Inventory Database Schema 7.5 Part 4 - Software and User

**Inventory Database Schema Part 4**

In Part 4 the Software and User Inventory data classes are covered. Whether you need to create a custom report, or a custom computer filter based on Inventory Data in Inventory Solution 7.5, knowing the database schema is important. In Inventory Solution 7.5 the data structure for inventory follows closely with previous schemas in the 7.x version family. This document is meant to help understand not only the dependencies between the hardware tables, but provide information on the columns and their purpose.

[Introduction 2](#_Toc400700325)

[Database Schema 2](#_Toc400700326)

[User Inventory 2](#_Toc400700327)

[Inv\_UG\_UserAccount – BASE Class 3](#_Toc400700328)

[Inv\_UG\_UserAccount\_UNIX – SUB Class 3](#_Toc400700329)

[Inv\_UG\_UserAccount\_Windows – SUB Class 3](#_Toc400700330)

[Inv\_UG\_AdminGroupMembers – STANDARD Class 4](#_Toc400700331)

[Inv\_UG\_Group – STANDARD Class 4](#_Toc400700332)

[Inv\_UG\_UserUsage\_UNIX – STANDARD Class 5](#_Toc400700333)

[Software Inventory 5](#_Toc400700334)

[Inv\_SW\_Antivirus – STANDARD Class 5](#_Toc400700335)

[Inv\_SW\_AuditResults – STANDARD Class 6](#_Toc400700336)

[Inv\_SW\_AvailableUpdates\_UNIX – STANDARD Class 6](#_Toc400700337)

[Inv\_SW\_BIOSElement – STANDARD Class 7](#_Toc400700338)

[Inv\_SW\_DeviceDriver\_Windows – STANDARD Class 7](#_Toc400700339)

[Inv\_SW\_DiskUsageByFileType – STANDARD Class 8](#_Toc400700340)

[Inv\_SW\_Fonts – STANDARD Class 8](#_Toc400700341)

[Inv\_SW\_Patch\_Windows – STANDARD Class 8](#_Toc400700342)

[Inv\_SW\_SMBIOSElement – STANDARD Class 9](#_Toc400700343)

[Inv\_SW\_Virtual\_Software\_Packages\_Windows – STANDARD Class 9](#_Toc400700344)

[Inv\_SW\_Virtual\_Software\_Sublayers\_Windows – STANDARD Class 10](#_Toc400700345)

[Conclusion 10](#_Toc400700346)

# Introduction

Documenting database schema is not an easy task. SQL can provide a table-column view of all selected tables, but this does not account for any interdependencies between tables in a normalized environment, or what the column is meant for. In 7.5 Normalization includes dependencies on base-class tables, or, in other words, subsequent tables are extensions of the base table.

*NOTE: The information in this document may change, though at the time of publication this is believed to be the accurate information for the release of Inventory Solution 7.5.*

# Database Schema

The following list reveals data classes and their structure. First, the name of the table is given, followed by a designation as a Base or Sub Class data class. The following label system is used:

* BASE Class – This is a data class that has no dependencies on other Inventory data classes
* SUB Class – This is a data class that has 1 dependency on a BASE data class
* STANDARD Class – This is a data class that contains no dependencies or sub classes

Note the following when reviewing the grids:

1. Display Name represents how the column will be labeled when working within the Symantec Management Console, including reports, Resource Manager details, Pickers, etc.
2. Not all values have descriptions, but the label of the table and column should provide data on what’s stored therein.
3. When a BASE Class is listed, all subsequent SUB classes are tied to that BASE Class, sequentially, below.
4. Qualifiers per Column/Value are provided based on applicability.

For help in navigating dependent data classes, use the following guide:

* Inv\_UG\_UserAccount
  + Inv\_UG\_UserAccount\_UNIX
  + Inv\_UG\_UserAccount\_Windows

# User Inventory

The following data classes are for User-based data:

