Inventory Database Schema 7.5 Part 2 - Hardware

**Inventory Database Schema Part 2**

In Part 2 the Hardware 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 1](#_Toc400699522)

[Database Schema 1](#_Toc400699523)

[Inv \_HW\_LogicalDevice – BASE Class 3](#_Toc400699524)

[Inv \_HW\_Baseboard – SUB Class 4](#_Toc400699525)

[Inv \_HW\_Battery – SUB Class 4](#_Toc400699526)

[Inv \_HW\_Chassis – SUB Class 5](#_Toc400699527)

[Inv \_HW\_ComputerSystem – SUB Class 6](#_Toc400699528)

[Inv \_HW\_DesktopMonitor – SUB Class 6](#_Toc400699529)

[Inv \_HW\_DisplayController – SUB Class 7](#_Toc400699530)

[Inv \_HW\_DisplayController\_Windows – TERTIARY Class 7](#_Toc400699531)

[Inv \_HW\_Keyboard – SUB Class 8](#_Toc400699532)

[Inv \_HW\_PlugAndPlayEntity\_Windows – SUB Class 8](#_Toc400699533)

[Inv \_HW\_NetworkPort – SUB Class 8](#_Toc400699534)

[Inv \_HW\_PhysicalMemory – SUB Class 8](#_Toc400699535)

[Inv \_HW\_PointingDevice – SUB Class 9](#_Toc400699536)

[Inv \_HW\_Printer – SUB Class 10](#_Toc400699537)

[Inv \_HW\_Printer\_Windows – TERTIARY Class 10](#_Toc400699538)

[Inv \_HW\_Processor – SUB Class 10](#_Toc400699539)

[Inv \_HW\_SCSIController – SUB Class 13](#_Toc400699540)

[Inv \_HW\_SoundDevice – SUB Class 13](#_Toc400699541)

[Inv \_HW\_Storage – SUB Class 13](#_Toc400699542)

[Inv \_HW\_USBDevice – SUB Class 14](#_Toc400699543)

[Inv \_HW\_DiskPartition – STANDARD Class 14](#_Toc400699544)

[Inv \_HW\_LogicalDisk – STANDARD Class 15](#_Toc400699545)

[Inv \_HW\_PhysicalMemoryArray – STANDARD Class 16](#_Toc400699546)

[Inv \_HW\_PhysicalMemoryLocation – STANDARD Class, *ASSOCIATION Class* 16](#_Toc400699547)

[Inv \_HW\_StorageToDiskPartition – ASSOCATION Class 17](#_Toc400699548)

[Inv \_HW\_LogicalDiskBasedOnPartition - ASSOCIATION Class 17](#_Toc400699549)

[Conclusion 18](#_Toc400699550)

# 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
* TERTIARY Class – This is a data class that has two dependent data class in the hierarchy
* STANDARD Class – This is a data class that contains no dependencies or sub classes
* ASSOCIATION Class – This data class associates two other data classes’ rows together (not dependencies)

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 or TERTIARY classes subsequently listed 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\_HW\_LogicalDevice
  + Inv\_HW\_Baseboard
  + Inv\_HW\_Battery
  + Inv\_HW\_Chassis
  + Inv \_HW\_ComputerSystem
  + Inv \_HW\_DesktopMonitor
  + Inv \_HW\_DisplayController
    - Inv \_HW\_DisplayController\_Windows
  + Inv \_HW\_Keyboard
  + Inv \_HW\_NetworkPort
  + Inv \_HW\_PhysicalMemory
  + Inv \_HW\_PlugAndPlayEntity\_Windows
  + Inv \_HW\_PointingDevice
  + Inv \_HW\_Printer
    - Inv \_HW\_Printer\_Windows
  + Inv \_HW\_Processor
  + Inv \_HW\_SCSIController
  + Inv \_HW\_SoundDevice
  + Inv \_HW\_Storage
  + Inv \_HW\_USBDevice

