

Miradore Management Suite Web Service

API Specification released with Miradore Management Suite 5.3.0

Recent Changes

API specification's version number was tied to the version of Miradore Management Suite in version 4.4.0.

Version	Comments
5.3.0	<p>Added support for reading the <i>MediaFolder</i> attribute from the Package item.</p> <p>Added support for reading <i>InitialInstallationEnabled</i> attribute from the OS item.</p> <p>Added new Asset extensions EndpointBackupJobConfiguration and EndpointBackupReport. With the information, it is for example possible to create reports for showing the history of endpoint backup jobs performed on a device or to find devices which are not backed up.</p> <p>Added new Asset extension InvLastLoggedInUser for reading the username of the user who has last logged into the device according to the inventory data gathered from the device.</p> <p>Example query for testing: <a href="http://<hostname>/<instancename>/connectors/rest.svc/Asset?select=Asset[AssetID,InvLastLoggedInUser]">http://<hostname>/<instancename>/connectors/rest.svc/Asset?select=Asset[AssetID,InvLastLoggedInUser]</p> <p>Added new Asset extension InvSecurityProduct for reading Windows-provided information about the device's security products such as firewall and antivirus solutions. Devices with Microsoft server operating system are not supported. Windows desktop versions are supported starting from Windows 7.</p> <p>Added SystemTask item and SystemTaskResult extension for reading data about Miradore Management Suite's system tasks. Data in these items can be used for example to monitor the execution of the system tasks based on their latest run results.</p> <p>Added a new item AssetSpecificPackage which can be used to read, write, update or delete asset-specific packages. When modifying the priorities of asset-specific packages through API, make sure that two packages do not have equal priorities.</p> <p>Example query for reading asset-specific packages (GET method): <a href="http://<hostname>/<instancename>/connectors/rest.svc/AssetSpecificPackage?select=AssetSpecificPackage[*]&options=type=embedded,debug&filters=AssetSpecificPackage.AssetID eq 10">http://<hostname>/<instancename>/connectors/rest.svc/AssetSpecificPackage?select=AssetSpecificPackage[*]&options=type=embedded,debug&filters=AssetSpecificPackage.AssetID eq 10</p> <p>Example query for adding an asset-specific package (POST method): <a href="http://<hostname>/<instancename>/connectors/rest.svc/AssetSpecificPackage">http://<hostname>/<instancename>/connectors/rest.svc/AssetSpecificPackage</p> <p>Message body:</p> <pre> <Request> <Items> <AssetSpecificPackage> <Asset> <DeviceName>L0000500</DeviceName> </Asset> <Licensee>Jim Straw</Licensee> <LicenseKey>LIC_123ABCaaaa</LicenseKey> <Package> <PackageID>244</PackageID> </Package> <Priority>7</Priority> <Reboot>False</Reboot> </AssetSpecificPackage> </Items> </Request> </pre>

Contents

1.	Introduction to Miradore web service	4
2.	URL and query string	5
2.1	Options	6
2.2	Select	13
2.3	Filters	15
2.4	OrderBy.....	18
2.5	ChildFilters	18
2.6	Aggregate.....	19
2.7	GroupBy.....	19
3.	Authentication and security	21
4.	Available operations.....	22
4.1	Get	22
4.2	Create	23
4.3	Update	25
4.4	Remove.....	26
4.5	Execute	27
5.	XML payload structure	30
5.1	Root elements	30
5.2	Child elements.....	31
6.	Testing the web service manually	34
	Appendix 1: Supported items and operations.....	35
	Appendix 2: Supported attributes per item	37
	AgentStatus	37
	Asset	38
	Asset extensions.....	40
	GroupDistribution.....	63
	HistorySeries extensions	64
	Location extensions.....	70

1. Introduction to Miradore web service

Miradore web service is an interface intended for integrating Miradore with external systems. This document describes the programming interface of the web service and includes lots of examples. The document is aimed for technical persons implementing the interface.

Miradore web service is a REST based web service which can be used to programmatically create, read, update and remove configuration items in Miradore CMDB, and also to execute certain actions for asset configuration items. It is used over HTTP(S) with POST, GET, PUT and DELETE methods. HTTP Basic authentication and Miradore user groups are used to authenticate web service requests. The web service uses XML as its payload. REST highly utilizes HTTP request URL in its operations.

Chapter [2 URL and query string](#) describes how URLs are formed and what additional keywords can be used to modify queries.

Chapter [3 Authentication and security](#) contains information about how web service calls can be authenticated.

Chapter [4 Available operations](#) lists available operations and their usage.

Chapter [5 XML payload structure](#) describes the structure of input and output XML documents used by the web service.

Chapter [6 Testing the web service manually](#) speaks about API testing and introduces some useful tools.

Appendices 1 and 2 contain lists of available configuration items and their attributes that can be accessed by the web service interface.

2. URL and query string

Miradore web service is used by sending HTTP requests to Miradore server. Depending on the operation, data is transferred in URL or the body of HTTP request. *Get*, *remove*, and *execute* operations transfer all required information in the HTTP request URL. *Create* and *update* operations require also some data in the HTTP request body. This chapter describes the structure of the web service HTTP request URL.