## Inv\_UG\_UserAccount – BASE Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Description** | string(256) |  |  |  |  |  |
| Description | Description of the object. | | | |
| **Domain** | string(256) |  |  |  |  |  |
| Description | User domain or computer hostname for local users. | | | |
| Key | true | | | |
| **FullName** | string(256) |  |  |  |  |  |
| Description | Full name of the user belonging to the network login profile. | | | |
| DisplayName | Full Name | | | |
| **HomeDirectory** | string(256) |  |  |  |  |  |
| Description | Path to the home directory of the user. | | | |
| DisplayName | Home Directory | | | |
| **PrimaryGroupID** | string(256) |  |  |  |  |  |
| Description | The identifier verifies the primary group to which the user's profile belongs. | | | |
| DisplayName | Primary Group ID | | | |
| **UserID** | string(256) |  |  |  |  |  |
| Description | User identifier. Win32\_NetworkLoginProfile.UserID for Windows, UID for UNIX. | | | |
| DisplayName | User ID | | | |
| **UserName** | string(256) |  |  |  |  |  |
| Description | User account on a particular domain or computer. | | | |
| DisplayName | User Name | | | |
| Key | true | | | |

### Inv\_UG\_UserAccount\_UNIX – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **LoginShell** | string(256) |  |  |  |  |  |
| Description | The user's initial shell . | | | |
| DisplayName | Login Shell | | | |

### Inv\_UG\_UserAccount\_Windows – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **AccountExpires** | datetime |  |  |  |  |  |
| Description | This gives when the Account will expire. | | | |
| DisplayName | Account Expires | | | |
| **CodePage** | uint32 |  |  |  |  |  |
| Description | Code page for the user's language of choice. | | | |
| DisplayName | Code Page | | | |
| **CountryCode** | uint32 |  |  |  |  |  |
| Description | Country/region code for the user's language of choice. | | | |
| DisplayName | Country Code | | | |
| **Flags** | uint32 |  |  |  |  |  |
| Description | The properties available to this network profile.(e.g. 2-Account disabled, 512, Normal account, 8388608-Password expired) | | | |
| valueMap | 1, 2, 8, 16, 32, 64, 128, 256, 512, 2048, 4096, 8192, 65536, 1131072, 262144, 524288, 1048576, 2097152, 4194304, 8388608 | | | |
| values | Script, Account Disabled, Home Directory Required, Lockout, Password Not Required, Password Cannot Change, Encrypted Test Password Allowed, Temp Duplicate Account, Normal Account, Interdomain Trust Account, Workstation Trust Account, Server Trust Account, Do Not Expire Password, MNS Logon Account, Smartcard Required, Trusted for Delegation, Not Delegated, Use DES Key Only, Do Not Require Preauthorization, Password Expired | | | |
| **LastLogin** | datetime |  |  |  |  |  |
| Description | User last logged on to the system. | | | |
| DisplayName | Last Login | | | |
| **LogonHours** | string(256) |  |  |  |  |  |
| Description | Times during the week when the user can log on. Each bit represents a unit of time specified by the UnitsPerWeek property. | | | |
| DisplayName | Logon Hours | | | |
| **LogonServer** | string(256) |  |  |  |  |  |
| Description | Name of the server to which logon requests are sent. | | | |
| DisplayName | Logon Server | | | |
| **MaximumStorage** | uint64 |  |  |  |  |  |
| Description | Maximum amount of disk space available to the user. | | | |
| DisplayName | Maximum Storage (Bytes) | | | |
| **NumberOfLogons** | uint32 |  |  |  |  |  |
| Description | Number of successful times the user tried to log on to this account. | | | |
| DisplayName | Number Of Logons | | | |
| **PasswordExpires** | datetime |  |  |  |  |  |
| Description | Date and time the password expires. | | | |
| DisplayName | Password Expires | | | |
| **Privileges** | uint32 |  |  |  |  |  |
| Description | Level of privilege assigned (e.g. 0-Guest, 1-User, 2-Administrator) | | | |
| ValueMap | 0, 1, 2 | | | |
| Values | Guest, User, Administrator | | | |
| **Profile** | string(256) |  |  |  |  |  |
| Description | Path to the user's profile. | | | |

## Inv\_UG\_AdminGroupMembers – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Domain** | string(256) |  |  |  |  |  |
| Description | This specifies the domain name of the user from Admin group or the computer hostname for local users. | | | |
| Key | true | | | |
| **MemberName** | string(256) |  |  |  |  |  |
| Description | This specifies the name of the user from Admin group. | | | |
| DisplayName | Member Name | | | |
| Key | true | | | |

## Inv\_UG\_Group – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Description** | string(256) |  |  |  |  |  |
| Description | Textual description of the object. | | | |
| **GroupID** | string(256) |  |  |  |  |  |
| Description | The group identifier, SID for Windows, numerical group ID for UNIX. | | | |
| DisplayName | Group ID | | | |
| Key | true | | | |
| **Name** | string(256) |  |  |  |  |  |
| Description | The name of the group. | | | |