## Inv \_HW\_LogicalDevice – BASE Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Description** | string(256) |  |  |  |  |  |
| Description | Textual description of the device. | | | |
| **DeviceClass** | uint16 |  |  |  |  |  |
| Description | The device class. | | | |
| DisplayName | Device Class | | | |
| ValueMap | 0x0000, 0x0001, 0x0100, 0x0101, 0x0102, 0x0103, 0x0104, 0x0105, 0x0106, 0x0107, 0x0180, 0x0200, 0x0201, 0x0202, 0x0203, 0x0204, 0x0280, 0x0300, 0x0301, 0x0302, 0x0380, 0x0400, 0x0401, 0x0402, 0x0403, 0x0480, 0x0500, 0x0501, 0x0580, 0x0600, 0x0601, 0x0602, 0x0603, 0x0604, 0x0605, 0x0606, 0x0607, 0x0608, 0x0609, 0x060a, 0x0680, 0x0700, 0x0701, 0x0702, 0x0703, 0x0780, 0x0800, 0x0801, 0x0802, 0x0803, 0x0804, 0x0880, 0x0900, 0x0901, 0x0902, 0x0903, 0x0904, 0x0980, 0x0a00, 0x0a80, 0x0b00, 0x0b01, 0x0b02, 0x0b10, 0x0b20, 0x0b30, 0x0b40, 0x0c00, 0x0c01, 0x0c02, 0x0c03, 0x0c04, 0x0c05, 0x0c06, 0x0d00, 0x0d01, 0x0d10, 0x0d80, 0x0e00, 0x0f00, 0x0f01, 0x0f03, 0x0f04, 0x1000, 0x1010, 0x1080, 0x1100, 0x1101, 0x1110, 0x1180, 0x1202, 0x1203, 0x1206, 0x1207, 0x1208, 0x1209, 0x120B, 0x120D, 0x120E | | | |
| Values | Non-VGA unclassified device, VGA compatible unclassified device, SCSI storage controller, IDE interface, Floppy disk controller, IPI bus controller, RAID bus controller, ATA controller, SATA controller, Serial Attached SCSI controller, Mass storage controller, Ethernet controller, Token ring network controller, FDDI network controller, ATM network controller, ISDN controller, Network controller, VGA compatible controller, XGA compatible controller, 3D controller, Display controller, Multimedia video controller, Multimedia audio controller, Computer telephony device, Audio device, Multimedia controller, RAM memory, FLASH memory, Memory controller, Host bridge, ISA bridge, EISA bridge, MicroChannel bridge, PCI bridge, PCMCIA bridge, NuBus bridge, CardBus bridge, RACEway bridge, Semi-transparent PCI-to-PCI bridge, InfiniBand to PCI host bridge, Bridge, Serial controller, Parallel controller, Multiport serial controller, Modem, Communication controller, PIC, DMA controller, Timer, RTC, PCI Hot-plug controller, System peripheral, Keyboard controller, Digitizer Pen, Mouse controller, Scanner controller, Gameport controller, Input device controller, Generic Docking Station, Docking Station, 386, 486, Pentium, Alpha, Power PC, MIPS, Co-processor, FireWire (IEEE 1394), ACCESS Bus, SSA, USB Controller, Fibre Channel, SMBus, InfiniBand, IRDA controller, Consumer IR controller, RF controller, Wireless controller, I2O, Satellite TV controller, Satellite audio communication controller, Satellite voice communication controller, Satellite data communication controller, Network and computing encryption device, Entertainment encryption device, Encryption controller, DPIO module, Performance counters, Communication synchronizer, Signal processing controller, Communication device, HID (Human Interface Device), Still Imaging device, Printer, Mass Storage, Hub, Smart Card, Content Security device, Video | | | |
| **DeviceID** | string(256) |  |  |  |  |  |
| Description | Address or other identifying information to uniquely name the logical device. | | | |
| DisplayName | Device ID | | | |
| Key | true | | | |
| **Manufacturer** | string(256) |  |  |  |  |  |
| Description | The name of the organization responsible for producing the device. | | | |
| **Model** | string(256) |  |  |  |  |  |
| Description | The name by which the device is known. | | | |

### Inv \_HW\_Baseboard – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **PartNumber** | string(256) |  |  |  |  |  |
| Description | The part number assigned by the Manufacturer that is responsible for producing or manufacturing the physical element. | | | |
| DisplayName | Part Number | | | |
| **SerialNumber** | string(256) |  |  |  |  |  |
| Description | Manufacturer-allocated number to identify the physical element. | | | |
| DisplayName | Serial Number | | | |
| **Version** | string(256) |  |  |  |  |  |
| Description | Version of the physical element. | | | |

### Inv \_HW\_Battery – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Chemistry** | uint16 |  |  |  |  |  |
| Description | An enumeration that describes the chemistry of the Battery (e.g. 4-Nickel Cadmium, 6-Lithium Ion) | | | |
| ValueMap | 1, 2, 3, 4, 5, 6, 7, 8 | | | |
| Values | Other, Unknown, Lead Acid, Nickel Cadmium, Nickel Metal Hydride, Lithium-ion, Zinc air, Lithium Polymer | | | |
| **DesignCapacity** | uint32 |  |  |  |  |  |
| Description | The design capacity of the battery in mWatt-hours. If this property is not supported, enter 0. | | | |
| DisplayName | Design Capacity (Milli-watt hours) | | | |
| Units | MilliWattHours | | | |
| **DesignVoltage** | uint64 |  |  |  |  |  |
| Description | The design voltage of the battery in mVolts. If this attribute is not supported, enter 0. | | | |
| DisplayName | Design Voltage (Milli-volts) | | | |
| Units | MilliVolts | | | |
| **MaxRechargeTime** | uint32 |  |  |  |  |  |
| Description | MaxRechargeTime indicates the maximum time, in minutes, to fully charge the Battery. This property represents the time to recharge a fully depleted Battery, not the current remaining charging time | | | |
| DisplayName | Max Recharge Time (Minutes) | | | |
| Units | Minutes | | | |

