMobilityApplies</code>	<i>Type:</i> boolean 1 if eligible for bringing your own license to cloud environment
<code>NumberOfOSEPerLicense</code>	<i>Type:</i> integer. Nullable Number of OSE per license
<code>NumberOfProcessorsPerOSE</code>	<i>Type:</i> integer. Nullable Number of processors per OSE
<code>TotalNumberOfCoresPerVMPerLicense</code>	<i>Type:</i> integer. Nullable Total number of cores per VM per license

Database Column	Details
NumberOfCoresPerSocket	<i>Type:</i> integer. Nullable Number of cores per socket
ThirdPartyAccessAllowed	<i>Type:</i> boolean Access to applications is allowed to third party users. This field is defaulted to <code>True</code>
PURLComment	<i>Type:</i> text. Nullable Additional information provided by PURL
AllowExternalRoamingUse	<i>Type:</i> boolean. Nullable Set this field to <code>True</code> if license allows external roaming use. This field is defaulted to <code>False</code> . This is applicable for both device and user licenses and is related to virtual application access. If 1, this license will consume 1 entitlement per each user. If 0, this license will consume 1 license per each user device. And, if NULL, ignore virtual application access. This can be used in conjunction with <code>VirtualApplicationAccessMaximumUsagePeriod</code> .
MeasurementDate	<i>Type:</i> datetime. Nullable The date of the license measurment.
ConsumptionUnit	<i>Type:</i> text. Nullable Unit description to describe the consumption amount.
TargetOperatingSystemTypeID	<i>Type:</i> integer Type of Operating Systems to target
VirtualApplicationAccessMaximumUsagePeriod	<i>Type:</i> integer. Nullable This is a rule for virtual application access. This is used in conjunction with the <code>AllowExternalRoamingUse</code> . For Device licenses, a license will consume 1 entitlement per each user device when used in period specified here. For user licenses, if 1, this license will consume only when used in period specified here.

## SoftwareLicenseUseRightIBM Table

SoftwareLicenseUseRightIBM contains IBM licensing rules most of which can be set by PURL.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 419: Database columns for SoftwareLicenseUseRightIBM table**

Database Column	Details
SoftwareLicenseUseRightIBMid	Type: integer. Key. Generated ID A unique identifier
SoftwareLicenseID	Type: integer. Key A unique identifier for a software license.
PVULimitApplies	Type: boolean If 1 then PVU limits apply
PVULimit	Type: integer. Nullable PVU limit

## SoftwareLicenseUseRightName Table

SoftwareLicenseUseRightName is a static table listing all of the use rights that can be applied to a software license.

**Table 420: Database columns for SoftwareLicenseUseRightName table**

Database Column	Details
SoftwareLicenseUseRightNameID	Type: integer. Key. Generated ID A unique identifier for each SoftwareLicenseUseRightName. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = License type</li> <li>• 2 = Cover installs on virtual machines</li> <li>• 3 = Limit number of virtual installs</li> <li>• 4 = Number of allowed virtual installs</li> <li>• 5 = Limit virtual installs includes host</li> <li>• 6 = Use host processor information</li> <li>• 7 = Allow IBM PVU sub-capacity from non ILMT</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 8 = Limit number of applications each license point covers</li> <li>• 9 = Number of application installs allowed per license point</li> <li>• 10 = Limit number of computers user license can be installed on</li> <li>• 11 = Number of computers allowed per license point</li> <li>• 12 = Minimum number of users</li> <li>• 13 = Minimum number of users multiplied by processors</li> <li>• 14 = Second usage work laptop</li> <li>• 15 = Second usage at home</li> <li>• 16 = Downgrade enabled</li> <li>• 17 = Downgrade to version</li> <li>• 18 = Downgrade to version ID</li> <li>• 19 = Downgrade to edition</li> <li>• 20 = Downgrade to edition ID</li> <li>• 21 = Upgrade enabled</li> <li>• 22 = Upgrade to version</li> <li>• 23 = Upgrade to version ID</li> <li>• 24 = Upgrade until</li> <li>• 25 = Upgrade until date</li> <li>• 26 = Reassignment time limit applies</li> <li>• 27 = Reassignment time limit</li> <li>• 28 = License mobility applies</li> <li>• 29 = Number of OSE per license</li> <li>• 30 = Number of processors per OSE</li> <li>• 31 = Total number of cores per VM per license</li> <li>• 32 = Number of cores per socket</li> <li>• 33 = Third party access allowed</li> <li>• 34 = PURL comment</li> <li>• 35 = Allow external roaming use</li> <li>• 36 = Measurement date</li> <li>• 37 = Consumption unit</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 38 = PVU limit applies</li> <li>• 39 = PVU limit</li> <li>• 40 = Points rule set</li> <li>• 41 = Minimum number of processors</li> <li>• 42 = Minimum number of licenses per virtual machine</li> <li>• 43 = Number of sockets</li> <li>• 44 = User multiplier external</li> <li>• 45 = User multiplier infrequent</li> <li>• 46 = Exempted roles</li> <li>• 47 = Exempted role limit</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing the proposed action. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 256 characters)</p> <p>The text to display if the state resource string has no translation.</p>

## SoftwareLicenseUseRightProposal Table

The `SoftwareLicenseUseRightProposal` table is used to store a summary of use right changes to a particular software license. The changes have been inferred from changes to the license definition used to create the software license.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 421: Database columns for `SoftwareLicenseUseRightProposal` table**

Database Column	Details
SoftwareLicenseUseRightProposalID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>Primary key for the <code>SoftwareLicenseUseRightProposal</code> table.</p>
SoftwareLicenseID	<p><i>Type:</i> integer. Key</p>

Database Column	Details
	Foreign key to the <code>SoftwareLicense</code> table.
<code>SoftwareLicenseDefinitionID</code>	<i>Type:</i> integer Foreign key to the <code>SoftwareLicenseDefinition</code> table.
<code>LicenseDefinitionVersion</code>	<i>Type:</i> integer The version of the license definition that has been used for these proposed changes.
<code>SoftwareLicenseUseRightNameID</code>	<i>Type:</i> integer. Key The proposed use right being changed on the software license.
<code>Enabled</code>	<i>Type:</i> boolean. Key. Nullable Is this use right being enabled?
<code>Value</code>	<i>Type:</i> text (max 256 characters). Key. Nullable The proposed value for this use right.
<code>OldValue</code>	<i>Type:</i> text (max 256 characters). Nullable The existing value for this use right.
<code>RelatedID</code>	<i>Type:</i> integer. Nullable The database ID of the proposed object associated with this use right.
<code>OldRelatedID</code>	<i>Type:</i> integer. Nullable The database ID of the old object associated with this use right.
<code>SoftwareLicenseProposalStatusID</code>	<i>Type:</i> integer The state of this software license change proposal.
<code>Conflicted</code>	<i>Type:</i> boolean Whether this license type change proposal conflicts with another type proposed for the same license.
<code>ContractInherited</code>	<i>Type:</i> boolean Whether this license type change proposal is for a use right currently inherited from contract by the license.
<code>CreationUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.



Database Column	Details
CreationDate	<i>Type:</i> datetime The date the record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The operator who updated the record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the record was updated.

## SoftwareRecognition Table

**Table 422: Database columns for SoftwareRecognition table**

Database Column	Details
SoftwareRecognitionID	<i>Type:</i> text (max 30 characters). Key Factory generated identity.
UpdateMode	<i>Type:</i> text (max 20 characters). Nullable Update behavior.
LastCollectiveUpdated	<i>Type:</i> datetime. Nullable Last updated datetime by ARL on all software titles and evidence
LastLinkUpdated	<i>Type:</i> datetime. Nullable Last updated datetime by ARL on the software title links
LastRecordUpdated	<i>Type:</i> datetime. Nullable Last updated datetime by ARL on the software title or evidence records. To know which record this column refers to, see TypeOfID.
LastCollectiveChecksum	<i>Type:</i> integer. Nullable Last collective checksum on successful ARL update
LastLinkChecksum	<i>Type:</i> integer. Nullable Last link checksum on successful ARL update
LastRecordChecksum	<i>Type:</i> integer. Nullable

Database Column	Details
	Last record checksum on successful ARL update. To know which record this column refers to, see TypeOfID.
LastCollectiveUpdateResult	Type: integer. Nullable Last collective ARL update result
LastLinkUpdateResult	Type: integer. Nullable Last ARL link update result
LastRecordUpdateResult	Type: integer. Nullable Last ARL record update result
RecordAdoptedByARL	Type: boolean When an existing customer record is updated by the ARL, this flag will be set
SoftwareTitleID	Type: integer. Key. Nullable The related SoftwareTitle
ChildSoftwareTitleID	Type: integer. Key. Nullable The related child SoftwareTitle
SoftwareTitleProductID	Type: integer. Key. Nullable The related SoftwareTitleProduct
SoftwareTitleVersionID	Type: integer. Key. Nullable The related SoftwareTitleVersion
SoftwareTitleEditionID	Type: integer. Key. Nullable The related SoftwareTitleEdition
SoftwareTitlePublisherID	Type: integer. Key. Nullable The related SoftwareTitlePublisher
FileEvidenceID	Type: integer. Key. Nullable The related FileEvidence
InstallerEvidenceID	Type: integer. Key. Nullable The related InstallerEvidence
WMIEvidenceID	Type: integer. Key. Nullable The related WMIEvidence

Database Column	Details
RegistryEvidenceID	<i>Type:</i> integer. Nullable The related registry <code>WMIEvidence</code>
SoftwareLicensePointsDefaultID	<i>Type:</i> integer. Key. Nullable The related <code>SoftwareLicensePointsDefault</code>
SoftwareLicensePointsRuleSetID	<i>Type:</i> integer. Key. Nullable The related <code>SoftwareLicensePointsRuleSet</code>
SoftwareLicensePointsRuleID	<i>Type:</i> integer. Key. Nullable The related <code>SoftwareLicensePointsRule</code>
TypeOfID	<i>Type:</i> text (max 32 characters). Key The type of the last updated ARL record

## SoftwareSKULookup Table

`SoftwareSKULookup` maps licenses imported from external source to SKU published by FNMS

**Table 423: Database columns for `SoftwareSKULookup` table**

Database Column	Details
SoftwareSKULookupID	<i>Type:</i> integer. Key. Generated ID A unique identifier for this record.
SourceType	<i>Type:</i> text (max 32 characters). Key
LookupName	<i>Type:</i> text (max 128 characters). Key
SKU	<i>Type:</i> text (max 100 characters) Holds the SKU value.

## SoftwareSku Table

`SoftwareSku` defines all software SKU (stock-keeping unit) numbers.

**Table 424: Database columns for SoftwareSku table**

Database Column	Details
SoftwareSkuID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a software SKU.
SKUFactoryUID	<i>Type:</i> text (max 30 characters). Key A FlexNet Manager Suite factory unique ID for this SKU.
SKU	<i>Type:</i> text (max 100 characters). Key Holds the SKU value.
SKUDefinition	<i>Type:</i> text Encrypted data that describes this SKU.
SoftwareLicenseDefinitionID	<i>Type:</i> integer. Key SKU license definition. Used to create new licenses and link them to applications. Foreign key to the <code>SoftwareLicenseDefinition</code> table.
SoftwareSkuTypeID	<i>Type:</i> integer. Key For internal use only. A numerical representation of the type of SKU.
MaintenanceTypeID	<i>Type:</i> integer For internal use only. A numerical representation of the maintenance type (if any) of the SKU.
Version	<i>Type:</i> integer. Key The current version of the SKU definition.
PreviousVersion	<i>Type:</i> integer. Key. Nullable The version of the SKU definition prior to the current version.
CreationDate	<i>Type:</i> datetime The date that this SKU definition was created.
UpdatedDate	<i>Type:</i> datetime. Nullable The date that this SKU definition was last updated.

## SoftwareTitle Table

The `SoftwareTitle` table contains the application titles managed by FlexNet Manager Suite.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 425: Database columns for `SoftwareTitle` table**

Database Column	Details
<code>SoftwareTitleID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a software record.
<code>SoftwareTitleTypeID</code>	<i>Type:</i> integer. Key The application type. Foreign key to the <code>SoftwareTitleType</code> table.
<code>SoftwareTitleProductID</code>	<i>Type:</i> integer. Key The application product, which also may specify a publisher. Foreign key to the <code>SoftwareTitleProduct</code> table.
<code>SoftwareTitleVersionID</code>	<i>Type:</i> integer. Key. Nullable The application version. Foreign key to the <code>SoftwareTitleVersion</code> table.
<code>SoftwareTitleEditionID</code>	<i>Type:</i> integer. Key. Nullable The application edition. Foreign key to the <code>SoftwareTitleEdition</code> table.
<code>OperatorManageStateID</code>	<i>Type:</i> integer. Key The management responsibility for this information. Foreign key to the <code>OperatorManageState</code> table.
<code>FullName</code>	<i>Type:</i> text (max 512 characters) By default, the full name of the application is the concatenation of the product, version, and edition fields. The operator may overwrite this with any preferred value.
<code>SoftwareTitleClassificationID</code>	<i>Type:</i> integer. Nullable The classification of the title. Defaults to <code>None</code> . Foreign key to the <code>SoftwareTitleClassification</code> table.
<code>IsMonitoringSessions</code>	<i>Type:</i> boolean Set this field to <code>True</code> if sessions are being monitored.
<code>UsageSessions</code>	<i>Type:</i> integer

Database Column	Details
	An application is considered used if it is opened more than this many times within the monitoring period.
IsMonitoringActiveTime	<i>Type:</i> boolean Set this field to <code>True</code> if active time is being monitored.
UsageActiveTime	<i>Type:</i> integer An application is considered used if the application active time (time it is in the foreground) exceeds this value during the monitoring period.
UsagePeriod	<i>Type:</i> integer The period in months over which to consider usage.
Comments	<i>Type:</i> text. Nullable Stores any comments an operator wants to make about a particular application title.
SKU	<i>Type:</i> text (max 200 characters). Nullable Deprecated: now use <code>LicensePartNo</code> of the <code>PurchaseOrderDetail</code> table. Stock Keeping Unit (SKU) for the application.
CategoryID	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise category associated with this application title. Foreign key to the <code>GroupEx</code> table.
IsLicensable	<i>Type:</i> boolean Set this field to <code>True</code> if this application needs a license. If <code>False</code> , the application doesn't need a license.
ReleaseDate	<i>Type:</i> datetime. Nullable The date the application was released.
IsSharableToLibrary	<i>Type:</i> boolean Set this field to <code>True</code> if the application is sharable to the FlexNet Manager Suite ARL library.
AutoManageLicensePriority	<i>Type:</i> boolean Set this field to <code>True</code> if the application should automatically manage the priority of attached licenses.
TitleRequiresStrictMatching	<i>Type:</i> boolean

Database Column	Details
	Set this field to <code>True</code> if the application should use stricter matching rules, requiring all evidence of all types to be present.
<code>SupportedUntil</code>	<i>Type:</i> datetime. Nullable The date the application will be supported
<code>ExtendedSupportUntil</code>	<i>Type:</i> datetime. Nullable The date the application will be supported, in extended case
<code>SoftwareTitleActionID</code>	<i>Type:</i> integer A categorization for the application in the enterprise. Defaults to <code>New.Foreign</code> key to the <code>SoftwareTitleAction</code> table.
<code>HasInstalls</code>	<i>Type:</i> boolean If this field is <code>True</code> this application has at least one installation. If <code>False</code> , the application has no installations.
<code>IsShared</code>	<i>Type:</i> boolean

## SoftwareTitleAction Table

`SoftwareTitleAction` is a static table listing action outcomes for the application in the enterprise.

**Table 426: Database columns for `SoftwareTitleAction` table**

Database Column	Details
<code>SoftwareTitleActionID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SoftwareTitleAction</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Unmanaged (recently created application, not yet categorized)</li> <li>• 2 = Authorized (application is authorized for use in the enterprise)</li> <li>• 3 = Unauthorized (application is not authorized for use)</li> <li>• 4 = Ignored (application will not be tracked by the enterprise)</li> <li>• 5 = Inactive (application is not in use in the enterprise).</li> <li>• 6 = Deferred (application installed in enterprise but marked for later attention).</li> </ul>
<code>ActionResourceName</code>	<i>Type:</i> text (max 256 characters). Key

Database Column	Details
	The unique name of the localizable resource string representing an action outcome. Foreign key to the <code>ComplianceResourceString</code> table.
<code>ActionDefaultValue</code>	<i>Type:</i> text (max 100 characters) The text to display if the action outcome resource string has no translation.

## SoftwareTitleClassification Table

`SoftwareTitleClassification` is a static table listing the possible classifications for software titles.

**Table 427: Database columns for `SoftwareTitleClassification` table**

Database Column	Details
<code>SoftwareTitleClassificationID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>SoftwareTitleClassification</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Shareware</li> <li>• 2 = Freeware</li> <li>• 3 = Commercial</li> <li>• 4 = Update</li> <li>• 5 = Malware</li> <li>• 6 = Beta</li> <li>• 7 = XRated</li> <li>• 8 = None</li> <li>• 9 = Component.</li> </ul>
<code>ResourceName</code>	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing an application classification. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DefaultValue</code>	<i>Type:</i> text (max 100 characters) The text to display if the classification resource string has no translation.



## SoftwareTitleEdition Table

A list of application editions, which must be unique for a given product. Examples include “Ultimate”, “Professional” and “32 bit”.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 428: Database columns for SoftwareTitleEdition table**

Database Column	Details
SoftwareTitleEditionID	<i>Type:</i> integer. Key. Generated ID The unique identifier for an edition.
SoftwareTitleProductID	<i>Type:</i> integer. Key The edition's product. Foreign key to the <i>SoftwareTitleProduct</i> table.
EditionName	<i>Type:</i> text (max 50 characters). Key The text for this application edition.
EditionWeight	<i>Type:</i> decimal Edition weight (for ordering, so we know which editions are upgrades/downgrades of other editions).
IsLocal	<i>Type:</i> boolean If this field is <i>False</i> , the edition has come from the ARL. If it is <i>True</i> , then the edition has been created by an operator.
IsShared	<i>Type:</i> boolean

## SoftwareTitleEx Table

The *SoftwareTitleEx* table contains additional information on the application titles managed by FlexNet Manager Suite.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 429: Database columns for SoftwareTitleEx table**

Database Column	Details
SoftwareTitleID	<i>Type:</i> integer. Key A unique identifier for a software record.
OperatorManageStateID	<i>Type:</i> integer. Nullable The management responsibility for this information. Foreign key to the <code>OperatorManageState</code> table.
AutoManageLicensePriority	<i>Type:</i> boolean. Nullable Set this field to <code>True</code> if the application should automatically manage the priority of attached licenses.
IsMonitoringSessions	<i>Type:</i> boolean. Nullable Set this field to <code>True</code> if sessions are being monitored.
UsageSessions	<i>Type:</i> integer. Nullable An application is considered used if it is opened more than this many times within the monitoring period.
IsMonitoringActiveTime	<i>Type:</i> boolean. Nullable Set this field to <code>True</code> if active time is being monitored.
UsageActiveTime	<i>Type:</i> integer. Nullable An application is considered used if the application active time (time it is in the foreground) exceeds this value during the monitoring period.
UsagePeriod	<i>Type:</i> integer. Nullable The period in months over which to consider usage.
SoftwareTitleActionID	<i>Type:</i> integer. Key. Nullable A categorization for the application in the enterprise. Defaults to <code>New</code> . Foreign key to the <code>SoftwareTitleAction</code> table.
HasInstalls	<i>Type:</i> boolean. Nullable If this field is <code>True</code> this application has at least one installation. If <code>False</code> , the application has no installations.

## SoftwareTitleFileEvidence Table

SoftwareTitleFileEvidence links software (application) titles to file evidence.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 430: Database columns for SoftwareTitleFileEvidence table**

Database Column	Details
SoftwareTitleID	<i>Type:</i> integer. Key The application title to which the file evidence is related. Foreign key to the SoftwareTitle table.
FileEvidenceID	<i>Type:</i> integer. Key The file evidence related to the software title. Foreign key to the FileEvidence table.
EvidenceExistenceRuleID	<i>Type:</i> integer The evidence existence rule related to the software title. Foreign key to the EvidenceExistenceRule table.
TrackUsage	<i>Type:</i> boolean If this field is <code>True</code> , the linked file evidence should be considered when calculating whether the application title is being used. If <code>False</code> , the file is not tracked for usage calculations.
IsLocal	<i>Type:</i> boolean If this field is <code>False</code> , the link has come from the ARL. If it is <code>True</code> , then the link has been created by an operator.
IsShared	<i>Type:</i> boolean

## SoftwareTitleHierarchy Table

SoftwareTitleHierarchy records a hierarchy of applications. This table records relationships between Oracle database and component applications, between suites and their members, and between generic titles and more specific ones that will replace them.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 431: Database columns for `SoftwareTitleHierarchy` table**

Database Column	Details
<code>ParentSoftwareTitleID</code>	<p><i>Type:</i> integer. Key</p> <p>The parent application. Foreign key to the <code>SoftwareTitle</code> table.</p>
<code>ChildSoftwareTitleID</code>	<p><i>Type:</i> integer. Key</p> <p>The child application. Foreign key to the <code>SoftwareTitle</code> table.</p>
<code>IsLicensable</code>	<p><i>Type:</i> boolean. Nullable</p> <p>This field is used for Oracle option titles. Set this field to <code>True</code> to indicate that the child application needs to be separately licensed. If this field is <code>False</code>, the child application does not need to be separately licensed when the parent application is present and licensed.</p>
<code>IsMandatory</code>	<p><i>Type:</i> boolean. Nullable</p> <p>This field is used on component applications of software suites. When the value is <code>True</code>, the child application must be installed for the suite to be recognized as installed. Otherwise, the application may or may not be installed for the suite to be recognized.</p>
<code>RemovalOfChild</code>	<p><i>Type:</i> boolean. Nullable</p> <p>This field is used to allow removal of titles when higher quality titles (with more specified evidence) are also found installed. When the value is <code>True</code>, the child application should be removed if evidence is found that both it and its parent title are installed. Otherwise, the child application is left in place.</p>
<code>IsLocal</code>	<p><i>Type:</i> boolean</p> <p>If this field is <code>False</code>, the link has come from the ARL. If it is <code>True</code>, then the link has been created by an operator.</p>
<code>IsMandatoryDefault</code>	<p><i>Type:</i> boolean. Nullable</p> <p>This field is used on component applications of software suites. This indicates the Default value of the Mandatory field and can be used to determine if this has been overridden by the user, in the case of an application with non-local membership to the suite (that is, the ARL specifies that the app belongs to the suite).</p>
<code>IsShared</code>	<p><i>Type:</i> boolean</p>

## SoftwareTitleHierarchyEx Table

The `SoftwareTitleHierarchyEx` table contains additional information on the suite by FlexNet Manager Suite.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 432: Database columns for `SoftwareTitleHierarchyEx` table**

Database Column	Details
<code>ParentSoftwareTitleID</code>	<p>Type: integer. Key</p> <p>The parent application. Foreign key to the <code>SoftwareTitle</code> table.</p>
<code>ChildSoftwareTitleID</code>	<p>Type: integer. Key</p> <p>The child application. Foreign key to the <code>SoftwareTitle</code> table.</p>
<code>IsMandatory</code>	<p>Type: boolean. Nullable</p> <p>This field is used on component applications of software suites. When the value is <code>True</code>, the child application must be installed for the suite to be recognized as installed. Otherwise, the application may or may not be installed for the suite to be recognized.</p>

## SoftwareTitleInstallerEvidence Table

`SoftwareTitleInstallerEvidence` links software (application) titles to installer evidence.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 433: Database columns for `SoftwareTitleInstallerEvidence` table**

Database Column	Details
<code>SoftwareTitleID</code>	<p>Type: integer. Key</p> <p>The software title to which the installer evidence is related. Foreign key to the <code>SoftwareTitle</code> table.</p>

Database Column	Details
InstallerEvidenceID	<i>Type:</i> integer. Key The installer evidence related to the software title. Foreign key to the <code>InstallerEvidence</code> table.
IsLocal	<i>Type:</i> boolean If this field is <code>False</code> , the link has come from the ARL. If it is <code>True</code> , then the link has been created by an operator.
IsShared	<i>Type:</i> boolean

## SoftwareTitleLicense Table

The `SoftwareTitleLicense` table links software (application) titles to licenses.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 434: Database columns for `SoftwareTitleLicense` table**

Database Column	Details
SoftwareTitleID	<i>Type:</i> integer. Key The application. Foreign key to the <code>SoftwareTitle</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key The license covering this application. Foreign key to the <code>SoftwareLicense</code> table.
CompliancePriority	<i>Type:</i> integer. Nullable Installations of this application will consume the linked licenses in this table in order of priority. When <code>NULL</code> , the default priority stored in <code>SoftwareLicenseType</code> table will be used.
LicenseKeyValue	<i>Type:</i> text (max 400 characters). Nullable The license (installation) key value to be used when this license covers an installation of this application.

Database Column	Details
SoftwareTitleLicenseReasonID	<i>Type:</i> integer The reason that this application has been added to this license. Foreign key to the <code>SoftwareTitleLicenseReason</code> table.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
CreationDate	<i>Type:</i> datetime The date the record was created.

## SoftwareTitleLicenseProposal Table

The `SoftwareTitleLicenseProposal` table is used to store a summary of application changes to a particular software license. The changes have been inferred from changes to the license definition used to create the software license.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 435: Database columns for `SoftwareTitleLicenseProposal` table**

Database Column	Details
SoftwareTitleLicenseProposalID	<i>Type:</i> integer. Key. Generated ID Primary key for the <code>SoftwareTitleLicenseProposal</code> table.
SoftwareTitleID	<i>Type:</i> integer. Key Foreign key to the <code>SoftwareTitle</code> table.
OldPrimarySoftwareTitleID	<i>Type:</i> integer. Nullable The existing primary application of the license. This can be null if there is no primary application.
SoftwareLicenseID	<i>Type:</i> integer. Key Foreign key to the <code>SoftwareLicense</code> table.
SoftwareLicenseDefinitionID	<i>Type:</i> integer

Database Column	Details
	Foreign key to the <code>SoftwareLicenseDefinition</code> table.
<code>LicenseDefinitionVersion</code>	<i>Type:</i> integer The version of the license definition that has been used for these proposed changes.
<code>SoftwareTitleLicenseProposalActionID</code>	<i>Type:</i> integer. Key The proposed action for the software title on the software license.
<code>SoftwareLicenseProposalStatusID</code>	<i>Type:</i> integer The state of this software license change proposal.
<code>Conflicted</code>	<i>Type:</i> boolean Whether this license title change proposal conflicts with another for the same license.
<code>CreationUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
<code>CreationDate</code>	<i>Type:</i> datetime The date the record was created.
<code>UpdatedUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who updated the record.
<code>UpdatedDate</code>	<i>Type:</i> datetime. Nullable The date the record was updated.

## SoftwareTitleLicenseProposalAction Table

`SoftwareTitleLicenseProposalAction` is a static table listing all of the actions that can be proposed for a software title on a software license.

**Table 436: Database columns for `SoftwareTitleLicenseProposalAction` table**

Database Column	Details
<code>SoftwareTitleLicenseProposalActionID</code>	<i>Type:</i> integer. Key. Generated ID
<code>ResourceName</code>	<i>Type:</i> text (max 256 characters). Key



Database Column	Details
	The unique name of the localizable resource string representing the SoftwareTitleLicenseProposalAction record. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 256 characters) The text to display if the state resource string has no translation.

## SoftwareTitleLicenseReason Table

SoftwareTitleLicenseReason is a static table listing valid reasons why a software title was added to a license.

**Table 437: Database columns for SoftwareTitleLicenseReason table**

Database Column	Details
SoftwareTitleLicenseReasonID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each SoftwareTitleLicenseReason. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Manual</li> <li>• 2 = Current</li> <li>• 3 = Edition Downgrade</li> <li>• 4 = Version Downgrade</li> <li>• 5 = Version Upgrade.</li> </ul>
ReasonResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing the reason a license was linked to a title. Foreign key to the ComplianceResourceString table.
ReasonDefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the reason resource string has no translation.

## SoftwareTitleOracle Table

The SoftwareTitleOracle table stores attributes of an application installation that are relevant to Oracle applications only. These characteristics are important for Oracle licensing.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 438: Database columns for SoftwareTitleOracle table**

Database Column	Details
SoftwareTitleID	Type: integer. Key The Oracle application. Foreign key to the SoftwareTitle table.
MaximumSockets	Type: integer. Nullable The maximum number of sockets allowed on a computer where the application is installed.
NUPProcessorMultiplier	Type: integer. Nullable The multiplier value to use when determining the minimum Named User Plus licenses for the application.
OverrideSoftwareTitleTypeID	Type: integer. Nullable If this is not null, then the application was initially created as non-Oracle, but the operator wants to license it as an Oracle title. Foreign key to the SoftwareTitleType table.
IsShared	Type: boolean

## SoftwareTitleProduct Table

The “product”, unique for a given publisher, is the common name of a set of applications, independent of version or edition (for example, “Acrobat”).



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 439: Database columns for SoftwareTitleProduct table**

Database Column	Details
SoftwareTitleProductID	Type: integer. Key. Generated ID

Database Column	Details
	The unique identifier for a product.
SoftwareTitlePublisherID	<i>Type:</i> integer. Key. Nullable The publisher of this product. Foreign key to the <code>SoftwareTitlePublisher</code> table.
ProductName	<i>Type:</i> text (max 200 characters). Key The application's product name.
IsLocal	<i>Type:</i> boolean If this field is <code>False</code> , the product has come from the ARL. If it is <code>True</code> , then the product has been created by an operator.
IsShared	<i>Type:</i> boolean

## SoftwareTitleProperty Table

`SoftwareTitleProperty` defines extra custom properties for all applications.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 440: Database columns for `SoftwareTitleProperty` table**

Database Column	Details
SoftwareTitlePropertyID	<i>Type:</i> integer. Key. Generated ID The unique identifier for a software title property.
PropertyName	<i>Type:</i> text (max 256 characters). Key The name of the property.
CustomPropertyDisplayXMLID	<i>Type:</i> integer. Nullable Foreign key to a record in the <code>CustomPropertyDisplayXML</code> table, describing how to show the property on a property dialog.

# SoftwareTitlePropertyValue Table

For each application, `SoftwareTitlePropertyValue` stores the values for the custom properties defined in `SoftwareTitleProperty`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 441: Database columns for `SoftwareTitlePropertyValue` table**

Database Column	Details
<code>SoftwareTitlePropertyValueID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for a property value.
<code>SoftwareTitleID</code>	<i>Type:</i> integer. Key The title for which the property is being stored. Foreign key to the <code>SoftwareTitle</code> table.
<code>SoftwareTitlePropertyID</code>	<i>Type:</i> integer. Key The property whose value is being stored. Foreign key to the <code>SoftwareTitleProperty</code> table.
<code>PropertyValue</code>	<i>Type:</i> text (max 4000 characters) The property value.
<code>CreationUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who created the record.
<code>CreationDate</code>	<i>Type:</i> datetime The date the record was created.
<code>UpdatedUser</code>	<i>Type:</i> text (max 128 characters). Nullable The operator who last updated the record.
<code>UpdatedDate</code>	<i>Type:</i> datetime. Nullable The date the record was last updated.

## SoftwareTitlePublisher Table

Publishers of software applications (for example, "Microsoft"). Note that only application records take the publisher name from this table. License and contract records take the publisher name from the `Vendor` table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 442: Database columns for SoftwareTitlePublisher table**

Database Column	Details
SoftwareTitlePublisherID	Type: integer. Key. Generated ID The unique identifier for a publisher.
PublisherName	Type: text (max 200 characters). Key The publisher name.
IsLocal	Type: boolean If this field is <code>False</code> , the publisher has come from the ARL. If it is <code>True</code> , then the publisher has been created by an operator.
IsShared	Type: boolean

## SoftwareTitleRegistryEvidence Table

Reserved for future use.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 443: Database columns for SoftwareTitleRegistryEvidence table**

Database Column	Details
SoftwareTitleID	Type: integer. Key

Database Column	Details
	The software title to which the registry evidence is related. Foreign key to the <code>SoftwareTitle</code> table.
<code>RegistryEvidenceID</code>	<i>Type:</i> integer. Key The registry evidence related to the software title. Foreign key to the <code>RegistryEvidence</code> table.
<code>IsShared</code>	<i>Type:</i> boolean

## SoftwareTitleSuite Table

For software that has been classed as a suite (because it has other applications linking to it as component applications), `SoftwareTitleSuite` identifies how many of its member applications must be present for the installation to count as a suite installation, using “application evidence” for suite recognition.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 444: Database columns for `SoftwareTitleSuite` table**

Database Column	Details
<code>SoftwareTitleID</code>	<i>Type:</i> integer. Key The suite. Foreign key to the <code>SoftwareTitle</code> table.
<code>MinNumberApps</code>	<i>Type:</i> integer The minimum number of member applications of the software suite that must be installed.
<code>MinNumberAppsDefault</code>	<i>Type:</i> integer. Nullable The original, default value of <code>MinNumberApps</code> before it was changed.
<code>IsShared</code>	<i>Type:</i> boolean

## SoftwareTitleSuiteEx Table

The `SoftwareTitleSuiteEx` table contains additional information on the suite by FlexNet Manager Suite.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 445: Database columns for SoftwareTitleSuiteEx table**

Database Column	Details
SoftwareTitleID	Type: integer. Key A unique identifier for a software record.
MinNumberApps	Type: integer. Nullable The minimum number of member applications of the software suite that must be installed.

## SoftwareTitleType Table

`SoftwareTitleType` is a static table listing possible types of software (application) titles. This is used particularly to identify types that need special processing. It is quite distinct from license types.

**Table 446: Database columns for SoftwareTitleType table**

Database Column	Details
SoftwareTitleTypeID	Type: integer. Key. Generated ID A unique identifier for a <code>SoftwareTitleType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = General</li> <li>• 2 = Oracle Database</li> <li>• 3 = Oracle Option</li> <li>• 4 = Oracle Application</li> <li>• 5 = Oracle EBS Server</li> <li>• 6 = Oracle EBS.</li> </ul>
ResourceName	Type: text (max 256 characters). Key The unique name of the localizable resource string representing a document type. Foreign key to the <code>ComplianceResourceString</code> table.

Database Column	Details
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the type resource string has no translation.
InstanceTypeID	<i>Type:</i> integer The type of instance that can be created for this application. Foreign key to the <code>InstanceType</code> table.

## SoftwareTitleVersion Table

A list of application versions, which must be unique for a given product. Examples include “6.4”, “XP”, “Vista” and “2003”.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 447: Database columns for SoftwareTitleVersion table**

Database Column	Details
SoftwareTitleVersionID	<i>Type:</i> integer. Key. Generated ID The unique identifier for a version.
SoftwareTitleProductID	<i>Type:</i> integer. Key The version's product. Foreign key to the <code>SoftwareTitleProduct</code> table.
VersionName	<i>Type:</i> text (max 50 characters). Key The text for this application version.
VersionWeight	<i>Type:</i> decimal Version weight (for ordering, so we know which versions are upgrades/downgrades of other versions).
IsLocal	<i>Type:</i> boolean If this field is <code>False</code> , the version has come from the ARL. If it is <code>True</code> , then the version has been created by an operator.
IsShared	<i>Type:</i> boolean



## SoftwareTitleWMIEvidence Table

SoftwareTitleWMIEvidence links software titles to WMI evidence.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 448: Database columns for SoftwareTitleWMIEvidence table**

Database Column	Details
SoftwareTitleID	Type: integer. Key The software title to which the WMI evidence is related. Foreign key to the SoftwareTitle table.
WMIEvidenceID	Type: integer. Key The WMI evidence related to the software title. Foreign key to the WMIEvidence table.
IsLocal	Type: boolean If this field is <code>False</code> , the link has come from the ARL. If it is <code>True</code> , then the link has been created by an operator.
IsShared	Type: boolean

## SoftwareUserLicensePointsConsumedData Table

SoftwareUserLicensePointsConsumed records how many software license entitlements have been consumed for a given license by a given end-user.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 449: Database columns for SoftwareUserLicensePointsConsumedData table**

Database Column	Details
ComplianceUserID	Type: integer. Key

Database Column	Details
	The end-user. Foreign key to the <code>ComplianceUserSnapshot</code> table.
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key The license. Foreign key to the <code>SoftwareLicenseSnapshot</code> table.
<code>LicensesConsumed</code>	<i>Type:</i> integer The number of points (or entitlements) consumed for the license by the end-user.
<code>LicensesUsed</code>	<i>Type:</i> integer How many of the points consumed are for installations that are actually being used.
<code>LicenseMeasurementID</code>	<i>Type:</i> integer. Key The license measurement ID. Foreign key to the <code>LicenseMeasurement</code> table.

## SoftwareUserLicensePointsConsumedSuggested Table

`SoftwareUserLicensePointsConsumedSuggested` records how many software license entitlements would be consumed by an end-user for an optimized (suggested) license. Currently used to track optimized license usage suggested by FlexNet Manager for SAP.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 450: Database columns for `SoftwareUserLicensePointsConsumedSuggested` table**

Database Column	Details
<code>ComplianceUserID</code>	<i>Type:</i> integer. Key The end-user. Foreign key to the <code>ComplianceUser</code> table.
<code>SuggestedSoftwareLicenseID</code>	<i>Type:</i> integer. Key The suggested or optimized license. Foreign key to the <code>SoftwareLicense</code> table.
<code>LicensesConsumed</code>	<i>Type:</i> integer

Database Column	Details
	The number of points (or entitlements) consumed for the license by the end-user.
LicensesUsed	Type: integer How many of the points consumed are for installations that are actually being used.
LicenseMeasurementID	Type: integer. Key The associated SAP license measurement snapshot. Foreign key to the <code>LicenseMeasurement</code> table.

## SoftwareUserLicensePointsConsumedSuggestedHistory Table

`SoftwareUserLicensePointsConsumedSuggestedHistory` table records the history of suggested (optimised) license consumption.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 451: Database columns for `SoftwareUserLicensePointsConsumedSuggestedHistory` table**

Database Column	Details
ComplianceUserID	Type: integer. Key The end-user. Foreign key to the <code>ComplianceUser</code> table.
SuggestedSoftwareLicenseID	Type: integer. Key The suggested or optimized license. Foreign key to the <code>SoftwareLicense</code> table.
LicensesConsumed	Type: integer The number of points (or entitlements) consumed for the license by the end-user.
LicensesUsed	Type: integer How many of the points consumed are for installations that are actually being used.

Database Column	Details
LicenseMeasurementID	<p><i>Type:</i> integer. Key</p> <p>The associated SAP license measurement snapshot. Foreign key to the <code>LicenseMeasurement</code> table.</p>

## SoftwareUserLicensePointsHistory Table

`SoftwareUserLicensePointsHistory` records history of license consumption by end-users.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 452: Database columns for `SoftwareUserLicensePointsHistory` table**

Database Column	Details
ComplianceUserID	<p><i>Type:</i> integer. Key</p> <p>The end-user. Foreign key to the <code>ComplianceUser</code> table.</p>
SoftwareLicenseID	<p><i>Type:</i> integer. Key</p> <p>The license. Foreign key to the <code>SoftwareLicense</code> table.</p>
LicensesConsumed	<p><i>Type:</i> integer</p> <p>The number of points (or entitlements) consumed for the license by an end-user.</p>
LicensesUsed	<p><i>Type:</i> integer</p> <p>How many of the points consumed are for installations that are actually being used.</p>
LicenseMeasurementID	<p><i>Type:</i> integer. Key</p> <p>The associated SAP license measurement snapshot. Foreign key to the <code>LicenseMeasurement</code> table.</p>

## Tag Table

Reserved for future development.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 453: Database columns for Tag table**

Database Column	Details
TagID	Type: integer. Key. Generated ID The unique ID for this tag.
Name	Type: text (max 128 characters). Key The name of this tag.
Description	Type: text Description of this tag and its purpose.

## TargetOperatingSystemType Table

TargetOperatingSystemType; is a static table listing all types of OSes that can be targeted by licensing.

**Table 454: Database columns for TargetOperatingSystemType table**

Database Column	Details
TargetOperatingSystemTypeID	Type: integer. Key. Generated ID A unique identifier for each TargetOperatingSystemType. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = All</li> <li>• 2 = Windows Server operating systems</li> <li>• 3 = Windows desktop operating systems</li> <li>• 4 = Non Windows Server operating systems</li> </ul>
ResourceName	Type: text (max 256 characters). Key The unique name of the localizable resource string representing an Operating System family. Foreign key to the ComplianceResourceString table.
DefaultValue	Type: text (max 100 characters)

Database Column	Details
	The text to display if the type resource string has no translation.

## VDI Table

VDI is the list of VDI devices



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 455: Database columns for VDI table**

Database Column	Details
VDIID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a VDI device.
ComputerName	<i>Type:</i> text (max 256 characters). Key The computer name of the VDI.
ComplianceDomainID	<i>Type:</i> integer. Key. Nullable The domain the VDI is a member of. Foreign key to the <code>ComplianceDomain</code> table.
VDIGroupID	<i>Type:</i> integer. Key The VDI group the VDI device belongs to. Foreign key to the <code>VDIGroup</code> table.
VDITemplateID	<i>Type:</i> integer. Key The master VM template of the VDI. Foreign key to the <code>VDITemplate</code> table.
RetiredDate	<i>Type:</i> datetime. Key. Nullable The date the VDI device was deleted.
ApplicationDeliveryOnly	<i>Type:</i> boolean. Key Determines whether the VDI device is used only to server applications.

## VDIEndPointAccess Table

VDIEndPointAccess is the list of endpoint devices that have accessed VDI devices



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 456: Database columns for VDIEndPointAccess table**

Database Column	Details
VDIEndPointAccessID	Type: integer. Key. Generated ID A unique identifier for an endpoint device accessing a VDI.
ComplianceComputerID	Type: integer. Key. Nullable A unique identifier for the endpoint. Foreign key to the ComplianceComputer table.
ComplianceUserID	Type: integer. Key. Nullable A unique identifier for the endpoint user. Foreign key to the ComplianceUser table.
VDIID	Type: integer. Key A unique identifier for the VDI. Foreign key to the VDI table.
LogonTime	Type: datetime. Nullable The date the user logged on to the VDI.

## VDIGroup Table

VDIGroup stores the list of available VDI groups in a VDI environment.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 457: Database columns for VDIGroup table**

Database Column	Details
VDIGroupID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a VDI Group record.
GroupName	<i>Type:</i> text (max 128 characters). Key The VDI Group name
VDISiteID	<i>Type:</i> integer. Key The VDI site ID
VDIGroupUUID	<i>Type:</i> unique identifier. Key. Nullable The UUID of the VDI group

## VDISite Table

VDISite stores the list of available VDI sites.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 458: Database columns for VDISite table**

Database Column	Details
VDISiteID	<i>Type:</i> integer. Key. Generated ID A unique identifier for a VDI site record.
SiteName	<i>Type:</i> text (max 256 characters). Key The VDI Group name
AccessModeID	<i>Type:</i> integer. Key The access mode of the VDI site. Foreign key to the <i>AccessMode</i> table.



## VDITemplate Table

VDITemplate stores the list of available VDI groups in a VDI environment.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 459: Database columns for VDI Template table**

Database Column	Details
VDITemplateID	Type: integer. Key. Generated ID A unique identifier for a VDI Group record.
TemplateName	Type: text (max 256 characters). Key The VDI template name.
VDISiteID	Type: integer. Key. Nullable The VDI template's site ID
ComplianceComputerID	Type: integer. Key. Nullable The VDI template's ComplianceComputerID

## VDIUser Table

VDIUser is the list of users that have access to VDI groups



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 460: Database columns for VDI User table**

Database Column	Details
VDIUserID	Type: integer. Key. Generated ID A unique identifier for a VDI User.
VDIGroupID	Type: integer. Key

Database Column	Details
	A unique identifier for the VDI Group. Foreign key to the <code>VDIGroup</code> table.
<code>ComplianceUserID</code>	<p>Type: integer. Key. Nullable</p> <p>A unique identifier for the user with access to a VDI Group. Foreign key to the <code>ComplianceUser</code> table.</p>

## WMIEvidence Table

`WMIEvidence` lists WMI evidence that is used to identify that a particular item of software (defined in the `SoftwareTitle` table) has been installed on a computer.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 461: Database columns for `WMIEvidence` table**

Database Column	Details
<code>WMIEvidenceID</code>	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for a WMI evidence record.</p>
<code>ClassName</code>	<p>Type: text (max 50 characters). Key</p> <p>The WMI class name of the WMI evidence.</p>
<code>PropertyName</code>	<p>Type: text (max 50 characters). Key</p> <p>The WMI property name of the WMI evidence.</p>
<code>PropertyValue</code>	<p>Type: text (max 256 characters). Key</p> <p>The value of the WMI evidence property.</p>
<code>Ignored</code>	<p>Type: boolean</p> <p>Set this field to <code>True</code> if this WMI evidence is ignored for application recognition.</p>
<code>IsShared</code>	Type: boolean

## WMIEvidenceMatchCount Table

WMIEvidenceMatchCount tracks the number of times that each WMI evidence (rule) has been detected as installed and recorded in the data source. A separate count is kept for each WMI evidence rule, and for each data source.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 462: Database columns for WMIEvidenceMatchCount table**

Database Column	Details
WMIEvidenceMatchCountID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A synthetic unique identifier is required, since <i>ComplianceConnectionID</i>, being nullable, cannot be included in the primary key.</p>
WMIEvidenceID	<p><i>Type:</i> integer. Key</p> <p>WMI evidence rule being matched. Foreign key to the <i>WMIEvidence</i> table.</p>
ComplianceConnectionID	<p><i>Type:</i> integer. Key. Nullable</p> <p>Data source where the match is occurring. Foreign key to the <i>ComplianceConnection</i> table.</p>
MatchedCount	<p><i>Type:</i> integer</p> <p>The number of installed WMI evidence records in this data source matching this WMI evidence rule.</p>
InstallCount	<p><i>Type:</i> integer</p> <p>The number of physical application installations recognized in this data source using this WMI evidence rule.</p>

## Compliance.Logic.Structure Tables

The complete set of database tables documented here includes:

- ComplianceDomain table (see *ComplianceDomain Table* on page 460)
- GroupEx table (see *GroupEx Table* on page 460)
- GroupExPathCultureType table (see *GroupExPathCultureType Table* on page 463)

- GroupType table (see *GroupType Table* on page 463)
- MemberEx table (see *MemberEx Table* on page 464)
- RoleRight table (see *RoleRight Table* on page 465)

## ComplianceDomain Table

Stores a list of domain names imported FlexNet Manager Suite.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 463: Database columns for ComplianceDomain table**

Database Column	Details
ComplianceDomainID	Type: integer. Key. Generated ID A unique identifier for a domain.
QualifiedName	Type: text (max 100 characters) The fully qualified name of the domain.
FlatName	Type: text (max 32 characters) The flat name of the domain.

## GroupEx Table

The GroupEx table stores information about enterprise groups and roles.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 464: Database columns for GroupEx table**

Database Column	Details
GroupID	Type: integer. Key. Generated ID

Database Column	Details
	A unique identifier for a group.
GroupTypeID	<i>Type:</i> integer. Key Identifies the group type. Foreign key to the <code>GroupType</code> table.
BusinessView	<i>Type:</i> boolean. Key Set this to <code>True</code> if the group is a business view (that is, is a group heading like <code>Roles</code> or <code>Categories</code> ).
Path	<i>Type:</i> text (max 500 characters) Complete path of the group.
NextChild	<i>Type:</i> integer The ID number for the next child to be created under this group. Internal use only: do not edit.
GroupExID	<i>Type:</i> text (max 128 characters). Key Unique string identifier for this extension record.
BusinessPhoneNumber	<i>Type:</i> text (max 30 characters). Nullable The business phone number of the group.
FaxPhoneNumber	<i>Type:</i> text (max 30 characters). Nullable The fax number of the group.
Address_Street	<i>Type:</i> text (max 200 characters). Nullable The street address of the group.
Address_City	<i>Type:</i> text (max 200 characters). Nullable The city of the group.
Address_State	<i>Type:</i> text (max 200 characters). Nullable The state of the group.
Address_ZIP	<i>Type:</i> text (max 20 characters). Nullable The ZIP or postal code of the group.
Address_Country	<i>Type:</i> text (max 100 characters). Nullable The country of the group.
Email	<i>Type:</i> text (max 200 characters). Nullable

Database Column	Details
	The email address of the group.
Comments	<i>Type:</i> text. Nullable Comments about the group.
IsStockLocation	<i>Type:</i> boolean For locations only. If this field is set to <code>True</code> , the location is considered to be a stock or storage location.
ContactID	<i>Type:</i> integer. Nullable A contact person for this group. This field is no longer in use in FlexNet Manager Suite
ManagerID	<i>Type:</i> integer. Nullable A manager for this group. This field is no longer in use in FlexNet Manager Suite
GroupCN	<i>Type:</i> text (max 256 characters). Nullable The common name for the group.
NameResourceName	<i>Type:</i> text (max 256 characters). Nullable The unique name of the localizable resource string representing an enterprise group name (GroupCN). Foreign key to the <code>ComplianceResourceString</code> table.
DescriptionResourceName	<i>Type:</i> text (max 256 characters). Nullable The unique name of the localizable resource string representing an enterprise group description (Comments). Foreign key to the <code>ComplianceResourceString</code> table.
ParentGroupExID	<i>Type:</i> text (max 128 characters). Key. Nullable Unique string identifier for the parent record.
TreeLevel	<i>Type:</i> integer. Nullable The level of this group in the hierarchy.
TreePath	<i>Type:</i> text (max 4000 characters). Key. Nullable A generated path that can be used to sort groups in tree order.
IsShared	<i>Type:</i> boolean

## GroupExPathCultureType Table

The `GroupExPathCultureType` table stores complete enterprise group paths per culture type for each enterprise group.

**Table 465: Database columns for `GroupExPathCultureType` table**

Database Column	Details
GroupID	<i>Type:</i> integer. Key The ID of the group the translated path belongs to.
CultureType	<i>Type:</i> text (max 12 characters). Key A unique identifier for a culture type.
Path	<i>Type:</i> text (max 500 characters) The translated group path for the specific culture type.
TreePath	<i>Type:</i> text (max 4000 characters) A generated path that can be used to sort groups in tree order.

## GroupType Table

The collection of types of enterprise groups, such as locations, departments, and cost centers.

**Table 466: Database columns for `GroupType` table**

Database Column	Details
GroupTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>GroupType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Location</li> <li>• 2 = Departments</li> <li>• 3 = Cost Center</li> <li>• 4 = Category</li> <li>• 5 = Role.</li> </ul>
Description	<i>Type:</i> text (max 255 characters). Key A description of the type of enterprise group.

Database Column	Details
ResourceName	<p>Type: text (max 256 characters). Key. Nullable</p> <p>The unique name of the localizable resource string representing a group type.</p> <p>Foreign key to the <code>ComplianceResourceString</code> table.</p>

## MemberEx Table

The `MemberEx` table stores the membership lists for every enterprise group or role.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 467: Database columns for `MemberEx` table**

Database Column	Details
GroupID	<p>Type: integer. Key</p> <p>The <code>GroupEx</code> to which the member belongs.</p>
TargetTypeID	<p>Type: integer. Key</p> <p>The <code>TargetType</code>. Possible values are:</p> <ul style="list-style-type: none"> <li>• 3 = Enterprise Group</li> <li>• 9 = Asset</li> <li>• 10 = Contract</li> <li>• 11 = Purchase Order</li> <li>• 12 = Software License</li> <li>• 13 = Software Title</li> <li>• 14 = Computer</li> <li>• 15 = User</li> <li>• 16 = Operator</li> <li>• 17 = SAP system landscapes</li> <li>• 18 = SAP systems</li> <li>• 19 = SAP rule sets</li> </ul>



Database Column	Details
TargetID	<p>Type: integer. Key</p> <p>The Asset, Contract, etc. identifier, depending on TargetType.</p>

## RoleRight Table

Each action by FlexNet Manager Suite requires the role to have one or more `RoleRights` to perform an `ActionClass` over a given `Resource`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 468: Database columns for RoleRight table**

Database Column	Details
GroupID	<p>Type: integer. Key</p> <p>The role to whom the right is granted or denied.</p>
ResourceID	<p>Type: integer. Key</p> <p>The Resource to which the RoleRight applies.</p>
ActionClassID	<p>Type: integer. Key</p> <p>The action class which applies (read or modify).</p>
Denied	<p>Type: boolean</p> <p>When TRUE (1), indicates that the specified right is denied.</p>
ScopeGroupID	<p>Type: integer. Key. Nullable</p> <p>The enterprise group to which the right for this role applies, if applicable.</p>

## Compliance.Logic.Users Tables

The complete set of database tables documented here includes:

- ComplianceUser table (see *ComplianceUser Table* on page 466)
- ComplianceUserConnection table (see *ComplianceUserConnection Table* on page 469)

- ComplianceUserInventorySourceType table (see *ComplianceUserInventorySourceType Table* on page 470)
- ComplianceUserStatus table (see *ComplianceUserStatus Table* on page 470)
- EmploymentStatus table (see *EmploymentStatus Table* on page 471)
- UserSuffix table (see *UserSuffix Table* on page 472)
- UserTitle table (see *UserTitle Table* on page 472)

## ComplianceUser Table

ComplianceUser stores information about end-users in the enterprise, including contact details, login details and inventory source details (if applicable). End-users in ComplianceUser will not be able to log in to FlexNet Manager Suite unless they have a corresponding record in the ComplianceOperator table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 469: Database columns for ComplianceUser table**

Database Column	Details
ComplianceUserID	Type: integer. Key. Generated ID A unique identifier for the end-user.
UserName	Type: text (max 512 characters). Nullable The end-user's full name. When creating a new end-user manually, defaults to a concatenation of title, first name, middle name, last name and suffix.
SAMAccountName	Type: text (max 64 characters). Key. Nullable The login name (SAM account name) of the end-user.
ComplianceDomainID	Type: integer. Key. Nullable Domain that the end-user belongs to. Foreign key to the ComplianceDomain table.
LocationID	Type: text (max 128 characters). Key. Nullable Any enterprise location associated with this end-user. Foreign key to the GroupEx table.
BusinessUnitID	Type: text (max 128 characters). Key. Nullable

Database Column	Details
	Any corporate unit in the enterprise associated with this end-user. Foreign key to the GroupEx table.
CostCenterID	Type: text (max 128 characters). Key. Nullable Any cost center in the enterprise associated with this end-user. Foreign key to the GroupEx table.
CategoryID	Type: text (max 128 characters). Key. Nullable No longer in use, but retained for legacy systems. Any enterprise category associated with this end-user. Foreign key to the GroupEx table.
EmployeeNumber	Type: text (max 128 characters). Key. Nullable The employee number of the end-user (as defined in an organization's own HR system).
UserTitleID	Type: integer. Nullable The title of the end-user. Foreign key to the UserTitle table.
FirstName	Type: text (max 128 characters). Nullable The first name of the end-user.
MiddleName	Type: text (max 128 characters). Nullable The middle name(s) of the end-user.
LastName	Type: text (max 128 characters). Nullable The last name (surname) of the end-user.
UserSuffixID	Type: integer. Nullable The suffix to the name of the end-user. Foreign key to the UserSuffix table.
JobTitle	Type: text (max 128 characters). Nullable The job title of the end-user.
BusinessPhoneNumber	Type: text (max 30 characters). Nullable The work phone number of the end-user.
MobilePhoneNumber	Type: text (max 30 characters). Nullable The mobile phone number of the end-user.
FaxPhoneNumber	Type: text (max 30 characters). Nullable The fax number of the end-user.