## Inv\_UG\_UserUsage\_UNIX – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **AccessPoint** | string(256) |  |  |  |  |  |
| Description | Where the connection was made from. For remote connections this field will contain the remote host name or IP address, for local connections will be always "local" | | | |
| DisplayName | Access Point | | | |
| **ConnectionType** | string(256) |  |  |  |  |  |
| Description | How the connection was made: console, terminal window and so on. | | | |
| DisplayName | Connection Type | | | |
| **Domain** | string(256) |  |  |  |  |  |
| Description | User domain or the computer hostname for local users. | | | |
| **InstanceID** | string(256) |  |  |  |  |  |
| Description | Unique identificator for class object | | | |
| DisplayName | Instance ID | | | |
| Key | true | | | |
| **LoginSessionEndTime** | datetime |  |  |  |  |  |
| Description | The datetime value when the user logged out. | | | |
| DisplayName | Login Session End Time | | | |
| **LoginSessionStartTime** | datetime |  |  |  |  |  |
| Description | The datetime value when the user logged in. | | | |
| DisplayName | Login Session Start Time | | | |
| **UserName** | string(256) |  |  |  |  |  |
| Description | The login user name. | | | |
| DisplayName | User Name | | | |

# Software Inventory

The following data classes are for Software-based data:

## Inv\_SW\_Antivirus – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **CentrallyManaged** | boolean |  |  |  |  |  |
| Description | Value specifying a managed / unmanaged / standalone antivirus machine. | | | |
| DisplayName | Centrally Managed | | | |
| **DisplayName** | string(256) |  |  |  |  |  |
| Description | Name of the antivirus. | | | |
| DisplayName | Name | | | |
| Key | true | | | |
| **DisplayVersion** | string(256) |  |  |  |  |  |
| Description | Version of the antivirus. | | | |
| DisplayName | Version | | | |
| **LastScanTime** | datetime |  |  |  |  |  |
| Description | The time when the last virus scan was performed scheduled or manual scan. | | | |
| DisplayName | Last Scan Time | | | |
| **LastVirusDefinitionCheck** | datetime |  |  |  |  |  |
| Description | The time when the virus definitions on a machine were last updated. | | | |
| DisplayName | Last Virus Definition Check | | | |
| **Parent** | string(256) |  |  |  |  |  |
| Description | This is the server name of the antivirus machine | | | |
| DisplayName | Server Name | | | |
| **PatternFileRevision** | string(256) |  |  |  |  |  |
| Description | This is the revision number of the virus definition file | | | |
| DisplayName | Virus Definition File Version | | | |
| **RealTimeEnabled** | boolean |  |  |  |  |  |
| Description | Value specifying if virus protection is enabled / disabled on a machine | | | |
| DisplayName | Real Time Enabled | | | |
| **VirusDefFileDate** | datetime |  |  |  |  |  |
| Description | This is the date when the virus definition file was originally created | | | |
| DisplayName | Virus Definition File Date | | | |

## Inv\_SW\_AuditResults – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **FilesExamined** | uint32 |  |  |  |  |  |
| Description | The total number of files examined. | | | |
| DisplayName | Files Examined | | | |
| **ScanDuration** | uint32 |  |  |  |  |  |
| Description | The scan duration in minutes. | | | |
| DisplayName | Scan Duration (Minutes) | | | |
| Units | Minutes | | | |
| **ScanTime** | datetime |  |  |  |  |  |
| Description | Datetime value indicates when the software scanning started. | | | |
| DisplayName | Scan Time | | | |
| Key | true | | | |
| **TotalProductsReported** | uint32 |  |  |  |  |  |
| Description | The total number of software products reported. | | | |
| DisplayName | Total Products Reported | | | |

## Inv\_SW\_AvailableUpdates\_UNIX – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Name** | string(256) |  |  |  |  |  |
| Description | The available software update name. | | | |
| DisplayName | Name | | | |
| **RequiresRestart** | boolean |  |  |  |  |  |
| Description | True if the update requires computer restart after install. | | | |
| DisplayName | Requires Restart | | | |
| **Severity** | uint32 |  |  |  |  |  |
| Description | Describes the severity of the available update. Example: "recommended", "optional". | | | |
| DisplayName | Severity | | | |
| **Size** | uint64 |  |  |  |  |  |
| Description | Total size of the update in bytes. | | | |
| DisplayName | Size (Bytes) | | | |
| Units | Bytes | | | |
| **UpdateKey** | string(64) |  |  |  |  |  |
| Description | An identifier for the update. | | | |
| DisplayName | Update Key | | | |
| Key | true | | | |
| **Version** | string(256) |  |  |  |  |  |
| Description | The version of the update. | | | |
| DisplayName | Version | | | |