### Inv \_HW\_Chassis – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **AudibleAlarm** | boolean |  |  |  |  |  |
| Description | Boolean value indicating whether the Frame is equipped with an audible alarm. | | | |
| DisplayName | Audible Alarm | | | |
| **ChassisPackageType** | uint16 |  |  |  |  |  |
| Description | ChassisPackageType indicates the physical form factor for the type of Chassis. This property may have a value when the PackageType property contains the value 3 "Chassis Frame". A value of 28 "Blade Enclosure" shall indicate the Chassis is designed to contain one or more PhysicalPackage(s) of PackageType 16 "Blade" or PackageType 17 "Blade Expansion". | | | |
| DisplayName | Chassis Package Type | | | |
| ValueMap | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, .., 0x8000..0xFFFF | | | |
| Values | Unknown, Other, SMBIOS Reserved, Desktop, Low Profile Desktop, Pizza Box, Mini Tower, Tower, Portable, LapTop, Notebook, Hand Held, Docking Station, All in One, Sub Notebook, Space-Saving, Lunch Box, Main System Chassis, Expansion Chassis, SubChassis, Bus Expansion Chassis, Peripheral Chassis, Storage Chassis, SMBIOS Reseved, Sealed-Case PC, SMBIOS Reserved, CompactPCI, AdvancedTCA, Blade Enclosure, DMTF Reserved, Vendor Reserved | | | |
| **LockPresent** | boolean |  |  |  |  |  |
| Description | Boolean indicating whether the frame is protected with a lock. | | | |
| DisplayName | Lock Present | | | |
| **PartNumber** | string(256) |  |  |  |  |  |
| Description | The part number assigned by the organization that is responsible for producing or manufacturing the Physical Element. | | | |
| DisplayName | Part Number | | | |
| **SecurityBreach** | uint16 |  |  |  |  |  |
| Description | SecurityBreach is an enumerated, integer-valued property indicating whether a physical breach of the Frame was attempted but unsuccessful (value=4) or attempted and successful (5). Also, the values, "Unknown", "Other" or "No Breach", can be specified. | | | |
| DisplayName | Security Breach | | | |
| ValueMap | 1, 2, 3, 4, 5 | | | |
| Values | Other, Unknown, No Breach, Breach Attempted, Breach Successful | | | |
| **SecurityStatus** | uint16 |  |  |  |  |  |
| Description | Security setting for external input, for example, a keyboard to this computer. | | | |
| DisplayName | Security Status | | | |
| ValueMap | 1, 2, 3, 4, 5 | | | |
| Values | Other, Unknown, None, External interface locked out, External interface enabled | | | |
| **SerialNumber** | string(256) |  |  |  |  |  |
| Description | Manufacturer-allocated number to identify the physical element. | | | |
| DisplayName | Serial Number | | | |

### Inv \_HW\_ComputerSystem – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **IdentifyingNumber** | string(64) |  |  |  |  |  |
| Description | Product identification such as an onboard serial number. | | | |
| DisplayName | Identifying Number | | | |
| **NumberOfProcessors** | uint32 |  |  |  |  |  |
| Description | Total number of processors. It should be equal to the number of processor instances in class Inv\_HW\_Processor. | | | |
| DisplayName | Number Of Processors | | | |
| **TotalPhysicalMemory** | uint64 |  |  |  |  |  |
| Description | Total size of physical memory in bytes. | | | |
| DisplayName | Total Physical Memory (Bytes) | | | |
| Units | Bytes | | | |

### Inv \_HW\_DesktopMonitor – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **FeatureSupport** | string(256) |  |  |  |  |  |
| Description | This property will report features supported by Monitor such as,Stand-by, Suspend, Active Off , Monochrome , RGB Color , Non-RGB Multicolor, sRGB Supported , Preferred Timing , Default GTF Supported. Property will consist of more than one feature. | | | |
| DisplayName | Feature Support | | | |
| **ManufacturingDate** | string(64) |  |  |  |  |  |
| Description | Manufacturing date of Monitor in month and year. | | | |
| DisplayName | Manufacturing Date | | | |
| **MonitorType** | string |  |  |  |  |  |
| Description | Type of monitor. | | | |
| **ScreenHeight** | uint32 |  |  |  |  |  |
| Description | The maximum logical height of the Display in screen coordinates. | | | |
| DisplayName | Screen Height | | | |
| **ScreenWidth** | uint32 |  |  |  |  |  |
| Description | The maximum logical width of the Display in screen coordinates. | | | |
| DisplayName | Screen Width | | | |
| **SerialNumber** | string(64) |  |  |  |  |  |
| Description | The assigned serial number of Monitor. | | | |
| DisplayName | Serial Number | | | |
| **VideoInputMode** | uint16 |  |  |  |  |  |
| Description | Input mode of Monitor. | | | |
| DisplayName | Video Input Mode | | | |
| ValueMap | 1, 2 | | | |
| Values | Analog, Digital | | | |

### Inv \_HW\_DisplayController – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **AdapterRAM** | uint32 |  |  |  |  |  |
| Description | Memory size of the video adapter. Example: 64000 | | | |
| DisplayName | Adapter RAM (Bytes) | | | |
| Units | Bytes | | | |
| **MaxMemorySupported** | uint32 |  |  |  |  |  |
| Description | Maximum amount of memory supported in bytes. | | | |
| DisplayName | Max Memory Supported (Bytes) | | | |
| Units | Bytes | | | |
| **MaxRefreshRate** | uint32 |  |  |  |  |  |
| Description | Maximum refresh rate of the video controller in hertz. | | | |
| DisplayName | Max Refresh Rate (Hertz) | | | |
| Units | Hertz | | | |
| **VideoProcessor** | string |  |  |  |  |  |
| Description | A free-form string describing the video processor/Controller. Example: RADEON 9250 AGP (0x5960) | | | |
| DisplayName | Video Processor | | | |

#### Inv \_HW\_DisplayController\_Windows – TERTIARY Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **InstalledDisplayDrivers** | string(256) |  |  |  |  |  |
| Description | Name of the installed display device driver. Example: ati2dvag.dll. | | | |
| DisplayName | Installed Display Drivers | | | |

### Inv \_HW\_Keyboard – SUB Class

This table does not contain any relevant data beyond what is held in the Inv\_HW\_Logical\_Device (for reference only).