Database Column	Details
Address_Street	<i>Type:</i> text (max 200 characters). Nullable The street address of the end-user.
Address_City	<i>Type:</i> text (max 200 characters). Nullable The city or suburb name of the end-user.
Address_State	<i>Type:</i> text (max 200 characters). Nullable The state or province of the end-user.
Address_ZIP	<i>Type:</i> text (max 20 characters). Nullable The ZIP or postal code of the end-user.
Address_Country	<i>Type:</i> text (max 100 characters). Nullable The country of the end-user.
Email	<i>Type:</i> text (max 200 characters). Key. Nullable The email address of the end-user.
Messenger	<i>Type:</i> text (max 200 characters). Nullable The instant messenger address of the end-user.
ManagerID	<i>Type:</i> integer. Key. Nullable The manager of the end-user. Foreign key to another end-user in the <code>ComplianceUser</code> table.
CurrencyID	<i>Type:</i> integer. Nullable No longer in use - default currency is now stored in the <code>OperatorTenantSetting</code> table.
UserStatusID	<i>Type:</i> integer The end-user's status. Foreign key to the <code>ComplianceUserStatus</code> table.
EmploymentStatusID	<i>Type:</i> integer. Nullable The end-user's employment status. Foreign key to the <code>EmploymentStatus</code> table.
IsIncluded	<i>Type:</i> boolean If <code>False</code> , the end-user's login name is in the list of excluded accounts, and this end-user will not consume licenses or be recorded as the last-logged-on or

Database Column	Details
	calculated end-user of a computer. This end-user will also not appear in many lists of end-users.
CreationUser	Type: text (max 128 characters). Nullable The operator who created the record.
CreationDate	Type: datetime The date the record was created.
UpdatedUser	Type: text (max 128 characters). Nullable The operator who last updated the record.
UpdatedDate	Type: datetime. Nullable The date the record was last updated.
ComplianceUser InventorySourceTypeID	Type: integer Whether this end-user has ever been reported in inventory, or has been manually created and maintained. Foreign key to the ComplianceUserInventorySourceType table.
InventoryAgent	Type: text (max 64 characters). Nullable If this end-user is reported in inventory, the name of the person or tool that performed the last inventory.

## ComplianceUserConnection Table

ComplianceUserConnection stores a link between end-users in ComplianceUser which have been reported in inventory, and external IDs which can be used to identify them in their inventory sources. End-users reported in multiple inventory sources will appear multiple times in this table.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 470: Database columns for ComplianceUserConnection table**

Database Column	Details
ComplianceUserID	Type: integer. Key

Database Column	Details
	A unique identifier for the end-user. Foreign key to the <code>ComplianceUser</code> table.
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key The inventory source where the end-user was reported. Foreign key to the <code>ComplianceConnection</code> table.
<code>ExternalID</code>	<i>Type:</i> big integer. Key A (hopefully unique) identifier for the end-user in the external inventory source.

## ComplianceUserInventorySourceType Table

`ComplianceUserInventorySourceType` is a static table used to define possible end-user inventory source values (that is, whether the end-user was created manually or reported by the compliance importer).

**Table 471: Database columns for `ComplianceUserInventorySourceType` table**

Database Column	Details
<code>ComplianceUserInventorySourceTypeID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>ComplianceUserInventorySourceType</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>1 = Automatic (end-user was recently updated during an inventory import)</li> <li>2 = Manual (end-user was created manually by an operator, using FlexNet Manager Suite, and has never been updated by the compliance importer).</li> </ul>
<code>ResourceName</code>	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing an inventory source. Foreign key to the <code>ComplianceResourceString</code> table.
<code>DefaultValue</code>	<i>Type:</i> text (max 100 characters) The text to display if the inventory source resource string has no translation.

## ComplianceUserStatus Table

`ComplianceUserStatus` is a static table listing status values for end-user.

**Table 472: Database columns for ComplianceUserStatus table**

Database Column	Details
ComplianceUserStatusID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>ComplianceUserStatus</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Active</li> <li>• 2 = Inactive</li> <li>• 3 = Retired</li> <li>• 4 = On leave</li> <li>• 5 = Pending (perhaps for an employee just starting with the company).</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an end-user status. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the status resource string has no translation.</p>
IsUserActive	<p><i>Type:</i> boolean. Key</p> <p>If this field is set to <code>False</code>, any end-user with this status will not consume licenses or be recorded as the last-logged-on or calculated end-user of a computer. This end-user will also not appear in many lists of end-users.</p>

## EmploymentStatus Table

`EmploymentStatus` is a static table listing possible employment statuses values for end-users.

**Table 473: Database columns for EmploymentStatus table**

Database Column	Details
EmploymentStatusID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>EmploymentStatus</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Employee</li> <li>• 2 = Consultant</li> <li>• 3 = Temporary</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 4 = Part time</li> <li>• 5 = Casual.</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing an employment status. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the employment status resource string has no translation.

## UserSuffix Table

`UserSuffix` is a static table listing possible name suffixes for end-users.

**Table 474: Database columns for `UserSuffix` table**

Database Column	Details
UserSuffixID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each <code>UserSuffix</code> . Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Jr.</li> <li>• 2 = Sr.</li> <li>• 3 = I</li> <li>• 4 = II</li> <li>• 5 = III.</li> </ul>
ResourceString	<i>Type:</i> text (max 256 characters). Key The unique name of the localizable resource string representing an end-user name suffix. Foreign key to the <code>ComplianceResourceString</code> table.
DefaultString	<i>Type:</i> text (max 100 characters) The text to display if the suffix resource string has no translation.

## UserTitle Table

`UserTitle` is a static table listing the possible titles of end-users.



**Table 475: Database columns for UserTitle table**

Database Column	Details
UserTitleID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for each <code>UserTitle</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Mr.</li> <li>• 2 = Miss</li> <li>• 3 = Mrs.</li> <li>• 4 = Ms.</li> <li>• 5 = Dr.</li> <li>• 6 = Prof.</li> </ul>
ResourceString	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The unique name of the localizable resource string representing an end-user title. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultString	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the title resource string has no translation.</p>

## Compliance.SAP Tables

The complete set of database tables documented here includes:

- `SAPActivityCheckMultipleLogons` table (see *SAPActivityCheckMultipleLogons Table* on page 476)
- `SAPActivityCheckSummary` table (see *SAPActivityCheckSummary Table* on page 477)
- `SAPActivityCheckWorkTime` table (see *SAPActivityCheckWorkTime Table* on page 478)
- `SAPCompositeRole` table (see *SAPCompositeRole Table* on page 479)
- `SAPConnectivityDirectionType` table (see *SAPConnectivityDirectionType Table* on page 479)
- `SAPConnectivityType` table (see *SAPConnectivityType Table* on page 480)
- `SAPConsolidatedUser` table (see *SAPConsolidatedUser Table* on page 481)
- `SAPConsolidatedUserDuplicate` table (see *SAPConsolidatedUserDuplicate Table* on page 482)
- `SAPConsumption` table (see *SAPConsumption Table* on page 482)
- `SAPContentEngine` table (see *SAPContentEngine Table* on page 483)
- `SAPContentEngineRule` table (see *SAPContentEngineRule Table* on page 484)

- SAPDuplicateUserRecommendation table (see *SAPDuplicateUserRecommendation Table* on page 485)
- SAPEngine table (see *SAPEngine Table* on page 487)
- SAPEngineConsumptionSummary table (see *SAPEngineConsumptionSummary Table* on page 487)
- SAPEngineMetric table (see *SAPEngineMetric Table* on page 489)
- SAPEngineMetricName table (see *SAPEngineMetricName Table* on page 489)
- SAPEngineName table (see *SAPEngineName Table* on page 490)
- SAPEnginePeriodType table (see *SAPEnginePeriodType Table* on page 490)
- SAPEngineSystemConsumption table (see *SAPEngineSystemConsumption Table* on page 491)
- SAPImportedInventoryFileDigest table (see *SAPImportedInventoryFileDigest Table* on page 492)
- SAPLicenseRatio table (see *SAPLicenseRatio Table* on page 493)
- SAPLicenseRecommendation table (see *SAPLicenseRecommendation Table* on page 494)
- SAPLicenseType table (see *SAPLicenseType Table* on page 496)
- SAPLicenseTypeHierarchy table (see *SAPLicenseTypeHierarchy Table* on page 497)
- SAPLicenseTypeName table (see *SAPLicenseTypeName Table* on page 498)
- SAPModule table (see *SAPModule Table* on page 498)
- SAPMultipleLogon table (see *SAPMultipleLogon Table* on page 499)
- SAPObject table (see *SAPObject Table* on page 499)
- SAPObjectType table (see *SAPObjectType Table* on page 500)
- SAPRFCCConnection table (see *SAPRFCCConnection Table* on page 500)
- SAPRFCCConnectionSummary table (see *SAPRFCCConnectionSummary Table* on page 502)
- SAPRecommendationAdjustmentReason table (see *SAPRecommendationAdjustmentReason Table* on page 503)
- SAPRecommendationProcessedStatus table (see *SAPRecommendationProcessedStatus Table* on page 504)
- SAPRecommendationSet table (see *SAPRecommendationSet Table* on page 504)
- SAPRecommendationSetStatus table (see *SAPRecommendationSetStatus Table* on page 506)
- SAPRecommendationSetSummary table (see *SAPRecommendationSetSummary Table* on page 506)
- SAPRole table (see *SAPRole Table* on page 508)
- SAPRoleConsumption table (see *SAPRoleConsumption Table* on page 508)
- SAPRoleTransactionCode table (see *SAPRoleTransactionCode Table* on page 509)
- SAPRule table (see *SAPRule Table* on page 510)
- SAPRuleAlgorithm table (see *SAPRuleAlgorithm Table* on page 511)

- SAPRuleCategory table (see *SAPRuleCategory Table* on page 512)
- SAPRuleMapping table (see *SAPRuleMapping Table* on page 512)
- SAPRuleSet table (see *SAPRuleSet Table* on page 513)
- SAPRuleSetMapping table (see *SAPRuleSetMapping Table* on page 515)
- SAPRuleType table (see *SAPRuleType Table* on page 516)
- SAPSecurityUser table (see *SAPSecurityUser Table* on page 516)
- SAPSystem table (see *SAPSystem Table* on page 517)
- SAPSystemActivityCheckSummary table (see *SAPSystemActivityCheckSummary Table* on page 521)
- SAPSystemEngineMetric table (see *SAPSystemEngineMetric Table* on page 522)
- SAPSystemEnvironment table (see *SAPSystemEnvironment Table* on page 523)
- SAPSystemGroup table (see *SAPSystemGroup Table* on page 524)
- SAPSystemLandscape table (see *SAPSystemLandscape Table* on page 525)
- SAPSystemLandscapeEngine table (see *SAPSystemLandscapeEngine Table* on page 526)
- SAPSystemLandscapeEngineMapping table (see *SAPSystemLandscapeEngineMapping Table* on page 528)
- SAPSystemLandscapeLicenseType table (see *SAPSystemLandscapeLicenseType Table* on page 529)
- SAPSystemLandscapeLicenseTypeHierarchy table (see *SAPSystemLandscapeLicenseTypeHierarchy Table* on page 530)
- SAPSystemLandscapeLicenseTypeImport table (see *SAPSystemLandscapeLicenseTypeImport Table* on page 531)
- SAPSystemLandscapeStatus table (see *SAPSystemLandscapeStatus Table* on page 532)
- SAPSystemModule table (see *SAPSystemModule Table* on page 532)
- SAPSystemObject table (see *SAPSystemObject Table* on page 533)
- SAPSystemPriceList table (see *SAPSystemPriceList Table* on page 534)
- SAPSystemPriceListName table (see *SAPSystemPriceListName Table* on page 534)
- SAPSystemRFCCConnectionSummary table (see *SAPSystemRFCCConnectionSummary Table* on page 535)
- SAPSystemType table (see *SAPSystemType Table* on page 536)
- SAPTransactionProfile table (see *SAPTransactionProfile Table* on page 536)
- SAPTransactionProfileObject table (see *SAPTransactionProfileObject Table* on page 537)
- SAPUser table (see *SAPUser Table* on page 538)
- SAPUserRole table (see *SAPUserRole Table* on page 541)
- SAPUserType table (see *SAPUserType Table* on page 542)

# SAPActivityCheckMultipleLogons Table

This table stores SAP activity check data related to work time.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 476: Database columns for SAPActivityCheckMultipleLogons table**

Database Column	Details
SAPActivityCheckMultipleLogonsID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP activity check multiple login data.
SAPActivityCheckMultipleLogonsUID	<i>Type:</i> text (max 32 characters). Key The SAP unique identifier for the SAP activity check multiple login data.
SAPUserID	<i>Type:</i> integer. Key Foreign key to the SAP user.
SAPSystemLandscapeID	<i>Type:</i> integer. Key Foreign key to the system landscape that the SAP activity check multiple login data belongs to.
MeasurementDate	<i>Type:</i> datetime The date that the SAP activity check multiple login data was measured.
MeasurementPeriodStartDate	<i>Type:</i> datetime The start date that the SAP activity check multiple login data was measured from.
MeasurementPeriodEndDate	<i>Type:</i> datetime The end date that the SAP activity check multiple login data was measured to.
NumberOfMultipleLogons	<i>Type:</i> integer The number of logons the user account has made from different systems at the same time during the measurement period.
MultipleLogonsPeakDate	<i>Type:</i> datetime

Database Column	Details
	The date where the number of logons the user account has made from different systems at the same time during the measurement period reached its peak value.

## SAPActivityCheckSummary Table

This table stores SAP activity check summary data.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 477: Database columns for SAPActivityCheckSummary table**

Database Column	Details
SAPActivityCheckSummaryID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP activity check summary.
SAPUserID	<i>Type:</i> integer. Key Foreign key to the SAP user.
SAPSystemLandscapeID	<i>Type:</i> integer. Key Foreign key to the system landscape that the SAP activity check work time data belongs to.
HasExceededBreakDuration	<i>Type:</i> boolean Indicates whether or not the user has exceeded the minimum required break duration.
HasMultipleLogons	<i>Type:</i> boolean Indicates whether or not the user has multiple logons.
IsHidden	<i>Type:</i> boolean Is this record marked as hidden in the UI.

# SAPActivityCheckWorkTime Table

This table stores SAP activity check data related to work time.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 478: Database columns for SAPActivityCheckWorkTime table**

Database Column	Details
SAPActivityCheckWorkTimeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP activity check work time data.
SAPActivityCheckWorkTimeUID	<i>Type:</i> text (max 32 characters). Key The SAP unique identifier for the SAP activity check work time data.
SAPUserID	<i>Type:</i> integer. Key Foreign key to the SAP user.
SAPSystemLandscapeID	<i>Type:</i> integer. Key Foreign key to the system landscape that the SAP activity check work time data belongs to.
MeasurementDate	<i>Type:</i> datetime The date that the SAP activity check work time data was measured.
MeasurementPeriodStartDate	<i>Type:</i> datetime The start date that the SAP activity check work time data was measured from.
MeasurementPeriodEndDate	<i>Type:</i> datetime The end date that the SAP activity check work time data was measured to.
BreakDurationSetting	<i>Type:</i> integer The minimum number of seconds that a user must not be running any transactions in a 24 hour period.
TableName	<i>Type:</i> text (max 256 characters). Key The name of the SAP table that was accessed during the minimum required break period.

Database Column	Details
BreakDurationResult	<i>Type:</i> integer The number of days that the user has not meet the minimum break duration setting during the measurement period.

## SAPCompositeRole Table

This table stores SAP composite roles.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 479: Database columns for SAPCompositeRole table**

Database Column	Details
SAPCompositeRoleID	<i>Type:</i> integer. Key. Generated ID A unique identifier for SAP composite role.
CompositeRoleID	<i>Type:</i> integer. Key Foreign key to SAP role which contain one or more single roles.
SingleRoleID	<i>Type:</i> integer. Key Foreign key to SAP role that is a member if the composite role.

## SAPConnectivityDirectionType Table

This table stores SAP connectivity direction type.

**Table 480: Database columns for SAPConnectivityDirectionType table**

Database Column	Details
SAPConnectivityDirectionTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP connectivity direction type.
TypeName	<i>Type:</i> text (max 64 characters). Key

Database Column	Details
	<p>A unique lookup for each <code>SAPConnectivityDirectionType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• Out</li> <li>• In</li> <li>• InOut</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>A localizable resource string representing a SAP connectivity type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the SAP connectivity type resource string has no translation.</p>

## SAPConnectivityType Table

This table stores SAP connectivity type.

**Table 481: Database columns for SAPConnectivityType table**

Database Column	Details
SAPConnectivityTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for the SAP connectivity type.</p>
TypeName	<p><i>Type:</i> text (max 64 characters). Key</p> <p>A unique lookup for each <code>SAPConnectivityType</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• Interactive</li> <li>• Batch</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>A localizable resource string representing a SAP connectivity type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the SAP connectivity type resource string has no translation.</p>



# SAPConsolidatedUser Table

This table stores the data specific to an SAP consolidated user.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 482: Database columns for SAPConsolidatedUser table**

Database Column	Details
SAPConsolidatedUserID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP consolidated user.
UserUID	<i>Type:</i> text (max 128 characters). Key A globally unique identifier for the SAP license recommendation.
SAPRecommendationSetID	<i>Type:</i> integer. Key Foreign key to the SAP recommendation set that the duplicate user recommendation belongs to.
SAPUserID	<i>Type:</i> integer. Key Foreign key to the SAP user that the duplicate user recommendation belongs to.
UserName	<i>Type:</i> text The user name of the user that the duplicate user recommendation belongs to.
DuplicateGroupNum	<i>Type:</i> integer The unique identifier showing which users are duplicates of one another.
LicenseType	<i>Type:</i> text (max 2 characters). Nullable The license code originally assigned to the user.
IsConsolidatedBySAP	<i>Type:</i> boolean Whether or not this user is consolidated by SAP.
OptimalLicenseType	<i>Type:</i> text (max 2 characters). Nullable The license code recommended the user be assigned ignoring license ratios and rebalancing.

## SAPConsolidatedUserDuplicate Table

This table stores the data specific to an SAP consolidated user duplicate.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 483: Database columns for SAPConsolidatedUserDuplicate table**

Database Column	Details
SAPConsolidatedUserDuplicateID	Type: integer. Key. Generated ID A unique identifier for the SAP consolidated duplicate user.
SAPRecommendationSetID	Type: integer. Key Foreign key to the SAP recommendation set that the duplicate user belongs to.
SAPConsolidatedUserID	Type: integer. Key Foreign key to the SAP consolidated user linked to a SAP user.
SAPUserID	Type: integer. Key Foreign key to the SAP user that the duplicate user belongs to.
IsConsolidatedBySAP	Type: boolean. Key Whether or not this user is consolidated by SAP.

## SAPConsumption Table

This table stores the data related to the definition of SAP consumption data.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 484: Database columns for SAPConsumption table**

Database Column	Details
SAPConsumptionID	Type: integer. Key. Generated ID

Database Column	Details
	The unique identifier for the SAP consumption.
SAPUserID	<i>Type:</i> integer. Key Foreign key to the SAP user that the consumption belongs to.
TimePeriodStartDate	<i>Type:</i> datetime. Key The date and time of the consumption
AccountObject	<i>Type:</i> text (max 40 characters). Key The account object
AccountObjectDetails	<i>Type:</i> text (max 40 characters). Key The account object details
EntryType	<i>Type:</i> text (max 1 characters). Key The consumption entry type
TaskType	<i>Type:</i> text (max 2 characters). Key The consumption task type
CPUTime	<i>Type:</i> decimal. Key. Nullable CPU usage in seconds
MemoryUsed	<i>Type:</i> big integer. Nullable Memory used
PrivateMemoryUsed	<i>Type:</i> big integer. Nullable Private memory used
AccessCount	<i>Type:</i> integer. Nullable Number of times the object has been used/accessed.

## SAPContentEngine Table

This table stores an engine from downloadable content.

**Table 485: Database columns for SAPContentEngine table**

Database Column	Details
SAPContentEngineID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP content engine table.
EngineContentUID	<i>Type:</i> text (max 128 characters). Key A global unique identifier for the engine.
EngineName	<i>Type:</i> text (max 128 characters) Name of engine.
EngineDescription	<i>Type:</i> text. Nullable Description of engine.
Comments	<i>Type:</i> text. Nullable Comments from factory.
ApplicationID	<i>Type:</i> integer. Nullable SAP internal application ID
ConsumptionUnit	<i>Type:</i> text. Nullable Unit description to describe the consumption amount.
CreationDate	<i>Type:</i> datetime The data and time the engine was created.
UpdatedDate	<i>Type:</i> datetime The date and time the engine was last updated.

## SAPContentEngineRule Table

This table stores an engine rule from downloadable content.

**Table 486: Database columns for SAPContentEngineRule table**

Database Column	Details
SAPContentEngineRuleID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP content engine rule table.

Database Column	Details
EngineContentUID	Type: text (max 128 characters) A global unique identifier for the engine.
RuleContentUID	Type: text (max 128 characters). Key A global unique identifier for the engine rule.
RuleName	Type: text (max 128 characters) Name of engine rule.
RuleDefinition	Type: text. Nullable Rule definition for calculating consumption of an engine.
IsDefault	Type: boolean Is this formula the default for created packages.
CreationDate	Type: datetime The data and time the engine rule was created.
UpdatedDate	Type: datetime The date and time the engine rule was last updated.

## SAPDuplicateUserRecommendation Table

This table stores the data specific to an SAP duplicate user recommendation.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 487: Database columns for SAPDuplicateUserRecommendation table**

Database Column	Details
SAPDuplicateUserRecommendationID	Type: integer. Key. Generated ID A unique identifier for the SAP duplicate user recommendation.
RecommendationUID	Type: text (max 128 characters). Key A globally unique identifier for the SAP license recommendation.

Database Column	Details
SAPRecommendationSetID	<i>Type:</i> integer. Key Foreign key to the SAP recommendation set that the duplicate user recommendation belongs to.
DuplicateGroupNum	<i>Type:</i> integer The unique identifier showing which users are duplicates of one another.
SAPUserID	<i>Type:</i> integer. Key. Nullable The unique identifier of the user that the duplicate user recommendation belongs to.
UserName	<i>Type:</i> text The user name of the user that the duplicate user recommendation belongs to.
SystemID	<i>Type:</i> text The ID of the system that the duplicate user recommendation belongs to.
ClientID	<i>Type:</i> text The ID of the client that the duplicate user recommendation belongs to.
IsConsolidatedBySAP	<i>Type:</i> boolean Whether or not this duplicate is consolidated by SAP.
SAPRuleID	<i>Type:</i> integer. Nullable The unique identifier of the rule used to produce the duplicate user recommendation.
RuleSetName	<i>Type:</i> text. Nullable The name of the rule set used to produce the duplicate user recommendation.
RuleName	<i>Type:</i> text. Nullable The name of the rule used to produce the duplicate user recommendation.
RuleSequenceNumber	<i>Type:</i> integer. Nullable The sequence number of the rule used to produce the duplicate user recommendation.
RuleMessage	<i>Type:</i> text. Nullable The message produced given by the rule used to produce the duplicate user recommendation.

Database Column	Details
SAPRecommendationProcessedStatusID	<i>Type:</i> integer Foreign key to the SAP recommendation processed status of the duplicate user recommendation.
RuleMessageResourceName	<i>Type:</i> text (max 256 characters). Nullable The resource name of the message produced given by the rule used to produce the duplicate user recommendation.
RuleMessageParameters	<i>Type:</i> text (max 256 characters). Nullable The parameters used by the message produced given by the rule used to produce the duplicate user recommendation.

## SAPEngine Table

This table stores the application engines used in SAP.

**Table 488: Database columns for SAPEngine table**

Database Column	Details
SAPEngineID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP engine table.
ApplicationID	<i>Type:</i> integer. Key The unique identifier given to the application engine by SAP.
SAPEnginePeriodTypeID	<i>Type:</i> integer A unique identifier for the SAP engine period type.

## SAPEngineConsumptionSummary Table

This table stores the total consumption of SAP package consumption recommendation.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 489: Database columns for SAPEngineConsumptionSummary table**

Database Column	Details
SAPEngineConsumptionSummaryID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP engine consumption.
LandscapeUID	<i>Type:</i> text (max 128 characters) A global unique identifier for the system landscape the summary belongs to.
RecommendationSetUID	<i>Type:</i> text (max 128 characters) A global unique identifier for the SAP recommendation set the summary belongs to.
SAPRecommendationSetStatusID	<i>Type:</i> integer The status of the recommendation set.
SAPSystemLandscapeEngineID	<i>Type:</i> integer. Key. Nullable A unique identifier for the SAP system landscape engine table.
EngineUID	<i>Type:</i> text (max 128 characters) A global unique identifier for the SAP engine in a system landscape.
EngineName	<i>Type:</i> text (max 128 characters) Name of engine.
Consumed	<i>Type:</i> decimal. Nullable The number of consumed units for the package (null = indeterminate)
ConsumptionUnit	<i>Type:</i> text. Nullable Unit description to describe the consumption amount.
ReasonMessage	<i>Type:</i> text. Nullable And optional message detailing the reason for the consumed result.
EntitlementsPurchased	<i>Type:</i> integer Total number of purchased license entitlements.
EngineUnitPrice	<i>Type:</i> currency. Nullable The unit price of a license entitlement.
EngineUnitPriceRateID	<i>Type:</i> integer. Nullable



Database Column	Details
	The unit price rate of a license entitlement.
CalculationDate	<i>Type:</i> datetime The date of the license postion calculation.
SystemMeasurementDate	<i>Type:</i> datetime The date the system measurement calculation was performed.

## SAPEngineMetric Table

This table stores the application engine metrics used in SAP.

**Table 490: Database columns for SAPEngineMetric table**

Database Column	Details
SAPEngineMetricID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP engine metric table.
MetricID	<i>Type:</i> integer. Key The unique identifier given to the application engine metric by SAP.

## SAPEngineMetricName Table

This table stores the name of applications engine metrics in different languages.

**Table 491: Database columns for SAPEngineMetricName table**

Database Column	Details
SAPEngineMetricNameID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP engine metric name table.
SAPEngineMetricID	<i>Type:</i> integer. Key The unique identifier of an SAP engine metric.
EngineMetricName	<i>Type:</i> text (max 128 characters). Key. Nullable The name of the application engine metric.
Language	<i>Type:</i> text (max 4 characters). Key

Database Column	Details
	A unique code to identify the language.

## SAPEngineName Table

This table stores the name of applications engines in different languages.

**Table 492: Database columns for SAPEngineName table**

Database Column	Details
SAPEngineNameID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP engine name table.
SAPEngineID	<i>Type:</i> integer. Key The unique identifier of an SAP engine.
EngineName	<i>Type:</i> text (max 128 characters). Key. Nullable The name of the application engine.
Language	<i>Type:</i> text (max 4 characters). Key A unique code to identify the language.

## SAPEnginePeriodType Table

This table stores the types of SAP applications engine measurement periods.

**Table 493: Database columns for SAPEnginePeriodType table**

Database Column	Details
SAPEnginePeriodTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP engine period type table.
PeriodTypeCode	<i>Type:</i> text (max 1 characters). Key A unique lookup for each SAPEnginePeriodType. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>Y = Last year</li> <li>C = Last calendar year</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• T = Year to date</li> <li>• M = This month</li> <li>• Q = This quarter</li> <li>• 6 = Last six months</li> <li>• U = Undefined</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters)</p> <p>A localizable resource string representing a SAP system type. Foreign key to the <code>ComplianceResourceString</code> table.</p>
DefaultValue	<p><i>Type:</i> text (max 100 characters)</p> <p>The text to display if the system type resource string has no translation.</p>

## SAPEngineSystemConsumption Table

This table stores the per-system consumption of SAP package consumption recommendation.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 494: Database columns for SAPEngineSystemConsumption table**

Database Column	Details
SAPEngineSystemConsumptionID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for the SAP engine consumption.</p>
SAPRecommendationSetID	<p><i>Type:</i> integer. Key</p> <p>Foreign key to the SAP recommendation set that the license recommendation belongs to.</p>
SAPSystemLandscapeEngineID	<p><i>Type:</i> integer. Key. Nullable</p> <p>A unique identifier for the SAP system landscape engine table.</p>
EngineUID	<p><i>Type:</i> text (max 128 characters)</p> <p>A global unique identifier for the SAP engine in a system landscape.</p>

Database Column	Details
EngineName	Type: text (max 128 characters) Name of engine.
SAPSystemID	Type: integer. Key. Nullable The unique identifier of the system that the consumed count belongs to.
SystemID	Type: text The ID of the system that the license recommendation belongs to.
ClientID	Type: text The ID of the client that the license recommendation belongs to.
Consumed	Type: decimal. Nullable The number of consumed units for the package (null = indeterminate)
ReasonMessage	Type: text. Nullable And optional message detailing the reason for the consumed result.
SystemMeasurementDate	Type: datetime. Nullable The date the system measurement calculation was performed.

## SAPImportedInventoryFileDigest Table

This table stores digests of imported SAP inventory files.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 495: Database columns for SAPImportedInventoryFileDigest table**

Database Column	Details
SAPImportedInventoryFileDigestID	Type: integer. Key. Generated ID A unique identifier for the SAP imported inventory file digest.
LandscapeUID	Type: text (max 128 characters). Key A global unique identifier for the system landscape.

Database Column	Details
SystemID	Type: text (max 64 characters). Key The System ID that is used to identify the SAP system.
ClientID	Type: text (max 32 characters). Key The Client ID that is to be used when connecting to the SAP system.
SystemNumber	Type: text (max 32 characters). Key. Nullable The SAP system number. This value will be used by the RFC connection.
MD5Hash	Type: text (max 64 characters). Key MD5 hash of imported SAP inventory file content.
CreationDate	Type: datetime The data and time the digest record was created.

## SAPLicenseRatio Table

This table stores SAP license ratios used for recommending optimizations for SAP.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 496: Database columns for SAPLicenseRatio table**

Database Column	Details
SAPLicenseRatioID	Type: integer. Key. Generated ID A unique identifier for the SAP license ratio.
LeftLicenseType	Type: text (max 2 characters) The type of license assigned to the left side of the license ratio.
LeftValue	Type: integer The value belonging to the left side of the license ratio.
RightLicenseType	Type: text (max 2 characters) The type of license assigned to the right side of the license ratio.

Database Column	Details
RightValue	<i>Type:</i> integer The value belonging to the right side of the license ratio.
SAPSystemLandscapeID	<i>Type:</i> integer. Key Foreign key to the system landscape that the license ratio belongs to.
IsActive	<i>Type:</i> boolean Whether or not this license ratio is used to automatically optimize SAP license assignments.
CreationUser	<i>Type:</i> text (max 256 characters) The user who created the license ratio.
CreationDate	<i>Type:</i> datetime The data and time the license ratio was created.
UpdatedUser	<i>Type:</i> text (max 256 characters) The last user who update the license ratio.
UpdatedDate	<i>Type:</i> datetime The date and time the license ratio was last updated.

## SAPLicenseRecommendation Table

This table stores the data specific to an SAP license recommendation.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 497: Database columns for SAPLicenseRecommendation table**

Database Column	Details
SAPLicenseRecommendationID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP license recommendation.
RecommendationUID	<i>Type:</i> text (max 128 characters). Key

Database Column	Details
	A globally unique identifier for the SAP license recommendation.
SAPRecommendationSetID	<i>Type:</i> integer. Key Foreign key to the SAP recommendation set that the license recommendation belongs to.
SAPUserID	<i>Type:</i> integer. Key. Nullable The unique identifier of the user that the license recommendation belongs to.
UserName	<i>Type:</i> text The user name of the user that the license recommendation belongs to.
SystemID	<i>Type:</i> text The ID of the system that the license recommendation belongs to.
ClientID	<i>Type:</i> text The ID of the client that the license recommendation belongs to.
OriginalLicenseType	<i>Type:</i> text (max 2 characters). Nullable The license code originally assigned to the user.
RecommendedLicenseType	<i>Type:</i> text (max 2 characters). Nullable The license code recommended the user be assigned.
SAPRuleID	<i>Type:</i> integer. Nullable The unique identifier of the rule used to produce the license recommendation.
RuleSetName	<i>Type:</i> text. Nullable The name of the rule set used to produce the license recommendation.
RuleName	<i>Type:</i> text. Nullable The name of the rule used to produce the license recommendation.
RuleSequenceNumber	<i>Type:</i> integer. Nullable The sequence number of the rule used to produce the license recommendation.
RuleMessage	<i>Type:</i> text. Nullable The message produced given by the rule used to produce the license recommendation.

Database Column	Details
SAPRecommendationProcessedStatusID	<i>Type:</i> integer Foreign key to the SAP recommendation processed status of the license recommendation.
OptimalLicenseType	<i>Type:</i> text (max 2 characters). Nullable The license code recommended the user be assigned ignoring license ratios and rebalancing.
SAPRecommendationAdjustmentReasonID	<i>Type:</i> integer. Nullable The unique identifier of the reason why the license recommendation differs from optimal.
RuleMessageResourceName	<i>Type:</i> text (max 256 characters). Nullable The resource name of the message produced given by the rule used to produce the license recommendation.
RuleMessageParameters	<i>Type:</i> text (max 256 characters). Nullable The parameters used by the message produced given by the rule used to produce the license recommendation.

## SAPLicenseType Table

This table stores the SAP license type on SAP systems.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 498: Database columns for SAPLicenseType table**

Database Column	Details
SAPLicenseTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP license type.
SAPSystemID	<i>Type:</i> integer. Key Foreign key to the system that the SAP license type belongs to.
Identifier	<i>Type:</i> text (max 2 characters). Key



Database Column	Details
	SAP license type identifier
SAPSpecialVersionID	Type: integer. Key. Nullable SAP special version ID
Active	Type: boolean Indicates whether the SAP license type is active or not active.
SpecialVersionAssignment	Type: boolean. Nullable Indicates whether the SAP license type is affected by special version.
SSCR_Allow	Type: boolean. Nullable
IsDeleted	Type: boolean Indicated whether the SAP license type has been deleted or not.

## SAPLicenseTypeHierarchy Table

This table stores SAP license type hierarchy.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 499: Database columns for SAPLicenseTypeHierarchy table**

Database Column	Details
SAPLicenseTypeHierarchyID	Type: integer. Key. Generated ID The unique identifier for the SAP license type hierarchy.
SAPLicenseTypeID	Type: integer. Key Parent license type. Foreign key to the SAP license type.
ChildSAPLicenseTypeID	Type: integer. Key Child license type. Foreign key to SAP license type.

## SAPLicenseTypeName Table

This table stores SAP license types in various languages.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 500: Database columns for SAPLicenseTypeName table**

Database Column	Details
SAPLicenseTypeNameID	Type: integer. Key. Generated ID Unique identifier for SAP license type name.
SAPLicenseTypeID	Type: integer. Key Foreign key to the SAP license type.
Language	Type: text (max 4 characters) The two letter language code.
ShortName	Type: text (max 128 characters). Nullable SAP license type short name.
LongName	Type: text (max 256 characters). Nullable SAP license type long name

## SAPModule Table

This table stores the modules used in SAP.

**Table 501: Database columns for SAPModule table**

Database Column	Details
SAPModuleID	Type: integer. Key. Generated ID A unique identifier for the SAP module table.
ModuleName	Type: text (max 64 characters). Key The name of the module.

Database Column	Details
SubModuleName	Type: text (max 64 characters). Key. Nullable The name of the sub module.
ObjectName	Type: text (max 40 characters). Key. Nullable The name of the object linked to the SAP system module.

## SAPMultipleLogon Table

This table stores logon metrics for SAP users.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 502: Database columns for SAPMultipleLogon table**

Database Column	Details
SAPMultipleLogonID	Type: integer. Key. Generated ID A unique identifier for the user's logon metrics
SAPUserID	Type: integer. Key Foreign key to the SAP user.
Year	Type: text (max 4 characters) The year of the logon metrics
NumberOfMultipleLogon	Type: integer. Nullable Number of multiple concurrent logon
MaxMultipleLogon	Type: integer. Nullable Maximum number of concurrent logon

## SAPObject Table

This table stores the SAP object

**Table 503: Database columns for SAPObject table**

Database Column	Details
SAPObjectID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP object.
ObjectName	<i>Type:</i> text (max 40 characters). Key Name of the object
SAPObjectTypeID	<i>Type:</i> integer. Key Foreign key to the SAP object type that identifies the object type.

## SAPObjectType Table

This table stores SAP object types

**Table 504: Database columns for SAPObjectType table**

Database Column	Details
SAPObjectTypeID	<i>Type:</i> integer. Key. Generated ID <ul style="list-style-type: none"> <li>• 1 = Transaction</li> <li>• 2 = Report</li> <li>• 3 = Job</li> <li>• 4 = NonSAP</li> </ul>
TypeName	<i>Type:</i> text (max 64 characters). Key A unique name for the SAP object type.
ResourceName	<i>Type:</i> text (max 256 characters). Nullable A localizable resource string representing a SAP object type. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the object type resource string has no translation.

## SAPRFCConnection Table

This table stores RFC connections made to the SAP system.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 505: Database columns for SAPRFCConnection table**

Database Column	Details
SAPRFCConnectionID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP RFC consumption.
SAPUserID	<i>Type:</i> integer. Key SAP user performing the RFC connection. Foreign key to the <i>SAPUser</i> table
TimePeriodStartDate	<i>Type:</i> datetime. Key The date and time of the RFC connection
RemoteSystem	<i>Type:</i> text (max 128 characters). Key Remote system name connecting to the SAP system.
ProgramName	<i>Type:</i> text (max 40 characters). Key Program Name associated to the function name.
FunctionName	<i>Type:</i> text (max 40 characters). Key The function executed by the RFC calls
TaskType	<i>Type:</i> text (max 2 characters). Key. Nullable Task type.
RFCDestination	<i>Type:</i> text (max 128 characters). Key The RFC destination string value.
TotalExecutionCount	<i>Type:</i> integer The number of times the function is executed.
TotalExecutionTime	<i>Type:</i> decimal Total execution time.
TotalCallTime	<i>Type:</i> decimal Total call time.

Database Column	Details
TotalDataSent	Type: big integer Total data sent by the RFC calls.
TotalDataReceived	Type: big integer Total data received b the RFC calls.

## SAPRFCConnectionSummary Table

This table stores the remote RFC consumption summary. It only includes Non-SAP remote system



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 506: Database columns for SAPRFCConnectionSummary table**

Database Column	Details
SAPRFCConnectionSummaryID	Type: integer. Key. Generated ID A unique identifier for the remote SAP RFC connection summary.
RemoteSystem	Type: text (max 128 characters). Key Remote system making the RFC calls.
NumberOfSAPSystems	Type: integer Number of SAP systems the Remote System is connecting to.
NumberOfDialogUsers	Type: integer Number of Dialog SAP users making the RFC call to the SAP system.
NumberOfNonDialogUsers	Type: integer Number of Service SAP users making the RFC call to the SAP system.
NumberOfExecutedPrograms	Type: integer The number of executed programs
NumberOfExecutedFunctions	Type: integer The number of executed functions

Database Column	Details
TotalExecutionCount	<i>Type:</i> integer The total excution count of all functions.
TotalExecutionTime	<i>Type:</i> decimal Total execution time.
TotalCallTime	<i>Type:</i> decimal Total call time.
TotalDataSent	<i>Type:</i> big integer Total data sent by the RFC calls.
TotalDataReceived	<i>Type:</i> big integer Total data received b the RFC calls.
IsHidden	<i>Type:</i> boolean Is this record marked as hidden in the UI.

## SAPRecommendationAdjustmentReason Table

This table stores SAP Recommendation adjustment reasons.

**Table 507: Database columns for SAPRecommendationAdjustmentReason table**

Database Column	Details
SAPRecommendationAdjustmentReasonID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each SAPRecommendationAdjustmentReason. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = License ratio enforced</li> <li>• 2 = Excess purchase(s) of covering license type applied.</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key A localizable resource string representing a SAP recommendation adjustment reason. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the adjustment reason resource string has no translation.

## SAPRecommendationProcessedStatus Table

This table stores SAP Recommendation Processed status.

**Table 508: Database columns for SAPRecommendationProcessedStatus table**

Database Column	Details
SAPRecommendationProcessedStatusID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for each SAPRecommendationProcessedStatus.</p> <p>Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• 1 = Pending</li> <li>• 2 = Accepted</li> <li>• 3 = Rejected</li> </ul>
ResourceName	<p>Type: text (max 256 characters). Key</p> <p>A localizable resource string representing a SAP recommendation processed status. Foreign key to the ComplianceResourceString table.</p>
DefaultValue	<p>Type: text (max 100 characters)</p> <p>The text to display if the status resource string has no translation.</p>

## SAPRecommendationSet Table

This table stores data specific to the definition of a recommendation set.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 509: Database columns for SAPRecommendationSet table**

Database Column	Details
SAPRecommendationSetID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for the SAP recommendation set.</p>
RecommendationSetUID	<p>Type: text (max 128 characters). Key</p> <p>A global unique identifier for the SAP recommendation set.</p>



Database Column	Details
RecommendationSetName	<i>Type:</i> text (max 128 characters) Name of recommendation set.
RecommendationSetDescription	<i>Type:</i> text. Nullable Description of recommendation set.
LandscapeUID	<i>Type:</i> text (max 128 characters) A global unique identifier for the system landscape the recommendation set belongs to.
SAPRecommendationSetStatusID	<i>Type:</i> integer. Key The status of the recommendation set.
CalculationDate	<i>Type:</i> datetime. Nullable The date of the license position calculation.
CreationUser	<i>Type:</i> text (max 256 characters) The user who created the recommendation set.
CreationDate	<i>Type:</i> datetime The data and time the recommendation set was created.
UpdatedUser	<i>Type:</i> text (max 256 characters) The last user who update the recommendation set.
UpdatedDate	<i>Type:</i> datetime The date and time the recommendation set was last updated.
ReviewedUser	<i>Type:</i> text (max 256 characters). Nullable The user who reviewed the recommendation set.
ReviewedDate	<i>Type:</i> datetime. Nullable The date and time the recommendation set was reviewed.
ReleasedUser	<i>Type:</i> text (max 256 characters). Nullable The user who released the recommendation set.
ReleasedDate	<i>Type:</i> datetime. Nullable The date and time the recommendation set was released.
Uploaded	<i>Type:</i> boolean

Database Column	Details
	Indicates whether the recommendation set was oplayed by FNM-SAP
UploadedDate	<i>Type:</i> datetime. Nullable The date the recommendation set was oplayed by FNM-SAP

## SAPRecommendationSetStatus Table

This table stores SAP Recommendation Set status.

**Table 510: Database columns for SAPRecommendationSetStatus table**

Database Column	Details
SAPRecommendationSet StatusID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each SAPRecommendationSetStatus. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = In Review</li> <li>• 2 = Archived</li> <li>• 3 = Released</li> <li>• 4 = New License Position</li> <li>• 5 = Rejected</li> <li>• 6 = Simulation.</li> <li>• 7 = Creating</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key A localizable resource string representing a SAP recommendation set status. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the status resource string has no translation.

## SAPRecommendationSetSummary Table

This table stores a history of SAP license positions.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 511: Database columns for SAPRecommendationSetSummary table**

Database Column	Details
SAPRecommendationSetSummaryID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP license recommendation summary.
LandscapeUID	<i>Type:</i> text (max 128 characters). Key A global unique identifier for the system landscape the summary belongs to.
RecommendationSetUID	<i>Type:</i> text (max 128 characters). Key A global unique identifier for the SAP recommendation set the summary belongs to.
SAPRecommendationSetStatusID	<i>Type:</i> integer The status of the recommendation set.
LicenseType	<i>Type:</i> text (max 2 characters). Key. Nullable The license code to which the position applies.
EntitlementsPurchased	<i>Type:</i> integer Total number of purchased license entitlements.
EntitlementsOriginal	<i>Type:</i> integer Total number of consumed license entitlements.
EntitlementsRecommended	<i>Type:</i> integer Total number of recommended license entitlements.
LicenseTypeUnitPrice	<i>Type:</i> currency. Nullable The unit price of a license entitlement.
LicenseTypeUnitPriceRateID	<i>Type:</i> integer. Nullable The unit price rate of a license entitlement.
CalculationDate	<i>Type:</i> datetime The date of the license position calculation.
EntitlementsOptimal	<i>Type:</i> integer

Database Column	Details
	Total number of recommended license entitlements without license ratio constraints.

## SAPRole Table

This table stores SAP roles



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 512: Database columns for SAPRole table**

Database Column	Details
SAPRoleID	Type: integer. Key. Generated ID A unique identifier for the SAP role.
SAPSystemID	Type: integer. Key Foreign key to the system that the role belongs to.
RoleName	Type: text (max 30 characters) The name of the role.
NumberOfTransactionCodes	Type: integer. Nullable Total number of transaction codes allowed to be executed by this role.
LicenseType	Type: text (max 2 characters). Nullable License type associated to this role

## SAPRoleConsumption Table

This table stores SAP roles and its link to SAP consumption.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 513: Database columns for SAPRoleConsumption table**

Database Column	Details
SAPRoleConsumptionID	Type: integer. Key. Generated ID A unique identifier for for SAP role consumption.
SAPUserID	Type: integer. Key Foreign key to the SAP user that the role consumption belongs to.
CompositeRoleID	Type: integer. Key. Nullable Foreign key to SAP role.
SingleRoleID	Type: integer. Key Foreign key to SAP role.
SingleRoleTransactionCodeID	Type: integer. Key Foreign key to SAP transaction code.
SAPConsumptionID	Type: integer. Key Foreign key to SAP consumption.

## SAPRoleTransactionCode Table

This table stores list of roles and its transaction codes.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 514: Database columns for SAPRoleTransactionCode table**

Database Column	Details
SAPRoleTransactionCodeID	Type: integer. Key. Generated ID A unique identifier for the roles and its transaction codes.

Database Column	Details
SAPRoleID	Type: integer. Key Foreign to the SAP Roles where transaction codes belong to.
TCodeLow	Type: text (max 80 characters). Key. Nullable Lower range of the transaction code.
TCodeHigh	Type: text (max 40 characters). Nullable Upper range of the transaction code.

## SAPRule Table

This table stores SAP rules used for recommending optimizations for SAP.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 515: Database columns for SAPRule table**

Database Column	Details
SAPRuleID	Type: integer. Key. Generated ID A unique identifier for the SAP rule.
RuleName	Type: text (max 128 characters) Name of the rule.
SAPRuleTypeID	Type: integer. Key Foreign key to the rule type of the SAP rule.
SAPRuleSetID	Type: integer. Key Foreign key to the rule set that the SAP rule belongs to.
RuleDefinition	Type: text The rule definition XML used to build the rule statement used by the SAP rules engine.
SequenceNumber	Type: integer

Database Column	Details
	The sequence number used to designate the order of the rules within the rule set.
SAPRuleCategoryID	<i>Type:</i> integer. Key Foreign key to the rule category of the SAP rule.
IsActive	<i>Type:</i> boolean Whether or not this rule is active for execution.
UseRuleSetMapping	<i>Type:</i> boolean Whether or not to use mapping from the SAP rule set
CreationUser	<i>Type:</i> text (max 256 characters) The user who created the system landscape.
CreationDate	<i>Type:</i> datetime The data and time the system landscape was created.
UpdatedUser	<i>Type:</i> text (max 256 characters) The last user who update the system landscape.
UpdatedDate	<i>Type:</i> datetime The date and time the system landscape was last updated.

## SAPRuleAlgorithm Table

This table stores the available SAP rule algorithms used by SAP rules.

**Table 516: Database columns for SAPRuleAlgorithm table**

Database Column	Details
SAPRuleAlgorithmID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP rule algorithm.
AlgorithmName	<i>Type:</i> text (max 100 characters). Key A unique name for the SAP category.
SAPRuleCategoryID	<i>Type:</i> integer. Key Foreign key to the rule category of the SAP rule algorithm.

Database Column	Details
TitleResourceName	<i>Type:</i> text (max 256 characters). Nullable A localizable resource string representing a SAP rule algorithm. Foreign key to the <code>ComplianceResourceString</code> table.
TitleDefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the rule type resource string has no translation.
AlgorithmType	<i>Type:</i> text Type associated with this algorithm
AlgorithmData	<i>Type:</i> text. Nullable Data associated with this algorithm, such as a custom SQL query to run.

## SAPRuleCategory Table

This table stores the different rule categories used in recommending optimizations for SAP.

**Table 517: Database columns for SAPRuleCategory table**

Database Column	Details
SAPRuleCategoryID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP rule category.
CategoryName	<i>Type:</i> text (max 100 characters). Key A unique name for the SAP category.

## SAPRuleMapping Table

This table stores mapping between SAP rule to either System Landscapes, System Groups or SAP systems.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.



**Table 518: Database columns for SAPRuleMapping table**

Database Column	Details
SAPRuleMappingID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP rule set mapping.
SAPRuleID	<i>Type:</i> integer. Key Foreign key to SAP Rule ID
SAPSystemLandscapeID	<i>Type:</i> integer. Key. Nullable Foreign key to System Landscape ID
SAPSystemGroupID	<i>Type:</i> integer. Key. Nullable Foreign key to System Group ID.
SAPSystemID	<i>Type:</i> integer. Key. Nullable Foreign key to the SAP system.
CreationUser	<i>Type:</i> text (max 256 characters) The user who created the system landscape.
CreationDate	<i>Type:</i> datetime The data and time the system landscape was created.
UpdatedUser	<i>Type:</i> text (max 256 characters) The last user who update the system landscape.
UpdatedDate	<i>Type:</i> datetime The date and time the system landscape was last updated.

## SAPRuleSet Table

This table stores SAP rule sets used for recommending optimizations for SAP.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 519: Database columns for SAPRuleSet table**

Database Column	Details
SAPRuleSetID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP rule set.
RuleSetName	<i>Type:</i> text (max 128 characters) Name of rule set.
RuleSetDescription	<i>Type:</i> text. Nullable Description of rule set.
SAPSystemLandscapeID	<i>Type:</i> integer. Key Foreign key to the system landscape that the SAP rule set belongs to.
IsActive	<i>Type:</i> boolean Whether or not this rule set is used to automatically optimize SAP license assignments.
SequenceNumber	<i>Type:</i> integer The sequence number used to designate the order of the rule sets within the landscape.
SAPRuleCategoryID	<i>Type:</i> integer. Key Foreign key to the rule category of the SAP rule set.
NumberOfConsumptionMonth	<i>Type:</i> integer
ConsumptionMonthEndDate	<i>Type:</i> datetime. Nullable End date of consumption period used for recommending optimizations. If null,
CreationUser	<i>Type:</i> text (max 256 characters) The user who created the system landscape.
CreationDate	<i>Type:</i> datetime The data and time the system landscape was created.
UpdatedUser	<i>Type:</i> text (max 256 characters) The last user who update the system landscape.
UpdatedDate	<i>Type:</i> datetime The date and time the system landscape was last updated.

Database Column	Details
SecurityTypeID	Type: integer Security type for this object. Foreign key to the SecurityType table.

## SAPRuleSetMapping Table

This table stores mapping between SAP rule sets to either System Landscapes, System Groups or SAP systems.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 520: Database columns for SAPRuleSetMapping table**

Database Column	Details
SAPRuleSetMappingID	Type: integer. Key. Generated ID A unique identifier for the SAP rule set mapping.
SAPRuleSetID	Type: integer. Key Foreign key to SAP Rule Set ID
SAPSystemLandscapeID	Type: integer. Key. Nullable Foreign key to System Landscape ID
SAPSystemGroupID	Type: integer. Key. Nullable Foreign key to System Group ID.
SAPSystemID	Type: integer. Key. Nullable Foreign key to the SAP system.
CreationUser	Type: text (max 256 characters) The user who created the system landscape.
CreationDate	Type: datetime The data and time the system landscape was created.
UpdatedUser	Type: text (max 256 characters) The last user who update the system landscape.

Database Column	Details
UpdatedDate	<i>Type:</i> datetime The date and time the system landscape was last updated.

## SAPRuleType Table

This table stores the available SAP rule types used for recommending optimizations for SAP.