Form of the web service URL depends on the used operation. Base URL for all operations is:

`http://<server>/<instance>/connectors/rest.svc/<Item>`

[Table 1](#) lists the available operations and corresponding HTTP methods. For more information about operations, see [4 Available operations](#).

Table 1: HTTP methods used in different operations

Operation	Method	URL
Get	GET	<code>/connectors/rest.svc/<Item></code>
		<code>/connectors/rest.svc/<Item>/<ID></code>
Create	POST	<code>/connectors/rest.svc/<Item></code>
Update	PUT	<code>/connectors/rest.svc/<Item>/<ID></code>
Remove	DELETE	<code>/connectors/rest.svc/<Item>/<ID></code>
		<code>/connectors/rest.svc/<Item>/<ID>/<Attachment></code>
		<code>/connectors/rest.svc/<Item>/<ID>/<Attachment>/<AttachmentID></code>
Execute	POST	<code>/connectors/rest.svc/Asset/<ID>/<Action name>/<Optional parameter></code>

The URL may contain an optional query string which contains specific settings to the query. Available settings depend on the operation. [Table 2](#) lists the available query string settings and the operations that support them. Query string settings are explained more comprehensively in the following chapters.

Table 2: Available query string setting (G=get, C=create, U=update, R=remove and E=execute)

Setting	G	C	U	R	E	Description
<i>options</i>	x	x	x	x	x	Can be used to define output and input formatting.
<i>select</i>	x	x	x			Defines what item attributes are selected to response.
<i>filters</i>	x					Defines item attribute based filters used to filter the items that query concerns.
<i>orderby</i>	x					Define the order in which the configuration items are listed within the response.
<i>childFilters</i>	x					Defines item attribute based filters used to filter the child items which will be returned by the query.
<i>groupby</i>	x					Defines how the configuration items are grouped within the response.

Query string is formatted just like any standard URL query string containing key-value pairs.

Example 2-1: Web service request with multiple keywords in query string

GET:

`connectors/rest.svc/asset?select=Asset[AssetTag,DeviceName,ResponsiblePerson],User[FirstName,LastName]&filters=Asset.Location.Name eq Helsinki&options=type=embedded,rows=500`

This request contains *select*, *filters* and *options* keywords. *Options* defines two options: *type* and *rows*.

2.1 Options

Miradore web service supports several options that can be used to modify query handling and response. [Table 3](#) lists the available options and the operations that support them. Detailed descriptions of the options can be found after the table.

Some of the options are on/off flags and some require value. The syntax for the options is the following:

/Item?options=option1,option2=option2value,option3=option3value

Table 3: Available query options (G=get, C=create, U=update, R=remove and E=execute)

Option	G	C	U	R	E	Description	Default
<i>addChildType</i>	x	x	x			Adds an item attribute, which defines the type of the child item, for each child item element to response XML.	Not set
<i>debug</i>	x	x	x	x	x	Adds debug information to response.	Not set
<i>df</i>	x	x	x			Defines the DateTime format string for input and output dates.	dd.MM.yyyy HH:mm:ss
<i>getCount</i>	x					If set, response contains only count of items that query would return.	Not set
<i>page</i>	x					Page number of the page which will be returned.	1
<i>rows</i>	x					Number of rows (root level items) on one page.	100
<i>Showother</i>	x					Adds "[OTHER]" group to the result. This option can be only used with the GroupBy query setting and it requires that rows option is used to limit the results.	
<i>suppressMultiChildError</i>		x	x			If set, suppresses the error which is returned when multiple child items match to the defined attributes.	Not set
<i>useNamespace</i>	x	x	x			If set, adds an XML namespace declaration to response documents.	Not set
<i>type</i>	x					Format of the response XML document.	get: basic create, update: embedded

AddChildType

AddChildType option adds type of the child item to the response.

- **Default value:** -
- **Usage:** options=addchildtype

Example 2-2: Output when addChildType option is set

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Basic" ExportDate="26.11.2012 15:41:15">
  <Items item="Asset" count="1">
    <Asset id="1190">
      <AssetTag>VM0001190</AssetTag>
      <Location id="8" item="Location">Helsinki</Location>
      <Model id="21" item="Model">Microsoft Virtual Machine </Model>
      <OS id="7" item="OS">Microsoft Windows 7 Enterprise x64 English</OS>
      <ResponsiblePerson id="2" item="User">Doe John</ResponsiblePerson>
    </Asset>
  </Items>
</Response>
```

Debug

Debug option adds an additional Debug section to the response document. It is aimed for troubleshooting performance issues.

- **Default value:** -
- **Usage:** options=debug

Example 2-3: Output when debug option is set

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Referred" ExportDate="26.11.2012 15:41:15">
  <!-- Items ... -->
  <Debug>
    <QueryTime>151ms</QueryTime>
    <XmlBuildTime>10ms</XmlBuildTime>
    <TotalTime>181ms</TotalTime>
    <Queries>9</Queries>
  </Debug>
</Response>
```

Df

Df (date format) option defines the representation of incoming and outgoing date-time values. With this option the web service can be used with a date format familiar for caller end. *Df* supports .Net date format strings. Refer to Microsoft's documentation for more information about available configurations. For example, information about custom date and time format strings can be found from

<http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx>.