### Inv \_HW\_PlugAndPlayEntity\_Windows – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **ClassGuid** | string(64) |  |  |  |  |  |
| Description | Globally unique identifier (GUID) of this Plug and Play device. | | | |
| DisplayName | Class Guid | | | |
| **Service** | string(256) |  |  |  |  |  |
| Description | Name of the service that supports this Plug and Play device. | | | |

### Inv \_HW\_NetworkPort – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **AdapterTypeID** | uint16 |  |  |  |  |  |
| Description | Network medium in use. | | | |
| DisplayName | Adapter Type ID | | | |
| ValueMap | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 | | | |
| Values | Ethernet 802.3, Token Ring 802.5, Fiber Distributed Data Interface (FDDI), Wide Area Network (WAN), LocalTalk, Ethernet using DIX header format, ARCNET, ARCNET (878.2), ATM, Wireless, Infrared Wireless, Bpc, CoWan, 1394 | | | |
| **FullDuplex** | boolean |  |  |  |  |  |
| Description | Boolean that indicates whether the port is operating in full duplex mode or not. | | | |
| DisplayName | Full Duplex | | | |
| **MaxSpeed** | uint64 |  |  |  |  |  |
| Description | The maximum bandwidth of the Port in Bits per Second. | | | |
| DisplayName | Max Speed (Bits per Second) | | | |
| Units | Bits per Second | | | |
| **NetworkAddress** | string |  |  |  |  |  |
| Description | Network address for the port. For example, the MAC address of an Ethernet port, World Wide Name (WWN) for Fibre Channel, local address (LID) for InfiniBand, and so on. | | | |
| DisplayName | Network Address | | | |

### Inv \_HW\_PhysicalMemory – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Capacity** | uint64 |  |  |  |  |  |
| Description | Total capacity of the physical memory in bytes. | | | |
| DisplayName | Capacity (Bytes) | | | |
| Units | Bytes | | | |
| **DataWidth** | uint16 |  |  |  |  |  |
| Description | Data width of the physical memory in bits. | | | |
| DisplayName | Data Width (Bits) | | | |
| Units | Bits | | | |
| **FormFactor** | uint16 |  |  |  |  |  |
| Description | The implementation form factor for the Chip. For example, values such as SIMM (7), DIMM (8) can be specified. | | | |
| DisplayName | Form Factor | | | |
| ValueMap | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 | | | |
| Values | Unknown, Other, SIP, DIP, ZIP, SOJ, Proprietary, SIMM, DIMM, TSOP, PGA, RIMM, SODIMM, SRIMM, SMD, SSMP, QFP, TQFP, SOIC, LCC, PLCC, BGA, FPBGA, LGA | | | |
| **MemoryType** | uint16 |  |  |  |  |  |
| Description | Type of physical memory (e.g. 2-DRAM, 3-SDRAM) | | | |
| DisplayName | Memory Type | | | |
| ValueMap | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24..32567, 32568..65535 | | | |
| Values | Unknown, Other, DRAM, Synchronous DRAM, Cache DRAM, EDO, EDRAM, VRAM, SRAM, RAM, ROM, Flash, EEPROM, FEPROM, EPROM, CDRAM, 3DRAM, SDRAM, SGRAM, RDRAM, DDR, DDR-2, BRAM, FB-DIMM, DMTF Reserved, Vendor Reserved | | | |
| **Removable** | boolean |  |  |  |  |  |
| Description | If TRUE, a physical component is removable (if it is designed to be taken in and out of the physical container in which it is normally found) | | | |
| **SerialNumber** | string(256) |  |  |  |  |  |
| Description | Manufacturer-allocated number to identify the physical element. | | | |
| DisplayName | Serial Number | | | |
| **Speed** | uint32 |  |  |  |  |  |
| Description | The speed of the physical memory, in megahertz. | | | |
| DisplayName | Speed (Mega-Hertz) | | | |
| Units | MegaHertz | | | |
| **TotalWidth** | uint16 |  |  |  |  |  |
| Description | Total width, in bits, of the physical memory, including check or error correction bits. If there are no error correction bits, the value in this property should match what is specified for the DataWidth property. | | | |
| DisplayName | Total Width (Bits) | | | |
| Units | Bits | | | |

### Inv \_HW\_PointingDevice – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **DeviceInterface** | uint16 |  |  |  |  |  |
| Description | Type of interface used for the pointing device. | | | |
| DisplayName | Device Interface | | | |
| ValueMap | 1, 2, 3, 4, 5, 6, 7, 8, 160, 161, 162 | | | |
| Values | Other, Unknown, Serial, PS/2, Infrared, HP-HIL, Bus mouse, ADB (Apple Desktop Bus), Bus mouse DB-9, Bus mouse micro-DIN, USB | | | |
| **NumberOfButtons** | uint8 |  |  |  |  |  |
| Description | Number of buttons. If the Pointing Device has no buttons, enter 0. | | | |
| DisplayName | Number Of Buttons | | | |
| **PointingType** | uint16 |  |  |  |  |  |
| Description | Type of pointing device (e.g. 3=Mouse, 7=Touch Pad, 9=Mouse - Optical Sensor). | | | |
| DisplayName | Pointing Type | | | |
| ValueMap | 1, 2, 3, 4, 5, 6, 7, 8, 9 | | | |
| Values | Other, Unknown, Mouse, Track Ball, Track Point, Glide Point, Touch Pad, Touch Screen, Mouse - Optical Sensor | | | |