**Table 521: Database columns for SAPRuleType table**

Database Column	Details
SAPRuleTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP rule type.
TypeName	<i>Type:</i> text (max 100 characters). Key A unique name for the SAP rule type.
SAPRuleCategoryID	<i>Type:</i> integer. Key Foreign key to the rule category of the SAP rule.
TitleResourceName	<i>Type:</i> text (max 256 characters). Nullable A localizable resource string representing a SAP rule type. Foreign key to the ComplianceResourceString table.
TitleDefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the rule type resource string has no translation.
RuleTemplate	<i>Type:</i> text The template used to build a rule for the SAP rules engine.
DefaultRuleDefinition	<i>Type:</i> text. Nullable Default rule definition for newly created SAP rule

## SAPSecurityUser Table

This table stores the operators allowed to access SAP objects.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 522: Database columns for SAPSecurityUser table**

Database Column	Details
SAPSecurityUserID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP security user table.
TargetTypeID	<i>Type:</i> integer. Key Target type of object with restricted access.
SAPSystemLandscapeID	<i>Type:</i> integer. Key. Nullable The unique identifier of a SAP system landscape.
SAPSystemID	<i>Type:</i> integer. Key. Nullable The unique identifier of a SAP system.
SAPRuleSetID	<i>Type:</i> integer. Key. Nullable The unique identifier of a SAP rule set.
ResourceID	<i>Type:</i> integer The unique identifier of a security resource.
ActionClassID	<i>Type:</i> integer The unique identifier of a security action class.
ComplianceOperatorID	<i>Type:</i> integer. Key The unique identifier of an operator.

## SAPSystem Table

This table stores the data specific to the definition of SAP systems.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 523: Database columns for SAPSystem table**

Database Column	Details
SAPSystemID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP system.
SystemName	<i>Type:</i> text (max 128 characters) The name of the SAP system.
SystemDescription	<i>Type:</i> text. Nullable A more detailed description of the SAP system.
SAPSystemLandscapeID	<i>Type:</i> integer. Key Foreign key to the system landscape that the SAP system belongs to.
SAPSystemGroupID	<i>Type:</i> integer. Key. Nullable Foreign key to the system group that the SAP system belongs to.
SAPSystemEnvironmentID	<i>Type:</i> integer. Key. Nullable The type of environment for the SAP system.
SystemID	<i>Type:</i> text (max 64 characters) The System ID that is used to identify the SAP system.
ClientID	<i>Type:</i> text (max 32 characters) The Client ID that is to be used when connecting to the SAP system.
ServerName	<i>Type:</i> text (max 256 characters). Nullable The DNS name of the SAP system. This value will be used by the RFC connection. This field can also store the SAP System's IP address.
SystemNumber	<i>Type:</i> text (max 32 characters). Nullable The SAP system number. This value will be used by the RFC connection.
Username	<i>Type:</i> text (max 256 characters). Nullable The user name that is to be used when connecting to the SAP system.
Password	<i>Type:</i> binary. Nullable The password that is to be used when connecting to the SAP system.
IsOfflineSystem	<i>Type:</i> boolean

Database Column	Details
	Indicates whether an SAP system is offline.
IsPortalSystem	<i>Type:</i> boolean Indicates whether the system is a portal system.
SystemStatus	<i>Type:</i> text (max 128 characters). Nullable The status of the SAP system.
UsersControlledByCUA	<i>Type:</i> boolean Identifies whether the uses on the SAP system are controlled by a CUA.
ModelView	<i>Type:</i> text (max 128 characters). Nullable Further clarification required.
CUACentralSystem	<i>Type:</i> boolean The status of the SAP system.
CUACentralSystemID	<i>Type:</i> text (max 128 characters). Nullable The System ID of the CUA system that this SAP system is controlled by.
FNMSAPRelease	<i>Type:</i> text (max 128 characters). Nullable The version of FNM for SAP installed on the SAP system.
LAWVersion	<i>Type:</i> text (max 128 characters). Nullable The version of the License Assignment Workbench module installed on the SAP system.
SAPRelease	<i>Type:</i> text (max 128 characters). Nullable The version of SAP installed on the SAP system.
SAPPatchRelease	<i>Type:</i> text (max 128 characters). Nullable The SAP patch version
STPIRelease	<i>Type:</i> text (max 128 characters). Nullable The ST-PI version
DBSystem	<i>Type:</i> text (max 128 characters). Nullable The database system running on the SAP system.
HardwareKey	<i>Type:</i> text (max 128 characters). Nullable The hardware key of the SAP system.

Database Column	Details
InstallationNumber	<i>Type:</i> text (max 128 characters). Nullable The SAP system installation number
LastChangedOn	<i>Type:</i> datetime. Nullable The date and time the SAP system data was last refreshed.
SupportPackage	<i>Type:</i> text (max 128 characters). Nullable The support package of the SAP system.
HRSystem	<i>Type:</i> text (max 128 characters). Nullable The SAP system which contains the HR data.
SystemType	<i>Type:</i> text (max 128 characters). Nullable Indicates whether the SAP system is an ABAP or JAVA based system.
DefaultLicenseType	<i>Type:</i> text (max 2 characters). Nullable Default license type for the SAP system.
ContactFirstName	<i>Type:</i> text (max 128 characters). Nullable First name of the contact for this system.
ContactLastName	<i>Type:</i> text (max 128 characters). Nullable Last name of the contact for this system.
ContactBusinessPhone Number	<i>Type:</i> text (max 30 characters). Nullable Business phone number of the contact for this system.
ContactMobilePhoneNumber	<i>Type:</i> text (max 30 characters). Nullable Mobile phone number of the contact for this system.
ContactEmail	<i>Type:</i> text (max 200 characters). Nullable Email address of the contact for this system.
Location	<i>Type:</i> text (max 128 characters). Nullable Location of this system.
InventoryDate	<i>Type:</i> datetime. Nullable The date and time the SAP system data was collected by SAP Reader.
CreationUser	<i>Type:</i> text (max 256 characters)



Database Column	Details
	The user who created the system landscape.
CreationDate	Type: datetime The data and time the system landscape was created.
UpdatedUser	Type: text (max 256 characters) The last user who update the system landscape.
UpdatedDate	Type: datetime The date and time the system landscape was last updated.
SecurityTypeID	Type: integer. Key Security type for this object. Foreign key to the <code>SecurityType</code> table.
AccessToModuleData	Type: boolean Indicates whether the system has access to module data.
SAPSystemTypeID	Type: integer The type of system for the system. Foreign key to the <code>SAPSystemType</code> table.
SAPConnectivityTypeID	Type: integer. Nullable The type of connectivity for the SAP system. Foreign key to the <code>SAPConnectivityType</code> table.
SAPConnectivityDirectionTypeID	Type: integer. Nullable The type of SAP connectivity direction for the SAP system. Foreign key to the <code>SAPConnectivityDirectionType</code> table.
BeaconUID	Type: unique identifier. Key. Nullable The inventory beacon where this connection is defined.

## SAPSystemActivityCheckSummary Table

This table stores the link between SAP System and SAP Activity Check Summary data.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 524: Database columns for SAPSystemActivityCheckSummary table**

Database Column	Details
SAPSystemActivityCheckSummaryID	<i>Type:</i> integer. Key. Generated ID A unique identifier.
SAPSystemID	<i>Type:</i> integer. Key The Non-SAP system foreign key.
SAPActivityCheckSummaryID	<i>Type:</i> integer. Key The SAP Activity Check Summary data foreign key.

## SAPSystemEngineMetric Table

This table stores the value of applications engine metrics per system.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 525: Database columns for SAPSystemEngineMetric table**

Database Column	Details
SAPSystemEngineMetricID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP system engine metric name table.
SAPEngineID	<i>Type:</i> integer. Key The unique identifier of an SAP engine.
SAPEngineMetricID	<i>Type:</i> integer. Key The unique identifier of an SAP engine metric.
SAPSystemID	<i>Type:</i> integer. Key The unique identifier of a SAP system.
SAPEnginePeriodTypeID	<i>Type:</i> integer. Key A unique identifier for the SAP engine period type.
MetricValue	<i>Type:</i> decimal

Database Column	Details
	The value of the application engine metric.
PeriodStartDate	<i>Type:</i> datetime. Key. Nullable The start date of the SAP application engine metric calculation period.
PeriodEndDate	<i>Type:</i> datetime. Key. Nullable The end date of the SAP application engine metric calculation period.
CalculationDate	<i>Type:</i> datetime. Key. Nullable The date the SAP application engine metric calculation was performed.

## SAPSystemEnvironment Table

This table stores SAP System Environment.

**Table 526: Database columns for SAPSystemEnvironment table**

Database Column	Details
SAPSystemEnvironmentID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP System Environment.
EnvironmentCode	<i>Type:</i> text (max 1 characters). Key A unique lookup for each SAPSystemEnvironment. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• P = Production</li> <li>• T = Test</li> <li>• C = Customizing</li> <li>• D = Demo</li> <li>• E = Training/Education</li> <li>• S = SAP reference</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters) A localizable resource string representing a SAP system environment name. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 100 characters)

Database Column	Details
	The text to display if the SAP system environment resource string has no translation.

## SAPSystemGroup Table

This table stores the data specific to the definition of SAP system groups.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 527: Database columns for SAPSystemGroup table**

Database Column	Details
SAPSystemGroupID	Type: integer. Key. Generated ID A unique identifier for the SAP system group.
GroupName	Type: text (max 128 characters). Key The name of the SAP system group.
GroupDescription	Type: text. Nullable A more detailed description of the SAP system group.
SAPSystemLandscapeID	Type: integer. Key Foreign key to the system landscape that the SAP system group belongs to.
ParentSAPSystemGroupID	Type: integer. Key. Nullable Foreign key to the SAP system group that is its parent. This field will be null if the SAP system group is itself a top level SAP system group.
CreationUser	Type: text (max 256 characters) The user who created the system landscape.
CreationDate	Type: datetime The data and time the system landscape was created.
UpdatedUser	Type: text (max 256 characters) The last user who update the system landscape.

Database Column	Details
UpdatedDate	<i>Type:</i> datetime The date and time the system landscape was last updated.

## SAPSystemLandscape Table

This table stores the data specific to the definition of system landscapes.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 528: Database columns for SAPSystemLandscape table**

Database Column	Details
SAPSystemLandscapeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the system landscape.
LandscapeUID	<i>Type:</i> text (max 128 characters). Key A global unique identifier for the system landscape.
LandscapeName	<i>Type:</i> text (max 128 characters) A unique identifier for the system landscape.
LandscapeDescription	<i>Type:</i> text. Nullable A more detailed description of the SAP system group.
SAPSystemLandscape StatusID	<i>Type:</i> integer Identifies whether this system landscape is actively being used in the license optimization process.
LocationID	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise location associated with this landscape. Foreign key to the GroupEx table.
BusinessUnitID	<i>Type:</i> text (max 128 characters). Key. Nullable Any corporate unit in the enterprise associated with this landscape. Foreign key to the GroupEx table.

Database Column	Details
CostCenterID	<i>Type:</i> text (max 128 characters). Key. Nullable Any cost center in the enterprise associated with this landscape. Foreign key to the <code>GroupEx</code> table.
CategoryID	<i>Type:</i> text (max 128 characters). Key. Nullable Any enterprise category associated with this landscape. Foreign key to the <code>GroupEx</code> table.
CreationUser	<i>Type:</i> text (max 256 characters) The user who created the system landscape.
CreationDate	<i>Type:</i> datetime The data and time the system landscape was created.
UpdatedUser	<i>Type:</i> text (max 256 characters) The last user who update the system landscape.
UpdatedDate	<i>Type:</i> datetime The date and time the system landscape was last updated.
SecurityTypeID	<i>Type:</i> integer Security type for this object. Foreign key to the <code>SecurityType</code> table.
CanRebalanceLicenseTypes	<i>Type:</i> boolean Indicates whether license types can be rebalanced to use excess purchases of higher license types.

## SAPSystemLandscapeEngine Table

This table stores an engine in the system landscape.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 529: Database columns for SAPSystemLandscapeEngine table**

Database Column	Details
SAPSystemLandscapeEngineID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP system landscape engine table.
SAPSystemLandscapeID	<i>Type:</i> integer. Key Foreign key to the system landscape that the SAP engine belongs to.
EngineUID	<i>Type:</i> text (max 128 characters). Key A global unique identifier for the SAP engine in a system landscape.
EngineName	<i>Type:</i> text (max 128 characters) Name of engine.
EngineDescription	<i>Type:</i> text. Nullable Description of engine.
ApplicationID	<i>Type:</i> integer. Nullable SAP internal application ID
IsActive	<i>Type:</i> boolean Whether or not the engine is active for inclusion in license position.
NumberPurchased	<i>Type:</i> integer. Nullable
UnitPrice	<i>Type:</i> currency. Nullable
UnitPriceRateID	<i>Type:</i> integer. Nullable
SAPContentEngineID	<i>Type:</i> integer. Key. Nullable A unique identifier for the SAP content engine table.
SAPContentEngineRuleID	<i>Type:</i> integer. Key. Nullable A unique identifier for the SAP content engine rule table.
CustomRuleDefinition	<i>Type:</i> text. Nullable Custom rule definition for calculating consumption of an engine.
CustomTotalConsumption	<i>Type:</i> integer. Nullable Self-declared total consumption.
UseCustomTotalConsumption	<i>Type:</i> boolean

Database Column	Details
	Use CustomTotalConsumption
ConsumptionUnit	Type: text. Nullable Unit description to describe the consumption amount.
CreationUser	Type: text (max 256 characters) The user who created the system landscape.
CreationDate	Type: datetime The data and time the system landscape was created.
UpdatedUser	Type: text (max 256 characters) The last user who update the system landscape.
UpdatedDate	Type: datetime The date and time the system landscape was last updated.

## SAPSystemLandscapeEngineMapping Table

This table stores mapping between SAP system landscape engines to either System Landscapes, System Groups or SAP systems.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 530: Database columns for SAPSystemLandscapeEngineMapping table**

Database Column	Details
SAPSystemLandscapeEngineMappingID	Type: integer. Key. Generated ID A unique identifier for the SAP system landscape engine mapping.
SAPSystemLandscapeEngineID	Type: integer. Key Foreign key to SAPSystemLandscapeEngine ID
SAPSystemLandscapeID	Type: integer. Key. Nullable Foreign key to System Landscape ID



Database Column	Details
SAPSystemGroupID	Type: integer. Key. Nullable Foreign key to System Group ID.
SAPSystemID	Type: integer. Key. Nullable Foreign key to the SAP system.
CreationUser	Type: text (max 256 characters) The user who created the system landscape.
CreationDate	Type: datetime The data and time the system landscape was created.
UpdatedUser	Type: text (max 256 characters) The last user who update the system landscape.
UpdatedDate	Type: datetime The date and time the system landscape was last updated.

## SAPSystemLandscapeLicenseType Table

This table stores SAP license types belonging to SAP system landscapes.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 531: Database columns for SAPSystemLandscapeLicenseType table**

Database Column	Details
SAPSystemLandscapeLicenseTypeID	Type: integer. Key. Generated ID A unique identifier for SAP system landscape license type.
SAPSystemLandscapeID	Type: integer. Key Foreign key to SAP system landscape.
Identifier	Type: text (max 2 characters). Key The SAP license type identifier.

Database Column	Details
ShortName	<i>Type:</i> text (max 128 characters). Nullable The SAP license type short name.
LongName	<i>Type:</i> text (max 256 characters). Nullable The SAP license type long name.
Active	<i>Type:</i> boolean Indicate whether the SAP license is active or not.
NumberPurchased	<i>Type:</i> integer. Nullable Number purchased.
UnitPrice	<i>Type:</i> currency. Nullable Unit price of a SAP license type.
UnitPriceRateID	<i>Type:</i> integer. Nullable The unit price rate of a SAP license type.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The user who created the SAP license type.
CreationDate	<i>Type:</i> datetime The data and time the SAP license type was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The last user who update the SAP license type.
UpdatedDate	<i>Type:</i> datetime The date and time the SAP license type was last updated.
AllowLicenseBalancing	<i>Type:</i> boolean Indicates whether license types can be rebalanced to use excess purchases of higher license types.

## SAPSystemLandscapeLicenseTypeHierarchy Table

This table stores the SAP license hierarchy for a SAP system landscape.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 532: Database columns for SAPSystemLandscapeLicenseTypeHierarchy table**

Database Column	Details
SAPSystemLandscapeLicenseTypeHierarchyID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the license type hierarchy
SAPSystemLandscapeLicenseTypeID	<i>Type:</i> integer. Key A parent system landscape license type. Foreign key to SAP system landscape license type.
ChildSAPSystemLandscapeLicenseTypeID	<i>Type:</i> integer. Key A child system landscape license type. Foreign key to SAP system landscape license type.

## SAPSystemLandscapeLicenseTypeImport Table

This table stores the imported SAP license type.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 533: Database columns for SAPSystemLandscapeLicenseTypeImport table**

Database Column	Details
SAPSystemLandscapeLicenseTypeImportID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the imported SAP license type.
SAPSystemLandscapeID	<i>Type:</i> integer. Key Foreign key to SAP system landscape.
SAPSystemID	<i>Type:</i> integer. Key Foreign key to SAP system
SystemName	<i>Type:</i> text (max 128 characters). Nullable

Database Column	Details
	The SAP system name.
ImportUser	<i>Type:</i> text (max 128 characters). Nullable The user who imported the SAP license type
ImportDate	<i>Type:</i> datetime The data and time the SAP license type was imported

## SAPSystemLandscapeStatus Table

This table stores SAP System Landscape status.

**Table 534: Database columns for SAPSystemLandscapeStatus table**

Database Column	Details
SAPSystemLandscape StatusID	<i>Type:</i> integer. Key. Generated ID A unique identifier for each SAPSystemLandscapeStatus. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• 1 = Inactive</li> <li>• 2 = Active</li> <li>• 3 = Archived</li> <li>• 4 = Simulation</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Key A localizable resource string representing a SAP System Landscape status. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the status resource string has no translation.

## SAPSystemModule Table

This table stores the modules used in SAP and the system they are used on.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 535: Database columns for SAPSystemModule table**

Database Column	Details
SAPSystemModuleID	Type: integer. Key. Generated ID A unique identifier for the SAP system module table.
SAPSystemID	Type: integer. Key The unique identifier of a SAP system.
SAPModuleID	Type: integer. Key The unique identifier of a SAP module.

## SAPSystemObject Table

This table stores objects belonging to SAP systems



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 536: Database columns for SAPSystemObject table**

Database Column	Details
SAPSystemObjectID	Type: integer. Generated ID A unique identifier for the SAP system object
SAPSystemID	Type: integer. Key Foreign key to the SAP system that the object belongs to.
SAPObjectID	Type: integer. Key Foreign key to the SAP object.

## SAPSystemPriceList Table

This table stores the SAP system price list.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 537: Database columns for SAPSystemPriceList table**

Database Column	Details
SAPSystemPriceListID	Type: integer. Key. Generated ID A unique identifier for the SAP system price list.
SAPSystemID	Type: integer. Key Foreign key to the system that the price list belongs to.
PriceListID	Type: text (max 2 characters). Key SAP Price List ID
DefaultLicenseType	Type: text (max 2 characters). Nullable LicenseType associated to this price list
IsActive	Type: boolean Indicates whether the price list is active or not active.
Surcharge	Type: boolean Indicates whether the price list affected by surcharge.

## SAPSystemPriceListName Table

This table stores the SAP system price name in multiple languages.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 538: Database columns for SAPSystemPriceListName table**

Database Column	Details
SAPSystemPriceListNameID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP system price list name.
SAPSystemPriceListID	<i>Type:</i> integer. Key Foreign key to the SAP price list.
Language	<i>Type:</i> text (max 4 characters) A unique code to identify the language.
PriceListName	<i>Type:</i> text (max 128 characters). Nullable The name of the SAP price list.

## SAPSystemRFCConnectionSummary Table

This table stores the link between SAP System and RFC Consumption.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 539: Database columns for SAPSystemRFCConnectionSummary table**

Database Column	Details
SAPSystemRFCConnectionSummaryID	<i>Type:</i> integer. Key. Generated ID A unique identifier.
SAPSystemID	<i>Type:</i> integer. Key The Non-SAP system
SAPRFCConnectionSummaryID	<i>Type:</i> integer. Key The RFC consumption.

## SAPSystemType Table

This table stores SAP system type.

**Table 540: Database columns for SAPSystemType table**

Database Column	Details
SAPSystemTypeID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP system type.
TypeName	<i>Type:</i> text (max 64 characters). Key A unique lookup for each SAPSystemType. Possible values and the corresponding default strings are: <ul style="list-style-type: none"> <li>• SAP</li> <li>• NonSAP</li> </ul>
ResourceName	<i>Type:</i> text (max 256 characters). Nullable A localizable resource string representing a SAP system type. Foreign key to the ComplianceResourceString table.
DefaultValue	<i>Type:</i> text (max 100 characters) The text to display if the SAP system type resource string has no translation.

## SAPTransactionProfile Table

This table stores SAP transaction profiles.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 541: Database columns for SAPTransactionProfile table**

Database Column	Details
SAPTransactionProfileID	<i>Type:</i> integer. Key. Generated ID A unique identifier for the SAP transaction profile.
SAPSystemLandscapeID	<i>Type:</i> integer. Key



Database Column	Details
	Foreign key to SAP system landscapes the SAP transaction profile belongs to.
TransactionProfileName	Type: text (max 128 characters) Name of the SAP transaction profile
Description	Type: text. Nullable Description of the SAP transaction profile
CreationUser	Type: text (max 128 characters). Nullable The user who created the SAP transaction profile.
CreationDate	Type: datetime The data and time the SAP transaction profile was created.
UpdatedUser	Type: text (max 128 characters). Nullable The last user who update the SAP transaction profile.
UpdatedDate	Type: datetime The date and time the SAP transaction profile was last updated.

## SAPTransactionProfileObject Table

This table stores the linking between SAP transaction profile and SAP object.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 542: Database columns for SAPTransactionProfileObject table**

Database Column	Details
SAPTransactionProfileObjectID	Type: integer. Key. Generated ID A unique identifier for the SAP transaction profile object
SAPTransactionProfileID	Type: integer. Key Foreign key to a SAP transaction profile.
ObjectName	Type: text (max 128 characters)

Database Column	Details
	The SAP object name
Description	<i>Type:</i> text. Nullable The SAP object description
IsTransaction	<i>Type:</i> boolean Indicates whether the object is of type Transaction
IsReport	<i>Type:</i> boolean Indicates whether the object is of type Report
IsJob	<i>Type:</i> boolean Indicates whether the object is of type Job
IsExcludedFromProfile	<i>Type:</i> boolean Indicates whether the object is marked as excluded from this profile.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The user who created the profile and object link.
CreationDate	<i>Type:</i> datetime The data and time the profile and object link was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The last user who update the profile and object link.
UpdatedDate	<i>Type:</i> datetime The date and time the profile and object link was last updated.
IsNonSAP	<i>Type:</i> boolean Indicates whether the object is of type Non-SAP

## SAPUser Table

This table stores the data specific to the definition of SAP users.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 543: Database columns for SAPUser table**

Database Column	Details
SAPUserID	Type: integer. Key. Generated ID A unique identifier for the SAP user.
SAPSystemID	Type: integer. Key Foreign key to the system that the user belongs to.
UserName	Type: text (max 256 characters). Key The SAP user's username.
FirstName	Type: text (max 40 characters). Nullable The SAP user's first name.
LastName	Type: text (max 40 characters). Nullable The SAP user's last name.
ValidFrom	Type: datetime. Nullable The date that the SAP user is valid from on the SAP system.
ValidTo	Type: datetime. Nullable The date that the SAP user is valid to on the SAP system.
UserType	Type: text (max 1 characters). Nullable The type of user the SAP user is.
LicenseType	Type: text (max 2 characters). Nullable The type of license assigned to the SAP user.
UserGroup	Type: text (max 12 characters). Nullable The user group the SAP user belongs to.
LastLogonDate	Type: datetime. Nullable The date when the SAP user last logged on to the SAP system.
IsDeveloper	Type: boolean

Database Column	Details
	Indicates whether the SAP user is a developer or not.
UserCreationDate	<i>Type:</i> datetime. Nullable The date the SAP user was created.
EmailAddress	<i>Type:</i> text (max 128 characters). Nullable The SAP user's email address.
TelephoneNumber	<i>Type:</i> text (max 30 characters). Nullable The SAP user's telephone number.
TelephoneExtension	<i>Type:</i> text (max 10 characters). Nullable The SAP user's telephone extension.
AccountID	<i>Type:</i> text (max 12 characters). Nullable The SAP user's account ID.
CostCenter	<i>Type:</i> text (max 8 characters). Nullable The cost center the SAP user belongs to.
CompanyName1	<i>Type:</i> text (max 40 characters). Nullable The name of the company the SAP user belongs to.
CompanyName2	<i>Type:</i> text (max 40 characters). Nullable The name of a second company the SAP user belongs to.
Department	<i>Type:</i> text (max 40 characters). Nullable The department the SAP user belongs to.
UserFunction	<i>Type:</i> text (max 40 characters). Nullable
UserLockStatus	<i>Type:</i> integer. Nullable User lock status.
SpecialVersionAssignment	<i>Type:</i> text (max 2 characters). Nullable
CountrySurcharge	<i>Type:</i> text (max 4 characters). Nullable
RepresentativeFromDate	<i>Type:</i> datetime. Nullable
RepresentativeToDate	<i>Type:</i> datetime. Nullable
IsDeleted	<i>Type:</i> boolean

Database Column	Details
	Indicated whether the SAP user has been deleted or not.
ChargeableUserClient	Type: text (max 32 characters). Nullable
ChargeableUserSysID	Type: text (max 32 characters). Nullable
ChargeableUserName	Type: text (max 12 characters). Nullable
RemoteServerUserName	Type: text (max 64 characters). Nullable Remote server user name

## SAPUserRole Table

This table stores SAP users and its SAP role memberships



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 544: Database columns for SAPUserRole table**

Database Column	Details
SAPUserRoleID	Type: integer. Key. Generated ID A unique identifier for SAP user role.
SAPUserID	Type: integer. Key Foreign key to the SAP user that the role belongs to.
SAPRoleID	Type: integer. Key Foreign key to SAP role.
ValidFrom	Type: datetime. Nullable The date that the SAP role is valid from.
ValidTo	Type: datetime. Nullable The date that the SAP role is valid to.

## SAPUserType Table

This table stores SAP User type.

**Table 545: Database columns for SAPUserType table**

Database Column	Details
SAPUserTypeID	Type: integer. Key. Generated ID
UserTypeCode	<p>Type: text (max 1 characters). Key</p> <p>A unique lookup for each SAPUserType. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>• A = Dialog</li> <li>• B = System</li> <li>• C = Communication Data</li> <li>• D = BDC</li> <li>• L = Reference</li> <li>• S = Service</li> </ul>
ResourceName	<p>Type: text (max 256 characters)</p> <p>A localizable resource string representing a SAP user type. Foreign key to the ComplianceResourceString table.</p>
DefaultValue	<p>Type: text (max 100 characters)</p> <p>The text to display if the system type resource string has no translation.</p>

## ManageSoft Tables

The complete set of database tables documented here includes:

- DatabaseConfiguration table (see *DatabaseConfiguration Table* on page 542)

## DatabaseConfiguration Table

The DatabaseConfiguration table contains configuration properties for the FlexNet Manager Suite database tables, which are used for ongoing maintenance of the database.

**Table 546: Database columns for DatabaseConfiguration table**

Database Column	Details
Property	<i>Type:</i> text (max 32 characters). Key The name of the property.
Value	<i>Type:</i> text (max 256 characters) The value of the property.
Created	<i>Type:</i> datetime The date and time the property was created.
LastUpdate	<i>Type:</i> datetime The date and time the property was last updated.

## ReferenceData Tables

The complete set of database tables documented here includes:

- Country table (see *Country Table* on page 543)
- Language table (see *Language Table* on page 544)
- Locale table (see *Locale Table* on page 544)
- OperatingSystem table (see *OperatingSystem Table* on page 545)

## Country Table

Stores country information, including their ISO country code and English names.

**Table 547: Database columns for Country table**

Database Column	Details
CountryCode	<i>Type:</i> text (max 2 characters). Key The two letter country code.
Name	<i>Type:</i> text (max 128 characters). Key The english name of the country.

# Language Table

Stores language information, including their English names, and various forms of language id.

**Table 548: Database columns for Language table**

Database Column	Details
LangCode3	<i>Type:</i> text (max 3 characters). Key The three letter language code.
LangCode2	<i>Type:</i> text (max 2 characters). Nullable The two letter language code.
EnglishName	<i>Type:</i> text (max 128 characters). Key The english name of the language.
LocalName	<i>Type:</i> text (max 128 characters). Nullable The name of the language, written in the local language.
MSLanguageID	<i>Type:</i> integer. Nullable The Microsoft language id, as specified in winnt.h in the Platform SDK.

# Locale Table

Stores locale information, which consists of country and language combinations. Use the `LocaleCode` column as the foreign key into this table.

**Table 549: Database columns for Locale table**

Database Column	Details
LocaleCode	<i>Type:</i> text (max 6 characters). Key A combination of the language code and country code, separated by a hyphen. If there is no country code, then there will be no hyphen added. This column MUST have the correct value when inserted, based on the values of the language and country codes.
LangCode3	<i>Type:</i> text (max 3 characters). Key The three letter language code.
CountryCode	<i>Type:</i> text (max 2 characters). Key. Nullable



Database Column	Details
	The two letter country code.
LocaleName	<i>Type:</i> text (max 128 characters) The name of the locale. For example, "English (United States)".
MSLocaleID	<i>Type:</i> integer. Nullable The Microsoft identifier for the locale. For example, 1033 for English (United States).

## OperatingSystem Table

This table stores the information about different types of OS available on the network devices

**Table 550: Database columns for OperatingSystem table**

Database Column	Details
OperatingSystemID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
OperatingSystemName	<i>Type:</i> text (max 128 characters). Key Name of operating system
Category	<i>Type:</i> integer. Nullable Reference to operating system category

## Rights Tables

The complete set of database tables documented here includes:

- ActionClass table (see *ActionClass Table* on page 545)
- PartitionType table (see *PartitionType Table* on page 546)
- Resource table (see *Resource Table* on page 546)

## ActionClass Table

The types of action on a `Resource` for which rights may be granted or denied.

**Table 551: Database columns for ActionClass table**

Database Column	Details
ActionClassID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number.
ActionClassName	<i>Type:</i> text (max 16 characters). Key The name of the ActionClass.

## PartitionType Table

Some secured *Resources* may be partitioned. Partitions are used to grant rights to one part of a *Resource* excluding other parts, for example limiting rights so that the operator can access only certain distribution servers, organizational units, or areas in the software library. There are three types of partitioning, defined by entries in this table.

**Table 552: Database columns for PartitionType table**

Database Column	Details
PartitionTypeID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number.
PartitionTypeName	<i>Type:</i> text (max 32 characters). Key Name of the PartitionType.

## Resource Table

Access rights are granted to the *Resources* defined in this table.

**Table 553: Database columns for Resource table**

Database Column	Details
ResourceID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number.
ResourceName	<i>Type:</i> text (max 16 characters). Key Name of the Resource.
PartitionTypeID	<i>Type:</i> integer. Nullable

Database Column	Details
	If not NULL, the type of partitioning used with this Resource.

## Targeting Tables

The complete set of database tables documented here includes:

- TargetType table (see *TargetType Table* on page 547)

## TargetType Table

The TargetType table contains a row for each type of object that can be targeted in FlexNet Manager Suite.

**Table 554: Database columns for TargetType table**

Database Column	Details
TargetTypeID	<p>Type: integer. Key. Generated ID</p> <p>The ID for the target type:</p> <ul style="list-style-type: none"> <li>• Computers</li> <li>• Users</li> <li>• Group</li> <li>• DistributionLocation</li> <li>• DistributionServer</li> <li>• Organization</li> <li>• Assets</li> <li>• Contracts</li> <li>• Purchase orders</li> <li>• Software licenses</li> <li>• Software titles</li> <li>• Compliance computers</li> <li>• Compliance users</li> <li>• Operators</li> <li>• SAP system landscapes</li> <li>• SAP systems</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• SAP rule sets</li> <li>• Discovered devices</li> <li>• Beacon</li> <li>• Vendor</li> <li>• Device</li> <li>• Rule</li> <li>• Inventory connection</li> <li>• FNMP Server</li> <li>• Fast Import</li> <li>• OLE DB Connection</li> <li>• ORACLE Connection</li> <li>• XML</li> <li>• Intermediate File</li> <li>• ADSI Connection</li> <li>• Web Service</li> <li>• SQL Connection</li> <li>• Software Title Evidence</li> <li>• FNMEA Agent</li> <li>• Installed Software</li> <li>• Baseline Import</li> </ul>
TargetTypeName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The name of the target type.</p>

## Tenants Tables

The complete set of database tables documented here includes:

- FlexeraLicense table (see *FlexeraLicense Table* on page 549)
- Tenant table (see *Tenant Table* on page 549)

## FlexeraLicense Table

The `FlexeraLicense` table contains the encoded contents of the Flexera Software licenses required for the tenants in the system. This table is also used by the system in the single-tenant setup where there is only one tenant.

**Table 555: Database columns for FlexeraLicense table**

Database Column	Details
<code>TenantUID</code>	<i>Type:</i> text (max 40 characters). Key The unique identifier of a tenant. A reference to the <code>Tenant</code> to which this license is attached.
<code>License</code>	<i>Type:</i> text The encoded contents of the Flexera Software license attached to a particular <code>Tenant</code> .
<code>LicenseChecksum</code>	<i>Type:</i> integer. Key The check sum of the license.
<code>LicenseDetails</code>	<i>Type:</i> XML. Nullable XML definition of the license details

## Tenant Table

The `Tenant` table contains the details of each tenant in multitenant FlexNet Manager Suite database tables.

**Table 556: Database columns for Tenant table**

Database Column	Details
<code>TenantID</code>	<i>Type:</i> integer. Key. Generated ID The tenant ID in a multi-tenant database.
<code>TenantUID</code>	<i>Type:</i> text (max 40 characters). Key The unique identifier of a tenant. This identifier is used to identify the tenant in environments where tenant information is stored on multiple databases.
<code>TenantName</code>	<i>Type:</i> text (max 256 characters). Key The name of the tenant.
<code>Comments</code>	<i>Type:</i> text. Nullable

Database Column	Details
	Operator comments about this tenant record.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the tenant record.
CreationDate	<i>Type:</i> datetime The date the tenant record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The name of the operator who last updated the tenant record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the tenant record was last updated.

---

# 2

---

## Compliance Reader Database Schema

### Topics:

- *Information Structure*
- *Compliance.InventoryReader.Logic Tables*
- *Compliance.InventoryWriter.Matching Tables*

This chapter describes the schema for the staging tables used by the importer (`ComplianceReader.exe`) in the process of importing data into the main FlexNet Manager Suite. Imports through these tables may come from many sources, including (but not limited to) the inventory data collected by the FlexNet inventory agent and rationalized in the inventory database (see *Inventory Database Schema* on page 634).

For each data source, data in these tables is over-written as each import.

# Information Structure

The following information is provided about database tables. Items appear only when relevant to the database column, and are suppressed where they do not apply. Two of these items (shown bold) are columns in the following pages, and the remainder are displayed within the **Details**.

Item	Comment
<b>Database Column</b>	The name of the column in the SQL table.
<i>Type</i>	The data type of the contents of the database column.
Size	For types that have a maximum capacity, the upper limit is provided in parentheses.
Key	The word "Key" appears when a column is a unique key field within the table. It is possible for several database columns to be part of the key, so that this indicator may appear for several columns in a table.
Generated ID	This indicates that a numeric ID is assigned by the database.
Nullable	If this indicator is present, the database column permits nulls.
Computed	This indicator appears for columns that are automatically computed by the database.
Default	If a column has a default value declared in the schema, this is specified at the end of the first set of details for the column.
<b>Details</b>	Describes the data stored in the database column, including many of the indicators described above.

## Compliance.InventoryReader.Logic Tables

The complete set of database tables documented here includes:

- ExpiredImportedComputer table (see *ExpiredImportedComputer Table* on page 555)
- ImportedARSLicense table (see *ImportedARSLicense Table* on page 561)
- ImportedActiveDirectoryComputer table (see *ImportedActiveDirectoryComputer Table* on page 563)
- ImportedActiveDirectoryDomain table (see *ImportedActiveDirectoryDomain Table* on page 564)
- ImportedActiveDirectoryExternalMember table (see *ImportedActiveDirectoryExternalMember Table* on page 565)
- ImportedActiveDirectoryGroup table (see *ImportedActiveDirectoryGroup Table* on page 565)
- ImportedActiveDirectoryMember table (see *ImportedActiveDirectoryMember Table* on page 566)



- ImportedActiveDirectoryUser table (see *ImportedActiveDirectoryUser Table* on page 566)
- ImportedActiveSyncDevice table (see *ImportedActiveSyncDevice Table* on page 567)
- ImportedAttributeMapping table (see *ImportedAttributeMapping Table* on page 569)
- ImportedCluster table (see *ImportedCluster Table* on page 570)
- ImportedClusterGroup table (see *ImportedClusterGroup Table* on page 571)
- ImportedClusterGroupMember table (see *ImportedClusterGroupMember Table* on page 572)
- ImportedClusterHostAffinityRule table (see *ImportedClusterHostAffinityRule Table* on page 572)
- ImportedClusterNode table (see *ImportedClusterNode Table* on page 573)
- ImportedComputer table (see *ImportedComputer Table* on page 574)
- ImportedComputerCustomProperty table (see *ImportedComputerCustomProperty Table* on page 582)
- ImportedCustomPropertyName table (see *ImportedCustomPropertyName Table* on page 582)
- ImportedDomain table (see *ImportedDomain Table* on page 583)
- ImportedEvidenceAttribute table (see *ImportedEvidenceAttribute Table* on page 583)
- ImportedFNMEAFeature table (see *ImportedFNMEAFeature Table* on page 584)
- ImportedFNMEAProduct table (see *ImportedFNMEAProduct Table* on page 585)
- ImportedFNMEAUsageStatus table (see *ImportedFNMEAUsageStatus Table* on page 586)
- ImportedFileEvidence table (see *ImportedFileEvidence Table* on page 587)
- ImportedFileEvidenceMapping table (see *ImportedFileEvidenceMapping Table* on page 588)
- ImportedGuidMapping table (see *ImportedGuidMapping Table* on page 589)
- ImportedILMTPVUCounts table (see *ImportedILMTPVUCounts Table* on page 590)
- ImportedILMTPVUCreatedLicenses table (see *ImportedILMTPVUCreatedLicenses Table* on page 591)
- ImportedILMTVMMMapping table (see *ImportedILMTVMMMapping Table* on page 592)
- ImportedInstalledFileEvidence table (see *ImportedInstalledFileEvidence Table* on page 592)
- ImportedInstalledFileEvidenceUsage table (see *ImportedInstalledFileEvidenceUsage Table* on page 593)
- ImportedInstalledInstallerEvidence table (see *ImportedInstalledInstallerEvidence Table* on page 594)
- ImportedInstalledInstallerEvidenceAttribute table (see *ImportedInstalledInstallerEvidenceAttribute Table* on page 595)
- ImportedInstalledInstallerEvidenceUsage table (see *ImportedInstalledInstallerEvidenceUsage Table* on page 596)
- ImportedInstalledWMIEvidence table (see *ImportedInstalledWMIEvidence Table* on page 597)
- ImportedInstallerEvidence table (see *ImportedInstallerEvidence Table* on page 598)
- ImportedInstallerEvidenceMapping table (see *ImportedInstallerEvidenceMapping Table* on page 599)

- ImportedInstallerEvidenceRepackageMapping table (see *ImportedInstallerEvidenceRepackageMapping Table* on page 599)
- ImportedInstance table (see *ImportedInstance Table* on page 600)
- ImportedInstanceUser table (see *ImportedInstanceUser Table* on page 601)
- ImportedMissingComputer table (see *ImportedMissingComputer Table* on page 602)
- ImportedMissingLicenseUser table (see *ImportedMissingLicenseUser Table* on page 603)
- ImportedMissingUser table (see *ImportedMissingUser Table* on page 604)
- ImportedProductCodeEvidenceMapping table (see *ImportedProductCodeEvidenceMapping Table* on page 604)
- ImportedRelatedInstalledInstallerEvidence table (see *ImportedRelatedInstalledInstallerEvidence Table* on page 605)
- ImportedRemoteApplication table (see *ImportedRemoteApplication Table* on page 606)
- ImportedRemoteApplicationAccess table (see *ImportedRemoteApplicationAccess Table* on page 607)
- ImportedRemoteApplicationInstallerData table (see *ImportedRemoteApplicationInstallerData Table* on page 607)
- ImportedRemoteApplicationServer table (see *ImportedRemoteApplicationServer Table* on page 608)
- ImportedRemoteServerFileEvidenceMapping table (see *ImportedRemoteServerFileEvidenceMapping Table* on page 609)
- ImportedRemoteUsage table (see *ImportedRemoteUsage Table* on page 610)
- ImportedRemoteUserToApplicationAccess table (see *ImportedRemoteUserToApplicationAccess Table* on page 611)
- ImportedSite table (see *ImportedSite Table* on page 612)
- ImportedSiteSubnet table (see *ImportedSiteSubnet Table* on page 613)
- ImportedStringMapping table (see *ImportedStringMapping Table* on page 613)
- ImportedStringMappingLatin1CS table (see *ImportedStringMappingLatin1CS Table* on page 614)
- ImportedUser table (see *ImportedUser Table* on page 615)
- ImportedVDI table (see *ImportedVDI Table* on page 616)
- ImportedVDIEndPointAccess table (see *ImportedVDIEndPointAccess Table* on page 617)
- ImportedVDITemplate table (see *ImportedVDITemplate Table* on page 618)
- ImportedVDIUser table (see *ImportedVDIUser Table* on page 619)
- ImportedVMHostManagedBySoftware table (see *ImportedVMHostManagedBySoftware Table* on page 620)
- ImportedVMPool table (see *ImportedVMPool Table* on page 621)
- ImportedVirtualMachine table (see *ImportedVirtualMachine Table* on page 622)

- ImportedWMIEvidence table (see *ImportedWMIEvidence Table* on page 625)
- ImportedWMIEvidenceRuleMapping table (see *ImportedWMIEvidenceRuleMapping Table* on page 626)
- ImporterValueMapping table (see *ImporterValueMapping Table* on page 626)
- InstalledApplications table (see *InstalledApplications Table* on page 627)
- RelatedInstalledApplications table (see *RelatedInstalledApplications Table* on page 628)

## ExpiredImportedComputer Table

The `ExpiredImportedComputer` table holds all of the computers which have been retrieved from the source connections and are expired.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 557: Database columns for `ExpiredImportedComputer` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The identifier for a data source connection in the <code>ComplianceConnection</code> table.</p>
<code>ExternalID</code>	<p><i>Type:</i> big integer. Key. Nullable</p> <p>The identifier used in the source connection for the computer.</p>
<code>ComputerName</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The name of the computer. In Windows, this is the NetBIOS name of the local computer, as returned by <code>GetComputerName()</code>. For UNIX, it is the host name of the machine, as returned by <code>gethostname(2)</code>.</p>
<code>Domain</code>	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>The domain of the computer.</p>
<code>OperatingSystem</code>	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The operating system of the computer.</p>
<code>ServicePack</code>	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The service pack installed for the operating system.</p>

Database Column	Details
NumberOfProcessors	<i>Type:</i> integer. Nullable The number of processors in the computer.
ProcessorType	<i>Type:</i> text (max 256 characters). Nullable The type of processor in the computer.
MaxClockSpeed	<i>Type:</i> integer. Nullable The maximum clock speed of the fastest processor in the computer.
NumberOfCores	<i>Type:</i> integer. Nullable The number of cores in the computer.
TotalMemory	<i>Type:</i> big integer. Nullable The total RAM in the computer, in bytes.
ChassisType	<i>Type:</i> text (max 128 characters). Nullable The type of case of the computer. The value must be a (case insensitive) exact match for one of the values shown. Note that some license types use this information to optimize the licensing position, particularly with desktop and laptop computers.
NumberOfHardDrives	<i>Type:</i> integer. Nullable The number of hard drives in the computer.
TotalDiskSpace	<i>Type:</i> big integer. Nullable The total size of all hard drives in the computer.
NumberOfNetworkCards	<i>Type:</i> integer. Nullable The number of network cards in the computer.
NumberOfDisplayAdapters	<i>Type:</i> integer. Nullable The number of graphics cards in the computer.
IPAddress	<i>Type:</i> text (max 256 characters). Nullable The IP address of the computer.
MACAddress	<i>Type:</i> text (max 256 characters). Nullable The MAC address of the computer.
Manufacturer	<i>Type:</i> text (max 128 characters). Nullable The manufacturer of the computer hardware. Some examples include:

Database Column	Details
	<ul style="list-style-type: none"> <li>On Windows, the SMBios manufacturer (the WMI Manufacturer property of the 'Win32_ComputerSystem' class).</li> <li>On Linux, 'Manufacturer' in the 'System Information' section resulting from the 'dmidecode' command. Sample command: 'dmidecode -s system-manufacturer'</li> <li>On Solaris x86, as for Linux, with failovers first to 'sysinfo SI_HW_PROVIDER' and then to 'ModelNo'.</li> <li>On Solaris SPARC, the 'sysinfo SI_HW_PROVIDER'. Typically this value is 'Sun_Microsystems' or, more recently, 'Oracle Corporation'. Failover to the 'ModelNo'.</li> <li>On HP-UX, the string literal 'HP'.</li> <li>On AIX, the 'modelname' system attribute preceding the comma character. For example, if the 'modelname' system attribute is 'IBM,8202-E4B', then use 'IBM'. This value is typically 'IBM'.</li> </ul>
ModelNo	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The model number of the computer.</p>
SerialNo	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>The hardware serial number of the computer. The goal of this value is to be tied to the physical hardware, partition or virtual machine and to be as unique as possible across all computers in the organization. This is due to its use in tracking computers, particularly after an operating system rebuild. This value is also used to socialize computer inventory from different inventory sources, and is used to map virtual machine guest operating system inventory to the VM host on which the virtual machine is running. Example sources:</p> <ul style="list-style-type: none"> <li>On Windows, the SMBios serial number. The WMI 'SerialNumber' property of the 'Win32_BIOS' class. Can fail over to the 'SerialNumber' property of the 'Win32_SystemEnclosure' class which is typically the same value.</li> <li>On Linux, the SMBios serial number read using the command 'dmidecode -s system-serial-number'. Specifically, the 'System Information' section and the 'Serial Number' in that section is used.</li> <li>On Solaris 10 8/07 or later, for a non-global zone, the UUID value from the /etc/zones/index file. For a global zone, the same as Solaris 10 releases earlier than 8/07.</li> <li>For Solaris 10 releases earlier than 8/07, the hexadecimal version of 'SI_HW_SERIAL' with an appended hyphen character followed by the Zone's name. For example, '838bfc7b-global' or '838bfc7b-myzone'.</li> <li>For Solaris 8 and 9, The hexadecimal version of 'SI_HW_SERIAL'.</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>For Mac OS X, the serial number of the machine as printed on the packaging and found in "About this Mac" from the desktop.</li> <li>For HP-UX, the 'confstr_CS_PARTITION_IDENT' partition identifier if it is an nPar or vPar, or '_CS_MACHINE_IDENT' if not; with a failover to the machine serial number, and a final failover to the 'uname' machine identification number.</li> <li>For AIX, the 'id_to_partition' system attribute, starting from the third character (strips a '0X' from the start). For example, if the 'id_to_partition' system attribute is '0X0473409002F7B201' then use '0473409002F7B201'.</li> </ul>
HostID	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>An identifier for the host of the computer (when the computer is a virtual machine).</p>
LastLoggedInUser	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The DOMAIN/SAMAccountName of the user last logged onto the computer.</p>
InventoryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date the computer last had inventory reported.</p>
HardwareInventoryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date (and optionally time) when the hardware was last inventoried. For automated/scheduled data uploads through an inventory beacon, make sure that inventory dates are kept current, as they are used to report out-of-date inventory sources. For a one-time upload to the central application server, leave inventory dates empty (null). At each import from the saved file, the import date is used as the inventory date, which prevents the inventory becoming stale.</p>
ServicesInventoryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date when services (for example, Oracle) were last scanned on this computer. For automated/scheduled data uploads through an inventory beacon, make sure that inventory dates are kept current, as they are used to report out-of-date inventory sources. For a one-time upload to the central application server, leave inventory dates empty (null). At each import from the saved file, the import date is used as the inventory date, which prevents the inventory becoming stale.</p>
InventoryAgent	<p><i>Type:</i> text (max 128 characters)</p> <p>The name of the person or tool that performed the last inventory.</p>
ComplianceComputerID	<p><i>Type:</i> integer. Nullable</p>

Database Column	Details
	Identifier of the computer in the <code>ComplianceComputer</code> table that this imported computer links to. This is populated by the import process and does not need to be provided by the source connections.
<code>ComplianceDomainID</code>	<i>Type:</i> integer. Nullable Identifier of the domain in the <code>ComplianceDomain</code> table that this computer belongs to. This is populated by the import process and does not need to be provided by the source connections.
<code>IncompleteRecord</code>	<i>Type:</i> boolean. Nullable Used to identify records which do not have all information specified. Primarily used for <code>ManageSoft</code> source connections where the domain name was not reliably reported.
<code>NumberOfSockets</code>	<i>Type:</i> integer. Nullable The number of sockets in the computer.
<code>PartialNumberOfProcessors</code>	<i>Type:</i> decimal. Nullable The fractional processor count available to this computer.
<code>UntrustedSerialNo</code>	<i>Type:</i> boolean Use when this computer is known to have a serial number from a data source that should not be trusted.
<code>FullDetailsFromExternalID</code>	<i>Type:</i> big integer. Nullable If this computer is marked as incomplete, and some of its properties are updated from another computer, record the external ID if the full computer.
<code>FullDetailsFromComplianceConnectionID</code>	<i>Type:</i> integer. Nullable If this computer is marked as incomplete, and some of its properties are updated from another computer, record the connection ID if the full computer.
<code>ComplianceComputerTypeID</code>	<i>Type:</i> integer. Nullable If you know that the computer is a virtual machine or VM host, record that data here. If you are unsure, leave this cell empty (NULL): this allows the system to infer the computer type (for example, a computer with VMs linked to it is inferred to be a VM host). If data comes from multiple inventory sources, leaving this value as null also allows the value to be inserted from another source. So, unless there is a very good reason, do not just specify 'Computer', but allow the inference rules to help.
<code>ILMTAgentID</code>	<i>Type:</i> big integer. Nullable

Database Column	Details
	Store the unique ID used by the ILMT agent on this device, if the inventory source is aware of this value.
HostIdentifyingNumber	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>Virtual hosts may have an identifier that is unique only across that hardware model. It is less unique than the true hardware serial number, for example.</p>
HostType	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The type of the physical host computer. This value is similar to the model number, but it is always for the physical server that an execution context may be running on. Therefore, this will generally be a known value for standalone machines and partitions, but it will not be known for virtual machines. This value is used for matching computers. Examples:</p> <ul style="list-style-type: none"> <li>• 'i86pc'</li> <li>• 'Sun-Fire-T1000'</li> <li>• 'rx7620'</li> <li>• '785' (for a 9000/785/C3700)</li> <li>• '8202' (for an IBM,8202-E4B).</li> </ul>
NumberOfLogicalProcessors	<p><i>Type:</i> integer. Nullable</p> <p>The number of logical processors in the computer.</p>
IsRemoteACLDevice	<p><i>Type:</i> boolean</p> <p>Used to determine if the current record is a remote ACL based device.</p>
IsDuplicate	<p><i>Type:</i> boolean</p> <p>Used to identify that imported computer is a duplicate of another, whereby a new computer will not created.</p>
LegacySerialNo	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>A previous serial number of this computer that can also be used for matching.</p>
UUID	<p><i>Type:</i> unique identifier. Nullable</p> <p>The BIOS UUID of the computer.</p>
IMEI	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>IMEI (International Mobile Equipment Identity) is a 15- or 17-digit code that uniquely identifies mobile phone sets. Leave blank (null) for other device types.</p>
PhoneNumber	<p><i>Type:</i> text (max 128 characters). Nullable</p>



Database Column	Details
	The phone number of the device. Used for mobile devices.
EmailAddress	<i>Type:</i> text (max 256 characters). Nullable The email address associated with the device. Typically used for mobile devices.
CalculatedUser	<i>Type:</i> text (max 128 characters). Nullable The domain/SAMAccountName of the calculated user. Some inventory systems calculate the user who owns a computer. For example, it might be the user who, over the last ten logins, logged in most often.
LastSuccessfulInventoryDate	<i>Type:</i> datetime. Nullable For incremental imports, this represents the inventory date of the computer in the source at the time this record was last successfully imported. If the import procedure has failed, this may be different to the inventory date. At the end of a successful incremental import, this value is updated to match the inventory date. If no value is present in this field, either there has not been a successful import of this computer or the reader for this record is not using an incremental update model.
MDScheduleGeneratedDate	<i>Type:</i> datetime. Nullable The last time the managed device schedule was regenerated.
MDScheduleContainsPVUScan	<i>Type:</i> boolean. Nullable Does this managed device include an event in its current schedule for running extra IBM PVU hardware scans.
FirmwareSerialNumber	<i>Type:</i> text (max 100 characters). Nullable Serial number in the system firmware such as BIOS, EEPROM etc.
MachineID	<i>Type:</i> text (max 100 characters). Nullable For AIX, it is the System ID. For HP-UX, it is the Machine/Software ID. It is unset for other platforms.
IgnoredDueToLicense	<i>Type:</i> boolean True if this machine is not imported into compliance computer table due to license limitation

## ImportedARSLicense Table

The `ImportedARSLicense` table stores Action Request System BMC licenses.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 558: Database columns for ImportedARSLicense table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key. Nullable The identifier of a data source connection in the ComplianceConnection table.
ComputerID	Type: big integer. Key The identifier used in the source connection to represent the computer.
SoftwareLicenseID	Type: integer. Nullable The identifier for the license in the SoftwareLicense table.
ARSLicenseID	Type: integer The identifier for the imported ARS license.
ComplianceComputerID	Type: integer. Nullable The identifier for the compliance computer in the ComplianceComputer table.
LicenseType	Type: text (max 128 characters). Key The ARS license name.
ECMLicenseName	Type: text (max 256 characters) The name of the license in the FlexNet Manager Suite.
LicenseKey	Type: text (max 32 characters). Key. Nullable The imported license key.
LicenseSubType	Type: text (max 16 characters). Key The license subtype (FlexNet Manager Suite license version).
IssueDate	Type: datetime. Key The identifier for the issue date.
ExpiryDate	Type: datetime. Key. Nullable The identifier for the expiry date.