## Inv\_SW\_BIOSElement – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **BuildNumber** | string(64) |  |  |  |  |  |
| Description | The internal identifier for this compilation of this software element. | | | |
| DisplayName | Build Number | | | |
| **IdentificationCode** | string(64) |  |  |  |  |  |
| Description | Often this will be a stock keeping unit (SKU) or a part number. | | | |
| DisplayName | Identification Code | | | |
| **Manufacturer** | string(256) |  |  |  |  |  |
| Description | Manufacturer of this BIOS. | | | |
| **Name** | string(256) |  |  |  |  |  |
| Description | The name used to identify this SoftwareElement. | | | |
| Key | true | | | |
| **ReleaseDate** | datetime |  |  |  |  |  |
| Description | Release date of the BIOS in the UTC format. | | | |
| DisplayName | Release Date | | | |
| **Version** | string(64) |  |  |  |  |  |
| Description | Version of the BIOS. | | | |

## Inv\_SW\_DeviceDriver\_Windows – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **DeviceID** | string(256) |  |  |  |  |  |
| Description | Identifier (unique to the computer system) for different devices. | | | |
| DisplayName | Device ID | | | |
| Key | true | | | |
| **DriverProvider** | string(256) |  |  |  |  |  |
| Description | Name of the Windows device driver. | | | |
| DisplayName | Driver Provider | | | |
| **DriverVersion** | string(64) |  |  |  |  |  |
| Description | Version number of the device driver. | | | |
| DisplayName | Driver Version | | | |
| **InfFileName** | string(50) |  |  |  |  |  |
| Description | Name of the .inf file for the Windows device. | | | |
| DisplayName | Inf File Name | | | |
| **InfSection** | string(50) |  |  |  |  |  |
| Description | Section of the .inf file where the Windows information resides. | | | |
| DisplayName | Inf Section | | | |
| **Service** | string(255) |  |  |  |  |  |
| Description |  | | | |
| DisplayName | Service | | | |

## Inv\_SW\_DiskUsageByFileType – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Drive** | string(1024) |  |  |  |  |  |
| Description | Drive name. | | | |
| DisplayName | Drive | | | |
| Key | true | | | |
| **FileType** | string(40) |  |  |  |  |  |
| Description | File type. | | | |
| DisplayName | File Type | | | |
| Key | true | | | |
| **TotalFileSizes** | uint64 |  |  |  |  |  |
| Description | Total file sizes in kilobytes. | | | |
| DisplayName | Total File Sizes (KiloBytes) | | | |
| Units | KiloBytes | | | |
| **TotalFilesReported** | uint32 |  |  |  |  |  |
| Description | Total files reported. | | | |
| DisplayName | Total Files Reported | | | |

## Inv\_SW\_Fonts – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **FileName** | string(256) |  |  |  |  |  |
| Description | The name of file containing font. | | | |
| DisplayName | File Name | | | |
| **FontName** | string(256) |  |  |  |  |  |
| Description | The font name. | | | |
| DisplayName | Font Name | | | |
| **FontType** | string(256) |  |  |  |  |  |
| Description | The font type. Example: "Bitmap/vector", "TrueType", "OpenType PostScrip", "PostScript Type 1" | | | |
| DisplayName | Font Type | | | |
| **Path** | string(256) |  |  |  |  |  |
| Description | The path to directory wrere the font file is located. | | | |
| **SoftwareElementID** | string(256) |  |  |  |  |  |
| Description | An identifier for the font. | | | |
| DisplayName | Instance ID | | | |
| Key | true | | | |

## Inv\_SW\_Patch\_Windows – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Description** | string(256) |  |  |  |  |  |
| Description | Description of the object. | | | |
| **InstalledBy** | string(256) |  |  |  |  |  |
| Description | Person who installed the update. If this value is unknown, the property is empty. | | | |
| DisplayName | Installed By | | | |
| **InstalledDate** | datetime |  |  |  |  |  |
| Description | Object was installed. | | | |
| DisplayName | Installed Date | | | |
| **PatchID** | string |  |  |  |  |  |
| Description | Unique identifier associated with a particular update. | | | |
| DisplayName | Patch ID | | | |
| Key | true | | | |
| **RegKey** | string(256) |  |  |  |  |  |
| Description | This specifies the unique registration key. | | | |
| DisplayName | Reg Key | | | |
| **ServicePack** | string(256) |  |  |  |  |  |
| Description | Service pack in effect when the update was applied. | | | |
| DisplayName | Service Pack | | | |
| **Type** | string(256) |  |  |  |  |  |
| Description | This specifies the type of services and hot-fixes installed on the computer. | | | |