### Inv \_HW\_Printer – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **DefaultPaperType** | string |  |  |  |  |  |
| Description | Specifies the paper type that the Printer will use by default. The values of the property SHOULD conform to the requirements of the PWG Media Standardized Names specification [PWG5101.1], which defines the normative values for this property. | | | |
| DisplayName | Default Paper Type | | | |
| **HorizontalResolution** | uint32 |  |  |  |  |  |
| Description | The maximum resolution of the Printer in Pixels per Inch in the cross-feed direction, i.e., short-edge in portrait feed mode. | | | |
| DisplayName | HorizontalResolution (Pixels per Inch) | | | |
| Units | Pixels per Inch | | | |
| **Network** | boolean |  |  |  |  |  |
| Description | The Network property indicates whether the printer is a network printer. | | | |
| **VerticalResolution** | uint32 |  |  |  |  |  |
| Description | The maximum resolution of the Printer in Pixels per Inch in the feed direction, i.e., long-edge in portrait feed mode. | | | |
| DisplayName | Vertical Resolution (Pixels per Inch) | | | |
| Units | Pixels per Inch | | | |

#### Inv \_HW\_Printer\_Windows – TERTIARY Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Default** | boolean |  |  |  |  |  |
| Description | If TRUE, the printer is the default printer. | | | |
| **Location** | string |  |  |  |  |  |
| Description | Physical location of the printer. Example: Bldg. 38, Room 1164 (this must be programmatically obtainable) | | | |
| **PortName** | string |  |  |  |  |  |
| Description | Port that is used to transmit data to a printer. If a printer is connected to more than one port, the names of each port are separated by commas. Example: LPT1:, LPT2:, LPT3: | | | |
| DisplayName | Port Name | | | |
| **ServerName** | string |  |  |  |  |  |
| Description | Name of the server that controls the printer. If this string is NULL, the printer is controlled locally. | | | |
| DisplayName | Server Name | | | |
| **ShareName** | string |  |  |  |  |  |
| Description | Share name of the Windows printing device. Example: \\PRINTSERVER1\PRINTER2. | | | |
| DisplayName | Share Name | | | |

### Inv \_HW\_Processor – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **AddressWidth** | uint16 |  |  |  |  |  |
| Description | Processor address width in bits. | | | |
| DisplayName | Address Width (Bits) | | | |
| Units | Bits | | | |
| **Architecture** | uint16 |  |  |  |  |  |
| Description | Processor architecture that the platform uses (e.g x86, MIPS, Alpha, PowerPC, Itanium, x64). | | | |
| ValueMap | 0, 1, 2, 3, 6, 9, 10, 11 | | | |
| Values | x86, MIPS, Alpha, PowerPC, ia64, x64, SPARC, PA-RISC | | | |
| **DataWidth** | uint16 |  |  |  |  |  |
| Description | Processor data width in bits. | | | |
| DisplayName | Data Width (Bits) | | | |
| Units | Bits | | | |
| **ExternalClockSpeed** | uint32 |  |  |  |  |  |
| Description | External clock frequency in MHz. | | | |
| DisplayName | External Clock Speed (MegaHertz) | | | |
| Units | MegaHertz | | | |
| **Family** | uint16 |  |  |  |  |  |
| Description | The Processor family type. For example, values include "Pentium(R) processor with MMX(TM) technology" (value=14) and "68040" (value=96). | | | |
| ValueMap | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 48, 49, 50, 51, 52, 53, 54, 55, 64, 65, 66, 67, 68, 69, 80, 81, 82, 83, 84, 85, 86, 87, 88, 96, 97, 98, 99, 100, 101, 112, 120, 121, 122, 128, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 160, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 200, 201, 202, 203, 204, 210, 211, 212, 213, 230, 250, 251, 260, 261, 280, 281, 300, 301, 302, 320, 350, 500 | | | |
| Values | Other, Unknown, 8086, 80286, 80386, 80486, 8087, 80287, 80387, 80487, Pentium(R) brand, Pentium(R) Pro, Pentium(R) II, Pentium(R) processor with MMX(TM) technology, Celeron(TM), Pentium(R) II Xeon(TM), Pentium(R) III, M1 Family, M2 Family, K5 Family, K6 Family, K6-2, K6-3, AMD Athlon(TM) Processor Family, AMD(R) Duron(TM) Processor, AMD29000 Family, K6-2+, Power PC Family, Power PC 601, Power PC 603, Power PC 603+, Power PC 604, Power PC 620, Power PC X704, Power PC 750, PowerPC 7400 (G4), PowerPC 970 (G5), Alpha Family, Alpha 21064, Alpha 21066, Alpha 21164, Alpha 21164PC, Alpha 21164a, Alpha 21264, Alpha 21364, MIPS Family, MIPS R4000, MIPS R4200, MIPS R4400, MIPS R4600, MIPS R10000, SPARC Family, SuperSPARC, microSPARC II, microSPARC IIep, UltraSPARC, UltraSPARC II, UltraSPARC IIi, UltraSPARC III, UltraSPARC IIIi, 68040, 68xxx Family, 68000, 68010, 68020, 68030, Hobbit Family, Crusoe(TM) TM5000 Family, Crusoe(TM) TM3000 Family, Efficeon(TM) TM8000 Family, Weitek, Itanium(TM) Processor, AMD Athlon(TM) 64 Processor Family, AMD Opteron(TM) Processor Family, AMD Sempron(TM) Processor Family, AMD Turion(TM) 64 Mobile Technology, Dual-Core AMD Opteron(TM) Processor Family, AMD Athlon(TM) 64 X2 Dual-Core Processor Family, AMD Turion(TM) 64 X2 Mobile Technology, Quad-Core AMD Opteron(TM) Processor Family, Third-Generation AMD Opteron(TM) Processor Family, AMD Phenom(TM) FX Quad-Core Processor Family, AMD Phenom(TM) X4 Quad-Core Processor Family, AMD Phenom(TM) X2 Dual-Core Processor Family, AMD Athlon(TM) X2 Dual-Core Processor Family, PA-RISC Family, PA-RISC 8500, PA-RISC 8000, PA-RISC 7300LC, PA-RISC 7200, PA-RISC 7100LC, PA-RISC 7100, V30 Family, Pentium(R) III Xeon(TM), Pentium(R) III Processor with Intel(R) SpeedStep(TM) Technology, Pentium(R) 4, Intel(R) Xeon(TM), AS400 Family, Intel(R) Xeon(TM) processor MP, AMD Athlon(TM) XP Family, AMD Athlon(TM) MP Family, Intel(R) Itanium(R) 2, Intel(R) Pentium(R) M processor, Intel(R) Celeron(R) D processor, Intel(R) Pentium(R) D processor, Intel(R) Pentium(R) Processor Extreme Edition, Intel(R) Core(TM) Solo Processor, K7, Intel(R) Core(TM)2 Duo Processor, Intel(R) Core(TM)2 Solo Processor, Intel(R) Core(TM) Duo Processor, S/390 and zSeries Family, ESA/390 G4, ESA/390 G5, ESA/390 G6, z/Architectur base, VIA C7(TM)-M Processor Family, VIA C7(TM)-D Processor Family, VIA C7(TM) Processor Family, VIA Eden(TM) Processor Family, Embedded AMD Opteron(TM) Quad-Core Processor Family, i860, i960, SH-3, SH-4, ARM, StrongARM, 6x86, MediaGX, MII, WinChip, DSP, Video Processor | | | |
| **MaxClockSpeed** | uint32 |  |  |  |  |  |
| Description | The maximum speed (in MHz) of this Processor. | | | |
| DisplayName | Max Clock Speed (MegaHertz) | | | |
| Units | MegaHertz | | | |
| **NumberOfCores** | uint32 |  |  |  |  |  |
| Description | Number of cores of the physical processor. | | | |
| DisplayName | Number Of Cores | | | |
| **NumberOfLogicalProcessors** | uint32 |  |  |  |  |  |
| Description | Number of logical cores of the physical processor. | | | |
| DisplayName | Number Of Logical Processors | | | |
| **OtherFamilyDescription** | string(64) |  |  |  |  |  |
| Description | A string that describes the Processor Family type. It is used when the Family property is set to 1 ("Other"). This string should be set to NULL when the Family property is any value other than 1. | | | |
| DisplayName | Other Family Description | | | |
| **Version** | string(64) |  |  |  |  |  |
| Description | An architecture-dependent processor revision number. Example: Model 2, Stepping 12. | | | |