Database Column	Details
SiteName	<i>Type:</i> text (max 64 characters) The identifier for the site name.
HostID	<i>Type:</i> text (max 64 characters) An identifier for the ARS host in the source connection (not used in FlexNet Manager Suite).
LicenseNum	<i>Type:</i> integer The purchase count for the ARS license.
TokenList	<i>Type:</i> text (max 128 characters). Nullable The ARS token list (not used in FlexNet Manager Suite).
Comment	<i>Type:</i> text. Nullable Extra information about the ARS license.
Deleted	<i>Type:</i> integer Set this flag if an ARS license is to be deleted.

## ImportedActiveDirectoryComputer Table

The `ImportedActiveDirectoryComputer` table stores the incoming active directory data for computers.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 559: Database columns for `ImportedActiveDirectoryComputer` table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key The identifier of a data source connection in the <code>ComplianceConnection</code> table.
GUID	<i>Type:</i> unique identifier. Key The GUID of the computer.
ComputerName	<i>Type:</i> text (max 64 characters)

Database Column	Details
	The name of the computer. In Windows, this is the NetBIOS name of the local computer, as returned by <code>GetComputerName()</code> . For UNIX, it is the host name of the machine, as returned by <code>gethostname(2)</code> .
DomainName	<i>Type:</i> text (max 100 characters) The domain name for the computer.
SID	<i>Type:</i> text (max 256 characters). Nullable The SID of the computer.

## ImportedActiveDirectoryDomain Table

The `ImportedActiveDirectoryDomain` table stores the incoming active directory domains for a connection source.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 560: Database columns for ImportedActiveDirectoryDomain table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key The identifier of a data source connection in the <code>ComplianceConnection</code> table.
DomainFQDN	<i>Type:</i> text (max 100 characters). Key The fully qualified name domain name of the AD domain
FlatName	<i>Type:</i> text (max 32 characters) The AD domain flat name
LastADImportTime	<i>Type:</i> datetime The last time the AD data was imported

## ImportedActiveDirectoryExternalMember Table

The `ImportedActiveDirectoryExternalMember` table stores the incoming active directory data for external AD member objects.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 561: Database columns for `ImportedActiveDirectoryExternalMember` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<p>Type: integer. Key</p> <p>The identifier of a data source connection in the <code>ComplianceConnection</code> table.</p>
<code>ParentGroupGUID</code>	<p>Type: unique identifier. Key</p> <p>The parent AD group GUID.</p>
<code>SID</code>	<p>Type: text (max 256 characters). Key</p> <p>The SID of the member object.</p>

## ImportedActiveDirectoryGroup Table

The `ImportedActiveDirectoryGroup` table stores the incoming active directory data for a connection source.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 562: Database columns for `ImportedActiveDirectoryGroup` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<p>Type: integer. Key</p> <p>The identifier of a data source connection in the <code>ComplianceConnection</code> table.</p>
<code>GUID</code>	<p>Type: unique identifier. Key</p>

Database Column	Details
	The GUID of the AD group.
SID	Type: text (max 256 characters). Nullable The SID of the AD group.
Name	Type: text (max 128 characters). Nullable The AD group name
DomainName	Type: text (max 100 characters) The domain name for the user.

## ImportedActiveDirectoryMember Table

The `ImportedActiveDirectoryMember` table stores the incoming active directory data for AD member objects.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 563: Database columns for ImportedActiveDirectoryMember table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key The identifier of a data source connection in the <code>ComplianceConnection</code> table.
GUID	Type: unique identifier. Key The GUID of the member object.
ParentGroupGUID	Type: unique identifier. Key The parent AD group GUID.

## ImportedActiveDirectoryUser Table

The `ImportedActiveDirectoryUser` table stores the incoming active directory data for users.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 564: Database columns for ImportedActiveDirectoryUser table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key The identifier of a data source connection in the ComplianceConnection table.
GUID	Type: unique identifier. Key The GUID of the user.
SAMAccountName	Type: text (max 20 characters) The user name.
DomainName	Type: text (max 100 characters) The domain name for the user.
Sid	Type: text (max 256 characters). Nullable The Sid for the user.

## ImportedActiveSyncDevice Table

The ImportedActiveSyncDevice table stores details of ActiveSync partnerships. A partnership is a user/device pair, so there may be multiple rows for one device.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 565: Database columns for ImportedActiveSyncDevice table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key. Nullable

Database Column	Details
	The identifier for a data source connection in the <code>ComplianceConnection</code> table.
<code>ExternalID</code>	<i>Type:</i> integer. Key. Generated ID The identifier used in the source connection for the computer.
<code>ActiveSyncID</code>	<i>Type:</i> text (max 512 characters). Key. Nullable The EASIdentity presented by the source, a combination of the AD user and the unique device ID.
<code>Domain</code>	<i>Type:</i> text (max 100 characters). Nullable The domain of the device. This may be a flat name or FQDN.
<code>DeviceID</code>	<i>Type:</i> text (max 100 characters). Nullable The unique device identifier.
<code>DeviceOS</code>	<i>Type:</i> text (max 100 characters). Nullable The device operating system.
<code>DeviceModel</code>	<i>Type:</i> text (max 100 characters). Nullable The device model.
<code>DeviceType</code>	<i>Type:</i> text (max 50 characters). Nullable The device type.
<code>DeviceUserAgent</code>	<i>Type:</i> text (max 100 characters). Nullable The device user agent; an ActiveSync client-specific value that may identify the device type.
<code>UserDisplayName</code>	<i>Type:</i> text (max 256 characters). Nullable The AD user display name.
<code>IMEI</code>	<i>Type:</i> text (max 256 characters). Nullable IMEI (International Mobile Equipment Identity) is a 15- or 17-digit code that uniquely identifies mobile phone sets. Leave blank (null) for other device types.
<code>PhoneNumber</code>	<i>Type:</i> text (max 128 characters). Nullable The phone number of the device. Used for mobile devices.
<code>EmailAddress</code>	<i>Type:</i> text (max 256 characters). Nullable The user's primary email address.



Database Column	Details
ExchangeServer	<i>Type:</i> text (max 256 characters). Nullable The source exchange server for this information.
WhenCreatedUTC	<i>Type:</i> datetime. Nullable The date/time this partnership was created, in UTC.
LastSyncAttemptTime	<i>Type:</i> datetime. Nullable The last attempted sync time for this partnership, in UTC.
LastSuccessSync	<i>Type:</i> datetime. Nullable The last successful sync time for this partnership, in UTC.

## ImportedAttributeMapping Table

The `ImportedAttributeMapping` table is used by the importer to link imported instance attributes with attributes in the `Attribute` table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 566: Database columns for ImportedAttributeMapping table**

Database Column	Details
AttributeID	<i>Type:</i> integer. Nullable The identifier for the instance attribute in the <code>Attribute</code> table.
ExternalAttributeID	<i>Type:</i> integer. Key. Nullable The identifier used in the source connection for the imported instance attribute.
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier of a data source connection in the <code>ComplianceConnection</code> table.

# ImportedCluster Table

The `ImportedCluster` table holds all of the clusters which have been retrieved from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 567: Database columns for ImportedCluster table**

Database Column	Details
<code>ExternalID</code>	<i>Type:</i> big integer. Key. Nullable The unique identifier for this imported cluster.
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
<code>ClusterID</code>	<i>Type:</i> integer. Nullable The unique identifier for this imported cluster. Note that this maps to the 'ExternalID' column in the 'ImportedCluster' table, and not to the 'ClusterID' column.
<code>ExternalName</code>	<i>Type:</i> text (max 256 characters). Nullable The identifier of the cluster in the external cluster management system.
<code>Name</code>	<i>Type:</i> text (max 256 characters) The user-visible name of the cluster.
<code>Namespace</code>	<i>Type:</i> text (max 256 characters). Nullable The name of the domain/datacenter containing the cluster.
<code>ClusterTypeID</code>	<i>Type:</i> integer The type of cluster.
<code>InventoryDate</code>	<i>Type:</i> datetime. Nullable The date the cluster last had inventory reported.
<code>InventoryAgent</code>	<i>Type:</i> text (max 64 characters). Nullable The name of the person or tool that performed the last inventory.

Database Column	Details
DRS	Type: boolean. Nullable Whether Distributed Resource Scheduler (DRS) is enabled
DPM	Type: boolean. Nullable Whether Distributed Power Management (DPM) is enabled

## ImportedClusterGroup Table

The `ImportedClusterGroup` table holds all of the group objects defined on clusters which have been retrieved from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 568: Database columns for `ImportedClusterGroup` table**

Database Column	Details
ExternalID	Type: big integer. Key. Nullable The unique identifier for this imported cluster group.
ComplianceConnectionID	Type: integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ClusterID	Type: integer. Nullable The assigned identifier for this cluster group.
ClusterExternalID	Type: big integer. Key The unique identifier for the imported cluster.
Name	Type: text (max 256 characters) The name of the cluster group.
ClusterTypeID	Type: integer Foreign key to the <code>ClusterType</code> table.

## ImportedClusterGroupMember Table

The `ImportedClusterGroupMember` table holds all of the group memberships defined on clusters which have been retrieved from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 569: Database columns for `ImportedClusterGroupMember` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<p>Type: integer. Key. Nullable</p> <p>The identifier for a data source connection in the <code>ComplianceConnection</code> table.</p>
<code>ClusterGroupExternalID</code>	<p>Type: big integer. Key</p> <p>The unique identifier for the imported cluster group.</p>
<code>ComputerExternalID</code>	<p>Type: big integer. Key. Nullable</p> <p>The identifier used in the source connection for the external computer which is a member of the group.</p>
<code>VCOBJECTID</code>	<p>Type: text (max 256 characters). Key. Nullable</p> <p>The identifier of the virtual machine in Virtual Center.</p>

## ImportedClusterHostAffinityRule Table

The `ImportedClusterHostAffinityRule` table holds all of the host affinity rules for a cluster which have been retrieved from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 570: Database columns for ImportedClusterHostAffinityRule table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ClusterExternalID	<i>Type:</i> big integer. Key The unique identifier for the imported cluster.
Name	<i>Type:</i> text (max 256 characters). Key The name of the cluster group.
ClusterHostGroup ExternalID	<i>Type:</i> big integer. Key The unique identifier for the imported cluster host group.
ClusterVMGroupExternalID	<i>Type:</i> big integer. Key The unique identifier for the imported cluster VM group.
ClusterHostAffinity RuleTypeID	<i>Type:</i> integer A unique identifier indicating a type of Cluster Host Affinity Rule.

## ImportedClusterNode Table

The `ImportedClusterNode` table holds all of the cluster nodes which have been retrieved from the source connections.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 571: Database columns for ImportedClusterNode table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ClusterExternalID	<i>Type:</i> big integer. Key

Database Column	Details
	The unique identifier for the imported cluster.
ComputerExternalID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the external computer which is a member of the cluster.
ClusterNodeTypeID	<i>Type:</i> integer Foreign key to the <code>ClusterNodeType</code> table.

## ImportedComputer Table

The `ImportedComputer` table holds all of the computers which have been retrieved from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 572: Database columns for `ImportedComputer` table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ExternalID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the computer.
ComputerName	<i>Type:</i> text (max 256 characters). Key. Nullable The name of the computer. In Windows, this is the NetBIOS name of the local computer, as returned by <code>GetComputerName()</code> . For UNIX, it is the host name of the machine, as returned by <code>gethostname(2)</code> .
Domain	<i>Type:</i> text (max 100 characters). Key. Nullable The domain of the computer.
OperatingSystem	<i>Type:</i> text (max 128 characters). Nullable The operating system of the computer.

Database Column	Details
ServicePack	<i>Type:</i> text (max 128 characters). Nullable The service pack installed for the operating system.
NumberOfProcessors	<i>Type:</i> integer. Nullable The number of processors in the computer.
ProcessorType	<i>Type:</i> text (max 256 characters). Nullable The type of processor in the computer.
MaxClockSpeed	<i>Type:</i> integer. Nullable The maximum clock speed of the fastest processor in the computer.
NumberOfCores	<i>Type:</i> integer. Nullable The number of cores in the computer.
TotalMemory	<i>Type:</i> big integer. Nullable The total RAM in the computer, in bytes.
ChassisType	<i>Type:</i> text (max 128 characters). Nullable The type of case of the computer. The value must be a (case insensitive) exact match for one of the values shown. Note that some license types use this information to optimize the licensing position, particularly with desktop and laptop computers.
NumberOfHardDrives	<i>Type:</i> integer. Nullable The number of hard drives in the computer.
TotalDiskSpace	<i>Type:</i> big integer. Nullable The total size of all hard drives in the computer.
NumberOfNetworkCards	<i>Type:</i> integer. Nullable The number of network cards in the computer.
NumberOfDisplayAdapters	<i>Type:</i> integer. Nullable The number of graphics cards in the computer.
IPAddress	<i>Type:</i> text (max 256 characters). Nullable The IP address of the computer.
MACAddress	<i>Type:</i> text (max 256 characters). Nullable

Database Column	Details
	The MAC address of the computer.
Manufacturer	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>The manufacturer of the computer hardware. Some examples include:</p> <ul style="list-style-type: none"> <li>On Windows, the SMBios manufacturer (the WMI Manufacturer property of the 'Win32_ComputerSystem' class).</li> <li>On Linux, 'Manufacturer' in the 'System Information' section resulting from the 'dmidecode' command. Sample command: 'dmidecode -s system-manufacturer'</li> <li>On Solaris x86, as for Linux, with failovers first to 'sysinfo SI_HW_PROVIDER' and then to 'ModelNo'.</li> <li>On Solaris SPARC, the 'sysinfo SI_HW_PROVIDER'. Typically this value is 'Sun_Microsystems' or, more recently, 'Oracle Corporation'. Failover to the 'ModelNo'.</li> <li>On HP-UX, the string literal 'HP'.</li> <li>On AIX, the 'modelname' system attribute preceding the comma character. For example, if the 'modelname' system attribute is 'IBM,8202-E4B', then use 'IBM'. This value is typically 'IBM'.</li> </ul>
ModelNo	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The model of the computer hardware or the virtual machine. This value is defined for the context of the current execution environment, rather than the physical server that may be hosting a virtual machine or partition. Examples:</p> <ul style="list-style-type: none"> <li>On Windows, the SMBios product name. The WMI Model property of the Win32_ComputerSystem class.</li> <li>On Linux, the SMBios product name read using the command 'dmidecode -s system-product-name'. Specifically, the 'System Information' section and the 'Product Name' in that section is used.</li> <li>On Solaris x86, as for Linux, with failover to the 'sysinfo SI_PLATFORM', stripping 'SUNW', and replacing hyphen characters with space characters.</li> <li>On Solaris SPARC, the 'openprom' "banner-name" value read from '/dev/openprom'. Failover to the 'sysinfo SI_PLATFORM', stripping 'SUNW', and replacing hyphen characters with space characters.</li> <li>On HP-UX, the 'confstr_CS_MACHINE_MODEL'.</li> <li>On AIX, the 'modelname' system attribute following the comma character. For example, if the 'modelname' system attribute is 'IBM,8202-E4B', then use '8202-E4B'.</li> </ul>



Database Column	Details
SerialNo	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>The hardware serial number of the computer. The goal of this value is to be tied to the physical hardware, partition or virtual machine and to be as unique as possible across all computers in the organization. This is due to its use in tracking computers, particularly after an operating system rebuild. This value is also used to socialize computer inventory from different inventory sources, and is used to map virtual machine guest operating system inventory to the VM host on which the virtual machine is running. Example sources:</p> <ul style="list-style-type: none"> <li>• On Windows, the SMBios serial number. The WMI 'SerialNumber' property of the 'Win32_BIOS' class. Can fail over to the 'SerialNumber' property of the 'Win32_SystemEnclosure' class which is typically the same value.</li> <li>• On Linux, the SMBios serial number read using the command 'dmidecode -s system-serial-number'. Specifically, the 'System Information' section and the 'Serial Number' in that section is used.</li> <li>• On Solaris 10 8/07 or later, for a non-global zone, the UUID value from the / etc/zones/index file. For a global zone, the same as Solaris 10 releases earlier than 8/07.</li> <li>• For Solaris 10 releases earlier than 8/07, the hexadecimal version of 'SI_HW_SERIAL' with an appended hyphen character followed by the Zone's name. For example, '838bfc7b-global' or '838bfc7b-myzone'.</li> <li>• For Solaris 8 and 9, The hexadecimal version of 'SI_HW_SERIAL'.</li> <li>• For Mac OS X, the serial number of the machine as printed on the packaging and found in "About this Mac" from the desktop.</li> <li>• For HP-UX, the 'confstr_CS_PARTITION_IDENT' partition identifier if it is an nPar or vPar, or '_CS_MACHINE_IDENT' if not; with a failover to the machine serial number, and a final failover to the 'uname' machine identification number.</li> <li>• For AIX, the 'id_to_partition' system attribute, starting from the third character (strips a '0X' from the start). For example, if the 'id_to_partition' system attribute is '0X0473409002F7B201' then use '0473409002F7B201'.</li> </ul>
HostID	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>An identifier for the host of the computer (when inventorying a machine partition such as Solaris Zone, AIX IPar, HP-UX nPar/vPar). Examples:</p> <ul style="list-style-type: none"> <li>• For a Zone on Solaris, the hexadecimal version of SI_HW_SERIAL.</li> <li>• For nPar/vPar on HP-UX, the 'confstr_CS_MACHINE_IDENT' unique machine identifier.</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>For IPar on AIX, the 'modelname' system attribute following the comma character. For example, if the 'modelname' system attribute is 'IBM,8202-E4B', then use '8202-E4B'.</li> </ul>
LastLoggedInUser	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The DOMAIN/SAMAccountName of the user last logged onto the computer.</p>
InventoryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date the computer last had inventory reported.</p>
HardwareInventoryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date (and optionally time) when the hardware was last inventoried. For automated/scheduled data uploads through an inventory beacon, make sure that inventory dates are kept current, as they are used to report out-of-date inventory sources. For a one-time upload to the central application server, leave inventory dates empty (null). At each import from the saved file, the import date is used as the inventory date, which prevents the inventory becoming stale. Notice that this value is not available in the web interface.</p>
ServicesInventoryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date when services (for example, Oracle) were last scanned on this computer. For automated/scheduled data uploads through an inventory beacon, make sure that inventory dates are kept current, as they are used to report out-of-date inventory sources. For a one-time upload to the central application server, leave inventory dates empty (null). At each import from the saved file, the import date is used as the inventory date, which prevents the inventory becoming stale.</p>
InventoryAgent	<p><i>Type:</i> text (max 128 characters)</p> <p>The name of the person or tool that performed the last inventory. For imported spreadsheets, you may wish to include the name of the person preparing the data, in case there is subsequent follow-up required.</p>
ComplianceComputerID	<p><i>Type:</i> integer. Key. Nullable</p> <p>Identifier of the computer in the <code>ComplianceComputer</code> table that this imported computer links to. This is populated by the import process and does not need to be provided by the source connections.</p>
ComplianceDomainID	<p><i>Type:</i> integer. Key. Nullable</p> <p>Identifier of the domain in the <code>ComplianceDomain</code> table that this computer belongs to. This is populated by the import process and does not need to be provided by the source connections.</p>

Database Column	Details
IncompleteRecord	<p><i>Type:</i> boolean. Nullable</p> <p>Used to identify records which do not have all information specified. Primarily used for ManageSoft source connections where the domain name was not reliably reported.</p>
NumberOfSockets	<p><i>Type:</i> integer. Nullable</p> <p>The number of sockets in the computer.</p>
PartialNumberOfProcessors	<p><i>Type:</i> decimal. Nullable</p> <p>The fractional processor count available to this computer.</p>
UntrustedSerialNo	<p><i>Type:</i> boolean</p> <p>Is this computer known to have a serial number from a data source that should not be trusted.</p>
FullDetailsFromExternalID	<p><i>Type:</i> big integer. Nullable</p> <p>If this computer is marked as incomplete, and some of its properties are updated from another computer, record the external ID if the full computer.</p>
FullDetailsFromComplianceConnectionID	<p><i>Type:</i> integer. Nullable</p> <p>If this computer is marked as incomplete, and some of its properties are updated from another computer, record the connection ID if the full computer.</p>
ComplianceComputerTypeID	<p><i>Type:</i> integer. Nullable</p> <p>If you know that the computer is a virtual machine or VM host, record that data here. If you are unsure, leave this cell empty (NULL): this allows the system to infer the computer type (for example, a computer with VMs linked to it is inferred to be a VM host). If data comes from multiple inventory sources, leaving this value as null also allows the value to be inserted from another source. So, unless there is a very good reason, do not just specify 'Computer', but allow the inference rules to help.</p>
ILMTAgentID	<p><i>Type:</i> big integer. Key. Nullable</p> <p>The unique ID used by the IBM License Metric Tool (ILMT) inventory agent on this device, if the inventory source is aware of this value. This can be used to track a computer over time and can be used to socialize different inventory sources. Currently the ILMT and ManageSoft inventory adapters report this value. To find these values:</p> <ul style="list-style-type: none"> <li>On Windows: The standalone and agent based ILMT configuration files are '\$(WindowsFolder)/itlm/tlmstandalone.ini' and '\$(WindowsFolder)/itlm/tlmagent.ini' respectively. Read the 'agentid' property from these files using a case-insensitive match against the property name.</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>On UNIX: The standalone and agent based ILMT configuration files are '/etc/tlmstandalone.ini' and '/etc/tlmagent.ini' respectively. Read the 'agentid' property from these files using a case-insensitive match against the property name.</li> </ul>
HostIdentifyingNumber	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>Virtual hosts may have an identifier that is unique only across that hardware model. It is less unique than the true hardware serial number, for example.</p>
HostType	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>The type of the physical host computer. This value is similar to the model number, but it is always for the physical server that an execution context may be running on. Therefore, this will generally be a known value for standalone machines and partitions, but it will not be known for virtual machines. This value is used for matching computers. Examples:</p> <ul style="list-style-type: none"> <li>'i86pc'</li> <li>'Sun-Fire-T1000'</li> <li>'rx7620'</li> <li>'785' (for a 9000/785/C3700)</li> <li>'8202' (for an IBM,8202-E4B).</li> </ul>
NumberOfLogicalProcessors	<p><i>Type:</i> integer. Nullable</p> <p>The number of logical processors in the computer.</p>
IsRemoteACLDevice	<p><i>Type:</i> boolean. Key</p> <p>Used to determine if the current record is a remote ACL based device.</p>
IsDuplicate	<p><i>Type:</i> boolean</p> <p>Used to identify that imported computer is a duplicate of another, whereby a new computer will not created.</p>
LegacySerialNo	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>A previous serial number of this computer that can also be used for matching.</p>
UUID	<p><i>Type:</i> unique identifier. Key. Nullable</p> <p>The BIOS UUID of the computer.</p>
IMEI	<p><i>Type:</i> text (max 256 characters). Nullable</p>

Database Column	Details
	IMEI (International Mobile Equipment Identity) is a 15- or 17-digit code that uniquely identifies mobile phone sets. Leave blank (null) for other device types.
PhoneNumber	<i>Type:</i> text (max 128 characters). Nullable The phone number of the device. Used for mobile devices.
EmailAddress	<i>Type:</i> text (max 256 characters). Nullable The email address associated with the device. Typically used for mobile devices.
CalculatedUser	<i>Type:</i> text (max 128 characters). Nullable The domain/SAMAccountName of the calculated user. Some inventory systems calculate the user who owns a computer. For example, it might be the user who, over the last ten logins, logged in most often.
LastSuccessfulInventoryDate	<i>Type:</i> datetime. Nullable For incremental imports, this represents the inventory date of the computer in the source at the time this record was last successfully imported. If the import procedure has failed, this may be different to the inventory date. At the end of a successful incremental import, this value is updated to match the inventory date. If no value is present in this field, either there has not been a successful import of this computer or the reader for this record is not using an incremental update model.
MDScheduleGeneratedDate	<i>Type:</i> datetime. Nullable The last time the managed device schedule was regenerated.
MDScheduleContainsPVUScan	<i>Type:</i> boolean. Nullable Does this managed device include an event in its current schedule for running extra IBM PVU hardware scans.
FirmwareSerialNumber	<i>Type:</i> text (max 100 characters). Nullable Serial number in the system firmware such as BIOS, EEPROM etc.
MachineID	<i>Type:</i> text (max 100 characters). Nullable For AIX, it is the System ID. For HP-UX, it is the Machine/Software ID. It is unset for other platforms.
IgnoredDueToLicense	<i>Type:</i> boolean True if this machine is not imported into compliance computer table due to license limitation

## ImportedComputerCustomProperty Table

The `ImportedComputerCustomProperty` table is used by the importer to import custom properties for computers.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 573: Database columns for `ImportedComputerCustomProperty` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key The identifier of a data source connection in the <code>ComplianceConnection</code> table.
<code>ExternalID</code>	<i>Type:</i> big integer. Key The identifier, in the source connection, of the computer that this property belongs to.
<code>PropertyNameID</code>	<i>Type:</i> integer. Key The identifier for custom property in the <code>ImportedCustomPropertyName</code> table.
<code>PropertyValue</code>	<i>Type:</i> text (max 256 characters) The value of the custom property.

## ImportedCustomPropertyName Table

The `ImportedCustomPropertyName` table is used by the importer to store the names of custom properties.

**Table 574: Database columns for `ImportedCustomPropertyName` table**

Database Column	Details
<code>PropertyNameID</code>	<i>Type:</i> integer. Key. Generated ID A unique identifier for custom property.
<code>PropertyName</code>	<i>Type:</i> text (max 256 characters). Key The name of the custom property.

## ImportedDomain Table

The `ImportedDomain` table holds all of the domains which have been retrieved from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 575: Database columns for `ImportedDomain` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<p>Type: integer. Key. Nullable</p> <p>The identifier for a data source connection in the <code>ComplianceConnection</code> table.</p>
<code>ComplianceDomainID</code>	<p>Type: integer. Nullable</p> <p>Identifier of the domain in the <code>ComplianceDomain</code> table that this imported domain links to. This is populated as part of the import process and does not need to be provided by the source connections.</p>
<code>QualifiedName</code>	<p>Type: text (max 200 characters). Key. Nullable</p> <p>The fully qualified name of the domain.</p>
<code>FlatName</code>	<p>Type: text (max 200 characters). Key. Nullable</p> <p>The flat name of the domain.</p>

## ImportedEvidenceAttribute Table

The `ImportedEvidenceAttribute` table holds all of the instance attributes from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 576: Database columns for `ImportedEvidenceAttribute` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<p>Type: integer. Key. Nullable</p>

Database Column	Details
	The identifier for a data source connection in the <code>ComplianceConnection</code> table.
<code>AttributeID</code>	<i>Type:</i> integer. Key. Nullable The identifier used in the source connection for the instance attribute.
<code>AttributeName</code>	<i>Type:</i> text (max 256 characters). Key. Nullable The name of the instance attribute.

## ImportedFNMEAFeature Table

The `ImportedFNMEAFeature` table is used by the importer to import FlexNet Manager for Engineering Applications features.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 577: Database columns for `ImportedFNMEAFeature` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<i>Type:</i> integer The identifier of a data source connection in the <code>ComplianceConnection</code> table.
<code>ExternalID</code>	<i>Type:</i> integer. Key. Generated ID The identifier of the feature from the external data source.
<code>Name</code>	<i>Type:</i> text (max 256 characters) The name for this feature.
<code>Version</code>	<i>Type:</i> text (max 32 characters). Nullable The version of this feature.
<code>Publisher</code>	<i>Type:</i> text (max 256 characters) The publisher of the feature.
<code>VendorDaemon</code>	<i>Type:</i> text (max 256 characters)



Database Column	Details
	The vendor daemon of the feature.
ConsumedQuantity	Type: integer The count of the feature installs.
OutOfComplianceQuantity	Type: integer The count of out-of-compliance feature installs, as calculated by FlexNet Manager for Engineering Applications.
ComplianceStatus	Type: text (max 32 characters) The compliance status of this feature, as calculated by FlexNet Manager for Engineering Applications.
FNMEAFeatureID	Type: integer. Nullable The identifier of the FlexNet Manager for Engineering Applications feature in the <code>FNMEAFeature</code> table that this imported FlexNet Manager for Engineering Applications feature links to. This is populated by the import process and does not need to be provided by the source connections.

## ImportedFNMEAProduct Table

The `ImportedFNMEAProduct` table is used by the importer to import FlexNet Manager for Engineering Applications products.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 578: Database columns for `ImportedFNMEAProduct` table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key The identifier of a data source connection in the <code>ComplianceConnection</code> table.
ExternalID	Type: text (max 256 characters). Key The identifier of the product from the external data source. This is the product number in FlexNet Manager for Engineering Applications.

Database Column	Details
FeatureID	<i>Type:</i> integer. Key The identifier (from the external data source) of the feature this product is associated with.
Name	<i>Type:</i> text (max 256 characters) The name for this product.
Version	<i>Type:</i> text (max 32 characters). Key The version of this product.
VendorDaemon	<i>Type:</i> text (max 256 characters). Key The vendor daemon of the products feature.
Publisher	<i>Type:</i> text (max 256 characters) The publisher of the product.
PurchasedQuantity	<i>Type:</i> integer The count of the products purchased.
OutOfComplianceQuantity	<i>Type:</i> integer The count of out-of-compliance product installs, as calculated by FlexNet Manager for Engineering Applications.
ComplianceStatus	<i>Type:</i> text (max 32 characters) The compliance status of this feature, as calculated by FlexNet Manager for Engineering Applications.
FeatureQuantity	<i>Type:</i> integer The count of the features available per product purchased.
SoftwareLicenseID	<i>Type:</i> integer. Nullable The identifier of the software license in the <code>SoftwareLicense</code> table that this imported FlexNet Manager for Engineering Applications product links to. This is populated by the import process and does not need to be provided by the source connections.

## ImportedFNMEAUsageStatus Table

The `ImportedFNMEAUsageStatus` table is used by the importer to import FlexNet Manager for Engineering Applications status values.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 579: Database columns for ImportedFNMEAUsageStatus table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key The identifier of a data source connection in the ComplianceConnection table.
ProductNumber	Type: text (max 256 characters). Key The identifier of the product from the external data source. This is the product number in FlexNet Manager for Engineering Applications.
Month	Type: integer The month of the usage for this product.
Year	Type: integer The year of the usage of this product.
HWMUsage	Type: integer The high water mark usage of this product.

## ImportedFileEvidence Table

The ImportedFileEvidence table holds all of the file evidence which has been retrieved from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 580: Database columns for ImportedFileEvidence table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key. Nullable

Database Column	Details
	The identifier for a data source connection in the <code>ComplianceConnection</code> table.
<code>ExternalFileID</code>	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the file evidence.
<code>FileName</code>	<i>Type:</i> text (max 256 characters). Key. Nullable The name of the file used as evidence of software installation.
<code>FileVersion</code>	<i>Type:</i> text (max 100 characters). Nullable The version number of the file used as evidence of software installation.
<code>ProductVersion</code>	<i>Type:</i> text (max 200 characters). Nullable The product version number in the file header.
<code>ProductName</code>	<i>Type:</i> text (max 200 characters). Nullable The product name in the file header.
<code>FilePath</code>	<i>Type:</i> text (max 400 characters). Nullable The path of the file used as evidence of software installation.
<code>Company</code>	<i>Type:</i> text (max 100 characters). Key. Nullable The company in the file header.
<code>Description</code>	<i>Type:</i> text (max 200 characters) The description in the file header.
<code>FileSize</code>	<i>Type:</i> integer. Nullable The size of the file.
<code>Language</code>	<i>Type:</i> text (max 200 characters). Nullable The language in the file header.
<code>AccessModeID</code>	<i>Type:</i> integer. Key. Nullable The access mode ID of the file evidence.

## ImportedFileEvidenceMapping Table

The `ImportedFileEvidenceMapping` table is used by the importer to link imported file evidence with evidence in the `FileEvidence` table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 581: Database columns for `ImportedFileEvidenceMapping` table**

Database Column	Details
<code>FileEvidenceID</code>	<p>Type: integer. Key. Nullable</p> <p>The identifier for the file evidence in the <code>NewFileEvidence</code> table.</p>
<code>ExternalFileID</code>	<p>Type: big integer. Key. Nullable</p> <p>The identifier used in the source connection for the imported file evidence.</p>
<code>ComplianceConnectionID</code>	<p>Type: integer. Key. Nullable</p> <p>The identifier of a data source connection in the <code>ComplianceConnection</code> table.</p>

## ImportedGuidMapping Table

The `ImportedGuidMapping` table is used by the importer to keep a history of entities that have been imported from a data source that uses GUID IDs rather than integer IDs.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 582: Database columns for `ImportedGuidMapping` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<p>Type: integer. Key</p> <p>The identifier of a data source connection in the <code>ComplianceConnection</code> table.</p>
<code>Category</code>	<p>Type: text (max 100 characters). Key</p> <p>The importer category applicable for this ID space.</p>
<code>OriginalID</code>	<p>Type: unique identifier. Key</p>

Database Column	Details
	The ID of this entity in the source database.
MappedID	<p>Type: big integer. Generated ID</p> <p>A unique integer value we can use as an 'external ID' safely in the <code>ImportedComputer</code> table.</p>

## ImportedILMTPVUCounts Table

This table allows the summarised PVU sub capacity numbers to be imported from ILMT. These numbers are calculated by ILMT for a particular date range as PVU "reports".



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 583: Database columns for ImportedILMTPVUCounts table**

Database Column	Details
ExternalNodeID	<p>Type: big integer. Key</p> <p>The external ID of the server to which these points apply.</p>
ExternalVMID	<p>Type: big integer. Key. Nullable</p> <p>The external ID of the virtual machine associated with the node (server).</p>
ComplianceConnectionID	<p>Type: integer. Key</p> <p>The current connection ID for this data source.</p>
TitleName	<p>Type: text (max 512 characters). Key</p> <p>The name of the title these points apply to.</p>
Publisher	<p>Type: text (max 254 characters). Key</p> <p>The name of the publisher of the title these points apply to.</p>
SubCapacityCores	<p>Type: integer</p> <p>The number of sub-capacity licensable cores for the license on the computer.</p>
FullCapacityCores	<p>Type: integer</p> <p>The number of full-capacity licensable cores for the license on the computer.</p>

Database Column	Details
SubCapacityPVU	<i>Type:</i> integer The number of sub-capacity PVU counts consumed for the license on the computer.
FullCapacityPVU	<i>Type:</i> integer The number of full-capacity PVU counts consumed for the license on the computer.
PeakSubCapacityPVU	<i>Type:</i> integer The peak number of sub-capacity PVU counts consumed for the license on the computer.
PeakFullCapacityPVU	<i>Type:</i> integer The peak number of full-capacity PVU counts consumed for the license on the computer.

## ImportedILMTPVUCreatedLicenses Table

This table stores a history of IBM PVU licenses that have been created by the ILMT adapter.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 584: Database columns for ImportedILMTPVUCreatedLicenses table**

Database Column	Details
SoftwareLicenseID	<i>Type:</i> integer The ID of the created license.
TitleName	<i>Type:</i> text (max 512 characters) The name of the title that triggered the creation of the license.
Publisher	<i>Type:</i> text (max 254 characters) The name of the publisher of the title that triggered the creation of the license.

## ImportedILMTVMMapping Table

The `ImportedILMTVMMapping` table is used by the importer to keep a history of all Virtual Machine IDs (adm.VM records) that have been imported from ILMT data sources.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 585: Database columns for `ImportedILMTVMMapping` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key The identifier of a data source connection in the <code>ComplianceConnection</code> table.
<code>OriginalID</code>	<i>Type:</i> big integer. Key The agent ID of this agent in the ILMT database.
<code>MappedID</code>	<i>Type:</i> integer. Generated ID A unique integer value we can use as an 'external ID' safely in the <code>ImportedComputer</code> table.

## ImportedInstalledFileEvidence Table

The `ImportedInstalledFileEvidence` table holds a record of the file evidence that has been installed on a computer from the source connections.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 586: Database columns for `ImportedInstalledFileEvidence` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key. Nullable



Database Column	Details
	The identifier for a data source connection in the <code>ComplianceConnection</code> table.
<code>ExternalID</code>	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the computer that the file evidence is installed on.
<code>ExternalFileID</code>	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the file evidence.
<code>ExternalFilePathID</code>	<i>Type:</i> big integer. Nullable The identifier used in the source connection for the path of the file evidence.

## ImportedInstalledFileEvidenceUsage Table

The `ImportedInstalledFileEvidenceUsage` table holds a record of end-users that are using file evidence from the source connection.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 587: Database columns for `ImportedInstalledFileEvidenceUsage` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
<code>StartDate</code>	<i>Type:</i> text (max 10 characters). Nullable The start date of the file evidence usage tracking period.
<code>ExternalID</code>	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the computer that the file evidence is installed on.
<code>ExternalUserID</code>	<i>Type:</i> big integer. Key. Nullable

Database Column	Details
	The identifier used in the source connection for the end-user that has used the file evidence.
ExternalFileID	Type: big integer. Key. Nullable The identifier used in the source connection for the file evidence.
ActiveTimeInSeconds	Type: big integer. Nullable The number of seconds that the file evidence was in use during the usage tracking period.
NumberOfSessions	Type: big integer. Nullable The number of sessions that the file evidence was in use during the usage tracking period.
LastUsedDate	Type: text (max 10 characters). Nullable The last used date of the file evidence.

## ImportedInstalledInstallerEvidence Table

The `ImportedInstalledInstallerEvidence` table holds a record of the installer evidence that has been installed on a computer from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 588: Database columns for ImportedInstalledInstallerEvidence table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ExternalInstallerEvidenceID	Type: big integer. Key. Nullable The identifier used in the source connection for the installer evidence.
ExternalComputerID	Type: big integer. Key. Nullable

Database Column	Details
	The identifier used in the source connection for the computer that the installer evidence is installed on.
ExternalInstanceID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the instance that the installer evidence is associated with.
InstallDate	<i>Type:</i> text (max 10 characters). Nullable The install date of the installer evidence.
DiscoveryDate	<i>Type:</i> text (max 10 characters). Nullable The date that the installer evidence was first seen.

## ImportedInstalledInstallerEvidenceAttribute Table

The `ImportedInstalledInstallerEvidenceAttribute` table holds a record of the values of the instance attributes for each installer evidence which is reported to be installed on a computer.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 589: Database columns for `ImportedInstalledInstallerEvidenceAttribute` table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ExternalInstallerEvidenceID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the installer evidence.
ExternalComputerID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the computer that the installer evidence is installed on.
ExternalInstanceID	<i>Type:</i> big integer. Key. Nullable

Database Column	Details
	The identifier used in the source connection for the instance that the installer evidence is associated with.
AttributeID	<i>Type:</i> integer. Key The identifier used in the source connection for the instance attribute.
Value	<i>Type:</i> text The value of the instance attribute for the installed installer evidence.

## ImportedInstalledInstallerEvidenceUsage Table

The `ImportedInstalledInstallerEvidenceUsage` table holds a record of installed evidence being used from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 590: Database columns for `ImportedInstalledInstallerEvidenceUsage` table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
StartDate	<i>Type:</i> text (max 10 characters). Nullable The start date of the installer evidence usage tracking period.
ExternalID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the computer that the installer evidence is installed on.
ExternalInstallerID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the installer evidence.
ExternalInstanceID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the instance that the installer evidence is associated with.

Database Column	Details
NumberOfSessions	<i>Type:</i> big integer. Nullable The number of sessions that the installer evidence was in use during the usage tracking period.
LastUsedDate	<i>Type:</i> text (max 10 characters). Nullable The last used date of the installed installer evidence.
ExternalUserID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the user that the installer evidence was used on.

## ImportedInstalledWMIEvidence Table

The `ImportedInstalledWMIEvidence` table holds a record of the WMI evidence that has been installed on a computer from the source connections.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 591: Database columns for ImportedInstalledWMIEvidence table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ExternalComputerID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the computer that the WMI evidence is installed on.
ExternalEvidenceID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the WMI evidence.
InstanceName	<i>Type:</i> text (max 256 characters). Key. Nullable The name of the WMI class instance used in the source connection for the WMI evidence

# ImportedInstallerEvidence Table

The `ImportedInstallerEvidence` table holds all of the installer evidence which has been retrieved from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 592: Database columns for `ImportedInstallerEvidence` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The identifier for a data source connection in the <code>ComplianceConnection</code> table.</p>
<code>ExternalInstallerID</code>	<p><i>Type:</i> big integer. Key. Nullable</p> <p>The identifier used in the source connection for the installer evidence.</p>
<code>DisplayName</code>	<p><i>Type:</i> text (max 256 characters). Key. Nullable</p> <p>The display name of the software as reported by the installer evidence.</p>
<code>Version</code>	<p><i>Type:</i> text (max 72 characters). Key. Nullable</p> <p>The version of the software as reported by the installer evidence.</p>
<code>Publisher</code>	<p><i>Type:</i> text (max 200 characters). Key. Nullable</p> <p>The publisher of the software as reported by the installer evidence.</p>
<code>Evidence</code>	<p><i>Type:</i> text (max 32 characters). Nullable</p> <p>Identifier for the type of installer evidence.</p>
<code>ProductCode</code>	<p><i>Type:</i> text (max 55 characters). Nullable</p> <p>The product code of the evidence. This is usually the MSI product code.</p>
<code>AccessModeID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The access mode ID of the file evidence.</p>

## ImportedInstallerEvidenceMapping Table

The `ImportedInstallerEvidenceMapping` table is used by the importer to link imported installer evidence with evidence in the `InstallerEvidence` table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 593: Database columns for `ImportedInstallerEvidenceMapping` table**

Database Column	Details
<code>InstallerEvidenceID</code>	Type: integer. Key. Nullable The identifier for the installer evidence in the <code>InstallerEvidence</code> table.
<code>ExternalInstallerID</code>	Type: big integer. Key. Nullable The identifier used in the source connection for the imported installer evidence.
<code>ComplianceConnectionID</code>	Type: integer. Key. Nullable The identifier of a data source connection in the <code>ComplianceConnection</code> table.

## ImportedInstallerEvidenceRepackageMapping Table

The `ImportedInstallerEvidenceRepackageMapping` table is used by the importer to map the original and current installer evidence of repackaged softwares as reported by the ISO tag evidence.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 594: Database columns for `ImportedInstallerEvidenceRepackageMapping` table**

Database Column	Details
<code>ComplianceConnectionID</code>	Type: integer. Key. Nullable The identifier of a data source connection in the <code>ComplianceConnection</code> table.

Database Column	Details
OrigDisplayName	<i>Type:</i> text (max 256 characters). Key. Nullable The original display name of the repackaged software as reported by the ISO tag evidence.
OrigVersion	<i>Type:</i> text (max 72 characters). Key. Nullable The original version of the repackaged software as reported by the ISO tag evidence.
OrigPublisher	<i>Type:</i> text (max 200 characters). Key. Nullable The original publisher of the repackaged software as reported by the ISO tag evidence.
CurrentDisplayName	<i>Type:</i> text (max 256 characters). Key. Nullable The current display name of the repackaged software as reported by the ISO tag evidence.
CurrentVersion	<i>Type:</i> text (max 72 characters). Key. Nullable The current version of the repackaged software as reported by the ISO tag evidence.
CurrentPublisher	<i>Type:</i> text (max 200 characters). Key. Nullable The current publisher of the repackaged software as reported by the ISO tag evidence.

## ImportedInstance Table

The `ImportedInstance` table holds all of the instances which have been retrieved from the source connections.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 595: Database columns for `ImportedInstance` table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable



Database Column	Details
	The identifier for a data source connection in the <code>ComplianceConnection</code> table.
<code>InstanceID</code>	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the instance.
<code>InstanceName</code>	<i>Type:</i> text (max 256 characters). Nullable The name of the instance.
<code>ParentInstanceID</code>	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the parent instance.
<code>ExternalComputerID</code>	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the computer.
<code>AuditEvidence</code>	<i>Type:</i> binary. Nullable Oracle LMS CVS files in zip archive.
<code>AuditEvidenceDate</code>	<i>Type:</i> datetime. Nullable Oracle LMS CSV files collection date.

## ImportedInstanceUser Table

The `ImportedInstanceUser` table holds all of the end-users of an instance which have been retrieved from the source connections.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 596: Database columns for `ImportedInstanceUser` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
<code>ExternalID</code>	<i>Type:</i> big integer. Key

Database Column	Details
	The identifier used in the source connection for the instance end-user.
ComputerID	<i>Type:</i> big integer. Key The identifier used in the source connection for the computer.
InstanceID	<i>Type:</i> big integer. Key The identifier used in the source connection for the instance.
AccountStatus	<i>Type:</i> text (max 256 characters). Nullable The current status of the end-user account.
CreationDate	<i>Type:</i> datetime. Nullable The date and time when the end-user was created.
LastLogonDate	<i>Type:</i> datetime. Nullable The date and time when the end-user last logged on to the computer.
DefaultTablespace	<i>Type:</i> text (max 256 characters). Nullable The default tablespace for an Oracle end-user.
TempTablespace	<i>Type:</i> text (max 256 characters). Nullable The temporary tablespace for an Oracle end-user.
ApplicationID	<i>Type:</i> text (max 400 characters). Key. Nullable The Oracle EBS application ID the user has access to.

## ImportedMissingComputer Table

The `ImportedMissingComputer` table holds all of the computers which no longer have inventory records in the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 597: Database columns for ImportedMissingComputer table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ExternalID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the computer.
ComplianceComputerID	<i>Type:</i> integer. Key. Nullable Identifier of the computer in the <code>ComplianceComputer</code> table that this imported computer links to.

## ImportedMissingLicenseUser Table

The `ImportedMissingLicenseUser` table holds all of the external end-users which no longer have inventory records in the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 598: Database columns for ImportedMissingLicenseUser table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ExternalID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the external end-user.
LicenseUserID	<i>Type:</i> integer. Key. Nullable The identifier for the external end-user in the <code>LicenseUser</code> table.

## ImportedMissingUser Table

The `ImportedMissingUser` table holds all of the end-users which no longer have inventory records in the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 599: Database columns for `ImportedMissingUser` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<p>Type: integer. Key. Nullable</p> <p>The identifier for a data source connection in the <code>ComplianceConnection</code> table.</p>
<code>ExternalID</code>	<p>Type: big integer. Key. Nullable</p> <p>The identifier used in the source connection for the end-user.</p>
<code>ComplianceUserID</code>	<p>Type: integer. Key. Nullable</p> <p>The identifier for the end-user in the <code>ComplianceUser</code> table.</p>

## ImportedProductCodeEvidenceMapping Table

The `ImportedProductCodeEvidenceMapping` table is used by the importer to link imported product code evidence with evidence in the `InstallerEvidence` table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 600: Database columns for `ImportedProductCodeEvidenceMapping` table**

Database Column	Details
<code>InstallerEvidenceID</code>	<p>Type: integer. Key. Nullable</p> <p>The identifier for the installer evidence in the <code>InstallerEvidence</code> table.</p>
<code>ExternalInstallerID</code>	<p>Type: big integer. Key. Nullable</p>

Database Column	Details
	The identifier used in the source connection for the imported installer evidence.
ComplianceConnectionID	<p>Type: integer. Key. Nullable</p> <p>The identifier of a data source connection in the <code>ComplianceConnection</code> table.</p>

## ImportedRelatedInstalledInstallerEvidence Table

The `ImportedRelatedInstalledInstallerEvidence` table holds parent-child relationship between installer evidence.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 601: Database columns for `ImportedRelatedInstalledInstallerEvidence` table**

Database Column	Details
ComplianceConnectionID	<p>Type: integer. Key. Nullable</p> <p>The identifier for a data source connection in the <code>ComplianceConnection</code> table.</p>
ParentExternal InstallerEvidenceID	<p>Type: big integer. Key. Nullable</p> <p>The identifier used in the source connection for the installer evidence.</p>
ParentExternalComputerID	<p>Type: big integer. Key. Nullable</p> <p>The identifier used in the source connection for the computer that the installer evidence is installed on.</p>
ChildExternalInstaller EvidenceID	<p>Type: big integer. Key. Nullable</p> <p>The identifier used in the source connection for the installer evidence.</p>
ChildExternalComputerID	<p>Type: big integer. Key. Nullable</p> <p>The identifier used in the source connection for the computer that the installer evidence is installed on.</p>
IsCharged	Type: boolean. Key. Nullable

Database Column	Details
	The identifier used in the source connection to determine the pricing relation between parent and child installer evidence (specifies if it is charged = 1 or free = 0).
ConfidenceLevel	Type: integer. Nullable Confidence level for each bundled installer evidence (as a percentage).

## ImportedRemoteApplication Table

This `ImportedRemoteApplication` table stores all the published applications from Citrix XenApp/App-V Management Server.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 602: Database columns for ImportedRemoteApplication table**

Database Column	Details
FarmName	Type: text (max 256 characters). Nullable The farm from which the application belongs to.
AppID	Type: text (max 256 characters). Key. Nullable The unique identifier for XenApp applications.
AppName	Type: text (max 256 characters). Nullable The application name available in XenApp.
AppFileName	Type: text (max 256 characters). Key. Nullable The application executable name.
AppFileVersion	Type: text (max 256 characters). Key. Nullable The application executable version.
AppFilePublisher	Type: text (max 256 characters). Key. Nullable The application publisher.
AppFileDescription	Type: text (max 256 characters). Key. Nullable

Database Column	Details
	The application description.
IsStreamingProfile	Type: boolean. Nullable Whether the application is a streaming profile.
AccessModeID	Type: integer. Key The access mode of the virtual application.

## ImportedRemoteApplicationAccess Table

This `ImportedRemoteApplicationAccess` table stores all users/groups with sid who have access to what virtual applications.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 603: Database columns for ImportedRemoteApplicationAccess table**

Database Column	Details
FarmName	Type: text (max 256 characters). Nullable The farm from which the virtual application belongs to.
AppID	Type: text (max 256 characters). Nullable The unique identifier for virtual applications.
Sid	Type: text (max 256 characters). Nullable The sid that has access to the application.
AccessModeID	Type: integer The access mode of the virtual application.

## ImportedRemoteApplicationInstallerData Table

This `ImportedRemoteApplicationInstallerData` table stores all the MSI information in a streamed profile.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 604: Database columns for ImportedRemoteApplicationInstallerData table**

Database Column	Details
FarmName	Type: text (max 256 characters). Nullable The farm from which the application belongs to.
AppID	Type: text (max 256 characters). Key. Nullable The unique identifier for virtual applications.
DisplayName	Type: text (max 256 characters). Key. Nullable The application name.
Publisher	Type: text (max 200 characters). Key. Nullable The application publisher name.
Version	Type: text (max 72 characters). Key. Nullable The application version.
ProductCode	Type: text (max 55 characters). Nullable The product code of the evidence. This is usually the MSI product code.
AccessModeID	Type: integer. Key The access mode of the virtual application.

## ImportedRemoteApplicationServer Table

This ImportedRemoteApplicationServer table stores the servers from which applications are published from.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.



**Table 605: Database columns for ImportedRemoteApplicationServer table**

Database Column	Details
FarmName	Type: text (max 256 characters). Nullable The farm from which the server belongs to.
AppID	Type: text (max 256 characters). Key. Nullable The unique identifier for XenApp applications.
ServerName	Type: text (max 256 characters). Key. Nullable The XenApp server the application is available under.
ServerDomainName	Type: text (max 256 characters). Key. Nullable The XenApp server domain name.
VDIGroupUUID	Type: unique identifier. Nullable The desktop group UUID from which the application is published
AccessModeID	Type: integer. Key The access mode of the virtual application.

## ImportedRemoteServerFileEvidenceMapping Table

The ImportedRemoteServerFileEvidenceMapping table stores the mapping between file evidence on servers to software titles



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 606: Database columns for ImportedRemoteServerFileEvidenceMapping table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key. Nullable The identifier for a data source connection in the ComplianceConnection table.
ExternalServerID	Type: big integer. Key. Nullable

Database Column	Details
	The External Server ID for the remote server.
ExternalFileID	Type: big integer. Key. Nullable The identifier used in the source connection for the file evidence.
SoftwareTitleID	Type: integer. Nullable The software title ID corresponding to the piece of file evidence.

## ImportedRemoteUsage Table

This ImportedRemoteUsage table stores the remote usage for applications in remote hosting environments



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 607: Database columns for ImportedRemoteUsage table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key. Nullable The identifier for a data source connection in the ComplianceConnection table.
ExternalServerID	Type: big integer. Key. Nullable The External Server ID for the remote server.
ExternalClientID	Type: big integer. Nullable The External client ID for the remote client machine.
ExternalFileID	Type: big integer. Key. Nullable The identifier used in the source connection for the file evidence.
ExternalInstallerEvidenceID	Type: big integer. Nullable The identifier used in the source connection for the installer evidence.
ExternalUserID	Type: big integer. Nullable The identifier used in the source connection for the end-user that has used the file evidence.

Database Column	Details
StartDate	<i>Type:</i> text (max 10 characters). Nullable The start date of the remote usage tracking period.
ActiveTimeInSeconds	<i>Type:</i> big integer. Nullable The number of seconds that the file evidence was in use during the usage tracking period.
NumberOfSessions	<i>Type:</i> big integer. Nullable The number of sessions that the file evidence was in use during the usage tracking period.
AccessModeID	<i>Type:</i> integer. Nullable The access mode ID for the remote usage.