## Inv\_SW\_SMBIOSElement – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **SMBIOSMajorVersion** | uint16 |  |  |  |  |  |
| Description | Identifies the major version of SMBIOS specification implemented in the table structures, e.g. the value will be 0Ah for revision 10.22 and 02h for revision 2.1. This property corresponds to the value with offset 06h of SMBIOS Structure Table Entry Point. | | | |
| DisplayName | SMBIOS Major Version | | | |
| Key | true | | | |
| **SMBIOSMinorVersion** | uint16 |  |  |  |  |  |
| Description | Identifies the minor version of this specification implemented in the table structures, e.g. the value will be 16h for revision 10.22 and 01h for revision 2.1.This property corresponds to the value with offset 07h of SMBIOS Structure Table Entry Point. | | | |
| DisplayName | SMBIOS Minor Version | | | |

## Inv\_SW\_Virtual\_Software\_Packages\_Windows – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **ActivatedTime** | datetime |  |  |  |  |  |
| Description | When the layer was last activated. Null if the layer has never been activated. The activated time will be reported in NS date time format, example: '2006-10-17T10:32:06'. | | | |
| DisplayName | Activated Time | | | |
| **Active** | boolean |  |  |  |  |  |
| Description | 1 if the layer is activated, 0 otherwise. | | | |
| DisplayName | Active | | | |
| **AutoActivate** | boolean |  |  |  |  |  |
| Description | 1 if the layer is auto activated 0 otherwise. | | | |
| DisplayName | Auto Activate | | | |
| Key | true | | | |
| **CreatedTime** | datetime |  |  |  |  |  |
| Description | When the layer was created. | | | |
| DisplayName | Created Time | | | |
| **Name** | string(260) |  |  |  |  |  |
| Description | Layer Name. SVS has two layers for each package one read only, other writable. We will report only the writeable layers. | | | |
| DisplayName | Name | | | |
| **PackageId** | string(64) |  |  |  |  |  |
| Description | This typically takes the form of a GUID | | | |
| DisplayName | Package Id | | | |
| Key | true | | | |
| **ResetTime** | datetime |  |  |  |  |  |
| Description | When the layer was last reset. Null if the layer has never been reset. The reset time will be reported in NS date time format, example: '2006-10-17T10:32:06' | | | |
| DisplayName | Reset Time | | | |
| **Type** | uint32 |  |  |  |  |  |
| Description | Type of the virtual software package. 0 = Application 1 = Data | | | |
| DisplayName | Type | | | |

## Inv\_SW\_Virtual\_Software\_Sublayers\_Windows – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **FileRedirectPath** | string(1024) |  |  |  |  |  |
| Description | File system path where the redirect area of the layer is located. | | | |
| DisplayName | File Redirect Path | | | |
| **LayerId** | string(64) |  |  |  |  |  |
| Description | This typically takes the form of a GUID. | | | |
| DisplayName | Layer Id | | | |
| Key | true | | | |
| **PackageId** | string(64) |  |  |  |  |  |
| Description | This typically takes the form of a GUID. | | | |
| DisplayName | Package Id | | | |
| Key | true | | | |
| **RegistryRedirectPath** | string(1024) |  |  |  |  |  |
| Description | Registry path where the redirect area of the layer is located. | | | |
| DisplayName | Registry Redirect Path | | | |
| **Type** | uint32 |  |  |  |  |  |
| Description | Type of the layer.0 = Normal 1 = Peer 2 = Data | | | |
| DisplayName | Type | | | |

# Conclusion

Hopefully this provides a guide that will arm you with the necessary data to manage your reports, filters, or anything else based off of Software and User data. Small updates to data types might have been made, for example an Integer to a Big Integer to allow more flexibility. For the most part these changes will not affect queries against them, but if something is not working as expected, check the data types for the columns you are querying.

This also concludes the series and now all data classes as part of Inventory Solution 7.5 are included (this does not include Server Inventory data classes, which will be covered in a separate article).