### Inv \_HW\_SCSIController – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **HardwareVersion** | string(64) |  |  |  |  |  |
| Description | Hardware version number of the SCSI controller. Example: 1.25 | | | |
| DisplayName | Hardware Version | | | |
| **Index** | uint32 |  |  |  |  |  |
| Description | Index number of the SCSI controller in the system registry. Example: 0 | | | |
| **MaxDataWidth** | uint32 |  |  |  |  |  |
| Description | Maximum data width (in bits) supported by the SCSI controller. | | | |
| DisplayName | Max Data Width (Bits) | | | |
| Units | Bits | | | |
| **MaxTransferRate** | uint64 |  |  |  |  |  |
| Description | Maximum data width (in bits) supported by the SCSI controller. | | | |
| DisplayName | Max Transfer Rate (Bits Per Second) | | | |
| Units | Bits Per Second | | | |
| **ProtocolSupported** | uint16 |  |  |  |  |  |
| Description | The protocol used by the Controller to access controlled Devices (e.g. 5-PCI, 6-ATAPI). | | | |
| DisplayName | Protocol Supported | | | |
| ValueMap | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 | | | |
| Values | Other, Unknown, EISA, ISA, PCI, ATA/ATAPI, Flexible Diskette, 1496, SCSI Parallel Interface, SCSI Fibre Channel Protocol, SCSI Serial Bus Protocol, SCSI Serial Bus Protocol-2 (1394), SCSI Serial Storage Architecture, VESA, PCMCIA, Universal Serial Bus, Parallel Protocol, ESCON, Diagnostic, I2C, Power, HIPPI, MultiBus, VME, IPI, IEEE-488, RS232, IEEE 802.3 10BASE5, IEEE 802.3 10BASE2, IEEE 802.3 1BASE5, IEEE 802.3 10BROAD36, IEEE 802.3 100BASEVG, IEEE 802.5 Token-Ring, ANSI X3T9.5 FDDI, MCA, ESDI, IDE, CMD, ST506, DSSI, QIC2, Enhanced ATA/IDE, AGP, TWIRP (two-way infrared), FIR (fast infrared), SIR (serial infrared), IrBus, Serial ATA | | | |

### Inv \_HW\_SoundDevice – SUB Class

This table does not contain any relevant data beyond what is held in the Inv \_HW\_Logical\_Device (for reference only).