## ImportedRemoteUserToApplicationAccess Table

The `ImportedRemoteUserToApplicationAccess` table stores the applications that remote users have access to



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 608: Database columns for `ImportedRemoteUserToApplicationAccess` table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ExternalServerID	<i>Type:</i> big integer. Key. Nullable The External Server ID for the remote server.
VDIGroupUUID	<i>Type:</i> unique identifier. Nullable The desktop group UUID from which the application is published
ExternalFileID	<i>Type:</i> big integer. Key. Nullable

Database Column	Details
	The identifier used in the source connection for the file evidence.
ExternalInstallerEvidenceID	Type: big integer. Key. Nullable The identifier used in the source connection for the installer evidence.
ExternalUserID	Type: big integer. Key. Nullable The identifier used in the source connection for the end-user that has used the file evidence.
AccessModeID	Type: integer. Key. Nullable The access mode ID for the remote application access.

## ImportedSite Table

The `ImportedSubnet` contains sites imported from Microsoft Active Directory



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 609: Database columns for ImportedSite table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key The identifier for a data source connection in the <code>ComplianceConnection</code> table.
Name	Type: text (max 256 characters). Key The site's name.
AutoPopulated	Type: boolean Is the site auto populated at source?
Enabled	Type: boolean Is the site enabled?

## ImportedSiteSubnet Table

The `ImportedSiteSubnet` contains sites and subnets imported from Microsoft Active Directory



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 610: Database columns for `ImportedSiteSubnet` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key The identifier for a data source connection in the <code>ComplianceConnection</code> table.
<code>SiteName</code>	<i>Type:</i> text (max 256 characters). Key The site's name.
<code>IPSubnet</code>	<i>Type:</i> text (max 64 characters). Key The IP subnet.
<code>IPSubnetBits</code>	<i>Type:</i> tiny integer. Key The IP subnet mask in CIDR notation.
<code>AutoPopulated</code>	<i>Type:</i> boolean Is the subnet auto populated at source?
<code>Enabled</code>	<i>Type:</i> boolean Is the subnet enabled?

## ImportedStringMapping Table

The `ImportedStringMapping` table is used by the importer to keep a history of entities that have been imported from a data source that uses string IDs rather than integer IDs.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 611: Database columns for ImportedStringMapping table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key The identifier of a data source connection in the <code>ComplianceConnection</code> table.
Category	<i>Type:</i> text (max 100 characters). Key The importer category applicable for this ID space.
OriginalID	<i>Type:</i> text (max 400 characters). Key The ID of this entity in the source database.
MappedID	<i>Type:</i> big integer. Generated ID A unique integer value we can use as an 'external ID' safely in the <code>ImportedComputer</code> table.

## ImportedStringMappingLatin1CS Table

The `ImportedStringMappingLatin1CS` table is used by the importer to keep a history of entities that have been imported from a data source that uses case sensitive string IDs rather than integer IDs.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 612: Database columns for ImportedStringMappingLatin1CS table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key The identifier of a data source connection in the <code>ComplianceConnection</code> table.
Category	<i>Type:</i> text (max 100 characters). Key The importer category applicable for this ID space.
OriginalID	<i>Type:</i> text (max 400 characters). Key The ID of this entity in the source database.

Database Column	Details
MappedID	<p>Type: big integer. Generated ID</p> <p>A unique integer value we can use as an 'external ID' safely in the <code>ImportedComputer</code> table.</p>

## ImportedUser Table

The `ImportedUser` table holds all of the end-users which have been retrieved from the source connections.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 613: Database columns for `ImportedUser` table**

Database Column	Details
ComplianceConnectionID	<p>Type: integer. Key. Nullable</p> <p>The identifier for a data source connection in the <code>ComplianceConnection</code> table.</p>
ExternalID	<p>Type: big integer. Key. Nullable</p> <p>The identifier used in the source connection for the end-user.</p>
UserName	<p>Type: text (max 64 characters). Nullable</p> <p>The account name of the end-user.</p>
Domain	<p>Type: text (max 100 characters). Key. Nullable</p> <p>The domain of the end-user.</p>
SAMAccountName	<p>Type: text (max 64 characters). Key. Nullable</p> <p>The SAM account name of the end-user.</p>
InventoryAgent	<p>Type: text (max 64 characters). Nullable</p> <p>The name of the person or tool that performed the last inventory. For imported spreadsheets, you may wish to include the name of the person preparing the data, in case there is subsequent follow-up required.</p>
FirstName	<p>Type: text (max 128 characters). Nullable</p>

Database Column	Details
	The first name of the end-user.
LastName	Type: text (max 128 characters). Nullable The last name or surname of the end-user.
Email	Type: text (max 200 characters). Nullable The email address of the end-user.
EmployeeNumber	Type: text (max 128 characters). Nullable The employee number of the end-user.
CostCenter	Type: text (max 128 characters). Nullable The cost center of the end-user, as reported in SAP. Does not necessarily map to a cost centre in the <code>GroupEx</code> table.
ComplianceUserID	Type: integer. Nullable Identifier of the end-user in the <code>ComplianceUser</code> table that this imported user links to. This is populated by the import process and does not need to be provided by the source connections.
ComplianceDomainID	Type: integer. Nullable Identifier of the domain in the <code>ComplianceDomain</code> table that this end-user belongs to. This is populated by the import process and does not need to be provided by the source connections.
IsBlacklisted	Type: boolean. Key This is populated by the import process and does not need to be provided by the source connections. The field is set to <code>True</code> if the end-user matches a record from the <code>UserNameBlacklist</code> table, meaning the account should not be included in compliance calculations.

## ImportedVDI Table

The `ImportedVDIUser` table stores the list of VDI devices, their master VM template and the VDI group the VDI device resides under.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.



**Table 614: Database columns for ImportedVDI table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer The identifier of a data source connection in the <code>ComplianceConnection</code> table.
ExternalDeviceID	<i>Type:</i> big integer. Nullable The identifier used in the source connection for the VDI device.
ComputerName	<i>Type:</i> text (max 64 characters). Nullable The computer name of the VDI.
Domain	<i>Type:</i> text (max 100 characters). Nullable The domain name of the VDI device.
VDIGroupName	<i>Type:</i> text (max 100 characters). Key. Nullable The VDI group the VDI device belongs to.
TemplateName	<i>Type:</i> text (max 100 characters). Key. Nullable The VDI template the VDI is cloned from.
SiteName	<i>Type:</i> text (max 256 characters). Key. Nullable The site name of the VDI.
BrokerType	<i>Type:</i> text (max 64 characters). Key. Nullable The broker type of the VDI device.
IsPersistent	<i>Type:</i> boolean. Key. Nullable Determine whether the VDI device is a persistent VDI device.
VDIGroupUUID	<i>Type:</i> unique identifier. Nullable The group UUID the VDI device belongs to.
ApplicationDeliveryOnly	<i>Type:</i> boolean. Nullable Determines whether the VDI device is used only to server applications.

## ImportedVDIEndPointAccess Table

The `ImportedVDIEndPointAccess` table stores the list of users on end-points that have accessed VDI devices.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 615: Database columns for ImportedVDIEndPointAccess table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Nullable The identifier of a data source connection in the <i>ComplianceConnection</i> table.
ExternalDeviceID	<i>Type:</i> big integer. Nullable The identifier used in the source connection for the device.
ExternalUserID	<i>Type:</i> big integer. Nullable The identifier used in the source connection for the user.
VDIDeviceName	<i>Type:</i> text (max 64 characters). Nullable The computer name of the VDI device.
VDIDeviceDomain	<i>Type:</i> text (max 100 characters). Nullable The domain name of the VDI device.
VDITemplateName	<i>Type:</i> text (max 256 characters). Nullable The VDI template the VDI device was cloned from.
LogonTime	<i>Type:</i> datetime. Key. Nullable The logon time of the VDI device by the user.
BrokerType	<i>Type:</i> text (max 64 characters). Nullable The broker type of the VDI device.

## ImportedVDITemplate Table

The *ImportedVDITemplate* table stores the list of VDI templates.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 616: Database columns for ImportedVDITemplate table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer The identifier of a data source connection in the <i>ComplianceConnection</i> table.
TemplateName	<i>Type:</i> text (max 64 characters). Key. Nullable The template name of the VDI template.
SiteName	<i>Type:</i> text (max 256 characters). Key. Nullable The site name of the VDI.
BrokerType	<i>Type:</i> text (max 64 characters). Key. Nullable The broker type of the VDI template.
VDITemplateExternalID	<i>Type:</i> big integer. Nullable The ExternalID of the VDI template in the <i>ImportedComputer</i> table.

## ImportedVDIUser Table

The *ImportedVDIUser* table stores the list of users that have been granted access to VDI groups.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 617: Database columns for ImportedVDIUser table**

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key The identifier of a data source connection in the <i>ComplianceConnection</i> table.
ExternalUserID	<i>Type:</i> big integer. Key. Nullable

Database Column	Details
	The identifier used in the source connection for the end-user that has access to the VDI.
VDIGroupName	Type: text (max 100 characters). Nullable The VDI group the end-user has access to.
SiteName	Type: text (max 256 characters). Nullable The site name of the VDI.
BrokerType	Type: text (max 64 characters). Nullable The broker type of the VDI for the end user.

## ImportedVMHostManagedBySoftware Table

The `ImportedVMHostManagedBySoftware` table contains relationships between installer evidence of management software and VM hosts it manages.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 618: Database columns for `ImportedVMHostManagedBySoftware` table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ExternalInstallerID	Type: big integer. Key The identifier used in the source connection for an installer evidence of management software.
ExternalComputerID	Type: big integer. Key The identifier used in the source connection for the computer that the management software installer evidence is installed on.
RelationType	Type: text (max 100 characters). Key

Database Column	Details
	Identifier for the type of relation, to be matched against ImporterString column of RelationType table.
ExternalVMHostID	<p>Type: big integer. Key</p> <p>The identifier used in the source connection for the VM host computer that is managed by a management software.</p>

## ImportedVMPool Table

The `ImportedVMPool` table holds all of the virtual machine pools which have been retrieved from the source connections and the number of processors and cores that are assigned to each pool.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 619: Database columns for ImportedVMPool table**

Database Column	Details
PoolName	<p>Type: text (max 100 characters). Key. Nullable</p> <p>The name of the pool.</p>
VCOBJECTID	<p>Type: text (max 256 characters). Nullable</p> <p>The identifier of the virtual machine folder in Virtual Center.</p>
ParentName	<p>Type: text (max 100 characters). Nullable</p> <p>The name of the parent pool. This is the PoolName property for the parent pool.</p>
PoolFriendlyName	<p>Type: text (max 256 characters). Nullable</p> <p>The friendly name of the pool.</p>
HostComputerID	<p>Type: big integer. Key. Nullable</p> <p>The identifier used in the source connection for the computer which is hosting the pool.</p>
ObjectType	<p>Type: text (max 256 characters). Key. Nullable</p> <p>The type of pool.</p>

Database Column	Details
ComplianceConnectionID	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ParentObjectType	<i>Type:</i> text (max 256 characters). Nullable The type of pool of the parent.
NumberOfProcessors	<i>Type:</i> decimal. Nullable The number of processors available to this pool.
NumberOfCores	<i>Type:</i> decimal. Nullable The number of cores available to this pool.

## ImportedVirtualMachine Table

The `ImportedVirtualMachine` table holds all of the virtual machines which have been retrieved from the source connections.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 620: Database columns for `ImportedVirtualMachine` table**

Database Column	Details
HostComputerID	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the virtual machine's host computer.
VirtualMachineType	<i>Type:</i> text (max 100 characters). Nullable The type of virtual machine.
VMName	<i>Type:</i> text (max 256 characters). Nullable The name of the virtual machine.
VCOBJECTID	<i>Type:</i> text (max 256 characters). Nullable The identifier of the virtual machine in Virtual Center.

Database Column	Details
FriendlyName	<i>Type:</i> text (max 256 characters). Nullable The friendly name of the virtual machine.
ComputerName	<i>Type:</i> text (max 256 characters). Nullable The computer name of the virtual machine.
UUID	<i>Type:</i> text (max 256 characters). Key. Nullable The UUID of the virtual machine.
TotalMemory	<i>Type:</i> big integer. Nullable The total RAM in the computer, in bytes.
PoolName	<i>Type:</i> text (max 100 characters). Nullable The name of the pool that the virtual machine belongs to.
CPUUsage	<i>Type:</i> integer. Nullable The maximum CPU usage of the virtual machine (MHz).
MemoryUsage	<i>Type:</i> big integer. Nullable The maximum memory usage of the virtual machine (bytes).
VMEabledStateID	<i>Type:</i> integer. Nullable The state of the machine (powered on, off, etc).
ModelNo	<i>Type:</i> text (max 128 characters). Nullable The model number of the virtual machine.
Manufacturer	<i>Type:</i> text (max 128 characters). Nullable The manufacturer of the computer hardware. Some examples include: <ul style="list-style-type: none"> <li>On Windows, the SMBios manufacturer (the WMI Manufacturer property of the 'Win32_ComputerSystem' class).</li> <li>On Linux, 'Manufacturer' in the 'System Information' section resulting from the 'dmidecode' command. Sample command: 'dmidecode -s system-manufacturer'</li> <li>On Solaris x86, as for Linux, with failovers first to 'sysinfo SI_HW_PROVIDER' and then to 'ModelNo'.</li> <li>On Solaris SPARC, the 'sysinfo SI_HW_PROVIDER'. Typically this value is 'Sun_Microsystems' or, more recently, 'Oracle Corporation'. Failover to the 'ModelNo'.</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>On HP-UX, the string literal 'HP'.</li> <li>On AIX, the 'modelname' system attribute preceding the comma character. For example, if the 'modelname' system attribute is 'IBM,8202-E4B', then use 'IBM'. This value is typically 'IBM'.</li> </ul>
NumberOfProcessors	<p><i>Type:</i> integer. Nullable</p> <p>The number of processors in the virtual machine.</p>
ProcessorType	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The type of processor in the virtual machine.</p>
NumberOfHardDrives	<p><i>Type:</i> integer. Nullable</p> <p>The number of hard drives in the virtual machine.</p>
NumberOfNetworkCards	<p><i>Type:</i> integer. Nullable</p> <p>The number of network cards in the virtual machine.</p>
InventoryAgent	<p><i>Type:</i> text (max 64 characters). Nullable</p> <p>The name of the person or tool that performed the last inventory.</p>
ComplianceConnectionID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The identifier for a data source connection in the <code>ComplianceConnection</code> table.</p>
VMLocation	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>Location of the virtual machine on the file system.</p>
GuestFullName	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>Configured operating system for the guest.</p>
VMComputerID	<p><i>Type:</i> big integer. Key. Nullable</p> <p>The identifier used in the source connection for the virtual machine's computer.</p>
PoolType	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>The type of the pool that the virtual machine belongs to.</p>
AffinityEnabled	<p><i>Type:</i> boolean</p> <p>Set this to <code>True</code> if this VM is unable to move to different host computers.</p>
CPUAffinity	<p><i>Type:</i> text (max 256 characters). Nullable</p>



Database Column	Details
	Contains the CPU Affinity value for virtual machine(Host Logical Processors)
CoreAffinity	Type: text (max 256 characters). Nullable Contains the Core Affinity value for virtual machine
PartitionID	Type: text (max 100 characters). Nullable Partition ID generated and used by the managing virtualization platform
PartitionNumber	Type: integer. Nullable Number of this partition

## ImportedWMIEvidence Table

The `ImportedWMIEvidence` table holds all of the WMI evidence which has been retrieved from the source connections.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 621: Database columns for ImportedWMIEvidence table**

Database Column	Details
ComplianceConnectionID	Type: integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table.
ClassName	Type: text (max 50 characters). Key. Nullable The WMI class name of the WMI evidence.
PropertyName	Type: text (max 50 characters). Key. Nullable The WMI property name of the WMI evidence.
PropertyValue	Type: text (max 256 characters). Key. Nullable The value of the property of the WMI evidence.
ExternalEvidenceID	Type: big integer. Key. Nullable The identifier used in the source connection for the WMI evidence.

## ImportedWMIEvidenceRuleMapping Table

The `ImportedWMIEvidenceRuleMapping` table is used by the importer to link imported WMI evidence with evidence in the `WMIEvidence` table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 622: Database columns for `ImportedWMIEvidenceRuleMapping` table**

Database Column	Details
<code>EvidenceRuleID</code>	<i>Type:</i> integer. Nullable The identifier for the WMI evidence in the <code>WMIEvidence</code> table.
<code>ExternalEvidenceID</code>	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the imported WMI evidence.
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key. Nullable The identifier of a data source connection in the <code>ComplianceConnection</code> table.

## ImporterValueMapping Table

The `ImporterValueMapping` table stores mapping pairs for use by importer tasks. It serves as a basic lookup translation table that is not connection-specific.

**Table 623: Database columns for `ImporterValueMapping` table**

Database Column	Details
<code>ImporterValueMappingID</code>	<i>Type:</i> integer. Key. Generated ID Unique auto-incrementing identifier.
<code>Category</code>	<i>Type:</i> text (max 100 characters). Key The importer section applicable for this key, uses dotted notation: e.g. "MobileDevice.Apple.Model".
<code>FromValue</code>	<i>Type:</i> text (max 256 characters). Key The value to translate.

Database Column	Details
ToValue	Type: text (max 256 characters). Nullable The required destination value for the Category/FromValue pair.

## InstalledApplications Table

The `InstalledApplications` table is populated by the import process to track which software has been installed.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 624: Database columns for `InstalledApplications` table**

Database Column	Details
ComplianceComputerID	Type: integer. Key The identifier for the computer in the <code>ComplianceComputer</code> table that the software is installed on.
SoftwareTitleID	Type: integer. Key The identifier for the software in the <code>SoftwareTitle</code> table that is installed.
InstanceName	Type: text (max 256 characters). Nullable The name of the instance that the software installation is associated with.
InstallerEvidence	Type: boolean This field is <code>True</code> if the installation is reported due to installer evidence.
FileEvidence	Type: boolean This field is <code>True</code> if the installation is reported due to file evidence.
WMIEvidence	Type: boolean This field is <code>True</code> if the installation is reported due to WMI evidence.
AccessModeID	Type: integer The access mode for which the installed application has been accessed. Foreign key to the <code>AccessMode</code> table.

Database Column	Details
IsACL	<p>Type: boolean</p> <p>Determines whether the access mode record came from ACL data.</p>

## RelatedInstalledApplications Table

The `RelatedInstalledApplications` table is populated by the import process to track which relationship between applications.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 625: Database columns for `RelatedInstalledApplications` table**

Database Column	Details
ParentComplianceComputerID	<p>Type: integer. Key</p> <p>The parent identifier for the computer in the <code>ComplianceComputer</code> table that the software is installed on.</p>
ParentSoftwareTitleID	<p>Type: integer. Key</p> <p>The parent identifier for the software in the <code>SoftwareTitle</code> table that is installed.</p>
ParentAccessModeID	<p>Type: integer. Key</p> <p>The access mode for which the installed application has been accessed. Foreign key to the <code>AccessMode</code> table.</p>
ChildComplianceComputerID	<p>Type: integer. Key</p> <p>The child identifier for the computer in the <code>ComplianceComputer</code> table that the software is installed on.</p>
ChildSoftwareTitleID	<p>Type: integer. Key</p> <p>The child identifier for the software in the <code>SoftwareTitle</code> table that is installed.</p>
ChildAccessModeID	<p>Type: integer. Key</p>

Database Column	Details
	The access mode for which the installed application has been accessed. Foreign key to the <code>AccessMode</code> table.
<code>IsCharged</code>	<i>Type:</i> boolean. Key  The identifier used in the source connection to determine the pricing relation between parent and child installer evidence (specifies if it is charged or free).
<code>ConfidenceLevel</code>	<i>Type:</i> integer. Nullable  Confidence level for each bundled installer evidence (as a percentage).

## Compliance.InventoryWriter.Matching Tables

The complete set of database tables documented here includes:

- `ComplianceComputerMatchResult` table (see *ComplianceComputerMatchResult Table* on page 629)
- `ImportedComputerMatchResult` table (see *ImportedComputerMatchResult Table* on page 630)
- `ImportedVirtualMachineMatchResult` table (see *ImportedVirtualMachineMatchResult Table* on page 631)
- `VirtualMachineMatchResult` table (see *VirtualMachineMatchResult Table* on page 632)

## ComplianceComputerMatchResult Table

The `ComplianceComputerMatchResult` table stores the results of performing matching between `ImportedComputers` and `ComplianceComputers`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 626: Database columns for `ComplianceComputerMatchResult` table**

Database Column	Details
<code>ComplianceConnectionID</code>	<i>Type:</i> integer. Key  The identifier for a data source connection in the <code>ComplianceConnection</code> table that supplied the <code>ImportedComputer</code> .
<code>ExternalID</code>	<i>Type:</i> big integer. Key

Database Column	Details
	The identifier used in the source connection for the <code>ImportedComputer</code> .
<code>ComplianceComputerID</code>	<i>Type:</i> integer. Key Identifier of the computer in the <code>ComplianceComputer</code> table that this <code>ImportedComputer</code> links to.
<code>MatchingRule</code>	<i>Type:</i> text (max 128 characters) The matching rule which determined the match between this <code>ImportedComputer</code> and <code>ComplianceComputer</code> .

## ImportedComputerMatchResult Table

The `ImportedComputerMatchResult` table stores the results of performing matching between `ImportedComputers`.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 627: Database columns for `ImportedComputerMatchResult` table**

Database Column	Details
<code>PrimaryComplianceConnectionID</code>	<i>Type:</i> integer. Key The identifier for a data source connection in the <code>ComplianceConnection</code> table that supplied the primary <code>ImportedComputer</code> .
<code>PrimaryExternalID</code>	<i>Type:</i> big integer. Key The identifier used in the source connection for the primary <code>ImportedComputer</code> .
<code>MatchedComplianceConnectionID</code>	<i>Type:</i> integer. Key. Nullable The identifier for a data source connection in the <code>ComplianceConnection</code> table that supplied the matched <code>ImportedComputer</code> .
<code>MatchedExternalID</code>	<i>Type:</i> big integer. Key. Nullable The identifier used in the source connection for the matched <code>ImportedComputer</code> .
<code>MatchingRule</code>	<i>Type:</i> text (max 128 characters)

Database Column	Details
	The matching rule which determined the match between these ImportedComputers.

## ImportedVirtualMachineMatchResult Table

The ImportedVirtualMachineMatchResult table stores the results of performing matching between ImportedVirtualMachines.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 628: Database columns for ImportedVirtualMachineMatchResult table**

Database Column	Details
PrimaryComplianceConnectionID	<i>Type:</i> integer. Key The identifier for a data source connection in the ComplianceConnection table that supplied the primary ImportedVirtualMachine.
PrimaryVMComputerID	<i>Type:</i> big integer. Key The identifier used in the source connection for the primary ImportedVirtualMachine.
PrimaryHostComputerID	<i>Type:</i> big integer. Key The identifier used in the source connection for the primary host ImportedVirtualMachine.
MatchedComplianceConnectionID	<i>Type:</i> integer. Key The identifier for a data source connection in the ComplianceConnection table that supplied the matched ImportedVirtualMachine.
MatchedVMComputerID	<i>Type:</i> big integer. Key The identifier used in the source connection for the matched ImportedVirtualMachine.
MatchedHostComputerID	<i>Type:</i> big integer. Key The identifier used in the source connection for the matched host ImportedVirtualMachine.

Database Column	Details
MatchingRule	<p><i>Type:</i> text (max 128 characters)</p> <p>The matching rule which determined the match between these ImportedVirtualMachines.</p>
NeedsCreation	<p><i>Type:</i> boolean</p> <p>Whether this ImportedVirtualMachine is awaiting creation as a VirtualMachine or not.</p>

## VirtualMachineMatchResult Table

The VirtualMachineMatchResult table stores the results of performing matching between ImportedVirtualMachines and VirtualMachines.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 629: Database columns for VirtualMachineMatchResult table**

Database Column	Details
ComplianceConnectionID	<p><i>Type:</i> integer. Key</p> <p>The identifier for a data source connection in the ComplianceConnection table that supplied the ImportedVirtualMachine.</p>
VMComputerID	<p><i>Type:</i> big integer. Key</p> <p>The identifier used in the source connection for the ImportedVirtualMachine.</p>
HostComputerID	<p><i>Type:</i> big integer. Key</p> <p>The identifier used in the source connection for the host of the ImportedVirtualMachine.</p>
VirtualMachineID	<p><i>Type:</i> integer. Key</p> <p>Identifier of the virtual machine in the VirtualMachine table that this ImportedVirtualMachine links to.</p>
MatchingRule	<p><i>Type:</i> text (max 128 characters)</p>



Database Column	Details
	The matching rule which determined the match between these VirtualMachines.

---

# 3

---

## Inventory Database Schema

### Topics:

- *Information Structure*
- *AD Tables*
- *Allocation Tables*
- *DirectoryObjects Tables*
- *Directory Tables*
- *Distribution Tables*
- *IM\_Right Tables*
- *Installation Tables*
- *Inventory Tables*
- *Licensing Tables*
- *ManageSoft Tables*
- *Networking Tables*
- *Packaging Tables*
- *ReferenceData Tables*
- *Rights Tables*
- *Status Tables*
- *Targeting Tables*
- *Tenants Tables*
- *Usage Tables*
- *WakeOnLAN Tables*
- *WorkFlow Tables*

This chapter describes the schema for the FlexNet Manager Suite database that collects inventory uploaded by the FlexNet inventory agent, either when installed on 'adopted' devices, or when executing a remote, zero-touch inventory.

This inventory data undergoes some rationalization within this schema. The import of the resulting clean inventory data from this database to the compliance database is the work of the Compliance Reader, making use of another intermediate schema (see *Compliance Reader Database Schema* on page 551).

# Information Structure

The following information is provided about database tables. Items appear only when relevant to the database column, and are suppressed where they do not apply. Two of these items (shown bold) are columns in the following pages, and the remainder are displayed within the **Details**.

Item	Comment
<b>Database Column</b>	The name of the column in the SQL table.
<i>Type</i>	The data type of the contents of the database column.
Size	For types that have a maximum capacity, the upper limit is provided in parentheses.
Key	The word "Key" appears when a column is a unique key field within the table. It is possible for several database columns to be part of the key, so that this indicator may appear for several columns in a table.
Generated ID	This indicates that a numeric ID is assigned by the database.
Nullable	If this indicator is present, the database column permits nulls.
Computed	This indicator appears for columns that are automatically computed by the database.
Default	If a column has a default value declared in the schema, this is specified at the end of the first set of details for the column.
<b>Details</b>	Describes the data stored in the database column, including many of the indicators described above.

## AD Tables

The complete set of database tables documented here includes:

- ADComputer table (see *ADComputer Table* on page 636)
- ADEternalMember table (see *ADEternalMember Table* on page 636)
- ADSDOU table (see *ADSDOU Table* on page 637)
- ADUser table (see *ADUser Table* on page 638)

## ADComputer Table

The `ADComputer` table is populated with data from Active Directory in preparation for an Active Directory reconciliation.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 630: Database columns for ADComputer table**

Database Column	Details
DomainID	Type: integer. Key OrganizationID of the domain in which the computer resides.
ComputerCN	Type: text (max 64 characters). Key The computer's common name.
ComputerOURDN	Type: text (max 384 characters). Key The relative distinguished name of the organizational unit or container holding this computer.
GUID	Type: binary (max 16 bytes). Key The <code>objectGUID</code> of the Active Directory object that represents this computer, if known.
SID	Type: text (max 256 characters). Nullable The computer's SID.

## ADExternalMember Table

The `&ADExternalMember;` table stores cross domain Active Directory objects.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 631: Database columns for ADExternalMember table**

Database Column	Details
GroupID	Type: integer. Key The GroupID the external member belongs to.
ExternalMemberSID	Type: text (max 256 characters). Key The external member SID.

## ADSDOU Table

The ADSDOU table is populated with domain, and organizational unit data from Active Directory in preparation for an Active Directory reconciliation.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 632: Database columns for ADSDOU table**

Database Column	Details
DomainID	Type: integer. Key The domain in which this object resides.
RDN	Type: text (max 400 characters). Key The relative distinguished name of this object.
GUID	Type: binary (max 16 bytes). Key The ObjectGUID of this Active Directory object.
BlockInheritance	Type: boolean True (1) if package allocations should not be inherited from parent OUs or Domain, unless no-override is set for the Allocation (in the policy group membership mode).

## ADUser Table

The `ADUser` table contains is populated with data from Active Directory in preparation for an Active Directory reconciliation. It is a temporary table.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 633: Database columns for ADUser table**

Database Column	Details
DomainID	Type: integer. Key The domain in which this user resides.
UserCN	Type: text (max 64 characters). Key The user's common name.
UserOURDN	Type: text (max 384 characters). Key The relative distinguished name of the organizational unit or container holding this user.
GUID	Type: binary (max 16 bytes). Key. Nullable The Active Directory GUID of this user.
SAMAccountName	Type: text (max 20 characters). Nullable The user's logon name used to support clients and servers from versions of Windows prior to Windows 2000.
Sid	Type: text (max 512 characters). Nullable User's Sid

## Allocation Tables

The complete set of database tables documented here includes:

- AllocationDetails table (see *AllocationDetails Table* on page 639)
- InstallationPostponement table (see *InstallationPostponement Table* on page 641)
- PackageAllocation table (see *PackageAllocation Table* on page 642)

- PackageApplies table (see *PackageApplies Table* on page 643)
- Policy table (see *Policy Table* on page 644)
- PolicyApplies table (see *PolicyApplies Table* on page 645)

## AllocationDetails Table

The `AllocationDetails` table contains various details that, when taken together, describe the rules under which a particular software policy (allocation) should be applied. One row is created in this table for each distinct set of values, and these collected details may apply to many packages and to many users or computers.

**Table 634: Database columns for AllocationDetails table**

Database Column	Details
<code>AllocationDetailsID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>Auto-generated unique identifier</p>
<code>Action</code>	<p><i>Type:</i> text (max 16 characters). Key. Nullable</p> <p>An action that indicates whether the package is mandatory or optional for the user or computer. Possible values are:</p> <ul style="list-style-type: none"> <li>• <code>install</code> (for mandatory installations)</li> <li>• <code>choose</code> (for optional installations)</li> <li>• <code>alwaysupdate</code> (for installations that are optional, but with updates to existing installations of the same package being mandatory).</li> </ul>
<code>EffectiveFrom</code>	<p><i>Type:</i> datetime. Key. Nullable</p> <p>Date and time at which the policy becomes effective. Prior to this date and time, the package will not be installed.</p>
<code>EffectiveFromIsLocal</code>	<p><i>Type:</i> boolean. Key. Nullable</p> <p>Indicates whether the date and time in the <code>EffectiveFrom</code> field is local time on the managed device, or UTC time. The possible values are zero for UTC time and one for local time.</p>
<code>EffectiveUntil</code>	<p><i>Type:</i> datetime. Key. Nullable</p> <p>Date and time at which the policy ceases to be effective. After this date and time, the package will not be installed. The time zone used is the same as for <code>EffectiveFrom</code>.</p>
<code>Wake</code>	<p><i>Type:</i> boolean. Key. Nullable</p>

Database Column	Details
	Indicates whether FlexNet Manager Suite should use the Wake on LAN feature to wake a managed device when the package is to be installed. For more information on this feature, see the documentation.
PostponeNoLaterThan	<i>Type:</i> datetime. Key. Nullable Latest absolute time until which an end-user may postpone installation of this package. Only mandatory package installations can be postponed (that is, packages for which <i>Action</i> is either <i>install</i> or <i>alwaysupdate</i> ).
PostponeNoLaterThanIsLocal	<i>Type:</i> boolean. Key Indicates whether <i>PostponeNoLaterThan</i> should be interpreted as a local time on each managed device (1), or as a UTC time (0). The value of this field should be ignored if <i>PostponeNoLaterThan</i> is NULL.
PostponePeriod	<i>Type:</i> integer. Key. Nullable Number of seconds for which package installations may be postponed by an end-user after this policy first applies. Only mandatory package installations can be postponed (that is, packages for which <i>Action</i> is either <i>install</i> or <i>alwaysupdate</i> ).
PostponeLatest	<i>Type:</i> boolean. Key Indicates whether package installation may be postponed to the latest (1) or earliest (0) date indicated by the <i>PostponeNoLaterThan</i> and <i>PostponePeriod</i> fields if both of those fields are set. The value of this field should be ignored if either <i>PostponeNoLaterThan</i> or <i>PostponePeriod</i> are NULL.
PostponeOKForLowBandwidth	<i>Type:</i> boolean. Key Indicates whether a valid reason for postponing installation of this package is because the managed device is connected to a distribution location via a "slow" network connection (as determined by the <i>NetworkHighSpeed</i> preference on the managed device).
PostponeOKForAnyReason	<i>Type:</i> boolean. Key Indicates whether installation of this package can be postponed for any reason at the discretion of the end-user on the managed device on which this package is to be installed.
Exclusive	<i>Type:</i> boolean. Key Whether (1) or not (0, default) to uninstall the package when it is removed from policy. This value is retrieved from Active Directory.
Removable	<i>Type:</i> boolean. Key



Database Column	Details
	Whether (1) or not (0, default) this mandatory package can be removed by the user once it has initially been installed. This value is retrieved from Active Directory.

## InstallationPostponement Table

The `InstallationPostponement` table stores the resultant set of policy (RSOP) for all users and computers. It represents what packages each user and computer should have installed, whereas `Installation` represents what they actually have installed.

**Table 635: Database columns for `InstallationPostponement` table**

Database Column	Details
<code>TargetTypeID</code>	<p><i>Type:</i> integer. Key</p> <p>The target type of the package. Possible values are:</p> <ul style="list-style-type: none"> <li>1 (computer policy)</li> <li>2 (user policy)</li> </ul> <p>This is a foreign key into the <code>TargetType</code> table.</p>
<code>TargetID</code>	<p><i>Type:</i> integer. Key</p> <p>The user or computer targeted by the package. This is a foreign key into the <code>User</code> or <code>Computer</code> table.</p>
<code>PackagePathID</code>	<p><i>Type:</i> integer. Key</p> <p>The package applied by the <code>Policy</code>.</p>
<code>PolicyGUID</code>	<p><i>Type:</i> binary (max 16 bytes). Key</p> <p>The GUID of the group policy in Active Directory that records whether the package is targeted to the user or computer.</p>
<code>AllocationDetailsID</code>	<p><i>Type:</i> integer. Key</p> <p>The details indicating how and when this package should be installed. This is a foreign key into the <code>AllocationDetails</code> table.</p>
<code>PostponePeriodStart</code>	<p><i>Type:</i> datetime. Nullable</p> <p>UTC time at which any postponement period for this policy started. The postponement period for installation of this package (that is, the period during which end-users may postpone installation of this package) will end at this time plus any period specified by <code>PostponePeriod</code>. After the postponement</p>

Database Column	Details
	<p>period ends, the installation agent will attempt to force the package to be installed. This field is NULL until and unless an installation event log is received from the managed device indicating that the installation of the package has in fact been postponed. If <code>PostponePeriod</code> is NULL, the value of <code>PostponePeriodStart</code> represents the time at which the installation of this package was first postponed.</p> <p>An example SQL query to determine the expected (UTC) time at which the postponement period for this package will expire can be written as shown below. Note that this query will convert any local time <code>PostponeNoLaterThan</code> value to UTC using the timezone configured on the SQL Server.</p> <pre>SELECT * , CASE WHEN PostponePeriod IS NULL OR PostponePeriodStart IS NULL OR (PostponeLatest = 1 AND PostponePeriodEndUTC &lt;= PostponeNoLaterThanUTC) OR (PostponeLatest = 0 AND PostponePeriodEndUTC &gt;= PostponeNoLaterThanUTC) THEN PostponeNoLaterThanUTC ELSE PostponePeriodEndUTC END AS PostponeLatestUTC FROM ( SELECT * , CASE PostponeNoLaterThanIsLocal WHEN 1 THEN DATEADD(s, DATEDIFF(s, GETDATE(), GETUTCDATE()), PostponeNoLaterThan) ELSE PostponeNoLaterThan END AS PostponeNoLaterThanUTC , DATEADD(s, PostponePeriod, PostponePeriodStart) AS PostponePeriodEndUTC FROM Targetedpackage ) tp</pre>

## PackageAllocation Table

A `PackageAllocation` row exists for every `PackagePath` which has been approved to a `Policy`.

**Table 636: Database columns for PackageAllocation table**

Database Column	Details
<code>PolicyGUID</code>	<p><i>Type:</i> binary (max 16 bytes). Key</p> <p>The <code>Policy</code> to which the <code>PackagePath</code> has been approved. This is a foreign key into the <code>Policy</code> table.</p>
<code>PackagePathID</code>	<p><i>Type:</i> integer. Key</p> <p>The <code>PackagePath</code> which has been approved. This is a foreign key into the <code>PackagePath</code> table.</p>
<code>AccessGroupID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The <code>Group</code> to which the package applies. This group will have a NULL <code>GroupCN</code> it's an Access Control List (ACL) group. This is a foreign key into the <code>Group</code> table.</p>

Database Column	Details
TargetTypeID	<p><i>Type:</i> integer. Key</p> <p>The target type of the package. Possible values are:</p> <ul style="list-style-type: none"> <li>• 1 = computer policy</li> <li>• 2 = user policy.</li> </ul> <p>This is a foreign key into the <code>TargetType</code> table.</p>
Precedence	<p><i>Type:</i> integer</p> <p>The order of application of this package within this policy. The default value is 0.</p>
AllocationDetailsID	<p><i>Type:</i> integer. Key</p> <p>The details indicating how and when this package should be installed. This is a foreign key into the <code>AllocationDetails</code> table.</p>

## PackageApplies Table

The `PackageApplies` table stores the resultant set of policy (RSoP) for all users and computers. It represents what packages each user and computer should have installed, whereas `Installation` represents what they actually have installed.

**Table 637: Database columns for `PackageApplies` table**

Database Column	Details
TargetTypeID	<p><i>Type:</i> integer. Key</p> <p>The target type of the package. Possible values are:</p> <ul style="list-style-type: none"> <li>• 1 (computer policy)</li> <li>• 2 (user policy)</li> </ul> <p>This is a foreign key into the <code>TargetType</code> table.</p>
TargetOUID	<p><i>Type:</i> integer. Key</p> <p>The organizational unit of the user or computer targeted by the package. This is a foreign key into the <code>Organization</code> table. This column is included in the table for clustering purposes. The value of <code>TargetOUID</code> could be determined by looking up the <code>User</code> or <code>Computer</code> table.</p>
TargetID	<p><i>Type:</i> integer. Key</p>

Database Column	Details
	The user or computer targeted by the package. This is a foreign key into the <code>User</code> or <code>Computer</code> table.
<code>PackagePathID</code>	<i>Type:</i> integer. Key The package applied by the <code>Policy</code> .
<code>PolicyGUID</code>	<i>Type:</i> binary (max 16 bytes). Key The GUID of the group policy in Active Directory that records whether the package is targeted to the user or computer.
<code>AllocationDetailsID</code>	<i>Type:</i> integer. Key The details indicating how and when this package should be installed. This is a foreign key into the <code>AllocationDetails</code> table.
<code>Precedence</code>	<i>Type:</i> integer The order of application of this policy for this target device, defaults to 0.
<code>PolicyVersion</code>	<i>Type:</i> integer Policy version number from the underlying directory service.

## Policy Table

The `Policy` table correlates the GUID of an Active Directory policy with its display name. This is used in reporting and in `.npl` files.

**Table 638: Database columns for `Policy` table**

Database Column	Details
<code>GUID</code>	<i>Type:</i> binary (max 16 bytes). Key The GUID of the policy in Active Directory.
<code>DomainID</code>	<i>Type:</i> integer. Key. Nullable Organization id of the domain in which the policy resides.
<code>DisplayName</code>	<i>Type:</i> text (max 512 characters). Key. Nullable The display name for the policy, for use in <code>.npl</code> files.
<code>AccessGroupID</code>	<i>Type:</i> integer. Key. Nullable The Access Control List for the policy, represented as an anonymous <code>Group</code> .

Database Column	Details
EnabledForUsers	<i>Type:</i> boolean This policy has been enabled for software management for users.
EnabledForComputers	<i>Type:</i> boolean This policy has been enabled for software management for computers.

## PolicyApplies Table

The `PolicyApplies` table stores the identities of the principals to whom each policy applies, whether or not any packages or schedules apply.

**Table 639: Database columns for `PolicyApplies` table**

Database Column	Details
DomainID	<i>Type:</i> integer. Key Organizational id of the domain in which the policy resides.
TargetOUID	<i>Type:</i> integer. Key The OUID of the user or computer to whom the policy applies. Foreign key (unchecked) into the <code>Organization</code> table.
TargetTypeID	<i>Type:</i> integer. Key The target type of the package. Possible values are 1 (computer policy), 2 (user policy). This is a foreign key into the <code>TargetType</code> table.
TargetID	<i>Type:</i> integer. Key The user or computer to whom the policy applies. Foreign key (unchecked) into the <code>User</code> or <code>Computer</code> table.
GUID	<i>Type:</i> binary (max 16 bytes). Key The GUID of the policy in Active Directory.

## DirectoryObjects Tables

The complete set of database tables documented here includes:

- Computer table (see *Computer Table* on page 646)
- User table (see *User Table* on page 646)

# Computer Table

The `Computer` table contains all computers that have ever reported information or have been targeted by policy in a FlexNet Manager Suite environment.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 640: Database columns for `Computer` table**

Database Column	Details
<code>ComputerID</code>	<p><i>Type:</i> integer. Key. Generated ID</p> <p>The ID for the computer. This is automatically generated by SQL Server.</p>
<code>ComputerOID</code>	<p><i>Type:</i> integer. Key</p> <p>The organizational unit of the computer in Active Directory. In an SMS organization, this is set to the OID of the unknown OU.</p>
<code>ComputerCN</code>	<p><i>Type:</i> text (max 64 characters). Key</p> <p>The computer's common name. In an Active Directory environment this is the common name attribute of the computer's distinguished name. This is the same as the SAM account name.</p>
<code>OperatingSystemID</code>	<p><i>Type:</i> integer. Nullable</p> <p>The operating system of the computer, if known. This allows efficient determination of the operating system breakdown of computers in an organization.</p>
<code>GUID</code>	<p><i>Type:</i> binary (max 16 bytes). Key. Nullable</p> <p>The <code>objectGUID</code> of the Active Directory object that represents this computer, if known.</p>

# User Table

The `User` table contains all of the users that have ever reported information in a FlexNet Manager Suite environment. A row has been added to cater for installations that occur from computer policy. This row has a `UserID` set to 1 and a `UserCN` of `SYSTEM`, and belongs to `Organization 1`, which is the universal Domain.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 641: Database columns for User table**

Database Column	Details
UserID	Type: integer. Key. Generated ID The ID for the user. This is automatically generated by SQL Server.
UserOID	Type: integer. Key The organizational unit of the user in Active Directory. This is a foreign key into the <i>Organization</i> table. In an SMS environment, this is always set to the unknown OU.
UserCN	Type: text (max 64 characters). Key The user's common name. In an Active Directory environment this is the common name attribute of the user's distinguished name.
GUID	Type: binary (max 16 bytes). Key. Nullable The <i>objectGUID</i> of the Active Directory object that represents this user, if known.
SAMAccountName	Type: text (max 20 characters). Key. Nullable The SAM account name used to uniquely identify this user in event logs and user inventories.

## Directory Tables

The complete set of database tables documented here includes:

- Domain table (see *Domain Table* on page 648)
- DomainConfiguration table (see *DomainConfiguration Table* on page 649)
- Group table (see *Group Table* on page 650)
- Member table (see *Member Table* on page 651)
- Organization table (see *Organization Table* on page 651)
- TransitiveMember table (see *TransitiveMember Table* on page 652)

# Domain Table

The `Domain` table, in combination with the `Organization` table, contains data about all of the domains, and organizational units that have ever had users or computers report information in a FlexNet Manager Suite environment.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 642: Database columns for `Domain` table**

Database Column	Details
<code>OrganizationID</code>	<i>Type:</i> integer. Key Organizational ID. This is a foreign key into the <code>Organization</code> table.
<code>DN</code>	<i>Type:</i> text (max 100 characters). Key. Nullable Fully qualified distinguished name.
<code>DomainType</code>	<i>Type:</i> text (max 4 characters). Key. Nullable The type of directory service running, for example AD, NT 4.
<code>FlatName</code>	<i>Type:</i> text (max 32 characters). Nullable The NT 4 domain name.
<code>PreferredDomainController</code>	<i>Type:</i> text (max 32 characters). Nullable Preferred domain controller to query.
<code>PreferredDomainControllerOnly</code>	<i>Type:</i> boolean Whether (0) or not to fail over to alternate server if the preferred domain controller is not contactable.
<code>ADReconcile</code>	<i>Type:</i> boolean Whether (1) or not (0) to reconcile the FlexNet Manager Suite database with Active Directory.
<code>ADLoadLatency</code>	<i>Type:</i> integer If reconciling Active Directory with the FlexNet Manager Suite database, the length of time in minutes before the Active Directory data is refreshed in the FlexNet Manager Suite database. The default value is 60 minutes. A value of 0 means load the Active Directory data into the FlexNet Manager Suite database



Database Column	Details
	at each reconciliation. Set this to a high value to minimize network traffic for domains for delayed reconciliation is acceptable.
MergePolicies	<i>Type:</i> boolean Whether (1) or not (0) to generate merged policies.
LastADReconcile	<i>Type:</i> datetime. Nullable The date and time of the last reconciliation of the FlexNet Manager Suite database with Active Directory.
LastADReconcileStatus	<i>Type:</i> boolean This field is currently unused.
LastADLoad	<i>Type:</i> datetime. Nullable The date and time of the last Active Directory load. A value of NULL indicates that Active Directory data should be loaded at the next reconcile operation.
LastPolicyMerge	<i>Type:</i> datetime. Nullable The date and time of the last generation of merged policy.
LastPolicyMergeStatus	<i>Type:</i> boolean This field is currently unused.
DNReverse	<i>Type:</i> text (max 100 characters). Key. Nullable Fully qualified distinguished name, in reverse order (to improve sub-domain search performance).

## DomainConfiguration Table

The `DomainConfiguration` table contains configuration properties for the `Domain` table

**Table 643: Database columns for DomainConfiguration table**

Database Column	Details
DomainID	<i>Type:</i> integer. Key OrganizationID of the domain in which the entry resides.
Property	<i>Type:</i> text (max 32 characters). Key The name of the property.

Database Column	Details
Value	<i>Type:</i> text (max 256 characters). Nullable The value of the property.
DateValue	<i>Type:</i> datetime. Nullable The date and time value of the property.
LastUpdate	<i>Type:</i> datetime The date and time the property was last updated.

## Group Table

Each `Group` identifies either a named group or an unnamed Access Control List (ACL). Each `Group` is associated with rows in the `Member` table.

**Table 644: Database columns for Group table**

Database Column	Details
GroupID	<i>Type:</i> integer. Key. Generated ID The ID for the group, automatically generated by SQL Server.
GUID	<i>Type:</i> binary (max 16 bytes). Key The Globally Unique Identifier for the group. In the case where this <code>Group</code> represents an Access Control List for a <code>Policy</code> or a <code>PackageAllocation</code> , the GUID is that of this object.
GroupCN	<i>Type:</i> text (max 128 characters). Key. Nullable The Common Name for the group. In the case where this <code>Group</code> represents an Access Control List for a <code>Policy</code> or a <code>PackageAllocation</code> , the <code>GroupCN</code> is NULL.
GroupOUID	<i>Type:</i> integer. Key A reference to the <code>Organization</code> to which the group belongs.
GroupType	<i>Type:</i> integer. Nullable The bitmask of flags defining the type of this <code>Group</code> .
SID	<i>Type:</i> text (max 256 characters). Nullable The security identifier of this <code>Group</code> .

## Member Table

The `Member` table stores the membership lists for every group. Each `Member` details a `User`, `Computer`, `Group`, or `Organization` (only `Policy ACL` groups), and whether the specified item is excluded (only `ACL` groups), included (the default) or included mandatorily (cannot be excluded - used only for `Organizations` in `Policy ACLs`).

**Table 645: Database columns for `Member` table**

Database Column	Details
<code>GroupID</code>	<p><i>Type:</i> integer. Key</p> <p>The <code>Group</code> of which this is a <code>Member</code>.</p>
<code>TargetTypeID</code>	<p><i>Type:</i> integer. Key</p> <p>The <code>TargetType</code>. Possible values are:</p> <ul style="list-style-type: none"> <li>• 1 = <code>Computer</code></li> <li>• 2 = <code>User</code></li> <li>• 3 = <code>Group</code></li> <li>• 8 = <code>OrgUnit</code></li> <li>• 16 = <code>Operator</code></li> </ul>
<code>TargetID</code>	<p><i>Type:</i> integer. Key</p> <p>The <code>ComputerID</code>, <code>UserID</code>, <code>GroupID</code> or <code>OrganizationID</code>.</p>
<code>MemberMode</code>	<p><i>Type:</i> integer</p> <p>The <code>MemberMode</code> is 0 for <code>Exclude</code> (regardless of any other memberships, the principals of this <code>Target</code> are excluded from this group), 1 for <code>Include</code>, and 2 for <code>Always - NoOverride</code>.</p>

## Organization Table

The `Organization` table contains data about organizational units used in a FlexNet Manager Suite environment.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 646: Database columns for Organization table**

Database Column	Details
OrganizationID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>The ID for the organizational unit. (1 is used for “unknown OU” in the universal domain). This is automatically generated by SQL Server.</p>
RDN	<p><i>Type:</i> text (max 400 characters). Key. Nullable</p> <p>The relative distinguished name of this organizational unit.</p>
GUID	<p><i>Type:</i> binary (max 16 bytes). Key. Nullable</p> <p>The <code>objectGUID</code> of the Active Directory object that represents this organizational unit, if known.</p>
DomainID	<p><i>Type:</i> integer. Key</p> <p><code>OrganizationID</code> of the domain in which the entry resides. For a domain, must be set to reference self.</p>
RDNReverse	<p><i>Type:</i> text (max 400 characters). Key. Nullable</p> <p>The relative distinguished name of the computer, reversed for superior performance on sub-organization searching.</p>
IsUnknown	<p><i>Type:</i> integer</p> <p>True (1) if the organizational unit cannot be resolved through Active Directory (for example, the unknown OU, which has a NULL RDN), false (0) otherwise (if the OU has a non-empty RDN).</p>
IsDomain	<p><i>Type:</i> integer</p> <p>True (1) if the organizational unit is a domain (has an empty - not NULL - RDN), false (0) otherwise.</p>
BlockInheritance	<p><i>Type:</i> boolean</p> <p>True (1) if package allocations should not be inherited from the parent Organization, unless <code>NoOverride</code> is set for the Policy. <code>NoOverride</code> is set using <code>MemberMode=2</code> (Always) on the Organization member in the Policy ACL group.</p>

## TransitiveMember Table

The `TransitiveMember` table stores data similar to the `Member` table, but is populated only when needed, such as to assist in evaluating the rights of a particular user. The difference is that for each user, it contains the full list of groups in which they are members either directly or indirectly through membership in other groups.

**Table 647: Database columns for TransitiveMember table**

Database Column	Details
GroupID	<p>Type: integer. Key</p> <p>The Group of which this is a Member.</p>
TargetTypeID	<p>Type: integer. Key</p> <p>The TargetType. Possible values are:</p> <ul style="list-style-type: none"> <li>• 1 = Computer</li> <li>• 2 = User</li> <li>• 3 = Group</li> <li>• 8 = OrgUnit</li> </ul>
TargetID	<p>Type: integer. Key</p> <p>The ComputerID, UserID, GroupID or OrganizationID.</p>
MemberMode	<p>Type: integer</p> <p>The MemberMode is 0 for Exclude (regardless of any other memberships, the principals of this Target are excluded from this group), 1 for Include, and 2 for Always - NoOverride.</p>

## Distribution Tables

The complete set of database tables documented here includes:

- DistributedPackage table (see *DistributedPackage Table* on page 653)
- DistributionGroup table (see *DistributionGroup Table* on page 655)
- DistributionGroupMember table (see *DistributionGroupMember Table* on page 655)
- DistributionServer table (see *DistributionServer Table* on page 656)
- DistributionServerStatus table (see *DistributionServerStatus Table* on page 657)
- DistributionServerType table (see *DistributionServerType Table* on page 658)

## DistributedPackage Table

The `DistributedPackage` table stores the status (both of current and pending distributions) of package distributions to distribution servers and distribution locations.

**Table 648: Database columns for DistributedPackage table**

Database Column	Details
DistributedPackageID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>Auto-generated identity number</p>
ServerUID	<p><i>Type:</i> binary (max 16 bytes). Key</p> <p>The distribution server or distribution location related to the status record. This is a foreign key into the <code>DistributionServer</code> table.</p>
RequestedVersionID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The id for the Requested <code>PackageVersion</code>.</p>
RequestState	<p><i>Type:</i> text (max 16 characters). Nullable</p> <p>The state of a package that is pending distribution. The possible values are:</p> <ul style="list-style-type: none"> <li>• empty (literal string)</li> <li>• pending</li> <li>• removing</li> </ul> <p>If the <code>RequestState</code> field contains a value other than the literal string <code>empty</code>, the <code>RequestState</code> overrides the <code>ConfirmedState</code> of the package.</p>
RequestDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date and time at which the package distribution began. Only used for distributions currently in progress.</p>
ConfirmedVersionID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The id for the Existing <code>PackageVersion</code></p>
ConfirmedState	<p><i>Type:</i> text (max 16 characters). Nullable</p> <p>The state of the package currently on the distribution server or distribution location. The possible values are:</p> <ul style="list-style-type: none"> <li>• available</li> <li>• unavailable</li> </ul> <p>If the <code>RequestState</code> field contains a value other than the literal string <code>empty</code>, then the <code>RequestState</code> overrides the <code>ConfirmedState</code> of the package.</p>
ConfirmedDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date and time that the current distribution status of a package was recorded.</p>

Database Column	Details
ConfirmedReason	<i>Type:</i> text. Nullable The reason that package distribution failed. This is only specified in the case of a failure.

## DistributionGroup Table

All defined distribution groups are stored in the `DistributionGroup` table.

**Table 649: Database columns for `DistributionGroup` table**

Database Column	Details
GroupUID	<i>Type:</i> binary (max 16 bytes). Key A unique identifier for this distribution group.
GroupName	<i>Type:</i> text (max 128 characters). Key The descriptive name assigned to this distribution group.

## DistributionGroupMember Table

Any distribution servers and distribution locations assigned to distribution groups are stored in the `DistributionGroupMember` table.

**Table 650: Database columns for `DistributionGroupMember` table**

Database Column	Details
GroupUID	<i>Type:</i> binary (max 16 bytes). Key A unique identifier for this distribution group. This UID is a foreign key to the GroupUID in the <code>DistributionGroup</code> table.
MemberID	<i>Type:</i> binary (max 16 bytes). Key A unique identifier for the distribution server or distribution location that is a member of this group. This UID is a foreign key to the ServerUID in the <code>DistributionServer</code> table.
MemberType	<i>Type:</i> integer An identifier for the type of this distribution group member. This identifier is a foreign key to the TargetTypeID in the <code>DistributionServerType</code> table.

# DistributionServer Table

The `DistributionServer` table stores all of the distribution servers and distribution locations in the FlexNet Manager Suite distribution hierarchy.

**Table 651: Database columns for `DistributionServer` table**

Database Column	Details
<code>ServerUID</code>	<p><i>Type:</i> binary (max 16 bytes). Key</p> <p>A unique identifier for the distribution server or distribution location. The core distribution server has a value of all zeroes.</p>
<code>DNSName</code>	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>DNS name of the server</p>
<code>ServerType</code>	<p><i>Type:</i> small integer</p> <p>The server type. The possible values are:</p> <ul style="list-style-type: none"> <li>• 0 for distribution location</li> <li>• 1 for distribution server</li> </ul>
<code>ServerName</code>	<p><i>Type:</i> text (max 64 characters)</p> <p>The name of the distribution server or distribution location.</p>
<code>PrimaryParentUID</code>	<p><i>Type:</i> binary (max 16 bytes). Key. Nullable</p> <p>The parent of the distribution server or distribution location. For the core distribution server, the <code>PrimaryParentUID</code> is NULL.</p>
<code>ConfigState</code>	<p><i>Type:</i> text (max 20 characters). Nullable</p> <p>The state of configuration of the distribution server. This is only set for distribution servers (<code>ServerType</code> is 1). This can be one of the following values:</p> <ul style="list-style-type: none"> <li>• configure</li> <li>• failed</li> <li>• pending</li> <li>• NULL</li> </ul>
<code>LastConfigStart</code>	<p><i>Type:</i> datetime. Nullable</p> <p>The date and time of the last configuration message sent to the distribution server. This is only set for distribution servers (<code>ServerType</code> is 1).</p>
<code>LastConfigJobId</code>	<p><i>Type:</i> text (max 40 characters). Nullable</p>



Database Column	Details
	The job identifier for the last configuration message sent to the distribution server. This is only set for distribution servers ( <code>ServerType</code> is 1).
<code>ConfigFailReason</code>	<i>Type:</i> text. Nullable The reason for a configuration failure for the distribution server.
<code>PolicyQuarantined</code>	<i>Type:</i> boolean Boolean value indicating whether this distribution location is quarantined from receiving policy distributions.
<code>TenantID</code>	<i>Type:</i> small integer The Tenant ID this Distribution Server has been assigned to.

## DistributionServerStatus Table

The `DistributionServerStatus` table stores status information for the distribution servers in the FlexNet Manager Suite distribution hierarchy.

**Table 652: Database columns for `DistributionServerStatus` table**

Database Column	Details
<code>ServerUID</code>	<i>Type:</i> binary (max 16 bytes). Key The distribution server related to the status record. This is a foreign key into the <code>DistributionServer</code> table.
<code>ReportedDate</code>	<i>Type:</i> datetime The date and time at which the distribution server last reported status information for this parameter.
<code>Type</code>	<i>Type:</i> text (max 32 characters). Key The type of the status parameter reported. Currently supported types are 'job' for jobs on the distribution server and 'logs' for log files awaiting upload from the distribution server.
<code>Name</code>	<i>Type:</i> text (max 64 characters). Key The name of the status parameter reported. This is an internal name for the parameter and is not intended for display.
<code>Count</code>	<i>Type:</i> integer

Database Column	Details
	The count of items for this status parameter currently awaiting processing by this distribution server.
DelayedCount	<i>Type:</i> integer The count of items for this status parameter that are older than a configurable time period that are currently awaiting processing by this distribution server. This will not necessarily have meaning for each status parameter.

## DistributionServerType Table

The available distribution server types are defined in the `DistributionServerType` table.

**Table 653: Database columns for `DistributionServerType` table**

Database Column	Details
DistributionServerTypeID	<i>Type:</i> integer. Key An identifier for this distribution server type.
DistributionServerType Name	<i>Type:</i> text (max 256 characters). Key The descriptive name assigned to this distribution server type.

## IM\_Right Tables

The complete set of database tables documented here includes:

- Right table (see *Right Table* on page 658)

## Right Table

Each action by FlexNet Manager Suite requires one or more `Rights` to perform an `ActionClass` over a given `Resource`.

**Table 654: Database columns for `Right` table**

Database Column	Details
RightID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number.

Database Column	Details
GroupID	<i>Type:</i> integer. Key The group to whom the <code>Right</code> is granted or denied (deny always takes precedence!).
ResourceID	<i>Type:</i> integer. Key The <code>Resource</code> to which the <code>Right</code> applies.
ActionClassID	<i>Type:</i> integer. Key The action class which applies (read or modify).
Denied	<i>Type:</i> boolean When TRUE (1), indicates that the specified right is denied.
Value	<i>Type:</i> integer. Key. Nullable The integer id of an item which depends on the <code>PartitionType</code> of the associated resource. In FlexNet Manager Suite 7.5, only <code>Organization</code> partitioning ( <code>PartitionTypeID</code> = 1) is used, so the <code>Value</code> field contains an <code>OrganizationID</code> . The <code>Right</code> applies to this organization and all child organizations, unless denied by another <code>Right</code> .

## Installation Tables

The complete set of database tables documented here includes:

- Installation table (see *Installation Table* on page 659)
- InstallationHistory table (see *InstallationHistory Table* on page 661)
- Reason table (see *Reason Table* on page 662)

## Installation Table

The `Installation` table contains the latest installation status of each package for each user and computer. Success or failure of installations is recorded. When a package is uninstalled, its installation status record is removed from the table. When an installation is successful, the successful installation status record replaces any earlier failure status records. Once an installation is successful, the `Installation` table retains the successful installation status record even if there are subsequent failed installation attempts. If there have been any subsequent failed installations, the latest of these failure records is also be retained. For example, if an installation fails, and then succeeds on a subsequent attempt, only the successful status is recorded. If an installation succeeds, but a later installation attempt fails, then both the success and failure status records are stored. All other installations are added to the `InstallationHistory` table as new status information is generated.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 655: Database columns for Installation table**

Database Column	Details
ComputerID	<p><i>Type:</i> integer. Key</p> <p>The computer that the installation event occurred on. This is a foreign key into the <code>Computer</code> table.</p>
UserID	<p><i>Type:</i> integer. Key</p> <p>The user associated with the installation event. This is a foreign key into the <code>User</code> table. If the <code>UserID</code> is 1 (system user), then the installation event occurred as part of computer policy. Otherwise, the installation event occurred as part of user policy.</p>
PackageVersionID	<p><i>Type:</i> integer. Key</p> <p>Package version that was installed</p>
OrganizationID	<p><i>Type:</i> integer. Key</p> <p>The organizational unit of the user or computer associated with the installation event. This is a foreign key into the <code>Organization</code> table. This column is included in the table for clustering purposes. The value of OUID could be determined by looking up the <code>User</code> or <code>Computer</code> table. If the <code>UserID</code> is 1 (system user), OUID represents the organizational unit of the computer that the installation event occurred on. Otherwise, it represents the organizational unit of the user associated with the installation event.</p>
Action	<p><i>Type:</i> text (max 10 characters)</p> <p>The action performed on the package. This is currently set to "install". In future, "upgrade", "update" and "selfheal" may be added.</p>
Reported	<p><i>Type:</i> datetime. Nullable</p> <p>The date and time that the installation event occurred.</p>
Received	<p><i>Type:</i> datetime. Nullable</p> <p>The date and time that the installation status event was received into the database.</p>
FailReasonID	<p><i>Type:</i> integer. Nullable</p>

Database Column	Details
	A reference to the reason for the installation failure. If the installation succeeded then this value is NULL.
Result	Type: text (max 16 characters). Nullable The result of the package installation. Possible values are success or failure.

## InstallationHistory Table

The action performed on the package, normally “install” or “uninstall”. In the event that installation event logs were lost, entries may be reconstructed here from data in inventory (cache tracking). Such entries may be less reliable (in particular the recorded date will be the date of the inventory) and will have one of the following *Action* values:

- “inv insert”
- “inv delete”
- “inv update”



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 656: Database columns for InstallationHistory table**

Database Column	Details
ComputerID	Type: integer. Key The computer that the installation event occurred on. This is a foreign key into the <i>Computer</i> table.
UserID	Type: integer. Key The user associated with the installation event. This is a foreign key into the <i>User</i> table. If the <i>UserID</i> is 1 then the installation event occurred as part of computer policy. Otherwise, the installation event occurred as part of user policy.
PackageVersionID	Type: integer. Key The id for the <i>PackageVersion</i> installed
Reported	Type: datetime. Key

Database Column	Details
	The date and time that the installation event occurred.
Action	Type: text (max 10 characters) The action performed on the package. This value can be either, install or uninstall.
Received	Type: datetime. Nullable The date and time that the installation status event was received into the database.
FailReasonID	Type: integer. Nullable A reference to the reason for the installation failure. If the installation succeeded then this value is NULL.
Result	Type: text (max 16 characters). Nullable The result of the package installation. Possible values are either success or failure.

## Reason Table

Stores extended text uploaded from logs to describe operational failures.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 657: Database columns for Reason table**

Database Column	Details
ReasonID	Type: integer. Key. Generated ID The ID for the Reason. This is automatically generated by SQL Server.
ReasonHash	Type: integer. Key The checksum of the ReasonText, calculated by SQL Server.
ReasonText	Type: text The Reason text.

# Inventory Tables

The complete set of database tables documented here includes:

- HardwareClass table (see *HardwareClass Table* on page 664)
- HardwareObject table (see *HardwareObject Table* on page 664)
- HardwareProperty table (see *HardwareProperty Table* on page 665)
- HardwareValue table (see *HardwareValue Table* on page 666)
- InventoryReport table (see *InventoryReport Table* on page 666)
- ServiceComponent table (see *ServiceComponent Table* on page 667)
- ServiceProvider table (see *ServiceProvider Table* on page 668)
- ServiceProviderApplicationOracle table (see *ServiceProviderApplicationOracle Table* on page 669)
- ServiceProviderApplicationUsagePerMonth table (see *ServiceProviderApplicationUsagePerMonth Table* on page 670)
- ServiceProviderApplicationUsageType table (see *ServiceProviderApplicationUsageType Table* on page 670)
- ServiceProviderApplicationUserOracle table (see *ServiceProviderApplicationUserOracle Table* on page 671)
- ServiceProviderComponent table (see *ServiceProviderComponent Table* on page 672)
- ServiceProviderComponentProperty table (see *ServiceProviderComponentProperty Table* on page 672)
- ServiceProviderComponentValue table (see *ServiceProviderComponentValue Table* on page 672)
- ServiceProviderName table (see *ServiceProviderName Table* on page 673)
- ServiceProviderProperty table (see *ServiceProviderProperty Table* on page 674)
- ServiceProviderType table (see *ServiceProviderType Table* on page 674)
- ServiceProviderValue table (see *ServiceProviderValue Table* on page 674)
- ServiceUser table (see *ServiceUser Table* on page 675)
- ServiceUserOracle table (see *ServiceUserOracle Table* on page 676)
- SoftwareDetails table (see *SoftwareDetails Table* on page 676)
- SoftwareFile table (see *SoftwareFile Table* on page 677)
- SoftwareFileName table (see *SoftwareFileName Table* on page 679)
- SoftwareFilePath table (see *SoftwareFilePath Table* on page 679)
- SoftwareFileProperty table (see *SoftwareFileProperty Table* on page 679)
- SoftwareIsoTagEntity table (see *SoftwareIsoTagEntity Table* on page 680)
- SoftwareIsoTagFile table (see *SoftwareIsoTagFile Table* on page 681)

- SoftwareIsoTagSoftwareVersion table (see *SoftwareIsoTagSoftwareVersion Table* on page 683)
- SoftwareIsoTagUnique table (see *SoftwareIsoTagUnique Table* on page 684)
- SoftwareOccurrence table (see *SoftwareOccurrence Table* on page 684)
- SoftwareOccurrenceSoftwareIsoTagFile table (see *SoftwareOccurrenceSoftwareIsoTagFile Table* on page 686)
- SoftwareProperty table (see *SoftwareProperty Table* on page 687)
- SoftwareValue table (see *SoftwareValue Table* on page 687)
- SoftwareVersion table (see *SoftwareVersion Table* on page 687)
- VirtualDesktopAccess table (see *VirtualDesktopAccess Table* on page 688)
- VirtualDesktopApplicationUsage table (see *VirtualDesktopApplicationUsage Table* on page 689)
- VirtualDesktopGroupAccess table (see *VirtualDesktopGroupAccess Table* on page 690)
- VirtualDesktopGroupAccessScan table (see *VirtualDesktopGroupAccessScan Table* on page 690)

## HardwareClass Table

`HardwareClass` contains a record for every class of hardware object found during hardware inventories, including mainly the WMI classes

**Table 658: Database columns for `HardwareClass` table**

Database Column	Details
<code>HardwareClassID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
<code>Class</code>	<i>Type:</i> text (max 256 characters). Key Hardware Class name
<code>SuperClassID</code>	<i>Type:</i> integer. Key. Nullable Reference to superclass, if any (and known)

## HardwareObject Table

The `HardwareObject` table entries describe a specific configuration item (usually a piece of physical hardware) associated with a computer. The information is represented in the database as Windows Management Instrumentation (WMI) classes.





**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 659: Database columns for `HardwareObject` table**

Database Column	Details
<code>HardwareObjectID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
<code>ComputerID</code>	<i>Type:</i> integer. Key The computer on which the hardware was found. It is a foreign key into the <code>Computer</code> table.
<code>HardwareName</code>	<i>Type:</i> text (max 256 characters). Key The hardware name as reported by the system.
<code>Occurrence</code>	<i>Type:</i> integer. Key The distinguishing identifier for the hardware. For example, if a computer has more than one memory card with the same <code>Class</code> and <code>HardwareName</code> , each memory card is assigned an <code>Occurrence</code> value (0, 1, 2...).
<code>HardwareClassID</code>	<i>Type:</i> integer. Key The id for the <code>HardwareClass</code> of the object.

## HardwareProperty Table

The `HardwareProperty` table provides property names and values for each hardware object. The information is represented in the database as Windows Management Instrumentation (WMI) properties.

**Table 660: Database columns for `HardwareProperty` table**

Database Column	Details
<code>HardwarePropertyID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
<code>Property</code>	<i>Type:</i> text (max 256 characters). Key The hardware property. A single hardware object can have many properties.

## HardwareValue Table

The value of a specified `HardwareProperty` of the specified `HardwareObject`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 661: Database columns for HardwareValue table**

Database Column	Details
HardwareObjectID	Type: integer. Key Object.
HardwarePropertyID	Type: integer. Key Property.
Value	Type: text (max 256 characters). Nullable Property value.

## InventoryReport Table

The `InventoryReport` table contains a record of every user and computer that has reported hardware or software inventory. It details the date and time when the hardware or software tracking was performed.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 662: Database columns for InventoryReport table**

Database Column	Details
ComputerID	Type: integer. Key  The computer that the inventory was tracked on. This is a foreign key into the <code>Computer</code> table.
UserID	Type: integer. Key

Database Column	Details
	The user for whom inventory was tracked. For computer inventory, the <code>UserID</code> is 1 (system user). This is a foreign key into the <code>User</code> table.
<code>SWDate</code>	<i>Type:</i> datetime. Nullable The time software was tracked, or is NULL if no tracking is recorded.
<code>HWDate</code>	<i>Type:</i> datetime. Nullable The time hardware was tracked, or is NULL if no tracking is recorded.
<code>FilesDate</code>	<i>Type:</i> datetime. Nullable The time files were tracked, or is NULL if no tracking is recorded.
<code>ServicesDate</code>	<i>Type:</i> datetime. Nullable The time Oracle services were tracked, or is NULL if no tracking is recorded.
<code>VMwareServicesDate</code>	<i>Type:</i> datetime. Nullable The time VMware services were tracked, or is NULL if no tracking is recorded.
<code>SequenceNumber</code>	<i>Type:</i> integer. Nullable Used when generating a differential inventory.
<code>OVMMDate</code>	<i>Type:</i> datetime. Nullable The time Oracle VM manager was interrogated, or is NULL if no interrogation is recorded.

## ServiceComponent Table

A software component installed to implement a `ServiceProvider`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 663: Database columns for ServiceComponent table**

Database Column	Details
<code>ServiceComponentID</code>	<i>Type:</i> integer. Key. Generated ID Unique ID for the service component.

Database Column	Details
Name	<i>Type:</i> text (max 128 characters). Key The name of the service component.
Version	<i>Type:</i> text (max 32 characters). Key The version of the service component.
Publisher	<i>Type:</i> text (max 128 characters). Key The publisher of the service component.
Edition	<i>Type:</i> text (max 128 characters). Key The edition of the service component.

## ServiceProvider Table

The inventoried providers of services.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 664: Database columns for ServiceProvider table**

Database Column	Details
ServiceProviderID	<i>Type:</i> integer. Key. Generated ID Unique ID for the service provider.
ComputerID	<i>Type:</i> integer. Key The <i>Computer</i> this service provider is hosted by.
ParentServiceProviderID	<i>Type:</i> integer. Nullable The <i>ServiceProvider</i> this provider is parented by.
ServiceProviderTypeID	<i>Type:</i> integer. Key The <i>ServiceProviderType</i> of the service provider.
ServiceProviderNameID	<i>Type:</i> integer. Key The <i>ServiceProviderName</i> of the service provider.

Database Column	Details
LastInventoryDate	<i>Type:</i> datetime The date and time that the service provider was last inventoried.
LastInventoryResult	<i>Type:</i> integer. Nullable The error code returned when the service provider was last inventoried.
LastInventoryError	<i>Type:</i> text (max 256 characters). Nullable The error message returned when the service provider was last inventoried.
CreationDate	<i>Type:</i> datetime. Nullable The date and time that the service provider was created.
AuditEvidence	<i>Type:</i> binary. Nullable The Oracle LMS audit evidence in zip archive.

## ServiceProviderApplicationOracle Table

An Oracle application.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 665: Database columns for ServiceProviderApplicationOracle table**

Database Column	Details
ServiceProviderApplicationOracleID	<i>Type:</i> integer. Key. Generated ID Unique ID for the Oracle application.
ServiceProviderID	<i>Type:</i> integer. Key Unique ID for the service provider.
Name	<i>Type:</i> text (max 240 characters). Key The application name.
Users	<i>Type:</i> integer The number of users.

Database Column	Details
ApplicationID	<i>Type:</i> integer. Key The ID of the application as assigned by Oracle.

## ServiceProviderApplicationUsagePerMonth Table

A count of oracle application usage items per month.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 666: Database columns for ServiceProviderApplicationUsagePerMonth table**

Database Column	Details
ServiceProviderApplicationUsagePerMonthID	<i>Type:</i> integer. Key. Generated ID Unique ID for the Oracle per month summary count.
ServiceProviderID	<i>Type:</i> integer. Key Unique ID for the service provider.
ServiceProviderApplicationUsageTypeID	<i>Type:</i> integer. Key The <i>ServiceProviderApplicationUsageType</i> of the service provider application usage.
YearMonth	<i>Type:</i> datetime. Key The year and month of the count.
ItemsUsed	<i>Type:</i> integer The number of items used.

## ServiceProviderApplicationUsageType Table

The types of inventoried *ServiceProviderApplicationUsagePerMonth* items.

**Table 667: Database columns for ServiceProviderApplicationUsageType table**

Database Column	Details
ServiceProviderApplicationUsageTypeID	<i>Type:</i> integer. Key. Generated ID Unique ID for the service provider application usage item type.
Type	<i>Type:</i> text (max 128 characters). Key The type of a service provider application usage item.

## ServiceProviderApplicationUserOracle Table

An Oracle applications User.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 668: Database columns for ServiceProviderApplicationUserOracle table**

Database Column	Details
ServiceProviderApplicationUserOracleID	<i>Type:</i> integer. Key. Generated ID Unique ID for the Oracle application user.
ServiceProviderApplicationOracleID	<i>Type:</i> integer. Key The application this user is associated with.
UserID	<i>Type:</i> integer. Key The application users user ID.
UserName	<i>Type:</i> text (max 100 characters) The application users user name.
Description	<i>Type:</i> text (max 240 characters). Nullable The application users description.
EMail	<i>Type:</i> text (max 240 characters). Nullable The application users email address.

## ServiceProviderComponent Table

A software component installed to implement a `ServiceProvider`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 669: Database columns for ServiceProviderComponent table**

Database Column	Details
<code>ServiceProviderID</code>	<i>Type:</i> integer. Key The <code>ServiceProvider</code> this component is associated with.
<code>ServiceComponentID</code>	<i>Type:</i> integer. Key The <code>ServiceComponent</code> this provider is associated with.

## ServiceProviderComponentProperty Table

The `ServiceProviderComponentProperty` table provides property names and values for each service component on a provider.

**Table 670: Database columns for ServiceProviderComponentProperty table**

Database Column	Details
<code>ServiceProviderComponentPropertyID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
<code>Property</code>	<i>Type:</i> text (max 256 characters). Key The service component property. A single service component on a provider can have many properties.

## ServiceProviderComponentValue Table

The value of a specified `ServiceProviderComponentProperty` of the specified `ServiceProviderComponent`.





**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 671: Database columns for ServiceProviderComponentValue table**

Database Column	Details
ServiceProviderID	Type: integer. Key Service provider.
ServiceComponentID	Type: integer. Key Service component.
ServiceProvider ComponentPropertyID	Type: integer. Key Property.
Value	Type: text (max 256 characters). Nullable Property value.

## ServiceProviderName Table

The names of inventoried `ServiceProviders`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 672: Database columns for ServiceProviderName table**

Database Column	Details
ServiceProviderNameID	Type: integer. Key. Generated ID Unique ID for the service provider name.
Name	Type: text (max 128 characters). Key The name of a service provider.

## ServiceProviderProperty Table

The `ServiceProviderProperty` table provides property names and values for each service provider.

**Table 673: Database columns for ServiceProviderProperty table**

Database Column	Details
<code>ServiceProviderPropertyID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
<code>Property</code>	<i>Type:</i> text (max 256 characters). Key The service provider property. A single service provider can have many properties.

## ServiceProviderType Table

The types of inventoried `ServiceProviders`.

**Table 674: Database columns for ServiceProviderType table**

Database Column	Details
<code>ServiceProviderTypeID</code>	<i>Type:</i> integer. Key. Generated ID Unique ID for the service provider type.
<code>Type</code>	<i>Type:</i> text (max 128 characters). Key The type of a service provider.

## ServiceProviderValue Table

The value of a specified `ServiceProviderProperty` of the specified `ServiceProvider`.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 675: Database columns for ServiceProviderValue table**

Database Column	Details
ServiceProviderID	Type: integer. Key Service provider.
ServiceProviderPropertyID	Type: integer. Key Property.
Value	Type: text (max 256 characters). Nullable Property value.

## ServiceUser Table

A user that uses a ServiceProvider.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 676: Database columns for ServiceUser table**

Database Column	Details
ServiceUserID	Type: integer. Key. Generated ID Unique ID for the service user.
ServiceProviderID	Type: integer. Key The <i>ServiceProvider</i> this user is associated with.
Name	Type: text (max 128 characters). Key The name of the service user.
Description	Type: text (max 256 characters). Nullable A textual description of the service user.
AccountStatus	Type: text (max 256 characters). Nullable Current status of user account.
CreationDate	Type: datetime. Nullable

Database Column	Details
	Date and time when user was created.
LastLogonDate	Type: datetime. Nullable Date and time when user last logged on.

## ServiceUserOracle Table

A specific kind of `ServiceUser`, specifically an Oracle user.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 677: Database columns for ServiceUserOracle table**

Database Column	Details
ServiceUserOracleID	Type: integer. Key. Generated ID Unique ID for the Oracle service user.
ServiceUserID	Type: integer. Key The service user this user is associated with.
DefaultTablespace	Type: text (max 256 characters). Nullable The default tablespace for the user.
TempTablespace	Type: text (max 256 characters). Nullable The temporary tablespace for the user.

## SoftwareDetails Table

The `SoftwareDetails` table contains a record of detailed data for each `SoftwareOccurrence` found.

**Table 678: Database columns for SoftwareDetails table**

Database Column	Details
SoftwareDetailsID	Type: integer. Key. Generated ID

Database Column	Details
	The id for the software details. This is automatically generated by SQL Server.
RawSoftwareName	<i>Type:</i> text (max 128 characters). Key The name of the software defined by the vendor, unprocessed by FlexNet Manager Suite.
RawVersion	<i>Type:</i> text (max 32 characters). Key The version of the software defined by the vendor, unprocessed by FlexNet Manager Suite.
Publisher	<i>Type:</i> text (max 256 characters). Key The publisher of the software defined by the vendor.
ProductID	<i>Type:</i> text (max 256 characters). Key The MSI product ID of the software defined by the vendor.

## SoftwareFile Table

The `SoftwareFile` table contains a record for each file associated with an application on each computer. File tracking is not enabled by default. For more information on configuring which files to track, see the section about the Inventory Agent.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 679: Database columns for `SoftwareFile` table**

Database Column	Details
SoftwareFileID	<i>Type:</i> integer. Key. Generated ID The id for the software file. This is automatically generated by SQL Server.
ComputerID	<i>Type:</i> integer. Key The computer on which the file was tracked. This is a foreign key into the <code>Computer</code> table.
SoftwareID	<i>Type:</i> integer. Key. Nullable

Database Column	Details
	The software containing the file that was tracked. This is a foreign key into the <code>SoftwareVersion</code> table.
<code>SoftwareIsoTagFileID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The software ID tag content of the file. This is a foreign key into the <code>SoftwareIsoTagFile</code> table.</p>
<code>Version</code>	<p><i>Type:</i> text (max 32 characters). Nullable</p> <p>The version of the software file defined by the vendor.</p>
<code>MD5</code>	<p><i>Type:</i> text (max 32 characters)</p> <p>The file's MD5 digest.</p>
<code>Size</code>	<p><i>Type:</i> integer</p> <p>The file's size in bytes.</p>
<code>DateTime</code>	<p><i>Type:</i> datetime. Nullable</p> <p>The last date and time the file was modified on the computer.</p>
<code>FileVersion</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The file version of the software file defined by the vendor.</p>
<code>FileDescription</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The file description of the software file defined by the vendor.</p>
<code>Language</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The language of the software file defined by the vendor.</p>
<code>CompanyName</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The company name of the software file defined by the vendor.</p>
<code>SoftwareFilePathID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The full path to the file that was tracked, minus the filename. This is a foreign key into the <code>SoftwareFilePath</code> table.</p>
<code>SoftwareFileNameID</code>	<p><i>Type:</i> integer. Key. Nullable</p> <p>The name of the file that was tracked, minus the path. This is a foreign key into the <code>SoftwareFileName</code> table.</p>

## SoftwareFileName Table

The `SoftwareFileName` table contains a record for each unique file name for files captured in inventory.

**Table 680: Database columns for `SoftwareFileName` table**

Database Column	Details
<code>SoftwareFileNameID</code>	<i>Type:</i> integer. Key. Generated ID The id for the software file name. This is automatically generated by SQL Server.
<code>Name</code>	<i>Type:</i> text (max 400 characters). Key The name of a file captured in inventory, minus the path.
<code>CreationDate</code>	<i>Type:</i> datetime. Key The creation date of the <code>SoftwareFileName</code> which will be used to cleanup the older unused records.

## SoftwareFilePath Table

The `SoftwareFilePath` table contains a record for each unique file path for files captured in inventory.

**Table 681: Database columns for `SoftwareFilePath` table**

Database Column	Details
<code>SoftwareFilePathID</code>	<i>Type:</i> integer. Key. Generated ID The id for the software file path. This is automatically generated by SQL Server.
<code>Path</code>	<i>Type:</i> text (max 400 characters). Key The full path to a file captured in inventory, minus the filename.
<code>CreationDate</code>	<i>Type:</i> datetime. Key The creation date of the <code>SoftwareFilePath</code> which will be used to cleanup the older unused records.

## SoftwareFileProperty Table

The `SoftwareFileProperty` table provides property names and values for each software file object.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 682: Database columns for SoftwareFileProperty table**

Database Column	Details
SoftwareFileID	Type: integer. Key The SoftwareFile that this property belongs to
Name	Type: text (max 256 characters). Key The software file property name.
Value	Type: text (max 256 characters) The software file property value.

## SoftwareIsoTagEntity Table

The SoftwareIsoTagEntity table provides property names and values for each unique entities on software ID tags.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 683: Database columns for SoftwareIsoTagEntity table**

Database Column	Details
SoftwareIsoTagEntityID	Type: integer. Key. Generated ID The SoftwareIsoTagEntity table unique ID for each records.
RegID	Type: text (max 200 characters). Key The unique registration ID value of an entity in an software ID tag.
Name	Type: text (max 200 characters). Key The entity name value in a software ID tag.



## SoftwareIsoTagFile Table

The `SoftwareIsoTagFile` table provides property names and values for each Software ID Tag in a normalized manner.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 684: Database columns for `SoftwareIsoTagFile` table**

Database Column	Details
<code>SoftwareIsoTagFileID</code>	<i>Type:</i> integer. Key. Generated ID The <code>SoftwareIsoTagFile</code> that this property belongs to
<code>MD5</code>	<i>Type:</i> text (max 32 characters). Key The MD5 property value of software ID tag file.
<code>TagContent</code>	<i>Type:</i> text The actual content of the software id tag file.
<code>EntitlementRequiredIndicator</code>	<i>Type:</i> boolean. Nullable The entitlement required indicator value of the software ID tag.
<code>SoftwareIsoTagSoftwareVersionID</code>	<i>Type:</i> integer. Key. Nullable The product version and name identifier for this software. This is a foreign key into the <code>SoftwareIsoTagSoftwareVersion</code> table.
<code>SoftwareCreatorEntityID</code>	<i>Type:</i> integer. Key. Nullable The software creator related data for software ID tag. This is a foreign key into the <code>SoftwareIsoTagEntity</code> table.
<code>SoftwareLicensorEntityID</code>	<i>Type:</i> integer. Key. Nullable The software licensor related data for software ID tag. This is a foreign key into the <code>SoftwareIsoTagEntity</code> table.
<code>TagCreatorEntityID</code>	<i>Type:</i> integer. Key. Nullable The tag creator related data for software ID tag. This is a foreign key into the <code>SoftwareIsoTagEntity</code> table.
<code>OriginalArpGuid</code>	<i>Type:</i> text (max 200 characters). Nullable

Database Column	Details
	The original GUID of add-remove programs values of a repackaged software.
OriginalArpPublisher	<i>Type:</i> text (max 200 characters). Nullable The original publisher of add-remove programs values of a repackaged software.
OriginalArpDisplayName	<i>Type:</i> text (max 200 characters). Nullable The original display name of add-remove programs values of a repackaged software.
OriginalArpDisplayVersion	<i>Type:</i> text (max 200 characters). Nullable The original display version of add-remove programs values of a repackaged software.
CurrentArpGuid	<i>Type:</i> text (max 200 characters). Nullable The current GUID of add-remove programs values of a repackaged software.
CurrentArpPublisher	<i>Type:</i> text (max 200 characters). Nullable The current publisher of add-remove programs values of a repackaged software.
CurrentArpDisplayName	<i>Type:</i> text (max 200 characters). Nullable The current display name of add-remove programs values of a repackaged software.
CurrentArpDisplayVersion	<i>Type:</i> text (max 200 characters). Nullable The current display version of add-remove programs values of a repackaged software.
AdminStudioAppCatalogID	<i>Type:</i> text (max 200 characters). Nullable Application catalog ID of a repackaged application in AdminStudio.
IsValidSchema	<i>Type:</i> boolean. Nullable Whether the software id tag has valid schema.
IsValidSignature	<i>Type:</i> boolean. Nullable Whether the software id tag has valid digital signature.
ActivationStatus	<i>Type:</i> text (max 50 characters). Nullable The activation status value of software ID tag.
ChannelType	<i>Type:</i> text (max 200 characters). Nullable

Database Column	Details
	The channel type value of software ID tag.
SerialNumber	Type: text (max 200 characters). Nullable The serial number value of software ID tag.
ParseErrorMessage	Type: text (max 1000 characters). Nullable The message of the error occurred while reading the software iso tag file.

## SoftwareIsoTagSoftwareVersion Table

The `SoftwareIsoTagSoftwareVersion` table provides property names and values for each software ID tag unique product related data.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 685: Database columns for SoftwareIsoTagSoftwareVersion table**

Database Column	Details
SoftwareIsoTagSoftwareVersionID	Type: integer. Key. Generated ID The <code>SoftwareIsoTagSoftwareVersion</code> table unique ID for each records.
TagCreatorEntityID	Type: integer. Key The tag creator related data for software ID tag. This is a foreign key into the <code>SoftwareIsoTagEntity</code> table.
TagSoftwareUniqueID	Type: integer. Key. Nullable The software unique ID related data for software ID tag. This is a foreign key into the <code>SoftwareIsoTagUnique</code> table.
ProductTitle	Type: text (max 200 characters). Key The product title value for software ID tag.
ProductVersionName	Type: text (max 200 characters). Key The product version name value for software ID tag.
ProductVersionMajor	Type: integer. Key

Database Column	Details
	The major version value of software ID tag.
ProductVersionMinor	<i>Type:</i> integer. Key The minor version value of software ID tag.
ProductVersionBuild	<i>Type:</i> integer. Key The build version value of software ID tag.
ProductVersionReview	<i>Type:</i> integer. Key The review version value of software ID tag.

## SoftwareIsoTagUnique Table

The `SoftwareIsoTagUnique` table provides property names and values for each unique id on software ID tags.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 686: Database columns for `SoftwareIsoTagUnique` table**

Database Column	Details
SoftwareIsoTagUniqueID	<i>Type:</i> integer. Key. Generated ID The <code>SoftwareIsoTagUniqueID</code> table unique ID for each records.
UniqueID	<i>Type:</i> text (max 200 characters). Key The unique ID value of a software ID tag.

## SoftwareOccurrence Table

The `SoftwareOccurrence` table contains the list (by computer and user) of applications that are installed. The applications may not have been installed through FlexNet Manager Suite. The information is obtained from managed devices from:

- FlexNet Manager Suite  
packages cache
- Add/Remove Programs registry entries

- Microsoft Installer
- ProductVersion resource strings in program files, if files are tracked



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 687: Database columns for SoftwareOccurrence table**

Database Column	Details
SoftwareOccurrenceID	<p>Type: integer. Key. Generated ID</p> <p>The id for the software occurrence. This is automatically generated by SQL Server.</p>
ComputerID	<p>Type: integer. Key</p> <p>The computer on which the software was tracked. For user inventory, this is the computer that the user was logged on to at the time of the Generate Inventory event. This is a foreign key into the <code>Computer</code> table.</p>
UserID	<p>Type: integer. Key</p> <p>User for whom the SoftwareVersion was installed. This is a foreign key to the <code>User</code> table.</p>
SoftwareID	<p>Type: integer. Key</p> <p>The software that has been tracked. This is a foreign key to the <code>SoftwareVersion</code> table.</p>
SoftwareDetailsID	<p>Type: integer. Key</p> <p>The details that have been tracked. This is a foreign key to the <code>SoftwareDetails</code> table.</p>
Evidence	<p>Type: text (max 32 characters). Nullable</p> <p>An indication of how the software was determined to be on the managed device. The valid entries are:</p> <ul style="list-style-type: none"> <li>• msi</li> <li>• managesoft</li> <li>• uninstall</li> <li>• exehdr (for file tracking only)</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>dllhdr (for file tracking only)</li> </ul>
PackagePathID	Type: integer. Key. Nullable FlexNet Manager Suite PackageFullName if known (not always!).
PolicyGUID	Type: binary (max 16 bytes). Nullable FlexNet Manager Suite Policy GUID if known.
InstallationDate	Type: datetime. Nullable The date and time that the software was installed.

## SoftwareOccurrenceSoftwareIsoTagFile Table

The SoftwareOccurrenceSoftwareIsoTagFile table is link table joining records in SoftwareOccurrence and SoftwareIsoTagFile tables.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 688: Database columns for SoftwareOccurrenceSoftwareIsoTagFile table**

Database Column	Details
SoftwareOccurrenceSoftwareIsoTagFileID	Type: integer. Key. Generated ID The SoftwareOccurrenceSoftwareIsoTagFile table unique ID for each records.
SoftwareOccurrenceID	Type: integer. Key This is a foreign key into the SoftwareOccurrence table.
SoftwareIsoTagFileID	Type: integer. Key This is a foreign key into the SoftwareIsoTagFile table.

## SoftwareProperty Table

The `SoftwareProperty` table contains a record for each unique property name captured in inventory.

**Table 689: Database columns for `SoftwareProperty` table**

Database Column	Details
<code>SoftwarePropertyID</code>	<i>Type:</i> integer. Key. Generated ID The id for the software property. This is automatically generated by SQL Server.
<code>Property</code>	<i>Type:</i> text (max 256 characters). Key The software property. A single software object can have many properties.

## SoftwareValue Table

The value of a specified `SoftwareProperty` of the specified `SoftwareOccurrence`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 690: Database columns for `SoftwareValue` table**

Database Column	Details
<code>SoftwareOccurrenceID</code>	<i>Type:</i> integer. Key Object.
<code>SoftwarePropertyID</code>	<i>Type:</i> integer. Key Property.
<code>Value</code>	<i>Type:</i> text (max 256 characters). Nullable Property value.

## SoftwareVersion Table

The `SoftwareVersion` table contains a record for each software name/version combination returned through inventory. The software names and versions are gathered from places such as Add/Remove Programs on

managed devices. They do not represent package names and versions from the software library, although correlation is likely.

**Table 691: Database columns for SoftwareVersion table**

Database Column	Details
SoftwareID	<i>Type:</i> integer. Key. Generated ID The id for the software version. This is automatically generated by SQL Server.
SoftwareName	<i>Type:</i> text (max 128 characters). Key The name of the software defined by the vendor.
Version	<i>Type:</i> text (max 32 characters). Key The version of the software defined by the vendor.
CreationDate	<i>Type:</i> datetime. Key The creation date of the SoftwareVersion which will be used to cleanup the older unused records.

## VirtualDesktopAccess Table

A VDI device a `User` has accessed on an end-point.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 692: Database columns for VirtualDesktopAccess table**

Database Column	Details
ComputerID	<i>Type:</i> integer. Key The end-point ComputerID. This is a foreign key into the <code>Computer</code> table.
UserID	<i>Type:</i> integer. Key The ID for the user accessing the VDI device. This is a foreign key into the <code>User</code> table.
MachineName	<i>Type:</i> text (max 64 characters). Key Computer name of the VDI device.



Database Column	Details
MachineDomain	<i>Type:</i> text (max 256 characters). Key. Nullable Fully qualified domain of the VDI device.
VDITemplateName	<i>Type:</i> text (max 256 characters). Key The template from which the VDI device was cloned.
Type	<i>Type:</i> text (max 64 characters). Key The type of VDI.
LogonTime	<i>Type:</i> datetime. Key The time the user logged on to the VDI device.
VirtualDesktopAccessID	<i>Type:</i> integer. Key. Generated ID The ID of the user session to the VDI device.

## VirtualDesktopApplicationUsage Table

A virtualized application is used from VDI.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 693: Database columns for VirtualDesktopApplicationUsage table**

Database Column	Details
VirtualDesktopApplicationUsageID	<i>Type:</i> integer. Key. Generated ID The ID of the application usage record.
VirtualDesktopAccessID	<i>Type:</i> integer. Key The ID of the corresponding VDI access record. This is a foreign key into the <i>VirtualDesktopAccess</i> table.
Name	<i>Type:</i> text (max 64 characters). Key The display name of the virtual application.
Version	<i>Type:</i> text (max 16 characters). Key

Database Column	Details
	The version of the virtual application.
PackageGUID	<i>Type:</i> unique identifier. Key The GUID of the package that the virtual application is associated with.
LastLaunchOnSystem	<i>Type:</i> datetime The last date and time that the virtual application was launched.
AccessMode	<i>Type:</i> text (max 100 characters). Key The access mode for the application.

## VirtualDesktopGroupAccess Table

A user with access to a particular VDI Group for a given site.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 694: Database columns for VirtualDesktopGroupAccess table**

Database Column	Details
VDISiteName	<i>Type:</i> text (max 256 characters). Key the VDI Site.
VDIGroupName	<i>Type:</i> text (max 256 characters). Key The name of the VDI Group.
Sid	<i>Type:</i> text (max 512 characters). Key. Nullable The Sid of the user.
VDIBrokerType	<i>Type:</i> text (max 64 characters). Key The type of VDI infrastructure.

## VirtualDesktopGroupAccessScan Table

The last scan time of the VDI to retrieve ACL information



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 695: Database columns for VirtualDesktopGroupAccessScan table**

Database Column	Details
VDIBrokerType	Type: text (max 64 characters). Key The type of VDI.
VDISiteName	Type: text (max 256 characters). Key The VDI Site.
ScanTime	Type: datetime The VDI Site.

## Licensing Tables

The complete set of database tables documented here includes:

- LicenseAllocation table (see *LicenseAllocation Table* on page 691)
- LicenseModel table (see *LicenseModel Table* on page 692)
- LicensePurchase table (see *LicensePurchase Table* on page 692)
- ProductContainsSoftware table (see *ProductContainsSoftware Table* on page 693)
- SoftwareProduct table (see *SoftwareProduct Table* on page 694)
- SoftwarePublisher table (see *SoftwarePublisher Table* on page 695)
- SoftwareReseller table (see *SoftwareReseller Table* on page 695)

## LicenseAllocation Table

The `LicenseAllocation` table specifies the allocation of licenses for each organizational unit. The same licensable product definition (from `SoftwareProduct`) may have license allocations for more than one organizational unit.

**Table 696: Database columns for LicenseAllocation table**

Database Column	Details
AllocationID	<i>Type:</i> integer. Key. Generated ID Unique identifier for the license allocation record. This is automatically generated by SQL Server.
SoftwareProductID	<i>Type:</i> integer. Key The license that maps to an application. This is a foreign key into the <code>SoftwareProduct</code> table.
OrganizationID	<i>Type:</i> integer. Key Id of the organizational unit to which the software is allocated.
UnitsAllocated	<i>Type:</i> integer. Nullable The number of units allocated for the application.
Expiry	<i>Type:</i> datetime. Nullable The date and time that the license allocation expires.

## LicenseModel Table

The `LicenseModel` table defines the license models available (for example, Site license). Each licensable product (listed in `SoftwareProduct`) is assigned a license model. A license model may apply to multiple licensable products.

**Table 697: Database columns for LicenseModel table**

Database Column	Details
ModelID	<i>Type:</i> integer. Key. Generated ID The unique identifier for a license model.
Name	<i>Type:</i> text (max 256 characters). Key The name of the license model.

## LicensePurchase Table

`LicensePurchase` records details of purchases of licenses for a specified `SoftwareProduct`.

**Table 698: Database columns for LicensePurchase table**

Database Column	Details
SoftwareProductID	<i>Type:</i> integer. Key The <code>SoftwareProduct</code> purchased.
ResellerID	<i>Type:</i> integer. Key The <code>Reseller</code> from which the software product was purchased.
OrganizationID	<i>Type:</i> integer. Key The organizational unit that owns the license for the product.
Purchased	<i>Type:</i> datetime. Key When the purchase was made.
Expires	<i>Type:</i> datetime. Nullable When the license expires.
Price	<i>Type:</i> integer. Nullable The price paid for the license.
Quantity	<i>Type:</i> integer Number of units licensed.
OrderNumber	<i>Type:</i> text (max 32 characters). Key Cross-reference to customer's purchase order number.

## ProductContainsSoftware Table

The `ProductContainsSoftware` table lists the applications returned by inventory (in the `SoftwareVersion` table) that are covered by licensable products (listed in `SoftwareProduct`). A license can map to multiple applications: if any of these applications is installed, a license is required.

**Table 699: Database columns for ProductContainsSoftware table**

Database Column	Details
SoftwareProductID	<i>Type:</i> integer. Key The license that maps to an application. This is a foreign key into the <code>SoftwareProduct</code> table.

Database Column	Details
SoftwareVersionID	<p><i>Type:</i> integer. Key</p> <p>The application maps to the license. This is a foreign key into the <code>SoftwareVersion</code> table.</p>

## SoftwareProduct Table

The `SoftwareProduct` table contains all of the licensable products (license definitions) for an organization. It represents all of the license agreements available for monitoring.

**Table 700: Database columns for `SoftwareProduct` table**

Database Column	Details
SoftwareProductID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>This is a unique identifier for the software product.</p>
ProductName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The name of the license. This normally corresponds to the name of the software product as defined by the vendor.</p>
ModelID	<p><i>Type:</i> integer. Key</p> <p>Reference to the Licensing model for FlexNet Manager Suite</p>
TrackedByID	<p><i>Type:</i> integer</p> <p>In what units are Licences counted?</p>
PublisherID	<p><i>Type:</i> integer. Key</p> <p>Reference to publisher</p>
Agreement	<p><i>Type:</i> text (max 256 characters)</p> <p>A URL to the license agreement for the product.[Comments]</p>
Comments	<p><i>Type:</i> text. Nullable</p> <p>Additional comments</p>

## SoftwarePublisher Table

The `SoftwarePublisher` table lists application publishers (for example, Microsoft). Each licensable product (listed in `SoftwareProduct`) is assigned a publisher. A publisher may be assigned to multiple licensable products.

**Table 701: Database columns for `SoftwarePublisher` table**

Database Column	Details
<code>PublisherID</code>	<i>Type:</i> integer. Key. Generated ID The unique identifier for a publisher.
<code>Name</code>	<i>Type:</i> text (max 256 characters). Key The name of the publisher.
<code>SupportURL</code>	<i>Type:</i> text (max 256 characters) The support URL.
<code>SupportPhone</code>	<i>Type:</i> text (max 256 characters) The support phone number.
<code>ContactName</code>	<i>Type:</i> text (max 256 characters) The name of the contact.
<code>Comments</code>	<i>Type:</i> text (max 512 characters) An arbitrary comment about the publisher.

## SoftwareReseller Table

The `SoftwareReseller` table lists application resellers (usually the organization listed on the purchase order for the product). Each licensable product (listed in `SoftwareProduct`) is assigned an application reseller. A reseller may be assigned to multiple licensable products.

**Table 702: Database columns for `SoftwareReseller` table**

Database Column	Details
<code>ResellerID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identifier of <code>Reseller</code>
<code>Name</code>	<i>Type:</i> text (max 256 characters). Key

Database Column	Details
	The name of the reseller.
ContactName	Type: text (max 256 characters) The name of the sales contact.
ContactPhone	Type: text (max 256 characters) The contact phone number.
Comments	Type: text (max 512 characters) An arbitrary comment about the reseller.

## ManageSoft Tables

The complete set of database tables documented here includes:

- DatabaseConfiguration table (see *DatabaseConfiguration Table* on page 696)

## DatabaseConfiguration Table

The DatabaseConfiguration table contains configuration properties for the FlexNet Manager Suite database tables, which are used for ongoing maintenance of the database.

**Table 703: Database columns for DatabaseConfiguration table**

Database Column	Details
Property	Type: text (max 32 characters). Key The name of the property.
Value	Type: text (max 256 characters) The value of the property.
Created	Type: datetime The date and time the property was created.
LastUpdate	Type: datetime The date and time the property was last updated.



# Networking Tables

The complete set of database tables documented here includes:

- NetworkLocation table (see *NetworkLocation Table* on page 697)
- Subnet table (see *Subnet Table* on page 698)

## NetworkLocation Table

The Location table contains data about Locations



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 704: Database columns for NetworkLocation table**

Database Column	Details
NetworkLocationID	<i>Type:</i> integer. Key. Generated ID The ID for the Location
Name	<i>Type:</i> text (max 256 characters). Key The name of the Location
DN	<i>Type:</i> text (max 1024 characters). Key. Nullable The Distinguished name of the Location
AutoPopulated	<i>Type:</i> boolean Specifies whether the row was populated automatically(1) or manually(0).
Enabled	<i>Type:</i> boolean Specifies whether the row will be used when mapping domains and devices to Locations
DomainID	<i>Type:</i> integer. Key DomainID of the domain in which the NetworkLocation resides

# Subnet Table

The Subnet table contains data about subnets in a location.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 705: Database columns for Subnet table**

Database Column	Details
SubnetID	Type: integer. Key. Generated ID The ID for the Subnet
IPSubnet	Type: text (max 64 characters). Key The IPSubnet of the Subnet
IPSubnetMask	Type: text (max 64 characters). Key The IPSubnetMask of the Subnet
NetworkLocationID	Type: integer. Key NetworkLocationID of the NetworkLocation in which the Subnet resides
AutoPopulated	Type: boolean Specifies whether the row was populated automatically(1) or manually(0).
Enabled	Type: boolean Specifies whether the row will be used when mapping domains and devices to Locations

## Packaging Tables

The complete set of database tables documented here includes:

- Architecture table (see *Architecture Table* on page 699)
- FileNameMap table (see *FileNameMap Table* on page 699)
- Media table (see *Media Table* on page 700)
- MediaContainsPackagePath table (see *MediaContainsPackagePath Table* on page 700)

- MediaContainsPackageVersion table (see *MediaContainsPackageVersion Table* on page 701)
- MediaType table (see *MediaType Table* on page 701)
- PackageFamily table (see *PackageFamily Table* on page 702)
- PackagePath table (see *PackagePath Table* on page 702)
- PackagePathType table (see *PackagePathType Table* on page 703)
- PackageProvides table (see *PackageProvides Table* on page 703)
- PackageRequires table (see *PackageRequires Table* on page 703)
- PackageState table (see *PackageState Table* on page 704)
- PackageVersion table (see *PackageVersion Table* on page 705)
- PackageVersionArchitecture table (see *PackageVersionArchitecture Table* on page 706)
- PackageVersionEnvironment table (see *PackageVersionEnvironment Table* on page 706)
- PackageVersionInState table (see *PackageVersionInState Table* on page 706)
- PackageVersionLocale table (see *PackageVersionLocale Table* on page 707)

## Architecture Table

*Architecture* identifies a target CPU (ABI), used to identify on what type of computer a package may be installed.

**Table 706: Database columns for Architecture table**

Database Column	Details
ArchitectureID	Type: integer. Key. Generated ID Auto-generated identity number
ArchitectureName	Type: text (max 64 characters). Key Name of the computer architecture

## FileNameMap Table

Stores mappings from a file on disk to a filename that will be used on the managed device. Currently used by the HPUX wizards to rename files that have non-Windows conforming names.

**Table 707: Database columns for FileNameMap table**

Database Column	Details
MediaID	<i>Type:</i> integer. Key The <code>Media</code> that the <code>SourceFile</code> exists on.
SourceFile	<i>Type:</i> text (max 256 characters). Key The file to be renamed.
DestFile	<i>Type:</i> text (max 256 characters) The final file name.
IsFile	<i>Type:</i> boolean Boolean field that specifies whether the row refers to a file or a directory.

## Media Table

Packages are stored on `Media` identified in this table.

**Table 708: Database columns for Media table**

Database Column	Details
MediaID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number, 1 = local administration server.
Title	<i>Type:</i> text (max 128 characters). Key Name of media (empty for local administration server).
MediaTypeID	<i>Type:</i> integer. Key What type of media?
Location	<i>Type:</i> text (max 256 characters). Nullable Where on the media?

## MediaContainsPackagePath Table

This table identifies which `Media` contains which `PackagePath`. A record exists here at least for every `PackagePath` currently in the local administration server.

**Table 709: Database columns for MediaContainsPackagePath table**

Database Column	Details
MediaID	<i>Type:</i> integer. Key What Media contains the package?
PackagePathID	<i>Type:</i> integer. Key What PackagePath?

## MediaContainsPackageVersion Table

This table identifies which Media contains which PackageVersion. A record exists here at least for every PackageVersion currently in the local administration server.

**Table 710: Database columns for MediaContainsPackageVersion table**

Database Column	Details
MediaID	<i>Type:</i> integer. Key What Media contains the package?
PackageVersionID	<i>Type:</i> integer. Key What PackageVersion?

## MediaType Table

Packages are stored on Media of various types. This table contains a record for each type.

**Table 711: Database columns for MediaType table**

Database Column	Details
MediaTypeID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number, 1 = Warehouse (administration server).
Description	<i>Type:</i> text (max 128 characters). Key Media type name (for example: Warehouse, Filesystem, CD).