### Inv \_HW\_Storage – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **InterfaceType** | string |  |  |  |  |  |
| Description | The InterfaceType property indicates the storage interface type. Example: SCSI | | | |
| DisplayName | Interface Type | | | |
| **MaxMediaSize** | uint64 |  |  |  |  |  |
| Description | Maximum size, in KBytes, of media supported by this Device. KBytes is interpreted as the number of bytes multiplied by 1000 (NOT the number of bytes multiplied by 1024). | | | |
| DisplayName | Max Media Size (KiloBytes) | | | |
| Units | KiloBytes | | | |
| **MediaType** | uint16 |  |  |  |  |  |
| Description | This will identify the different Storage Media Types. Values include - Floppy, Tape, DVD, CD, Disk, Optical, Other. | | | |
| DisplayName | Media Type | | | |
| ValueMap | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66 | | | |
| Values | Unknown, Other, Tape Cartridge, QIC Cartridge, AIT Cartridge, DTF Cartridge, DAT Cartridge, 8mm Tape Cartridge, 19mm Tape Cartridge, DLT Cartridge, Half-Inch Magnetic Tape Cartridge, Cartridge Disk, JAZ Disk, ZIP Disk, SyQuest Disk, Winchester Removable Disk, CD-ROM, CD-ROM/XA, CD-I, CD Recordable, WORM, Magneto-Optical, DVD, DVD-RW+, DVD-RAM, DVD-ROM, DVD-Video, Divx, Floppy/Diskette, Hard Disk, Memory Card, Hard Copy, Clik Disk, CD-RW, CD-DA, CD+, DVD Recordable, DVD-RW, DVD-Audio, DVD-5, DVD-9, DVD-10, DVD-18, Magneto-Optical Rewriteable, Magneto-Optical Write Once, Magneto-Optical Rewriteable (LIMDOW), Phase Change Write Once, Phase Change Rewriteable, Phase Change Dual Rewriteable, Ablative Write Once, Near Field Recording, MiniQic, Travan, 8mm Metal Particle, 8mm Advanced Metal Evaporate, NCTP, LTO Ultrium, LTO Accelis, 9 Track Tape, 18 Track Tape, 36 Track Tape, Magstar 3590, Magstar MP, D2 Tape, Tape - DST Small, Tape - DST Medium, Tape - DST Large | | | |
| **SerialNumber** | string(256) |  |  |  |  |  |
| Description | Manufacturer-allocated number to identify the physical element. | | | |
| DisplayName | Serial Number | | | |

### Inv \_HW\_USBDevice – SUB Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **DeviceSpeed** | uint16 |  |  |  |  |  |
| Description | Data rate speed of the USB device. | | | |
| DisplayName | Device Speed | | | |
| ValueMap | 0, 1, 2, 3, 4, 5 | | | |
| Values | Unknown, Other, Low Speed, Full Speed, Hi-Speed, Super-Speed | | | |
| **SerialNumber** | string(256) |  |  |  |  |  |
| Description | Serial Number of the USB device. | | | |
| DisplayName | Serial Number | | | |
| **USBVersion** | string(64) |  |  |  |  |  |
| Description | Indicates the latest USB Version supported by the USB Device. | | | |
| DisplayName | USB Version | | | |

## Inv \_HW\_DiskPartition – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **BlockSize** | uint64 |  |  |  |  |  |
| Description | Size in bytes of the blocks which form this partition. If variable block size, then the maximum block size in bytes should be specified. If the block size is unknown or if a block concept is not valid (for example, for AggregateExtents, Memory or LogicalDisks), enter a 1. | | | |
| DisplayName | Block Size (Bytes) | | | |
| Units | Bytes | | | |
| **Bootable** | boolean |  |  |  |  |  |
| Description | Boolean indicating that the Partition is labeled as bootable. (Note that this does not mean that an Operating System is actually loaded on the Partition.) | | | |
| **DeviceID** | string(256) |  |  |  |  |  |
| Description | An address or other identifying information used to uniquely identify the LogicalDevice. | | | |
| DisplayName | Device ID | | | |
| Key | true | | | |
| **Name** | string(1024) |  |  |  |  |  |
| Description | The Name property defines the label by which the object is known. | | | |
| **NumberOfBlocks** | uint64 |  |  |  |  |  |
| Description | Total number of logically contiguous blocks, of size BlockSize, which form this Extent. The total size of the Extent can be calculated by multiplying BlockSize by NumberOfBlocks. If the BlockSize is 1, this property is the total size of the Extent. | | | |
| DisplayName | Number Of Blocks | | | |
| **PrimaryPartition** | boolean |  |  |  |  |  |
| Description | Boolean indicating that the DiskPartition is labelled as the primary partition for a ComputerSystem. | | | |
| DisplayName | Primary Partition | | | |

## Inv \_HW\_LogicalDisk – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **Compressed** | boolean |  |  |  |  |  |
| Description | The Compressed property indicates whether the logical volume exists as a single compressed entity, such as a DoubleSpace volume. | | | |
| **Description** | string |  |  |  |  |  |
| Description | The Description property provides a textual description of the object. | | | |
| **DeviceID** | string(256) |  |  |  |  |  |
| Description | An address or other identifying information used to uniquely identify the logical disk. | | | |
| DisplayName | Device ID | | | |
| Key | true | | | |
| **FileSystem** | string(64) |  |  |  |  |  |
| Description | The FileSystem property indicates the file system on the logical disk. Example: NTFS. | | | |
| DisplayName | File System | | | |
| **FreeSpace** | uint64 |  |  |  |  |  |
| Description | The FreeSpace property indicates in bytes how much free space is available on the logical disk. | | | |
| DisplayName | Free Space (Bytes) | | | |
| Units | Bytes | | | |
| **LogicalDiskType** | uint16 |  |  |  |  |  |
| Description | Provides the type of logical disk that this class represents (e.g. 2= Removable, 3=Local, 4=Network, 5=CD, 6=RAM) This is the same as the Win32\_LogicalDisk.DriveType. | | | |
| DisplayName | Logical Disk Type | | | |
| ValueMap | 0, 1, 2, 3, 4, 5, 6 | | | |
| Values | Unknown, No Root Directory, Removable Disk, Local Disk, Network Drive, Compact Disc, RAM Disk | | | |
| **MountPoint** | string(1024) |  |  |  |  |  |
| Description | Mounted path to the logical disk. For Windows this is the association between a Win32\_Volume and a Win32\_Directory using a Win32\_MountPoint class. If more than one MountPoint exists, use a semi-colon delimiter to include each point. | | | |
| DisplayName | Mount Point | | | |
| **Name** | string(1024) |  |  |  |  |  |
| Description | The Name property defines the label by which the object is known. | | | |
| **ProviderName** | string(256) |  |  |  |  |  |
| Description | Network path to the logical device. | | | |
| DisplayName | Provider Name | | | |
| **Size** | uint64 |  |  |  |  |  |
| Description | The Size property indicates in bytes, the size of the logical disk. If unknown, enter 0. | | | |
| DisplayName | Size (Bytes) | | | |
| Units | Bytes | | | |