## PackageFamily Table

`PackageFamily` is a short name used by the client to decide where a package to be downloaded to and whether it's an upgrade or downgrade of a previous package. Only one package version of a family may be installed in a given context.

**Table 712: Database columns for `PackageFamily` table**

Database Column	Details
<code>PackageFamilyID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
<code>PackageName</code>	<i>Type:</i> text (max 64 characters). Key Package family name

## PackagePath Table

Package Path identifies a filesystem path where the package will be stored in the software library and in transit. As such, the administration server and each distribution server may only contain a single package version having a given Path

**Table 713: Database columns for `PackagePath` table**

Database Column	Details
<code>PackagePathID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
<code>PackageFullName</code>	<i>Type:</i> text (max 256 characters). Key. Nullable Package Path string
<code>ParentPathID</code>	<i>Type:</i> integer. Key. Nullable When a package is a variant of a parent package, this contains a reference to the parent package's path.
<code>PackagePathTypeID</code>	<i>Type:</i> integer The type of <code>PackagePath</code> that this row represents.

## PackagePathType Table

This table contains the list of different types of packages, which also corresponds to the main areas in the Software Library.

**Table 714: Database columns for PackagePathType table**

Database Column	Details
PackagePathTypeID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
Description	<i>Type:</i> text (max 128 characters). Key This describes the type of the package

## PackageProvides Table

PackageProvides is used when a package can satisfy a virtual dependency, like “web-browser”.

**Table 715: Database columns for PackageProvides table**

Database Column	Details
PackageProvidesID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
PackageVersionID	<i>Type:</i> integer. Key The package which provides the interface
PackageFamilyID	<i>Type:</i> integer. Key The (virtual) package which is provided
Version	<i>Type:</i> text (max 32 characters). Key. Nullable The version provided, if necessary

## PackageRequires Table

PackageRequires is used when a package requires another package or some other configuration, like a piece of hardware for example.

**Table 716: Database columns for PackageRequires table**

Database Column	Details
PackageVersionID	<i>Type:</i> integer. Key The <code>PackageVersion</code> which has the requirement.
RequiredType	<i>Type:</i> text (max 8 characters). Key Requirement type: for example, software, hardware.
RequiredObject	<i>Type:</i> text (max 64 characters). Key Required object: for example, <code>PackageFamily</code> name.
Strength	<i>Type:</i> integer. Nullable Strength of the requirement.
Property	<i>Type:</i> text (max 64 characters). Nullable The required property of the object (for example, package version).
Value	<i>Type:</i> text (max 64 characters). Nullable The value of the required property.
Match	<i>Type:</i> integer. Key How to match the required value.

## PackageState Table

This table contains the package states that may be assigned to a package in the software library. The default set of states are based on ITIL release management processes. The state names are internationalized when displayed on the MMC console.

**Table 717: Database columns for PackageState table**

Database Column	Details
PackageStateID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
Name	<i>Type:</i> text (max 64 characters). Key Package State Name
CanAddToPolicy	<i>Type:</i> boolean



Database Column	Details
	Whether a package in this state can be added to policy

## PackageVersion Table

The `PackageVersion` table contains information about all of the packages in the software library. It is primarily used to map between `Installation` and `PackageApplies` for the purpose of comparing what users and computers should have versus what they actually have installed. This table only stores the details of one version of each package. This will change in future releases.

**Table 718: Database columns for `PackageVersion` table**

Database Column	Details
<code>PackageVersionID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
<code>PackagePathID</code>	<i>Type:</i> integer. Key Reference to Path (Full name) of Package
<code>Version</code>	<i>Type:</i> text (max 32 characters). Key The version number of the package. The <code>Installation</code> table also has <code>PackageName</code> and <code>Version</code> columns. This value can be used to find the corresponding <code>PackageFullName</code> so that <code>Installation</code> can be mapped to <code>PackageApplies</code> .
<code>Update</code>	<i>Type:</i> text (max 64 characters). Key The current update (or patch) number of the package
<code>PackageFamilyID</code>	<i>Type:</i> integer. Key A managed device may only have one <code>PackageVersion</code> in a family.
<code>Title</code>	<i>Type:</i> text (max 64 characters). Nullable The friendly name for the package.
<code>MD5</code>	<i>Type:</i> text (max 40 characters). Nullable The MD5 digest of the project file (.ndp) for the package. This is updated in the database when the package is packed or distributed.
<code>Size</code>	<i>Type:</i> integer. Nullable If set, contains the size in bytes of the distributable form of the package

Database Column	Details
Category	<i>Type:</i> text (max 128 characters). Nullable A category or class used to group packages

## PackageVersionArchitecture Table

`PackageVersionArchitecture` specifies all the architectures that a particular package version applies to.

**Table 719: Database columns for `PackageVersionArchitecture` table**

Database Column	Details
PackageVersionID	<i>Type:</i> integer. Key Foreign key into the <code>PackageVersion</code> table.
ArchitectureID	<i>Type:</i> integer. Key Foreign key into the <code>Architecture</code> table.

## PackageVersionEnvironment Table

`PackageVersionEnvironment` specifies all the environments (operating systems) that a particular package version applies to.

**Table 720: Database columns for `PackageVersionEnvironment` table**

Database Column	Details
PackageVersionID	<i>Type:</i> integer. Key Foreign key into the <code>PackageVersion</code> table.
Environment	<i>Type:</i> text (max 128 characters). Key Name of the environment that is used in the package. This refers to the environments used in the Packer.

## PackageVersionInState Table

This table contains a history of changes made to the state of a package. Note that the username is recorded as a `nvarchar` rather than a foreign key to the user table so that if a user is deleted, there is still a record of the changes that were made.

**Table 721: Database columns for PackageVersionInState table**

Database Column	Details
PackageVersionID	<i>Type:</i> integer. Key The package that has been changed
PackageStateID	<i>Type:</i> integer. Key The state that was set
UserName	<i>Type:</i> text (max 64 characters). Key The user that made the state change
Changed	<i>Type:</i> datetime. Key The date/time that the change was made
Comments	<i>Type:</i> text (max 256 characters) A user defined set of comments relating to the state change

## PackageVersionLocale Table

`PackageVersionLocale` specifies all the locales (language and country combinations) that a particular package version applies to.

**Table 722: Database columns for PackageVersionLocale table**

Database Column	Details
PackageVersionID	<i>Type:</i> integer. Key Foreign key into the <code>PackageVersion</code> table.
LocaleCode	<i>Type:</i> text (max 6 characters). Key Foreign key into the <code>Locale</code> table.

## ReferenceData Tables

The complete set of database tables documented here includes:

- Country table (see *Country Table* on page 708)
- Language table (see *Language Table* on page 708)

- Locale table (see *Locale Table* on page 709)
- OperatingSystem table (see *OperatingSystem Table* on page 709)

## Country Table

Stores country information, including their ISO country code and English names.

**Table 723: Database columns for Country table**

Database Column	Details
CountryCode	<i>Type:</i> text (max 2 characters). Key The two letter country code.
Name	<i>Type:</i> text (max 128 characters). Key The english name of the country.

## Language Table

Stores language information, including their English names, and various forms of language id.

**Table 724: Database columns for Language table**

Database Column	Details
LangCode3	<i>Type:</i> text (max 3 characters). Key The three letter language code.
LangCode2	<i>Type:</i> text (max 2 characters). Nullable The two letter language code.
EnglishName	<i>Type:</i> text (max 128 characters). Key The english name of the language.
LocalName	<i>Type:</i> text (max 128 characters). Nullable The name of the language, written in the local language.
MSLanguageID	<i>Type:</i> integer. Nullable The Microsoft language id, as specified in winnt.h in the Platform SDK.

## Locale Table

Stores locale information, which consists of country and language combinations. Use the `LocaleCode` column as the foreign key into this table.

**Table 725: Database columns for `Locale` table**

Database Column	Details
<code>LocaleCode</code>	<i>Type:</i> text (max 6 characters). Key A combination of the language code and country code, separated by a hyphen. If there is no country code, then there will be no hyphen added. This column <b>MUST</b> have the correct value when inserted, based on the values of the language and country codes.
<code>LangCode3</code>	<i>Type:</i> text (max 3 characters). Key The three letter language code.
<code>CountryCode</code>	<i>Type:</i> text (max 2 characters). Key. Nullable The two letter country code.
<code>LocaleName</code>	<i>Type:</i> text (max 128 characters) The name of the locale. For example, "English (United States)".
<code>MSLocaleID</code>	<i>Type:</i> integer. Nullable The Microsoft identifier for the locale. For example, 1033 for English (United States).

## OperatingSystem Table

This table stores the information about different types of OS available on the network devices

**Table 726: Database columns for `OperatingSystem` table**

Database Column	Details
<code>OperatingSystemID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
<code>OperatingSystemName</code>	<i>Type:</i> text (max 128 characters). Key Name of operating system
<code>Category</code>	<i>Type:</i> integer. Nullable

Database Column	Details
	Reference to operating system category

## Rights Tables

The complete set of database tables documented here includes:

- ActionClass table (see *ActionClass Table* on page 710)
- PartitionType table (see *PartitionType Table* on page 710)
- Resource table (see *Resource Table* on page 711)

## ActionClass Table

The types of action on a `Resource` for which rights may be granted or denied.

**Table 727: Database columns for ActionClass table**

Database Column	Details
ActionClassID	Type: integer. Key. Generated ID Auto-generated identity number.
ActionClassName	Type: text (max 16 characters). Key The name of the ActionClass.

## PartitionType Table

Some secured `Resources` may be partitioned. Partitions are used to grant rights to one part of a `Resource` excluding other parts, for example limiting rights so that the operator can access only certain distribution servers, organizational units, or areas in the software library. There are three types of partitioning, defined by entries in this table.

**Table 728: Database columns for PartitionType table**

Database Column	Details
PartitionTypeID	Type: integer. Key. Generated ID Auto-generated identity number.
PartitionTypeName	Type: text (max 32 characters). Key

Database Column	Details
	Name of the <code>PartitionType</code> .

## Resource Table

Access rights are granted to the `Resources` defined in this table.

**Table 729: Database columns for Resource table**

Database Column	Details
<code>ResourceID</code>	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number.
<code>ResourceName</code>	<i>Type:</i> text (max 16 characters). Key Name of the <code>Resource</code> .
<code>PartitionTypeID</code>	<i>Type:</i> integer. Nullable If not NULL, the type of partitioning used with this <code>Resource</code> .

## Status Tables

The complete set of database tables documented here includes:

- `AMTEventLog` table (see *AMTEventLog Table* on page 711)

## AMTEventLog Table

Records the entries in the AMT event log for a `NetworkDevice`.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 730: Database columns for AMTEventLog table**

Database Column	Details
<code>AMTEventLogID</code>	<i>Type:</i> integer. Key. Generated ID

Database Column	Details
	Auto-generated identity number.
DeviceID	Type: integer. Key. Nullable NetworkDevice identity number.
Reported	Type: datetime Date and time the event log entry was reported at.
PETDeviceAddress	Type: small integer The device address from the PET message format.
PETEventSensorType	Type: small integer The event sensor type from the PET message format.
PETEventType	Type: small integer The event type from the PET message format.
PETEventOffset	Type: small integer The event offset from the PET message format.
PETEventSourceType	Type: small integer The event source type from the PET message format.
PETEventSeverity	Type: small integer The event severity from the PET message format.
PETSensorNumber	Type: small integer The sensor number from the PET message format.
PETEntity	Type: small integer The entity from the PET message format.
PETEntityInstance	Type: small integer The entity instance address from the PET message format.
PETEventData	Type: text (max 32 characters) The event data from the PET message format.



# Targeting Tables

The complete set of database tables documented here includes:

- TargetType table (see *TargetType Table* on page 713)

## TargetType Table

The `TargetType` table contains a row for each type of object that can be targeted in FlexNet Manager Suite.

**Table 731: Database columns for TargetType table**

Database Column	Details
TargetTypeID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>The ID for the target type:</p> <ul style="list-style-type: none"> <li>• Computers</li> <li>• Users</li> <li>• Group</li> <li>• DistributionLocation</li> <li>• DistributionServer</li> <li>• Organization</li> <li>• Assets</li> <li>• Contracts</li> <li>• Purchase orders</li> <li>• Software licenses</li> <li>• Software titles</li> <li>• Compliance computers</li> <li>• Compliance users</li> <li>• Operators</li> <li>• SAP system landscapes</li> <li>• SAP systems</li> <li>• SAP rule sets</li> <li>• Discovered devices</li> <li>• Beacon</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• Vendor</li> <li>• Device</li> <li>• Rule</li> <li>• Inventory connection</li> <li>• FNMP Server</li> <li>• Fast Import</li> <li>• OLE DB Connection</li> <li>• ORACLE Connection</li> <li>• XML</li> <li>• Intermediate File</li> <li>• ADSI Connection</li> <li>• Web Service</li> <li>• SQL Connection</li> <li>• Software Title Evidence</li> <li>• FNMEA Agent</li> <li>• Installed Software</li> <li>• Baseline Import</li> </ul>
TargetTypeName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The name of the target type.</p>

## Tenants Tables

The complete set of database tables documented here includes:

- FlexeraLicense table (see *FlexeraLicense Table* on page 714)
- Tenant table (see *Tenant Table* on page 715)

## FlexeraLicense Table

The `FlexeraLicense` table contains the encoded contents of the Flexera Software licenses required for the tenants in the system. This table is also used by the system in the single-tenant setup where there is only one tenant.

**Table 732: Database columns for FlexeraLicense table**

Database Column	Details
TenantUID	<i>Type:</i> text (max 40 characters). Key The unique identifier of a tenant. A reference to the <code>Tenant</code> to which this license is attached.
License	<i>Type:</i> text The encoded contents of the Flexera Software license attached to a particular <code>Tenant</code> .
LicenseChecksum	<i>Type:</i> integer. Key The check sum of the license.
LicenseDetails	<i>Type:</i> XML. Nullable XML definition of the license details

## Tenant Table

The `Tenant` table contains the details of each tenant in multitenant FlexNet Manager Suite database tables.

**Table 733: Database columns for Tenant table**

Database Column	Details
TenantID	<i>Type:</i> integer. Key. Generated ID The tenant ID in a multi-tenant database.
TenantUID	<i>Type:</i> text (max 40 characters). Key The unique identifier of a tenant. This identifier is used to identify the tenant in environments where tenant information is stored on multiple databases.
TenantName	<i>Type:</i> text (max 256 characters). Key The name of the tenant.
Comments	<i>Type:</i> text. Nullable Operator comments about this tenant record.
CreationUser	<i>Type:</i> text (max 128 characters). Nullable The operator who created the tenant record.

Database Column	Details
CreationDate	<i>Type:</i> datetime The date the tenant record was created.
UpdatedUser	<i>Type:</i> text (max 128 characters). Nullable The name of the operator who last updated the tenant record.
UpdatedDate	<i>Type:</i> datetime. Nullable The date the tenant record was last updated.

## Usage Tables

The complete set of database tables documented here includes:

- ComputerUsage table (see *ComputerUsage Table* on page 716)
- SoftwareFileUsage table (see *SoftwareFileUsage Table* on page 717)
- SoftwareUsagePerWeek table (see *SoftwareUsagePerWeek Table* on page 718)

## ComputerUsage Table

Each time usage information is received, the `ComputerUsage` table is updated with the current day's time-stamp.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 734: Database columns for `ComputerUsage` table**

Database Column	Details
ComputerID	<i>Type:</i> integer. Key The id of the computer this information applies to. This id is a foreign key to the <code>Computer</code> table. It forms part of the unique index that identifies each row of data.
UserID	<i>Type:</i> integer. Key

Database Column	Details
	The id of the user context in which the application was detected. This is a foreign key to the <code>User</code> table. It forms part of the unique index that identifies each row of data.
LastReported	<i>Type:</i> datetime. Nullable  The date that the user last reported usage information from the specified computer.

## SoftwareFileUsage Table

This table contains information about each file relevant to reporting software usage information on each computer.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 735: Database columns for SoftwareFileUsage table**

Database Column	Details
SoftwareFileUsageID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
ComputerID	<i>Type:</i> integer. Key  The id of the computer this information applies to. This id is a foreign key to the <code>Computer</code> table. It forms part of the unique index that identifies each row of data.
UserID	<i>Type:</i> integer. Key  The id of the user context in which the application was detected. This is a foreign key to the <code>User</code> table. It forms part of the unique index that identifies each row of data.
Version	<i>Type:</i> text (max 32 characters). Key  The version of the software file defined by the vendor.
SoftwareFileNameID	<i>Type:</i> integer. Key  The name of the file that was tracked, minus the path. This is a foreign key into the <code>SoftwareFileName</code> table.

Database Column	Details
LongName	Type: text (max 4000 characters). Nullable The full path and file that was tracked.
CompanyName	Type: text (max 50 characters). Key The company name of the software.
Description	Type: text (max 1024 characters). Key The file description of the software.
ProductName	Type: text (max 50 characters). Key The product name of the software file.
ProductVersion	Type: text (max 32 characters). Key The version of the product of the software file defined by the vendor.

## SoftwareUsagePerWeek Table

Software usage information is stored in weekly batches. Information received by the server is stored in the `SoftwareUsagePerWeek` table. Each row in the table represents usage information received from a specified user, on a specified managed device, regarding usage of specified software, during the week where the Monday is the specified date.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 736: Database columns for `SoftwareUsagePerWeek` table**

Database Column	Details
SoftwareUsagePerWeekID	Type: integer. Key. Generated ID Auto-generated identity number
ComputerID	Type: integer. Key The id of the computer this information applies to. This id is a foreign key to the <code>Computer</code> table. It forms part of the unique-clustered-index that identifies each row of data.
UserID	Type: integer. Key

Database Column	Details
	The id of the user context in which the application was detected. This id is a foreign key to the <code>User</code> table. It forms part of the unique-clustered-index that identifies each row of data.
<code>SoftwareID</code>	<i>Type:</i> integer. Key  The id of the software that was used. This is a foreign key to the <code>SoftwareVersion</code> table. It forms part of the unique-clustered-index that identifies each row of data.
<code>SoftwareFileUsageID</code>	<i>Type:</i> integer. Key. Nullable  The id of the software file usage that was used. This is a foreign key to the <code>SoftwareFileUsage</code> table. It forms part of the unique-clustered-index that identifies each row of data.
<code>StartOfWeek</code>	<i>Type:</i> datetime. Key  The first day for the week. This date identifies the week that usage data applies to.
<code>Duration</code>	<i>Type:</i> integer. Nullable  The total duration, in seconds, that the application was run. It represents the total spanning across many sessions.
<code>ActiveTime</code>	<i>Type:</i> integer. Nullable  The total active time, in seconds, that the application was in the foreground. It represents the total spanning across many sessions.
<code>Sessions</code>	<i>Type:</i> integer. Nullable  The number of sessions the in which the application was used within the week.
<code>Days</code>	<i>Type:</i> integer. Nullable  The number of distinct days the application was used within the week.

## WakeOnLAN Tables

The complete set of database tables documented here includes:

- WakeOnLANDistributionJob table (see *WakeOnLANDistributionJob Table* on page 720)
- WakeOnLANStatus table (see *WakeOnLANStatus Table* on page 720)
- WakeOnLANTask table (see *WakeOnLANTask Table* on page 721)

## WakeOnLANDistributionJob Table

Wake on LAN distribution jobs control the distribution of a Wake on LAN task to the nearest distribution server for the targeted managed devices. The status of these distribution jobs is stored in the `WakeOnLANDistributionJob` table. Each row in the table represents a Wake on LAN job, which is any Wake on LAN task (or a subset of a Wake on LAN task), that has been distributed to a distribution server. Be aware: There can be multiple distribution jobs for a given Wake on LAN task.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 737: Database columns for WakeOnLANDistributionJob table**

Database Column	Details
DistJobUID	Type: binary (max 16 bytes). Key A unique identifier for this distribution job.
TaskUID	Type: binary (max 16 bytes). Key A unique identifier for the task that created this distribution job. This is a foreign key linked to the TaskUID in the WakeOnLANTask table.
ServerUID	Type: binary (max 16 bytes). Key A unique identifier for the distribution server that this distribution job targets. This foreign key links to the ServerUID in the DistributionServer table.
State	Type: text (max 16 characters) The state of this distribution job. This can be one of the following values: + Pending + Failed + Success

## WakeOnLANStatus Table

All managed devices targeted by a Wake on LAN task have a status associated with them. The status of the managed devices is stored in the `WakeOnLANStatus` table. Each row in the table represents a managed device to be woken by a Wake on LAN task from a distribution job.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.



**Table 738: Database columns for WakeOnLANStatus table**

Database Column	Details
DistJobUID	<p><i>Type:</i> binary (max 16 bytes). Key</p> <p>A unique identifier for a distribution job. This foreign key links to the DistJobUID in the WakeOnLANDistributionJob table. It forms part of the unique index that identifies each row of data.</p>
ComputerID	<p><i>Type:</i> integer. Key</p> <p>The id for the managed device. It forms part of the unique index that identifies each row of data.</p>
State	<p><i>Type:</i> text (max 16 characters)</p> <p>The state of this managed device. This can be one of the following values:</p> <ul style="list-style-type: none"> <li>• Pending</li> <li>• Failed</li> <li>• Woken</li> <li>• Awake</li> </ul>

## WakeOnLANTask Table

Wake on LAN tasks control any targeted managed devices. The details of these tasks are stored in the WakeOnLANTask table. Each row in the table represents a Wake on LAN task.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 739: Database columns for WakeOnLANTask table**

Database Column	Details
TaskUID	<p><i>Type:</i> binary (max 16 bytes). Key</p> <p>A unique identifier for the task that created a Wake on LAN job.</p>
FriendlyName	<p><i>Type:</i> text (max 400 characters)</p> <p>The descriptive name assigned to the Wake on LAN task.</p>
StartTime	<p><i>Type:</i> datetime. Nullable</p>

Database Column	Details
	The time at which the managed devices will be woken.

## WorkFlow Tables

The complete set of database tables documented here includes:

- Action table (see *Action Table* on page 722)
- ActionApplies table (see *ActionApplies Table* on page 723)
- ActionState table (see *ActionState Table* on page 724)
- Job table (see *Job Table* on page 725)
- Task table (see *Task Table* on page 726)
- TaskSchedule table (see *TaskSchedule Table* on page 726)
- TaskType table (see *TaskType Table* on page 727)

## Action Table

An **Action** arising from a **Task**, to be applied (possibly repeatedly) by an actor (often a distribution server) to a set of target devices.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 740: Database columns for Action table**

Database Column	Details
ActionUID	Type: binary (max 16 bytes). Key The unique identifier for the Action.
TaskID	Type: integer. Key The Task which gave rise to this Action.
ServerUID	Type: binary (max 16 bytes). Key. Nullable True if this Action has been delegated to a distribution server.

Database Column	Details
JobUID	<i>Type:</i> binary (max 16 bytes). Key. Nullable The <code>Job</code> which instructed the DS to perform the <code>Action</code> , if the <code>Job</code> still exists.
ActionStateID	<i>Type:</i> integer .One of the action states defined in the <code>ActionState</code> table.
PackageVersionID	<i>Type:</i> integer. Nullable If <code>Task</code> is of type <code>Distribution</code> , a <code>PackageVersion</code> applies.
FailureReason	<i>Type:</i> text. Nullable If not empty, text describing the reason the <code>Action</code> failed.
LastUpdate	<i>Type:</i> datetime The last time that the <code>ActionState</code> was updated. This value is the UTC date time of the event.
DSVersion	<i>Type:</i> text (max 32 characters). Nullable The version of the DS used to execute the <code>Action</code> .

## ActionApplies Table

An action applies/applied to this computer, which can be identified by its computer id, device id, DNS, IP or MAC address. One of the five related cross-references must be non-null. If more than one is non-null, precedence is applied top to bottom in the order documented below.



**Note •** To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 741: Database columns for ActionApplies table**

Database Column	Details
ActionAppliesID	<i>Type:</i> integer. Key. Generated ID Auto-generated identity number
ActionUID	<i>Type:</i> binary (max 16 bytes). Key The <code>Action</code> which applies.

Database Column	Details
ComputerID	<i>Type:</i> integer. Key. Nullable The computer id of the device to which the <code>Action</code> applies. Index into the <code>Computer</code> table.
DeviceID	<i>Type:</i> integer. Key. Nullable Index into the <code>NetworkDevice</code> table for this device.
MACAddress	<i>Type:</i> text (max 18 characters). Key. Nullable The network hardware address of the device.
DNSName	<i>Type:</i> text (max 128 characters). Key. Nullable The DNS name of the device.
IPAddress	<i>Type:</i> text (max 64 characters). Key. Nullable The IP Address of the device.
ActionStateID	<i>Type:</i> integer One of the action states defined in the <code>ActionState</code> table.
FailureReason	<i>Type:</i> text. Nullable If not empty, text describing the reason the action failed.
LastUpdate	<i>Type:</i> datetime The last time that the state of this action was updated. This value is the UTC date-time of the event.

## ActionState Table

All possible states for an action are reflected in a record here.

**Table 742: Database columns for ActionState table**

Database Column	Details
ActionStateID	<i>Type:</i> integer. Key. Generated ID The id for the action state.
ActionStateName	<i>Type:</i> text (max 32 characters). Key The name for the action state. Possible id-name pairs are:

Database Column	Details
	<ul style="list-style-type: none"> <li>• 1 = Created</li> <li>• 2 = DistributionInProgress</li> <li>• 3 = DistributionFailed</li> <li>• 4 = Distributed</li> <li>• 5 = SchedulePending</li> <li>• 6 = ScheduledFailed</li> <li>• 7 = Scheduled</li> <li>• 8 = Applied</li> <li>• 9 = ApplyFailed</li> <li>• 10 = CancelPending</li> <li>• 11 = CancelFailed</li> <li>• 12 = Cancelled</li> <li>• 13 = NotSupported</li> </ul>

## Job Table

This table stores the information about the jobs.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 743: Database columns for Job table**

Database Column	Details
JobUID	<i>Type:</i> binary (max 16 bytes). Key The unique id for the job.
TaskID	<i>Type:</i> integer. Key The id for the task.
ServerUID	<i>Type:</i> binary (max 16 bytes). Key The unique id for the server.

## Task Table

This table stores the information about the tasks.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 744: Database columns for Task table**

Database Column	Details
TaskID	Type: integer. Key. Generated ID The id of the task.
TaskUID	Type: binary (max 16 bytes). Key. Nullable The id of the task.
TaskTypeID	Type: integer The id for the task type.
TaskName	Type: text (max 128 characters). Key The name for the task.
PackagePathID	Type: integer. Key. Nullable For a distribution task, which package.
TaskScheduleID	Type: integer The id for the task schedule.
MinimumVersion	Type: text (max 16 characters). Nullable The minimum version required to execute the task.

## TaskSchedule Table

This table stores the required information about the task schedule, such as the start and finish times number of retries, delays and other related information.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 745: Database columns for TaskSchedule table**

Database Column	Details
TaskScheduleID	Type: integer. Key. Generated ID The id for the task schedule.
StartTime	Type: datetime. Nullable The time that the scheduled task must start.
EndTime	Type: datetime. Nullable The time that the scheduled task must end.
RetryCount	Type: integer. Nullable Number of times for task retries.
MinRetryDelay	Type: integer. Nullable Number of seconds before a retry occurs in case of a failure.
RepeatDelay	Type: integer. Nullable Number of seconds before the task is repeated.
NumParallelTasks	Type: integer. Nullable Number of tasks that can be run in parallel.
SleepBetweenTasks	Type: integer. Nullable Amount of time before the next task can start.

## TaskType Table

This table stores the information about different types of tasks and their associated IDs.

**Table 746: Database columns for TaskType table**

Database Column	Details
TaskTypeID	Type: integer. Key. Generated ID

Database Column	Details
	The id for the task.
TaskTypeName	Type: text (max 32 characters). Key The name of the task.



---

# 4

---

## License Portal Database Schema

### Topics:

- *Information Structure*
- *Compliance.ECM.Logic Tables*

This chapter describes additions made to the database schema for FlexNet Manager Suite to accommodate a separate licensing portal. With the entire product now presented in a web interface, this separation is entirely historical. The tables described in this chapter continue to appear in the database for all implementations.

# Information Structure

The following information is provided about database tables. Items appear only when relevant to the database column, and are suppressed where they do not apply. Two of these items (shown bold) are columns in the following pages, and the remainder are displayed within the **Details**.

Item	Comment
<b>Database Column</b>	The name of the column in the SQL table.
<i>Type</i>	The data type of the contents of the database column.
Size	For types that have a maximum capacity, the upper limit is provided in parentheses.
Key	The word "Key" appears when a column is a unique key field within the table. It is possible for several database columns to be part of the key, so that this indicator may appear for several columns in a table.
Generated ID	This indicates that a numeric ID is assigned by the database.
Nullable	If this indicator is present, the database column permits nulls.
Computed	This indicator appears for columns that are automatically computed by the database.
Default	If a column has a default value declared in the schema, this is specified at the end of the first set of details for the column.
<b>Details</b>	Describes the data stored in the database column, including many of the indicators described above.

## Compliance.ECM.Logic Tables

The complete set of database tables documented here includes:

- ComplianceActionHistory table (see *ComplianceActionHistory Table* on page 731)
- ComplianceActionHistoryResource table (see *ComplianceActionHistoryResource Table* on page 732)
- EcmSettings table (see *EcmSettings Table* on page 733)
- SoftwareLicenseUsageHistory table (see *SoftwareLicenseUsageHistory Table* on page 734)
- TrackGroup table (see *TrackGroup Table* on page 735)
- TrackSoftwareLicenseUsage table (see *TrackSoftwareLicenseUsage Table* on page 735)
- TrackSoftwareTitle table (see *TrackSoftwareTitle Table* on page 736)

- TrackSoftwareTitleUsage table (see *TrackSoftwareTitleUsage Table* on page 737)

## ComplianceActionHistory Table

ComplianceActionHistory records actions performed in the Compliance portal on a contract or software license, including usage activation/deactivation.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 747: Database columns for ComplianceActionHistory table**

Database Column	Details
ComplianceActionHistoryID	<i>Type:</i> integer. Key. Generated ID Unique identifier for the record.
ComplianceActionHistoryResourceID	<i>Type:</i> integer. Key Identifies the type of action performed. Foreign key to the ComplianceActionHistoryResource table.
History	<i>Type:</i> text Detailed information about the action performed.
HistoryParameters	<i>Type:</i> text Details of parameters changed and their changed values.
AssociatedObjectID	<i>Type:</i> integer The ID of the contract or license associated with the action.
AssociatedObjectName	<i>Type:</i> text (max 512 characters) The name of the contract or license associated with the action.
Comment	<i>Type:</i> text (max 1024 characters) Comments recorded about the change by the operator.
CreationUser	<i>Type:</i> text (max 512 characters) The username of the operator who made the change.
CreationDate	<i>Type:</i> datetime

Database Column	Details
	The date of the change.

## ComplianceActionHistoryResource Table

ComplianceActionHistoryResource table stores string resources required by the ComplianceActionHistory table.

**Table 748: Database columns for ComplianceActionHistoryResource table**

Database Column	Details
ComplianceActionHistoryResourceID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>Unique identifier for each record. Possible values and the corresponding default strings that may be written into a history list are:</p> <ul style="list-style-type: none"> <li>• 1 = Payment made</li> <li>• 2 = Payment edited</li> <li>• 3 = Payment cancelled</li> <li>• 4 = Activated application usage tracking for contract</li> <li>• 5 = Deactivated application usage tracking for contract</li> <li>• 6 = Activated application usage tracking for software license</li> <li>• 7 = Deactivated application usage tracking for software license</li> <li>• 8 = Modified application usage tracking for software license</li> <li>• 9 = Modified application usage tracking for contract</li> <li>• 10 = Not defined</li> <li>• 11 = Obligated to pay: (amount)</li> <li>• 12 = Actual amount was set to: (amount)</li> <li>• 13 = Actual amount currency rate was set to: (rate)</li> <li>• 14 = Estimated amount was set to: (amount)</li> <li>• 15 = Estimated amount currency rate was set to: (rate)</li> <li>• 16 = Budgeted amount was set to: (amount)</li> <li>• 17 = Budgeted amount currency rate was set to: (amount)</li> <li>• 18 = Payment status was set to: (status)</li> <li>• 19 = Payment amount: (amount); Payment date: (date)</li> </ul>

Database Column	Details
	<ul style="list-style-type: none"> <li>• 20 = Payment date was set to: (date)</li> <li>• 21 = Software license: (license name)</li> <li>• 22 = Software title: (application name)</li> <li>• 23 = Contract: (contract name)</li> <li>• 24 = Tracked: (yes/no); Track group: (group); Track start date: (date); Track end date: (date)</li> <li>• 25 = Applications tracked: (number).</li> </ul>
ResourceName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The name of the resource that determines the text to display on the user interface.</p>
DefaultValue	<p><i>Type:</i> text (max 512 characters)</p> <p>The default value to display if there is no resource string available to define the history action.</p>

## EcmSettings Table

EcmSettings stores operator-specific settings for the Compliance portal.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 749: Database columns for EcmSettings table**

Database Column	Details
EcmSettingID	<p><i>Type:</i> integer. Key. Generated ID</p> <p>A unique identifier for the record.</p>
ComplianceOperatorID	<p><i>Type:</i> integer. Key</p> <p>An operator of the Compliance portal. Foreign key to the ComplianceOperator table.</p>
SettingKey	<p><i>Type:</i> text (max 512 characters). Key</p> <p>A resource describing the operator setting.</p>

Database Column	Details
SettingType	Type: text (max 512 characters) The data type of the operator setting.
SettingValueString	Type: text Serialized value of the operator setting.
LastUpdated	Type: datetime Date and time when this setting was last updated.

## SoftwareLicenseUsageHistory Table

SoftwareLicenseUsageHistory records snapshots of software license utilization.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 750: Database columns for SoftwareLicenseUsageHistory table**

Database Column	Details
SoftwareLicenseUsageHistoryID	Type: integer. Key. Generated ID A unique identifier for each record in this table.
SnapshotDate	Type: datetime Date that the snapshot was recorded and the projected usage was calculated.
SoftwareLicenseID	Type: integer. Key SoftwareLicenseID that identifies the software license. This field is a foreign key to the SoftwareLicense table.
NumberPurchased	Type: integer Total number of licenses purchased, as of the Snapshot Date.
NumberInstalled	Type: integer Total number of installations for the license, as of the Snapshot Date.
NumberUsedActual	Type: integer. Nullable

Database Column	Details
	Total consumption of the license, as of the Snapshot Date. If application usage is not being tracked, this field is blank.
NumberUsedProjected	<p>Type: integer. Nullable</p> <p>The projected usage calculated for this license, based on patterns of usage over time.</p>

## TrackGroup Table

The `TrackGroup` table contains a list of the different tracking groups that tracked computer belong to.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 751: Database columns for TrackGroup table**

Database Column	Details
TrackGroupID	<p>Type: integer. Key. Generated ID</p> <p>A unique identifier for each <code>TrackGroup</code>. Possible values and the corresponding default strings are:</p> <ul style="list-style-type: none"> <li>1 = Sample</li> <li>2 = Enterprise.</li> </ul>
ResourceName	<p>Type: text (max 50 characters). Nullable</p> <p>The name of the resource that determines the text to display on the user interface.</p>
GroupName	<p>Type: text (max 64 characters). Key</p> <p>The default name of the <code>TrackGroup</code>. This is the value displayed if there is no resource string available to define the <code>TrackGroup</code>.</p>

## TrackSoftwareLicenseUsage Table

`TrackSoftwareLicenseUsage` keeps track of usage for each license.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 752: Database columns for TrackSoftwareLicenseUsage table**

Database Column	Details
TrackSoftwareLicenseUsageID	Type: integer. Key. Generated ID Unique identifier for each record.
SoftwareLicenseID	Type: integer. Key Identifies a license. This field is a foreign key to the SoftwareLicense table.
TrackGroupID	Type: integer. Key. Nullable Identifies the track group associated with the license. This field is a foreign key to the TrackGroup table.
SampleSize	Type: integer. Nullable Number of computers in sample group.
UsedPercentage	Type: decimal. Nullable Percentage of computers within the tracking group that reported use of applications associated with this license.
LastUpdated	Type: datetime Date and time when software license usage was updated.

## TrackSoftwareTitle Table

TrackSoftwareTitle stores details related to tracking software usage for a software title.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database *TenantID* has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 753: Database columns for TrackSoftwareTitle table**

Database Column	Details
TrackSoftwareTitleID	Type: integer. Key. Generated ID



Database Column	Details
	Unique identifier for each record. This field is a foreign key to the <code>SoftwareTitle</code> table.
<code>SoftwareTitleID</code>	<i>Type:</i> integer. Key. Nullable Identifies the application for which usage is being tracked. This field is a foreign key to the <code>SoftwareTitle</code> table.
<code>SoftwareLicenseID</code>	<i>Type:</i> integer. Key. Nullable Identifies the license associated with the application. This field is a foreign key to the <code>SoftwareLicense</code> table.
<code>TrackGroupID</code>	<i>Type:</i> integer. Key Identifies if usage tracking has been activated for the Sample or Enterprise tracking group. This field is a foreign key to the <code>TrackGroup</code> table.
<code>LastTrackStartDate</code>	<i>Type:</i> datetime. Nullable Date that tracking was last turned on.
<code>LastTrackEndDate</code>	<i>Type:</i> datetime. Nullable Date that tracking was last turned off. This field may be null if the operator cleared the end date when activating application usage.
<code>TrackEndDueDate</code>	<i>Type:</i> datetime. Nullable Date that the current tracking period ends. Should be null when <code>IsTracked</code> is False.
<code>IsTracked</code>	<i>Type:</i> boolean. Key Indicates whether usage tracking is enabled for this application entry.

## TrackSoftwareTitleUsage Table

`TrackSoftwareTitleUsage` keeps track of whether licensed software is being used on a computer.



**Note** • To cater for multi-tenant mode, this table may contain data for multiple tenants. Access requires that the database `TenantID` has been set in the SQL Server connection context information. That setting filters an underlying table to produce this view of data for the single, selected tenant.

**Table 754: Database columns for TrackSoftwareTitleUsage table**

Database Column	Details
TrackSoftwareTitleUsageID	<i>Type:</i> integer. Key. Generated ID Unique identifier for each record.
ComplianceComputerID	<i>Type:</i> integer. Key Identifies the computer on which usage tracking details were recorded. This field is a foreign key to the <code>ComplianceComputer</code> table.
SoftwareTitleID	<i>Type:</i> integer. Key Identifier for the application that was installed on the computer. This field is a foreign key to the <code>SoftwareTitle</code> table.
SoftwareLicenseID	<i>Type:</i> integer. Key. Nullable Identifier for the license associated with the installed application on the computer. This field is a foreign key to the <code>SoftwareLicense</code> table.
TrackGroupID	<i>Type:</i> integer. Key. Nullable Identifies the track group to which the computer has been assigned.
IsUsed	<i>Type:</i> boolean. Nullable Indicates whether the application is used on the computer.
LastUsed	<i>Type:</i> datetime. Nullable Date and time when software was last used on computer.

---

# 5

---

## Inventory Spreadsheet Templates

### Topics:

- *Information Structure for Spreadsheet Inventory Imports*
- *Compliance.InventoryReader.Logic Tables*

In contrast with other chapters in this document, this chapter takes a different approach: rather than documenting the schema of the central database for FlexNet Manager Suite, it describes the formats acceptable for spreadsheet (.xlsx) or comma-separated value (.csv) files that can be used to import various kinds of inventory information into the central database. For each data element, it shows which database table, and which column in that table, is the final destination for the imported data. (For details about importing inventory as spreadsheets or CSV files, see the chapter *Importing Inventory Spreadsheets and CSV Files* in the companion volume, *FlexNet Manager Suite System Reference*.)

Such spreadsheet (including CSV) files can be imported through two different paths:

- Using the web interface for FlexNet Manager Suite, the data may be uploaded directly to the central application server(s) as a one-time upload
- Optionally with a repeatable schedule, the data may also be uploaded through an inventory beacon.

The same templates are used for inventory imports through either of these channels.

# Information Structure for Spreadsheet Inventory Imports

The following information is provided about the structure of spreadsheet (.xlsx) and comma-separated value (.csv) template files that can be prepared as a data source for importing inventory. The items listed below appear only when relevant to the spreadsheet column, and are suppressed where they do not apply. Four of these items (shown bold) are columns in the following pages, and the remainder are displayed within the **Details** column.


Below this key is a mapping between:


- The file name of the downloaded template
- The prompt in the web interface of FlexNet Manager Suite for upload of the completed spreadsheet
- The topic below that covers this data (topic names are driven by the underlying database schema).



**Remember** • The template files are fixed format. While adding data to each file, you may not change:

- The file name
- The names of columns
- The number of columns
- The order of columns.

Item	Comment
<b>Column</b>	<p>The name of the column in the spreadsheet template (and uploaded data file).</p>  <p><b>Important</b> • Some column names are long, and must be wrapped over more than one line in this document. In all cases, the wrapped text should be continuous on a single line without white space in the template column names.</p>
<b>Example values</b>	Some sample data, or in some cases the list of supported values. When such a list is present, ensure that each row has a value that is an exact match for one of the available values (except that the validation is case insensitive).
<b>Details</b>	Describes the data required in the spreadsheet column, including many of the indicators described below.
<i>Type</i>	The data type of the contents of the spreadsheet column.
max	For types that have a maximum capacity, the upper limit is provided in parentheses.

Item	Comment
Key	The word “Key” appears when a column is a unique key field for data matching between the row of the spreadsheet and the data in the central database table (the destination for the data). Keep in mind that a single spreadsheet may include data destined for multiple database tables; and even within a single database table, it is possible for several database columns to be part of the key. For these reasons, this indicator may appear in several rows in the documentation list.
Nullable	If this indicator is present, the spreadsheet column may be left blank (and the target database entity allows nulls). Be careful about spaces in a cell of your spreadsheet: white space is a valid value, and is not equivalent to a null.
Destination	<p>Where the imported data is eventually saved in the central database for FlexNet Manager Suite. This is given with a dot separating the database table and the column name within the table, in the format <i>tableName.columnName</i>. For further details on these database tables and columns, see the other chapters in this volume.</p>  <p><b>Tip •</b> A single value in the imported spreadsheet may update data in more than one database column. Where that happens, this <b>Destination</b> listing shows the multiple destinations for the individual row.</p>

## Mapping templates to topics

The following table relates the template names (and the related prompts in the web interface) to the topics in this section that describe the individual columns within the templates. Templates are listed alphabetically. The naming of the following topics is driven by the related table names in the underlying database schema, so this list helps map the real world presentation to the database.



**Tip •** Templates are provided in matching pairs of XLSX and CSV files. As these are structurally identical, only the base file name (without an extension) is listed here.

Template file name	Web prompt	See topic
Cluster	Cluster evidence	ConsolidatedCluster Template
ClusterGroup	Cluster group data	ConsolidatedClusterGroup Template
ClusterHostAffinityRule	Cluster host affinity rule data	ConsolidatedClusterHostAffinityRule Template
Computer	Computers and VMs	ConsolidatedComputer Template
FileEvidence	File evidence	ConsolidatedFileEvidence Template

Template file name	Web prompt	See topic
InstallerEvidence	<b>Installation evidence</b>	<i>ConsolidatedInstallerEvidence Template</i>
OracleDatabaseUser	<b>Oracle database user</b>	<i>ConsolidatedOracleDatabaseUser Template</i>
RemoteAccessFile	<b>Access shown by file evidence</b>	<i>ConsolidatedRemoteAccessFile Template</i>
RemoteAccessInstaller	<b>Access shown by installer evidence</b>	<i>ConsolidatedRemoteAccessInstaller Template</i>
VMPool	<b>Virtual machine pool data</b>	<i>ConsolidatedVMPool Template</i>
WMIEvidence	<b>WMI evidence</b>	<i>ConsolidatedWMIEvidence Template</i>

## Compliance.InventoryReader.Logic Tables

The complete set of database tables documented here includes:

- ConsolidatedCluster table (see *ConsolidatedCluster Template* on page 742)
- ConsolidatedClusterGroup table (see *ConsolidatedClusterGroup Template* on page 745)
- ConsolidatedClusterHostAffinityRule table (see *ConsolidatedClusterHostAffinityRule Template* on page 746)
- ConsolidatedComputer table (see *ConsolidatedComputer Template* on page 747)
- ConsolidatedFileEvidence table (see *ConsolidatedFileEvidence Template* on page 761)
- ConsolidatedInstallerEvidence table (see *ConsolidatedInstallerEvidence Template* on page 764)
- ConsolidatedOracleDatabaseUser table (see *ConsolidatedOracleDatabaseUser Template* on page 768)
- ConsolidatedRemoteAccessFile table (see *ConsolidatedRemoteAccessFile Template* on page 772)
- ConsolidatedRemoteAccessInstaller table (see *ConsolidatedRemoteAccessInstaller Template* on page 775)
- ConsolidatedVMPool table (see *ConsolidatedVMPool Template* on page 777)
- ConsolidatedWMIEvidence table (see *ConsolidatedWMIEvidence Template* on page 779)

## ConsolidatedCluster Template

The Cluster spreadsheet provides a simple interface for defining server clustering. It is useful when combined with the ClusterGroup and ClusterHostAffinityRule spreadsheets.

**Table 755: Columns included with ConsolidatedCluster templates**

Column	Details
ClusterID	<p><i>Type:</i> big integer. Key</p> <p>The unique identifier for this imported cluster. This may be a string or an integer.</p> <p>Destination:</p> <p><code>ImportedCluster.ExternalID</code></p>
ClusterName	<p><i>Type:</i> text (max 128 characters)</p> <p>The name of the cluster in the external cluster management system.</p> <p>Destination:</p> <p><code>ImportedCluster.ExternalName</code></p> <p><code>ImportedCluster.Name</code></p>
Namespace	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>Where the cluster is contained: + The fully-qualified domain name (for HyperV clusters) - example: 'france.thc.myenterprise.com' + The datacenter name (for VMWare clusters) - example: 'MelProdDataCenter'</p> <p>Destination:</p> <p><code>ImportedCluster.Namespace</code></p>
ClusterType	<p><i>Type:</i> text (max 128 characters)</p> <p>The kind of cluster. The value must be an exact case-insensitive match to one of the permitted values.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• vMotion Cluster</li> <li>• Hyper-V Cluster</li> <li>• Host Affinity Group</li> <li>• VM Affinity Group</li> <li>• Oracle VM</li> </ul> <p>Destination:</p> <p><code>ImportedCluster.ClusterTypeID</code></p>
InventoryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date (with optional time) that the cluster last had inventory reported. The date must be entered in one of the supported formats.</p>

Column	Details
	<p>Possible values:</p> <ul style="list-style-type: none"> <li>• yyyy/MM/dd</li> <li>• yyyy/MM/dd HH:mm:Ss</li> <li>• yyyy/MM/dd HH:mm</li> <li>• yyyy-MM-dd</li> <li>• yyyy-MM-dd HH:mm:Ss</li> <li>• yyyy-MM-dd HH:mm</li> <li>• yyyyMMdd</li> <li>• yyyyMMdd HH:mm:Ss</li> <li>• yyyyMMdd HH:mm</li> </ul> <p>Destination:</p> <p><code>ImportedCluster.InventoryDate</code></p>
InventoryAgent	<p><i>Type:</i> text (max 64 characters). Nullable</p> <p>The name of the person or tool that performed the last inventory. For imported spreadsheets, you may wish to include the name of the person preparing the data, in case there is subsequent follow-up required.</p> <p>Destination:</p> <p><code>ImportedCluster.InventoryAgent</code></p>
DRS	<p><i>Type:</i> boolean. Nullable</p> <p>Whether Distributed Resource Scheduler (DRS) is enabled on the cluster.</p> <p>Possible values:</p> <p>true, false, 0 or 1</p> <p>Destination:</p> <p><code>ImportedCluster.DRS</code></p>
DPM	<p><i>Type:</i> boolean. Nullable</p> <p>Whether Distributed Power Management (DPM) is enabled on the cluster.</p> <p>Possible values:</p> <p>true, false, 0 or 1</p> <p>Destination:</p> <p><code>ImportedCluster.DPM</code></p>



# ConsolidatedClusterGroup Template

The ClusterGroup spreadsheet uses data from the Cluster spreadsheet and defines groups of servers as well as computers that are members of these groups.

**Table 756: Columns included with ConsolidatedClusterGroup templates**

Column	Details
ClusterID	<p><i>Type:</i> big integer. Key</p> <p>The unique identifier for the imported cluster. This may be a string or an integer and must match a value for the ClusterID in the cluster spreadsheet.</p> <p>Destination:</p> <p>ImportedClusterGroup.ClusterExternalID</p>
ClusterGroupID	<p><i>Type:</i> big integer. Key</p> <p>The unique identifier for this cluster group. This may be a string or an integer.</p> <p>Destination:</p> <p>ImportedClusterGroup.ExternalID</p> <p>ImportedClusterGroupMember.ClusterGroupExternalID</p>
ClusterGroupName	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The name of the cluster group. Depending on the value of the ClusterGroupType this will be a group of hosts or virtual machines.</p> <p>Destination:</p> <p>ImportedClusterGroup.Name</p>
ClusterGroupType	<p><i>Type:</i> text (max 128 characters)</p> <p>The kind of cluster included in the group. The value must be an exact case-insensitive match to one of the permitted values.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• vMotion Cluster</li> <li>• Hyper-V Cluster</li> <li>• Host Affinity Group</li> <li>• VM Affinity Group</li> <li>• Oracle VM</li> </ul> <p>Destination:</p> <p>ImportedClusterGroup.ClusterTypeID</p>

Column	Details
ComputerID	<p><i>Type:</i> big integer. Key</p> <p>The identifier used in the 'Computer' spreadsheet for a computer which is a member of the group. To identify all the members of the group, repeat as many lines as required in your spreadsheet where the other values in the row are identical, and only the 'ComputerID' value changes. Values in this column must match a ComputerID in the computer spreadsheet or the row will be skipped.</p> <p>Destination:</p> <p><code>ImportedClusterGroupMember.ComputerExternalID</code></p>

## ConsolidatedClusterHostAffinityRule Template

The ClusterHostAffinity spreadsheet defines the groups of virtual machines which may run on groups of host servers.

**Table 757: Columns included with ConsolidatedClusterHostAffinityRule templates**

Column	Details
ClusterID	<p><i>Type:</i> big integer. Key</p> <p>The unique identifier for the imported cluster, to which this affinity rule applies. This may be a string or an integer and must match a ClusterID from the cluster spreadsheet.</p> <p>Destination:</p> <p><code>ImportedClusterHostAffinityRule.ClusterExternalID</code></p>
Name	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The name of the cluster host affinity rule.</p> <p>Destination:</p> <p><code>ImportedClusterHostAffinityRule.Name</code></p>
ClusterHostGroupName	<p><i>Type:</i> big integer. Key</p> <p>The name of the group of hosts that the ClusterVMGroupName virtual machines may run on.</p> <p>Destination:</p> <p><code>ImportedClusterHostAffinityRule.ClusterHostGroupExternalID</code></p>
ClusterVMGroupName	<p><i>Type:</i> big integer. Key</p>

Column	Details
	<p>The name of the virtual machine group that may run on the ClusterHostGroupName hosts.</p> <p>Destination:</p> <p><code>ImportedClusterHostAffinityRule.ClusterVMGroupExternalID</code></p>
ClusterHostAffinityRuleType	<p><b>Type:</b> text (max 128 characters)</p> <p>The type of affinity rule. The value must be an exact case-insensitive match to one of the permitted values.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• must run on</li> <li>• must not run on</li> </ul> <p>Destination:</p> <p><code>ImportedClusterHostAffinityRule.ClusterHostAffinityRuleTypeID</code></p>

## ConsolidatedComputer Template

'ConsolidatedComputer' consolidates data for the Computer, VirtualMachine, Domain, User and Cluster objects, providing a simpler way to populate this information. Any spreadsheet row that includes a 'HostComputerID' is making that row a virtual machine, and the import process expects that virtualization data will be provided.