## Inv \_HW\_PhysicalMemoryArray – STANDARD Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **MaxCapacity** | uint32 |  |  |  |  |  |
| Description | Maximum memory size (in Kilobytes) installable for this particular memory array. If the size is unknown, the property is given a value of 0 (zero). | | | |
| DisplayName | Max Capacity (Kilobytes) | | | |
| Units | Kilobytes | | | |
| **MemoryDevices** | uint16 |  |  |  |  |  |
| Description | Number of physical slots or sockets available in this memory array. | | | |
| DisplayName | Memory Devices | | | |
| **Tag** | string(256) |  |  |  |  |  |
| Description | An arbitrary string that uniquely identifies the Physical Element and serves as the key of the Element. The Tag property can contain information such as asset tag or serial number data. | | | |
| Key | true | | | |
| **Use** | uint16 |  |  |  |  |  |
| Description | The Use property indicates how the memory is used in the computer system. | | | |
| ValueMap | 0, 1, 2, 3, 4, 5, 6, 7 | | | |
| Values | Reserved, Other, Unknown, System memory, Video memory, Flash memory, Non-volatile RAM, Cache memory | | | |

## Inv \_HW\_PhysicalMemoryLocation – STANDARD Class, ASSOCIATION Class

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Qualifiers** | | | | |
| **Name** | **Value** | | | |
| **LocationWithinContainer** | string(64) |  |  |  |  |  |
| Description | Position of the physical element within the physical package. Example: BANK 1/DIMM1 | | | |
| DisplayName | Location With in Container | | | |

**ASSOCIATIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Class Origin** | **Reference Class** | **Description** |
| **GroupComponent** | Inv\_HW\_PhysicalMemoryLocation | [Inv\_HW\_PhysicalMemoryArray](http://mydevelopment/Products/Document%20Library/Inventory%20Solution/Inventory%20Solution%20Data%20classes/dataclasses/Altiris_HW_PhysicalMemoryArray.html) | Reference to the Inv\_HW\_PhysicalMemoryArray instance that represents the physical memory array that contains the physical memory. |
| **PartComponent** | Inv\_HW\_PhysicalMemoryLocation | [Inv\_HW\_PhysicalMemory](http://mydevelopment/Products/Document%20Library/Inventory%20Solution/Inventory%20Solution%20Data%20classes/dataclasses/Altiris_HW_PhysicalMemory.html) | Reference to the Inv\_HW\_PhysicalMemory instance that represents the physical memory contained in the physical memory array. |

#### Inv \_HW\_StorageToDiskPartition – ASSOCATION Class

**ASSOCIATIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Class Origin** | **Reference Class** | **Description** |
| **Antecedent** | Inv\_HW\_StorageToDiskPartition | [Inv\_HW\_Storage](http://mydevelopment/Products/Document%20Library/Inventory%20Solution/Inventory%20Solution%20Data%20classes/dataclasses/Altiris_HW_Storage.html) | The Antecedent reference represents the Inv\_HW\_Storage containing the properties of the storage where the partition exists. |
| **Dependent** | Inv\_HW\_StorageToDiskPartition | [Inv\_HW\_DiskPartition](http://mydevelopment/Products/Document%20Library/Inventory%20Solution/Inventory%20Solution%20Data%20classes/dataclasses/Altiris_HW_DiskPartition.html) | The Dependent reference represents the Inv\_HW\_DiskPartition containing the properties of a disk partition residing on the storage. |

#### Inv \_HW\_LogicalDiskBasedOnPartition - ASSOCIATION Class

**ASSOCIATIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Class Origin** | **Reference Class** | **Description** |
| **Antecedent** | Inv\_HW\_LogicalDiskBasedOnPartition | [Inv\_HW\_DiskPartition](http://mydevelopment/Products/Document%20Library/Inventory%20Solution/Inventory%20Solution%20Data%20classes/dataclasses/Altiris_HW_DiskPartition.html) | The Antecedent reference represents the Inv\_HW\_DiskPartition instance containing the properties of the disk partition the logical disk resides. |
| **Dependent** | Inv\_HW\_LogicalDiskBasedOnPartition | [Inv\_HW\_LogicalDisk](http://mydevelopment/Products/Document%20Library/Inventory%20Solution/Inventory%20Solution%20Data%20classes/dataclasses/Altiris_HW_LogicalDisk.html) | The Dependent reference represents the Inv\_HW\_LogicalDisk containing the properties of a logical disk that resides on a physical disk partition. |

# 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 Detailed Hardware 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.