**Table 758: Columns included with ConsolidatedComputer templates**

Column	Details
ComputerID	<p><b>Type:</b> big integer. Key</p> <p>The unique identifier for a computer (either physical or virtual). This identifier can either be an integer or a string. Keep this consistent across multiple imports: it is used to track the computer over time.</p> <p>Destination:</p> <p><code>ImportedComputer.ExternalID</code></p> <p><code>ImportedVirtualMachine.VMComputerID</code></p> <p><code>ImportedClusterNode.ComputerExternalID</code></p>
ComputerName	<p><b>Type:</b> text (max 256 characters)</p> <p>The name of the computer. In Windows, this is the NetBIOS name of the local computer, as returned by <code>GetComputerName()</code>. For UNIX, it is the host name of the machine, as returned by <code>gethostname(2)</code>.</p>

Column	Details
	<b>Destination:</b> <code>ImportedComputer.ComputerName</code>
DomainFlatName	<b>Type:</b> text (max 100 characters). Key. Nullable The flatname of the domain of the computer. Example: 'mycompany'. <b>Destination:</b> <code>ImportedDomain.FlatName</code>
DomainQualifiedName	<b>Type:</b> text (max 100 characters). Key. Nullable The fully qualified domain name for the computer. Example: 'prod.mycompany.eu'. <b>Destination:</b> <code>ImportedComputer.Domain</code> <code>ImportedDomain.QualifiedName</code>
BIOSUUID	<b>Type:</b> unique identifier. Nullable The BIOS UUID of the computer (physical or virtual), as provided by the operating system. Possible values: 93B5BE3B-88B0-450E-9F75-F6294210DFA0 <b>Destination:</b> <code>ImportedComputer.UUID</code>
OperatingSystem	<b>Type:</b> text (max 128 characters). Nullable The operating system of the computer. For virtual machines, it is the configured operating system of the guest. Note that this operating system identification is not used for licensing. <b>Destination:</b> <code>ImportedComputer.OperatingSystem</code> <code>ImportedVirtualMachine.GuestFullName</code>
ServicePack	<b>Type:</b> text (max 128 characters). Nullable The service pack installed for the operating system. <b>Destination:</b> <code>ImportedComputer.ServicePack</code>
EmailAddress	<b>Type:</b> text (max 256 characters). Nullable

Column	Details
	<p>The email address associated with the device. Typically used for mobile devices.</p> <p>Destination:</p> <p><code>ImportedComputer.EmailAddress</code></p>
PhoneNumber	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The phone number of the device. Used for mobile devices.</p> <p>Destination:</p> <p><code>ImportedComputer.PhoneNumber</code></p>
Manufacturer	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The manufacturer of the computer hardware. Some examples include:</p> <ul style="list-style-type: none"> <li>On Windows, the SMBios manufacturer (the WMI Manufacturer property of the 'Win32_ComputerSystem' class).</li> <li>On Linux, 'Manufacturer' in the 'System Information' section resulting from the 'dmidecode' command. Sample command: 'dmidecode -s system-manufacturer'</li> <li>On Solaris x86, as for Linux, with failovers first to 'sysinfo SI_HW_PROVIDER' and then to 'ModelNo'.</li> <li>On Solaris SPARC, the 'sysinfo SI_HW_PROVIDER'. Typically this value is 'Sun_Microsystems' or, more recently, 'Oracle Corporation'. Failover to the 'ModelNo'.</li> <li>On HP-UX, the string literal 'HP'.</li> <li>On AIX, the 'modelname' system attribute preceding the comma character. For example, if the 'modelname' system attribute is 'IBM,8202-E4B', then use 'IBM'. This value is typically 'IBM'.</li> </ul> <p>Destination:</p> <p><code>ImportedComputer.Manufacturer</code></p> <p><code>ImportedVirtualMachine.Manufacturer</code></p>
ModelNo	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The model of the computer hardware or the virtual machine. This value is defined for the context of the current execution environment, rather than the physical server that may be hosting a virtual machine or partition. Examples:</p> <ul style="list-style-type: none"> <li>On Windows, the SMBios product name. The WMI Model property of the Win32_ComputerSystem class.</li> </ul>

Column	Details
	<ul style="list-style-type: none"> <li>On Linux, the SMBios product name read using the command 'dmidecode -s system-product-name'. Specifically, the 'System Information' section and the 'Product Name' in that section is used.</li> <li>On Solaris x86, as for Linux, with failover to the 'sysinfo SI_PLATFORM', stripping 'SUNW', and replacing hyphen characters with space characters.</li> <li>On Solaris SPARC, the 'openprom' "banner-name" value read from '/dev/openprom'. Failover to the 'sysinfo SI_PLATFORM', stripping 'SUNW', and replacing hyphen characters with space characters.</li> <li>On HP-UX, the 'confstr _CS_MACHINE_MODEL'.</li> <li>On AIX, the 'modelname' system attribute following the comma character. For example, if the 'modelname' system attribute is 'IBM,8202-E4B', then use '8202-E4B'.</li> </ul> <p>Destination:</p> <p><code>ImportedComputer.ModelNo</code></p> <p><code>ImportedVirtualMachine.ModelNo</code></p>
SerialNo	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>The hardware serial number of the computer. The goal of this value is to be tied to the physical hardware, partition or virtual machine and to be as unique as possible across all computers in the organization. This is due to its use in tracking computers, particularly after an operating system rebuild. This value is also used to socialize computer inventory from different inventory sources, and is used to map virtual machine guest operating system inventory to the VM host on which the virtual machine is running. Example sources:</p> <ul style="list-style-type: none"> <li>On Windows, the SMBios serial number. The WMI 'SerialNumber' property of the 'Win32_BIOS' class. Can fail over to the 'SerialNumber' property of the 'Win32_SystemEnclosure' class which is typically the same value.</li> <li>On Linux, the SMBios serial number read using the command 'dmidecode -s system-serial-number'. Specifically, the 'System Information' section and the 'Serial Number' in that section is used.</li> <li>On Solaris 10 8/07 or later, for a non-global zone, the UUID value from the /etc/zones/index file. For a global zone, the same as Solaris 10 releases earlier than 8/07.</li> <li>For Solaris 10 releases earlier than 8/07, the hexadecimal version of 'SI_HW_SERIAL' with an appended hyphen character followed by the Zone's name. For example, '838bfc7b-global' or '838bfc7b-myzone'.</li> <li>For Solaris 8 and 9, The hexadecimal version of 'SI_HW_SERIAL'.</li> </ul>

Column	Details
	<ul style="list-style-type: none"> <li>For Mac OS X, the serial number of the machine as printed on the packaging and found in "About this Mac" from the desktop.</li> <li>For HP-UX, the 'confstr_CS_PARTITION_IDENT' partition identifier if it is an nPar or vPar, or '_CS_MACHINE_IDENT' if not; with a failover to the machine serial number, and a final failover to the 'uname' machine identification number.</li> <li>For AIX, the 'id_to_partition' system attribute, starting from the third character (strips a '0X' from the start). For example, if the 'id_to_partition' system attribute is '0X0473409002F7B201' then use '0473409002F7B201'.</li> </ul> <p>Destination:</p> <p><code>ImportedComputer.SerialNo</code></p>
ChassisType	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The type of case of the computer. The value must be a (case insensitive) exact match for one of the values shown. Note that some license types use this information to optimize the licensing position, particularly with desktop and laptop computers.</p> <p>Destination:</p> <p><code>ImportedComputer.ChassisType</code></p>
TotalMemory	<p><i>Type:</i> big integer. Nullable</p> <p>The total RAM in the computer, in bytes.</p> <p>Destination:</p> <p><code>ImportedComputer.TotalMemory</code></p>
NumberOfDisplayAdapters	<p><i>Type:</i> integer. Nullable</p> <p>The number of graphics cards in the computer.</p> <p>Destination:</p> <p><code>ImportedComputer.NumberOfDisplayAdapters</code></p>
VirtualMachineUUID	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The unique identifier of the virtual machine provided by the virtualization infrastructure. (This may have the same value as the 'BIOSUUID', or have byte order reversed, or be altogether different.)</p> <p>Destination:</p> <p><code>ImportedVirtualMachine.UUID</code></p>
IMEI	<p><i>Type:</i> text (max 256 characters). Nullable</p>

Column	Details
	<p>IMEI (International Mobile Equipment Identity) is a 15- or 17-digit code that uniquely identifies mobile phone sets. Leave blank (null) for other device types.</p> <p>Destination:</p> <p><code>ImportedComputer.IMEI</code></p>
NumberOfProcessors	<p><i>Type:</i> integer. Nullable</p> <p>The total number of physical processors (CPU) in the computer. Note that a number of server-based licenses depend on complete details of the processor types, counts and speeds to calculate a correct license position.</p> <p>Destination:</p> <p><code>ImportedComputer.NumberOfProcessors</code></p> <p><code>ImportedVirtualMachine.NumberOfProcessors</code></p>
ProcessorType	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The descriptive string of the processor(s) in the computer. This may be a comma-separated list in the case where there is more than one physical processor in the system. Note that a number of server-based licenses depend on complete details of the processor types, counts and speeds to calculate a correct license position.</p> <p>Destination:</p> <p><code>ImportedComputer.ProcessorType</code></p> <p><code>ImportedVirtualMachine.ProcessorType</code></p>
MaxClockSpeed	<p><i>Type:</i> integer. Nullable</p> <p>The maximum clock speed of the fastest processor in the computer in kHz. Note that a number of server-based licenses depend on complete details of the processor types, counts and speeds to calculate a correct license position.</p> <p>Destination:</p> <p><code>ImportedComputer.MaxClockSpeed</code></p>
NumberOfCores	<p><i>Type:</i> integer. Nullable</p> <p>The total number of cores in the computer. If there is more than one physical processor in the computer, then this would be the sum of the core counts for all the processors. For example, in a computer with two quad-core processors, this value would be 8. Note that a number of server-based licenses depend on complete details of the processor types, counts and speeds to calculate a correct license position.</p> <p>Destination:</p> <p><code>ImportedComputer.NumberOfCores</code></p>



Column	Details
NumberOfSockets	<p><b>Type:</b> integer. Nullable</p> <p>The number of physical sockets into which a processor may be placed in the computer. It is rare that an inventory source can know this value. If unset, it is typically approximated by the number of processors.</p> <p><b>Destination:</b></p> <p><code>ImportedComputer.NumberOfSockets</code></p>
NumberOfLogicalProcessors	<p><b>Type:</b> integer. Nullable</p> <p>The number of logical processors in the computer. This is the number of 'execution contexts' the operating system has access to. It will commonly be equivalent to the number processors in a single core, non-multi-threaded processor architecture, to the number of cores in a multi-core single threaded processor architecture, and to the number of threads in a multi-threaded processor architecture. For example, in a two processor, quad-core and hyper-threaded computer, this value would be 16. Note that a number of server-based licenses depend on complete details of the processor types, counts and speeds to calculate a correct license position.</p> <p><b>Destination:</b></p> <p><code>ImportedComputer.NumberOfLogicalProcessors</code></p>
PartialNumberOfProcessors	<p><b>Type:</b> decimal. Nullable</p> <p>Used in processor-based licensing, this is the non-integer number of cores allocated to this partition or virtual machine. When this property is null, the 'NumberOfCores' is used. Note that a number of server-based licenses depend on complete details of the processor types, counts and speeds to calculate a correct license position.</p> <p><b>Possible values:</b></p> <p>120.45</p> <p><b>Destination:</b></p> <p><code>ImportedComputer.PartialNumberOfProcessors</code></p>
NumberOfHardDrives	<p><b>Type:</b> integer. Nullable</p> <p>The number of physical hard drives in the computer. While the intent is physical drives, often this can end up being the number of disk partitions.</p> <p><b>Destination:</b></p> <p><code>ImportedComputer.NumberOfHardDrives</code></p> <p><code>ImportedVirtualMachine.NumberOfHardDrives</code></p>
TotalDiskSpace	<p><b>Type:</b> big integer. Nullable</p>

Column	Details
	<p>The total size of all hard drives in the computer in bytes. Note that this can be a very large number on modern systems. The maximum value for a bigint is 9,223,372,036,854,775,807, which can represent about 9.2 exabyte. While in practice it is unlikely that this size of storage capacity is reached for a single system, some systems can end up with large values through virtualized drives. Therefore, it is worth considering capping values when calculating total disk space, particularly when converting values from kilobytes or megabytes to bytes.</p> <p>Destination:</p> <p><code>ImportedComputer.TotalDiskSpace</code></p>
NumberOfNetworkCards	<p><i>Type:</i> integer. Nullable</p> <p>The number of network cards in the computer.</p> <p>Destination:</p> <p><code>ImportedComputer.NumberOfNetworkCards</code></p> <p><code>ImportedVirtualMachine.NumberOfNetworkCards</code></p>
IPAddress	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The IP address of the computer in IPv4 or IPv6 format. This may be a comma-separated list if there is more than one active network adapter in the system. Do not include inactive network adapters and network adapters with invalid IP addresses. Examples:</p> <ul style="list-style-type: none"> <li>• '69.89.31.226'</li> <li>• '2002:4559:1FE2::4559:1FE2'</li> </ul> <p>Destination:</p> <p><code>ImportedComputer.IPAddress</code></p>
MACAddress	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The MAC address of the computer. This may be a comma-separated list if there is more than one active network adapter in the system. Do not include inactive network adapters and network adapters with invalid MAC addresses.</p> <p>Destination:</p> <p><code>ImportedComputer.MACAddress</code></p>
LastLoggedInUser	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>The DOMAIN/SAMAccountName of the user last logged onto the computer.</p> <p>Destination:</p> <p><code>ImportedComputer.LastLoggedInUser</code></p>

Column	Details
	<p><code>ImportedUser.ExternalID</code></p> <p><code>ImportedUser.UserName</code> (Element 2 after splitting on '\\')</p> <p><code>ImportedUser.Domain</code> (Element 1 after splitting on '\\')</p> <p><code>ImportedUser.SAMAccountName</code> (Element 2 after splitting on '\\')</p>
LastLogonDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date and time when the user last logged on to the computer. The date must be entered in one of the supported formats.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• yyyy/MM/dd</li> <li>• yyyy/MM/dd HH:mm:Ss</li> <li>• yyyy/MM/dd HH:mm</li> <li>• yyyy-MM-dd</li> <li>• yyyy-MM-dd HH:mm:Ss</li> <li>• yyyy-MM-dd HH:mm</li> <li>• yyyyMMdd</li> <li>• yyyyMMdd HH:mm:Ss</li> <li>• yyyyMMdd HH:mm</li> </ul> <p>Destination:</p>
CalculatedUser	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The domain/SAMAccountName of the calculated user. Some inventory systems calculate the user who owns a computer. For example, it might be the user who, over the last ten logins, logged in most often.</p> <p>Destination:</p> <p><code>ImportedComputer.CalculatedUser</code></p>
HostComputerID	<p><i>Type:</i> text (max 256 characters). Key. Nullable</p> <p>The ComputerID of the server this virtual machine is hosted on. This may be a string or an integer and must match the ComputerID for another computer in this spreadsheet.</p> <p>Destination:</p> <p><code>ImportedVirtualMachine.HostComputerID</code></p>
VirtualMachineType	<p><i>Type:</i> text (max 100 characters). Nullable</p>

Column	Details
	<p>The type of the virtual machine. If present, the value must be a (case insensitive) exact match to one of the values shown.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• VMware</li> <li>• HyperV</li> <li>• LPAR</li> <li>• WPAR</li> <li>• nPar</li> <li>• vPar</li> <li>• SRP</li> <li>• Zone</li> <li>• Unknown</li> <li>• Oracle VM</li> </ul> <p>Destination:</p> <p><code>ImportedVirtualMachine.VirtualMachineType</code></p>
VMEnabledState	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The operational state of the virtual machine. If present, the value must be a (case insensitive) exact match to one of the values shown.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• Started</li> <li>• Stopped</li> <li>• Suspended</li> <li>• Unknown</li> </ul> <p>Destination:</p> <p><code>ImportedVirtualMachine.VMEnabledStateID</code></p>
AffinityEnabled	<p><i>Type:</i> boolean</p> <p>Set this to <code>true</code> (or 1) if this VM has affinity for its current host (so that it is unable to move to different host computers).</p> <p>Possible values:</p> <p><code>true</code>, <code>false</code>, 0 or 1</p> <p>Destination:</p>

Column	Details
	<code>ImportedVirtualMachine.AffinityEnabled</code>
<code>CPUAffinity</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>Contains a comma-separated list of processor numbers (Host Logical Processors) or ranges for which this virtual machine has affinity. Example: 1,3-5,8</p> <p><i>Destination:</i></p> <p><code>ImportedVirtualMachine.CPUAffinity</code></p>
<code>CoreAffinity</code>	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>Contains a comma-separated list of core numbers (or ranges) for which this virtual machine has affinity. Cores are numbered sequentially up the sequence of processors. Example: 1,5-8,10</p> <p><i>Destination:</i></p> <p><code>ImportedVirtualMachine.CoreAffinity</code></p>
<code>ComplianceComputerType</code>	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>If you know that the computer is a virtual machine or VM host, record that data here. If you are unsure, leave this cell empty (NULL): this allows the system to infer the computer type (for example, a computer with VMs linked to it is inferred to be a VM host). If data comes from multiple inventory sources, leaving this value as null also allows the value to be inserted from another source. So, unless there is a very good reason, do not just specify 'Computer', but allow the inference rules to help.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• Computer</li> <li>• VM Host</li> <li>• Virtual Machine</li> <li>• Remote Device</li> <li>• Mobile Device</li> <li>• VDI Template</li> </ul> <p><i>Destination:</i></p> <p><code>ImportedComputer.ComplianceComputerTypeID</code></p>
<code>HostIdentifyingNumber</code>	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>A physical server may have an identifier that is unique only across that hardware model, and may be less unique than the true hardware serial number, for example. This value is typically set for physical machines only, which</p>

Column	Details
	<p>include virtualization hosts, partitioned server hosts, and standalone machines. For a partitioned server, this value can be reported by each of the partitions on that server, such that a record of the physical computer can be created using one of the instances of this value. This value is used for matching computers.</p> <p>Destination:</p>
HostType	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The type of the physical host computer. This value is similar to the model number, but it is always for the physical server that an execution context may be running on. Therefore, this will generally be a known value for standalone machines and partitions, but it will not be known for virtual machines. This value is used for matching computers. Examples:</p> <ul style="list-style-type: none"> <li>• 'i86pc'</li> <li>• 'Sun-Fire-T1000'</li> <li>• 'rx7620'</li> <li>• '785' (for a 9000/785/C3700)</li> <li>• '8202' (for an IBM,8202-E4B).</li> </ul> <p>Destination:</p>
VMLocation	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>Location of the virtual machine on the file system.</p> <p>Destination:</p> <p><code>ImportedVirtualMachine.VMLocation</code></p>
PoolName	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>The name of the pool that the virtual machine belongs to.</p> <p>Destination:</p> <p><code>ImportedVirtualMachine.PoolName</code></p>
PoolType	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>The type of the pool that the virtual machine belongs to.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• Folder</li> <li>• Datacenter</li> <li>• ComputeResource</li> <li>• HostSystem</li> </ul>

Column	Details
	<ul style="list-style-type: none"> <li>• ResourcePool</li> <li>• VirtualMachine</li> <li>• PhysicalSharedPool</li> <li>• VirtualSharedPool</li> <li>• LPAR</li> <li>• RSET</li> <li>• ClusterComputeResource</li> </ul> <p>Destination:</p> <p><code>ImportedVirtualMachine.PoolType</code></p>
CPUUsage	<p><i>Type:</i> integer. Nullable</p> <p>The maximum CPU usage of the virtual machine (MHz).</p> <p>Destination:</p> <p><code>ImportedVirtualMachine.CPUUsage</code></p>
MemoryUsage	<p><i>Type:</i> big integer. Nullable</p> <p>The maximum memory usage of the virtual machine (bytes).</p> <p>Destination:</p> <p><code>ImportedVirtualMachine.MemoryUsage</code></p>
InventoryDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date (and optionally time) the computer last had inventory reported. This field is generally used for differential updates (that is, if the date/time has not changed since the previous import, the data record is not imported/updated). The date must be entered in one of the supported formats.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• yyyy/MM/dd</li> <li>• yyyy/MM/dd HH:mm:Ss</li> <li>• yyyy/MM/dd HH:mm</li> <li>• yyyy-MM-dd</li> <li>• yyyy-MM-dd HH:mm:Ss</li> <li>• yyyy-MM-dd HH:mm</li> <li>• yyyyMMdd</li> <li>• yyyyMMdd HH:mm:Ss</li> </ul>

Column	Details
	<ul style="list-style-type: none"> <li>• yyyyMMdd HH:mm</li> </ul> <p>Destination:</p> <p><code>ImportedComputer.InventoryDate</code></p>
ClusterID	<p><i>Type:</i> big integer. Key. Nullable</p> <p>The unique identifier for the cluster containing this computer. This must match the ClusterID used in the Cluster spreadsheet. If both the ClusterID and the ClusterNodeType do not match the data provided in the Cluster spreadsheet then the computer will not be associated with a cluster.</p> <p>Destination:</p> <p><code>ImportedClusterNode.ClusterExternalID</code></p>
ClusterNodeType	<p><i>Type:</i> text (max 128 characters). Nullable</p> <p>The Cluster node type of the computer. Must be a (case insensitive) exact match for one of the values shown. If both the ClusterID and the ClusterNodeType do not match the data provided in the Cluster spreadsheet then the computer will not be associated with a cluster.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• Active</li> <li>• Passive</li> <li>• Hot</li> <li>• Warm</li> <li>• Cold</li> </ul> <p>Destination:</p> <p><code>ImportedClusterNode.ClusterNodeTypeID</code></p>
HostID	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>The HostID hardware property for the server hosting this machine partition (when inventorying a machine partition such as Solaris Zone, AIX IPar, HP-UX nPar/vPar).</p> <p>Destination:</p> <p><code>ImportedComputer.HostID</code></p>
FirmwareSerialNumber	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>The Serial number in the system firmware such as BIOS, EEPROM etc.</p> <p>Destination:</p>



Column	Details
	<code>ImportedComputer.FirmwareSerialNumber</code>
MachineID	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>For AIX, it is the System ID. For HP-UX, it is the Machine/Software ID. It is unset for other platforms.</p> <p><i>Destination:</i></p> <p><code>ImportedComputer.MachineID</code></p>

## ConsolidatedFileEvidence Template

ConsolidatedFileEvidence provides a simpler interface to specify files and their usage on computers. It combines the computer, file evidence and usage details into a single row.

**Table 759: Columns included with ConsolidatedFileEvidence templates**

Column	Details
ComputerID	<p><i>Type:</i> big integer. Key</p> <p>The identifier used in the source connection for the computer. It must match the ComputerID from the Computer spreadsheet or the row will be ignored.</p> <p><i>Destination:</i></p> <p><code>ImportedInstalledFileEvidence.ExternalID</code></p> <p><code>ImportedInstalledFileEvidenceUsage.ExternalID</code></p>
FileName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The name of the file used as evidence of software installation. For unix operating systems include the full path in the file name, including the opening '/'. For Windows operating systems the file path is specified in the FilePath column and this column must only contain the file name.</p> <p><i>Destination:</i></p> <p><code>ImportedFileEvidence.ExternalFileID</code></p> <p><code>ImportedFileEvidence.FileName</code></p> <p><code>ImportedInstalledFileEvidence.ExternalFileID</code></p> <p><code>ImportedInstalledFileEvidenceUsage.ExternalFileID</code></p>
FileVersion	<p><i>Type:</i> text (max 100 characters). Key. Nullable</p> <p>The version number of the file used as evidence of software installation.</p> <p><i>Destination:</i></p>

Column	Details
	<p>ImportedFileEvidence.ExternalFileID</p> <p>ImportedFileEvidence.FileVersion</p> <p>ImportedInstalledFileEvidence.ExternalFileID</p> <p>ImportedInstalledFileEvidenceUsage.ExternalFileID</p>
ProductVersion	<p><b>Type:</b> text (max 200 characters). Nullable</p> <p>The product version number in the file header.</p> <p><b>Destination:</b></p> <p>ImportedFileEvidence.ProductVersion</p>
ProductName	<p><b>Type:</b> text (max 200 characters). Nullable</p> <p>The product name in the file header.</p> <p><b>Destination:</b></p> <p>ImportedFileEvidence.ProductName</p>
FilePath	<p><b>Type:</b> text (max 400 characters). Nullable</p> <p>The path of the file used as evidence of software installation.</p> <p><b>Destination:</b></p> <p>ImportedFileEvidence.FilePath</p>
Company	<p><b>Type:</b> text (max 100 characters). Key. Nullable</p> <p>The company in the file header.</p> <p><b>Destination:</b></p> <p>ImportedFileEvidence.ExternalFileID</p> <p>ImportedFileEvidence.Company</p> <p>ImportedInstalledFileEvidence.ExternalFileID</p> <p>ImportedInstalledFileEvidenceUsage.ExternalFileID</p>
Description	<p><b>Type:</b> text (max 200 characters). Key. Nullable</p> <p>The description in the file header.</p> <p><b>Destination:</b></p> <p>ImportedFileEvidence.ExternalFileID</p> <p>ImportedFileEvidence.Description</p> <p>ImportedInstalledFileEvidence.ExternalFileID</p> <p>ImportedInstalledFileEvidenceUsage.ExternalFileID</p>

Column	Details
FileSize	<p><i>Type:</i> integer. Key. Nullable</p> <p>The size of the file in bytes.</p> <p><b>Destination:</b></p> <p><code>ImportedFileEvidence.ExternalFileID</code></p> <p><code>ImportedFileEvidence.FileSize</code></p> <p><code>ImportedInstalledFileEvidence.ExternalFileID</code></p> <p><code>ImportedInstalledFileEvidenceUsage.ExternalFileID</code></p>
Language	<p><i>Type:</i> text (max 200 characters). Nullable</p> <p>The language in the file header.</p> <p><b>Destination:</b></p> <p><code>ImportedFileEvidence.Language</code></p>
AccessMode	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>The access mode of the file evidence. Leave this blank unless this row is a virtualized application. In that case choose one of the values below that matches your application or desktop virtualization infrastructure.</p> <p><b>Possible values:</b></p> <ul style="list-style-type: none"> <li>• Local</li> <li>• App-V</li> <li>• XenApp</li> <li>• XenDesktop</li> <li>• VMware View</li> </ul> <p><b>Destination:</b></p> <p><code>ImportedFileEvidence.ExternalFileID</code></p> <p><code>ImportedFileEvidence.AccessModeID</code></p> <p><code>ImportedInstalledFileEvidence.ExternalFileID</code></p> <p><code>ImportedInstalledFileEvidenceUsage.ExternalFileID</code></p>
NumberOfSessions	<p><i>Type:</i> big integer. Nullable</p> <p>The number of sessions that the file evidence was in use by the user specified in the UserID column during the usage tracking period. If multiple users used the same application on the computer, create one row for each user with usage.</p> <p><b>Destination:</b></p>

Column	Details
	<code>ImportedInstalledFileEvidenceUsage.NumberOfSessions</code>
<code>StartDate</code>	<p><i>Type:</i> text (max 10 characters). Nullable</p> <p>The start date of the usage. The date must be specified in the following format: 'yyyyMMdd'.</p> <p>Destination:</p> <p><code>ImportedInstalledFileEvidenceUsage.StartDate</code></p>
<code>LastUsedDate</code>	<p><i>Type:</i> text (max 10 characters). Nullable</p> <p>The last used date of the usage. The date must be specified in the following format: 'yyyyMMdd'.</p> <p>Destination:</p> <p><code>ImportedInstalledFileEvidenceUsage.LastUsedDate</code></p>
<code>UserID</code>	<p><i>Type:</i> big integer. Key. Nullable</p> <p>The DOMAIN/SAMAccountName for the user that the file evidence was used by. If this software was used by multiple users, create one row for each user of the software on the computer.</p> <p>Destination:</p> <p><code>ImportedInstalledFileEvidenceUsage.ExternalUserID</code></p> <p><code>ImportedUser.ExternalID</code></p> <p><code>ImportedUser.UserName</code> (Element 2 after splitting on '\')</p> <p><code>ImportedUser.Domain</code> (Element 1 after splitting on '\')</p> <p><code>ImportedUser.SAMAccountName</code> (Element 2 after splitting on '\')</p>

## ConsolidatedInstallerEvidence Template

ConsolidatedInstallerEvidence provides a simpler interface to specify installed applications and their usage on computers. It combines the computer, installer evidence and usage details into a single row.

**Table 760: Columns included with ConsolidatedInstallerEvidence templates**

Column	Details
<code>ComputerID</code>	<p><i>Type:</i> big integer. Key</p> <p>The identifier used in the source connection for the computer. It must match the ComputerID from the Computer spreadsheet or the row will be ignored.</p>

Column	Details
	<b>Destination:</b> ImportedInstalledInstallerEvidence.ExternalComputerID ImportedInstalledInstallerEvidenceUsage.ExternalID ImportedInstance.ExternalComputerID
DatabaseName	<b>Type:</b> big integer. Key. Nullable If this installer evidence is an Oracle database, then this field specifies the name of the database. <b>Destination:</b> ImportedInstalledInstallerEvidence.ExternalInstanceID ImportedInstalledInstallerEvidenceUsage.ExternalInstanceID ImportedInstance.InstanceID ImportedInstance.ParentInstanceID
InstanceName	<b>Type:</b> big integer. Key. Nullable If this installer evidence is an Oracle database, then this field specifies the name of the database instance. If there are multiple instances, create a row for each instance in this spreadsheet. <b>Destination:</b> ImportedInstalledInstallerEvidence.ExternalInstanceID ImportedInstalledInstallerEvidenceUsage.ExternalInstanceID ImportedInstance.InstanceID ImportedInstance.InstanceName
DisplayName	<b>Type:</b> text (max 256 characters). Key The display name of the software as reported by the installer evidence. <b>Destination:</b> ImportedInstallerEvidence.ExternalInstallerID ImportedInstallerEvidence.DisplayName ImportedInstalledInstallerEvidence.ExternalInstallerEvidenceID ImportedInstalledInstallerEvidenceUsage.ExternalInstallerID
Version	<b>Type:</b> text (max 72 characters). Key. Nullable The version of the software as reported by the installer evidence. <b>Destination:</b> ImportedInstallerEvidence.ExternalInstallerID

Column	Details
	<p>ImportedInstallerEvidence.Version</p> <p>ImportedInstalledInstallerEvidence.ExternalInstallerEvidenceID</p> <p>ImportedInstalledInstallerEvidenceUsage.ExternalInstallerID</p>
Publisher	<p><b>Type:</b> text (max 200 characters). Key. Nullable</p> <p>The publisher of the software as reported by the installer evidence.</p> <p><b>Destination:</b></p> <p>ImportedInstallerEvidence.ExternalInstallerID</p> <p>ImportedInstallerEvidence.Publisher</p> <p>ImportedInstalledInstallerEvidence.ExternalInstallerEvidenceID</p> <p>ImportedInstalledInstallerEvidenceUsage.ExternalInstallerID</p>
Evidence	<p><b>Type:</b> text (max 32 characters). Key. Nullable</p> <p>Identifier for the type of installer evidence.</p> <p><b>Destination:</b></p> <p>ImportedInstallerEvidence.ExternalInstallerID</p> <p>ImportedInstallerEvidence.Evidence</p> <p>ImportedInstalledInstallerEvidence.ExternalInstallerEvidenceID</p> <p>ImportedInstalledInstallerEvidenceUsage.ExternalInstallerID</p>
ProductCode	<p><b>Type:</b> text (max 55 characters). Nullable</p> <p>The product code of the evidence. This is usually the MSI product code.</p> <p><b>Destination:</b></p> <p>ImportedInstallerEvidence.ProductCode</p>
AccessMode	<p><b>Type:</b> text (max 128 characters). Key. Nullable</p> <p>The access mode of the installer evidence. Leave this blank unless this row is a virtualized application. In that case choose one of the values below that matches your application or desktop virtualization infrastructure.</p> <p><b>Possible values:</b></p> <ul style="list-style-type: none"> <li>• Local</li> <li>• App-V</li> <li>• XenApp</li> <li>• XenDesktop</li> <li>• VMware View</li> </ul>

Column	Details
	<p><b>Destination:</b></p> <p><code>ImportedInstallerEvidence.ExternalInstallerID</code></p> <p><code>ImportedInstallerEvidence.AccessModeID</code></p> <p><code>ImportedInstalledInstallerEvidence.ExternalInstallerEvidenceID</code></p> <p><code>ImportedInstalledInstallerEvidenceUsage.ExternalInstallerID</code></p>
InstallDate	<p><b>Type:</b> text (max 10 characters). Nullable</p> <p>The install date of the installer evidence. The date must be specified in the following format: 'yyyyMMdd'.</p> <p><b>Destination:</b></p> <p><code>ImportedInstalledInstallerEvidence.InstallDate</code></p>
DiscoveryDate	<p><b>Type:</b> text (max 10 characters). Nullable</p> <p>The date that the installer evidence was first seen. The date must be specified in the following format: 'yyyyMMdd'.</p> <p><b>Destination:</b></p> <p><code>ImportedInstalledInstallerEvidence.DiscoveryDate</code></p>
NumberOfSessions	<p><b>Type:</b> big integer. Nullable</p> <p>The number of sessions that the installer evidence was in use by the user specified in the UserID column during the usage tracking period. If multiple users used the same application on the computer, create one row for each user with usage.</p> <p><b>Destination:</b></p> <p><code>ImportedInstalledInstallerEvidenceUsage.NumberOfSessions</code></p>
StartDate	<p><b>Type:</b> text (max 10 characters). Nullable</p> <p>The start date of the usage. The date must be specified in the following format: 'yyyyMMdd'.</p> <p><b>Destination:</b></p> <p><code>ImportedInstalledInstallerEvidenceUsage.StartDate</code></p>
LastUsedDate	<p><b>Type:</b> text (max 10 characters). Nullable</p> <p>The last used date of the usage. The date must be specified in the following format: 'yyyyMMdd'.</p> <p><b>Destination:</b></p> <p><code>ImportedInstalledInstallerEvidenceUsage.LastUsedDate</code></p>

Column	Details
UserID	<p><i>Type:</i> big integer. Key. Nullable</p> <p>The DOMAIN/SAMAccountName for the user that the installer evidence was used by. If this software was used by multiple users, create one row for each user of the software on the computer.</p> <p>Destination:</p> <p>ImportedInstalledInstallerEvidenceUsage.ExternalUserID</p> <p>ImportedUser.ExternalID</p> <p>ImportedUser.UserName (Element 2 after splitting on '\')</p> <p>ImportedUser.Domain (Element 1 after splitting on '\')</p> <p>ImportedUser.SAMAccountName (Element 2 after splitting on '\')</p>

## ConsolidatedOracleDatabaseUser Template

ConsolidatedOracleDatabaseUser provides a list of the users for each Oracle database instance.

**Table 761: Columns included with ConsolidatedOracleDatabaseUser templates**

Column	Details
UserID	<p><i>Type:</i> big integer. Key</p> <p>The identifier used in the source connection for the instance end-user. This may be an integer or a string.</p> <p>Destination:</p> <p>ImportedInstanceUser.ExternalID</p> <p>ImportedLicenseUser.ExternalID</p>
ComputerID	<p><i>Type:</i> big integer. Key</p> <p>The identifier used in the source connection for the computer. It must match the ComputerID from the Computer spreadsheet or the row will be ignored.</p> <p>Destination:</p> <p>ImportedInstanceUser.ExternalID</p> <p>ImportedInstanceUser.ComputerID</p> <p>ImportedLicenseUser.ExternalID</p>
DatabaseName	<p><i>Type:</i> big integer. Key</p>



Column	Details
	<p>This field specifies the name of the database. It must match a row in the InstallerEvidence spreadsheet for the same ComputerID or this row will be skipped.</p> <p><b>Destination:</b></p> <p><code>ImportedInstanceUser.ExternalID</code></p> <p><code>ImportedInstanceUser.InstanceID</code></p> <p><code>ImportedLicenseUser.ExternalID</code></p>
InstanceName	<p><b>Type:</b> big integer. Key</p> <p>This field specifies the name of the database instance. If there are multiple instances, create a row for each instance in this spreadsheet. It must match a row in the InstallerEvidence spreadsheet for the same ComputerID and DatabaseName or this row will be skipped.</p> <p><b>Destination:</b></p> <p><code>ImportedInstanceUser.ExternalID</code></p> <p><code>ImportedInstanceUser.InstanceID</code></p> <p><code>ImportedLicenseUser.ExternalID</code></p>
Name	<p><b>Type:</b> text (max 256 characters)</p> <p>The name of the user.</p> <p><b>Destination:</b></p> <p><code>ImportedLicenseUser.UserName</code></p>
AccountStatus	<p><b>Type:</b> text (max 256 characters). Nullable</p> <p>The current status of the end-user account.</p> <p><b>Destination:</b></p> <p><code>ImportedInstanceUser.AccountStatus</code></p>
CreationDate	<p><b>Type:</b> datetime. Nullable</p> <p>The date and time when the end-user was created. The date must be entered in one of the supported formats.</p> <p><b>Possible values:</b></p> <ul style="list-style-type: none"> <li>• yyyy/MM/dd</li> <li>• yyyy/MM/dd HH:mm:Ss</li> <li>• yyyy/MM/dd HH:mm</li> <li>• yyyy-MM-dd</li> </ul>

Column	Details
	<ul style="list-style-type: none"> <li>• yyyy-MM-dd HH:mm:Ss</li> <li>• yyyy-MM-dd HH:mm</li> <li>• yyyyMMdd</li> <li>• yyyyMMdd HH:mm:Ss</li> <li>• yyyyMMdd HH:mm</li> </ul> <p>Destination:</p> <p><code>ImportedInstanceUser.CreationDate</code></p>
LastLogonDate	<p><i>Type:</i> datetime. Nullable</p> <p>The date and time when the end-user last logged on to the computer. The date must be entered in one of the supported formats.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• yyyy/MM/dd</li> <li>• yyyy/MM/dd HH:mm:Ss</li> <li>• yyyy/MM/dd HH:mm</li> <li>• yyyy-MM-dd</li> <li>• yyyy-MM-dd HH:mm:Ss</li> <li>• yyyy-MM-dd HH:mm</li> <li>• yyyyMMdd</li> <li>• yyyyMMdd HH:mm:Ss</li> <li>• yyyyMMdd HH:mm</li> </ul> <p>Destination:</p> <p><code>ImportedInstanceUser.LastLogonDate</code></p>
DefaultTablespace	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The default tablespace for an Oracle end-user.</p> <p>Destination:</p> <p><code>ImportedInstanceUser.DefaultTablespace</code></p>
TempTablespace	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The temporary tablespace for an Oracle end-user.</p> <p>Destination:</p> <p><code>ImportedInstanceUser.TempTablespace</code></p>

Column	Details
DisplayName	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The display name of the software as reported by the installer evidence. It must match a row in the InstallerEvidence spreadsheet for the same ComputerID, Version, Publisher, DatabaseName and InstanceName or this row will be skipped.</p> <p>Destination:</p> <p><code>ImportedInstanceUser.ApplicationID</code></p>
Version	<p><i>Type:</i> text (max 72 characters). Key</p> <p>The version of the software as reported by the installer evidence. It must match a row in the InstallerEvidence spreadsheet for the same ComputerID, DisplayName, Publisher, DatabaseName and InstanceName or this row will be skipped.</p> <p>Destination:</p> <p><code>ImportedInstanceUser.ApplicationID</code></p>
Publisher	<p><i>Type:</i> text (max 200 characters). Key</p> <p>The publisher of the software as reported by the installer evidence. It must match a row in the InstallerEvidence spreadsheet for the same ComputerID, DisplayName, Version, DatabaseName and InstanceName or this row will be skipped.</p> <p>Destination:</p> <p><code>ImportedInstanceUser.ApplicationID</code></p>
Evidence	<p><i>Type:</i> text (max 32 characters). Key. Nullable</p> <p>Identifier for the type of installer evidence.</p> <p>Destination:</p> <p><code>ImportedInstanceUser.ApplicationID</code></p>
AccessMode	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>The access mode of the installer evidence. Leave this blank unless this row is a virtualized application. In that case choose one of the values below that matches your application or desktop virtualization infrastructure.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• Local</li> <li>• App-V</li> <li>• XenApp</li> </ul>

Column	Details
	<ul style="list-style-type: none"> <li>XenDesktop</li> <li>VMware View</li> </ul> <p>Destination:</p> <p><code>ImportedInstanceUser.ApplicationID</code></p>

## ConsolidatedRemoteAccessFile Template

The RemoteAccessFile spreadsheet is used for capturing application virtualization information. Systems such as Microsoft AppV and Citrix XenApp allow a user to access applications that are not installed on a local computer. This object allows you to provide applications that a user may access by specifying the file evidence.

When populating the RemoteAccessFile template, please note that an application can be identified by file evidence. If the evidence does not match the ARL then no application will be created. The evidence not recognised will appear under the 'Unrecognised Evidence' screen within Flexnet Manager Suite. From there, you may create applications for any unrecognised evidence as required, and lastly ensure any new application relates to a license. This results in the evidence now being recognised for the new application and may cause license consumption after the next reconciliation. This application virtualization access using files is a special case in application matching. It does not require a mandatory file link to the application and can use a 'not for recognition' file to link to an application. This is because application and desktop virtualization systems rarely provide enough file information for more complex application recognition rules to function.

If entering file evidence, you must provide the following key identifier fields. + 1 = FileName

The following identifier fields are typically required for matching evidence in the ARL, however are not mandatory.  
+ 1 = Company + 2 = FileVersion + 3 = Description + 4 = FileSize

File evidence does not have to be specified in the FileEvidence spreadsheet as well as here.

**Table 762: Columns included with ConsolidatedRemoteAccessFile templates**

Column	Details
ServerID	<p><i>Type:</i> big integer. Key</p> <p>This is the ComputerID of the server that publishes this virtual application. The ComputerID must match a computer from the Computer spreadsheet, and that computer must have an installation of the application this file is part of. If the server does not have an installation of an appropriate application then the user will not be shown as having access to that application. This is a mandatory field.</p> <p>Destination:</p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalServerID</code></p>
FileName	<p><i>Type:</i> text (max 256 characters). Key</p>

Column	Details
	<p>The name of the file used as evidence of software installation. For unix operating systems include the full path in the file name, including the opening '/'. For Windows operating systems the file path is specified in the FilePath column and this column must only contain the file name.</p> <p>Destination:</p> <p>ImportedRemoteUserToApplicationAccess.ExternalFileID</p> <p>ImportedFileEvidence.ExternalFileID</p> <p>ImportedFileEvidence.FileName</p>
FileVersion	<p><i>Type:</i> text (max 100 characters). Key. Nullable</p> <p>The version number of the file used as evidence of software installation.</p> <p>Destination:</p> <p>ImportedRemoteUserToApplicationAccess.ExternalFileID</p> <p>ImportedFileEvidence.ExternalFileID</p> <p>ImportedFileEvidence.FileVersion</p>
ProductVersion	<p><i>Type:</i> text (max 200 characters). Nullable</p> <p>The product version number in the file header.</p> <p>Destination:</p> <p>ImportedFileEvidence.ProductVersion</p>
ProductName	<p><i>Type:</i> text (max 200 characters). Nullable</p> <p>The product name in the file header.</p> <p>Destination:</p> <p>ImportedFileEvidence.ProductName</p>
FilePath	<p><i>Type:</i> text (max 400 characters). Nullable</p> <p>The path of the file used as evidence of software installation.</p> <p>Destination:</p> <p>ImportedFileEvidence.FilePath</p>
Company	<p><i>Type:</i> text (max 100 characters). Key. Nullable</p> <p>The company in the file header.</p> <p>Destination:</p> <p>ImportedRemoteUserToApplicationAccess.ExternalFileID</p> <p>ImportedFileEvidence.ExternalFileID</p>

Column	Details
	<code>ImportedFileEvidence.Company</code>
Description	<p><b>Type:</b> text (max 200 characters). Key. Nullable</p> <p>The description in the file header.</p> <p><b>Destination:</b></p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalFileID</code></p> <p><code>ImportedFileEvidence.ExternalFileID</code></p> <p><code>ImportedFileEvidence.Description</code></p>
FileSize	<p><b>Type:</b> integer. Key. Nullable</p> <p>The size of the file in bytes.</p> <p><b>Destination:</b></p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalFileID</code></p> <p><code>ImportedFileEvidence.ExternalFileID</code></p> <p><code>ImportedFileEvidence.FileSize</code></p>
Language	<p><b>Type:</b> text (max 200 characters). Nullable</p> <p>The language in the file header.</p> <p><b>Destination:</b></p> <p><code>ImportedFileEvidence.Language</code></p>
UserID	<p><b>Type:</b> big integer. Key</p> <p>The UserID must be populated with the fully qualified name e.g. Mydomain \JohnSmith. If not then a User is not created.</p> <p>If fully qualified then this field populates the following user related fields. + 1 = The user name of the end-user from the text following the "\". + 2 = The login name (SAM account name) of the end-user from the text following the "\". + 3 = The domain name of the end-user from the text before the "\".</p> <p><b>Destination:</b></p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalUserID</code></p> <p><code>ImportedUser.ExternalID</code></p> <p><code>ImportedUser.UserName (Element 2 after splitting on '\')</code></p> <p><code>ImportedUser.Domain (Element 1 after splitting on '\')</code></p> <p><code>ImportedUser.SAMAccountName (Element 2 after splitting on '\')</code></p>
AccessMode	<b>Type:</b> text (max 128 characters). Key. Nullable

Column	Details
	<p>The AccessMode states how an application has been accessed.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• Local</li> <li>• App-V</li> <li>• XenApp</li> <li>• XenDesktop</li> <li>• VMware View</li> </ul> <p>Destination:</p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalFileID</code></p> <p><code>ImportedRemoteUserToApplicationAccess.AccessModeID</code></p> <p><code>ImportedFileEvidence.ExternalFileID</code></p> <p><code>ImportedFileEvidence.AccessModeID</code></p>

## ConsolidatedRemoteAccessInstaller Template

The RemoteAccessInstaller spreadsheet is used for capturing application virtualization information. Systems such as Microsoft AppV and Citrix XenApp allow a user to access applications that are not installed on a local computer. This object allows you to provide applications that a user may access by specifying the installer evidence.

When populating the RemoteAccessInstaller, please note that an application can be identified by installer evidence. If the evidence does not match the ARL then no application will be created. The evidence not recognised will appear under the 'Unrecognised Evidence' screen within Flexnet Manager Suite. From there, you may create applications for any unrecognised evidence as required, and lastly ensure any new application relates to a license. This results in the evidence now being recognised for the new application and may cause license consumption after the next reconciliation.

If entering installer evidence, you must provide the following key identifier fields. + 1 = DisplayName

The following identifier fields are typically required for matching evidence in the ARL, however are not mandatory.  
+ 1 = Version + 2 = Publisher + 3 = Evidence

Installer evidence does not have to be specified in the InstallerEvidence spreadsheet as well as here.

**Table 763: Columns included with ConsolidatedRemoteAccessInstaller templates**

Column	Details
DisplayName	Type: text (max 256 characters). Key

Column	Details
	<p>The display name of the software as reported by the installer evidence and is part of the unique identifier for installer evidence.</p> <p><b>Destination:</b></p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalInstallerEvidenceID</code></p> <p><code>ImportedInstallerEvidence.ExternalInstallerID</code></p> <p><code>ImportedInstallerEvidence.DisplayName</code></p>
Version	<p><b>Type:</b> text (max 72 characters). Key</p> <p>The version of the software as reported by the installer evidence and is part of the unique identifier for installer evidence.</p> <p><b>Destination:</b></p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalInstallerEvidenceID</code></p> <p><code>ImportedInstallerEvidence.ExternalInstallerID</code></p> <p><code>ImportedInstallerEvidence.Version</code></p>
Publisher	<p><b>Type:</b> text (max 200 characters). Key</p> <p>Publishers of software applications (for example, "Microsoft") as reported by the installer evidence and publisher is part of the unique identifier for installer evidence.</p> <p><b>Destination:</b></p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalInstallerEvidenceID</code></p> <p><code>ImportedInstallerEvidence.ExternalInstallerID</code></p> <p><code>ImportedInstallerEvidence.Publisher</code></p>
Evidence	<p><b>Type:</b> text (max 32 characters). Key</p> <p>The evidence type of the software as reported by the installer evidence and is part of the unique identifier for installer evidence.</p> <p><b>Destination:</b></p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalInstallerEvidenceID</code></p> <p><code>ImportedInstallerEvidence.ExternalInstallerID</code></p> <p><code>ImportedInstallerEvidence.Evidence</code></p>
ProductCode	<p><b>Type:</b> text (max 55 characters). Nullable</p> <p>The product code of the evidence. This is usually the MSI product code and is not part of the unique identifier.</p> <p><b>Destination:</b></p>



Column	Details
	<code>ImportedInstallerEvidence.ProductCode</code>
UserID	<p><i>Type:</i> big integer. Key</p> <p>The UserID must be populated with the fully qualified name e.g. Mydomain\JohnSmith. If not then a User is not created.</p> <p>If fully qualified then this field populates the following user related fields. + 1 = The user name of the end-user from the text following the ". + 2 = The login name (SAM account name) of the end-user from the text following the ". + 3 = The domain name of the end-user from the text before the ".</p> <p><b>Destination:</b></p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalUserID</code></p> <p><code>ImportedUser.ExternalID</code></p> <p><code>ImportedUser.UserName</code> (Element 2 after splitting on '\')</p> <p><code>ImportedUser.Domain</code> (Element 1 after splitting on '\')</p> <p><code>ImportedUser.SAMAccountName</code> (Element 2 after splitting on '\')</p>
AccessMode	<p><i>Type:</i> text (max 128 characters). Key. Nullable</p> <p>The AccessMode states how an application has been accessed.</p> <p><b>Possible values:</b></p> <ul style="list-style-type: none"> <li>• Local</li> <li>• App-V</li> <li>• XenApp</li> <li>• XenDesktop</li> <li>• VMware View</li> </ul> <p><b>Destination:</b></p> <p><code>ImportedRemoteUserToApplicationAccess.ExternalInstallerEvidenceID</code></p> <p><code>ImportedRemoteUserToApplicationAccess.AccessModeID</code></p> <p><code>ImportedInstallerEvidence.ExternalInstallerID</code></p> <p><code>ImportedInstallerEvidence.AccessModeID</code></p>

## ConsolidatedVMPool Template

The VMPool spreadsheet provides a simple method to associate virtual machines with groups (pools) on their host.

**Table 764: Columns included with ConsolidatedVMPool templates**

Column	Details
PoolName	<p><i>Type:</i> text (max 100 characters). Key</p> <p>The name of the pool.</p> <p>Destination:</p> <p><code>ImportedVMPool.PoolName</code></p>
ParentName	<p><i>Type:</i> text (max 100 characters). Nullable</p> <p>The name of the parent pool.</p> <p>Destination:</p> <p><code>ImportedVMPool.ParentName</code></p>
PoolFriendlyName	<p><i>Type:</i> text (max 256 characters)</p> <p>The friendly name of the pool.</p> <p>Destination:</p> <p><code>ImportedVMPool.PoolFriendlyName</code></p>
HostComputerID	<p><i>Type:</i> big integer. Key</p> <p>The identifier used in the source connection for the computer which is hosting the pool. The HostComputerID should match the ComputerID in the Computer spreadsheet. Otherwise the record will be ignored.</p> <p>Destination:</p> <p><code>ImportedVMPool.HostComputerID</code></p>
ObjectType	<p><i>Type:</i> text (max 256 characters). Key. Nullable</p> <p>The type of pool.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• Folder</li> <li>• Datacenter</li> <li>• ComputeResource</li> <li>• HostSystem</li> <li>• ResourcePool</li> <li>• VirtualMachine</li> <li>• PhysicalSharedPool</li> <li>• VirtualSharedPool</li> </ul>

Column	Details
	<ul style="list-style-type: none"> <li>• LPAR</li> <li>• RSET</li> <li>• ClusterComputeResource</li> </ul> <p>Destination:</p> <p><code>ImportedVMPool.ObjectType</code></p>
ComplianceConnectionID	<p><i>Type:</i> integer. Key. Nullable</p> <p>The identifier for a data source connection in the <code>ComplianceConnection</code> table.</p> <p>Destination:</p> <p><code>ImportedVMPool.ComplianceConnectionID</code></p>
ParentObjectType	<p><i>Type:</i> text (max 256 characters). Nullable</p> <p>The type of pool of the parent.</p> <p>Destination:</p> <p><code>ImportedVMPool.ParentObjectType</code></p>
NumberOfProcessors	<p><i>Type:</i> decimal. Nullable</p> <p>The number of processors in this pool.</p> <p>Possible values:</p> <p>120.45</p> <p>Destination:</p> <p><code>ImportedVMPool.NumberOfProcessors</code></p>
NumberOfCores	<p><i>Type:</i> decimal. Nullable</p> <p>The number of cores in this pool.</p> <p>Possible values:</p> <p>120.45</p> <p>Destination:</p> <p><code>ImportedVMPool.NumberOfCores</code></p>

## ConsolidatedWMIEvidence Template

ConsolidatedWMIEvidence provides a simpler interface to specify Windows Management Instrumentation (WMI) properties on computers. Other Web-Based Enterprise Management (WBEM) properties are supported from

Unix computers as well. The most important data to provide in this spreadsheet is operating system installs. The 'Win32\_OperatingSystem' class and the 'Name' property contains this data.

**Table 765: Columns included with Consolidated WMI Evidence templates**

Column	Details
ComputerID	<p><i>Type:</i> big integer. Key</p> <p>The identifier used in the source connection for the computer. It must match the ComputerID from the Computer spreadsheet or the row will be ignored.</p> <p><b>Destination:</b></p> <p><code>ImportedInstalledWMIEvidence.ExternalComputerID</code></p>
ClassName	<p><i>Type:</i> text (max 50 characters). Key</p> <p>The WMI class name of the evidence. An example is 'Win32_OperatingSystem'.</p> <p><b>Destination:</b></p> <p><code>ImportedWMIEvidence.ExternalEvidenceID</code></p> <p><code>ImportedWMIEvidence.ClassName</code></p> <p><code>ImportedInstalledWMIEvidence.ExternalEvidenceID</code></p>
PropertyName	<p><i>Type:</i> text (max 50 characters). Key</p> <p>The WMI property name of the WMI evidence. An example is 'Name'.</p> <p><b>Destination:</b></p> <p><code>ImportedWMIEvidence.ExternalEvidenceID</code></p> <p><code>ImportedWMIEvidence.PropertyName</code></p> <p><code>ImportedInstalledWMIEvidence.ExternalEvidenceID</code></p>
PropertyValue	<p><i>Type:</i> text (max 256 characters). Key</p> <p>The value of the property of the WMI evidence. An example is 'Microsoft Windows 7 Enterprise'</p> <p><b>Destination:</b></p> <p><code>ImportedWMIEvidence.ExternalEvidenceID</code></p> <p><code>ImportedWMIEvidence.PropertyValue</code></p> <p><code>ImportedInstalledWMIEvidence.ExternalEvidenceID</code></p>
InstanceName	<p><i>Type:</i> text (max 256 characters). Key. Nullable</p> <p>The name of the WMI class instance. This is important when there a multiple instances of a WMI class on a computer. An example is the 'Win32_VideoController' class that may have many instances with the same</p>

Column	Details
	<p>properties. In this case you need to specify the name of the instance here, 'Intel(R) HD Graphics Family' or 'NVIDIA Quadro K2100M' for example.</p> <p>Destination:</p> <p><code>ImportedWMIEvidence.ExternalEvidenceID</code></p> <p><code>ImportedInstalledWMIEvidence.ExternalEvidenceID</code></p> <p><code>ImportedInstalledWMIEvidence.InstanceName</code></p>