- **Default value:** dd.MM.yyyy HH:mm:ss
- **Usage:** options=df=<.Net DateTime format string>

Example 2-4: Output when option df=MM/dd/yyyy is set

```
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Referred" ExportDate="11/27/2012">
  <Items item="Asset" count="1">
    <Asset root="1" id="1190">
      <AssetTag>DCM0001232</AssetTag>
      <ModifyDate>12/02/2011</ModifyDate>
      <WarrantyEnd>09/07/2014</WarrantyEnd>
      <WarrantyStart>09/07/2011</WarrantyStart>
    </Asset>
  </Items>
</Response>
```

GetCount

GetCount returns only the number of items in total that match the query and filters. This can be used to reduce network traffic if only count of matching items is needed.

- **Default value:** -
- **Usage:** options=getcount

Example 2-5: Usage of getCount option

Get operation is used with the following query string: *filters=Asset.Location.Name eq Helsinki&options=getcount*. Response contains only number of the matching items:

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Basic" ExportDate="27.11.2012 10:23:17">
  <Items item="Asset" count="103" />
</Response>
```

Paging

Paging can be used to limit the number of items returned by the query. If the query returns lots of items, it might be useful to improve query performance by processing data in smaller pieces. Paging can be set with the following options:

- **Rows:** Defines the number of items returned per page (default: 100)
- **Page:** Page number (default: 1)

Example 2-6: Usage of paging with rows and page options

For example, if your query returns 550 items, you can request the data in three consecutive queries:

- **Query #1:** options=rows=250,page=1
- **Output:** Items from 1 to 250

- **Query #2:** options=rows=250,page=2
- **Output:** Items from 251 to 500

- **Query #3:** options=rows=250,page=3
- **Output:** Items from 501 to 550

showother

Adds “[OTHER]” group to the result. This option can be only used with the [groupby](#) query setting and it requires that *rows* option is used to limit the results.

Example 2-7: Usage of showother

```
Asset?groupby=[Location],aggregate[Count as Assets desc]&options=rows=3,showother
<?xml version="1.0" encoding="utf-8" ?>
<Response MDVersion="4.5.0" Type="GroupBy" Time="27.04.2017 09:35:33" StatusCode="200">
<Items item="Asset">
<Items item="Asset" count="4">
<Item>
<Location>Skinnarila</Location>
<Assets>195</Assets>
</Item>
<Item>
<Location>Lappeenranta</Location>
<Assets>85</Assets>
</Item>
<Item>
<Location>Erottajankatu</Location>
<Assets>60</Assets>
</Item>
<Item>
<Location>[OTHER]</Location>
<Assets>169</Assets>
</Item>
</Items>
</Items>
</Response>
```

Interpretation: There are 195 assets in Skinnarila, 85 in Lappeenranta, 60 in Erottajankatu and 169 in other locations.

SuppressMultiChildError

Create and *update* operations query child items with attributes defined in the request XML document. If these attributes match to multiple child items, an error is returned by default. *SuppressMultiChildError* option can be used to suppress this error. In this case the first child item returned by the query is selected. Beware though, that with this option, an error in the integrating implementation may go unnoticed.

- **Default value:** -
- **Usage:** options= suppressMultiChildError

Example 2-8: Usage of suppressMultiChildError option

For example, *update* operation is used to update asset's *ResponsiblePerson* with following request:

```
<Request>
  <Items>
    <Asset>
      <ResponsiblePerson>
        <FirstName>John</FirstName>
        <LastName>Doe</LastName>
      </ResponsiblePerson>
    </Asset>
  </Items>
</Request>
```

If there is multiple persons named John Doe in database, error is returned by default. In this case the responsible person should be identified more precisely or option *suppressMultiChildError* must be set to allow the the web service to return the first matching user.

UseNamespace

This option can be used to add default namespace declaration to the response documents. See [5.1 Root elements](#) for more information.

- **Default value:** -
- **Usage:** options= useNamespace

Example 2-9: Usage of useNamespace option

GET: *connectors/rest.svc/asset/1835?Options=usenamespace&select=Asset[AssetTag]*

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.4" Type="Basic"
Time="11.06.2013 09:54:47" StatusCode="200">
  <Items item="Asset" count="1">
    <Asset id="1835">
      <AssetTag>A0001835</AssetTag>
    </Asset>
  </Items>
</Response>
```

Type

Type option can be used to modify the structure of XML document. Miradore web service supports three different types: *Basic*, *Embedded* and *Referred*. This chapter describes only the effect of *Type* option. For more information about XML payload structure, refer to [5 XML payload structure](#).

Basic

Basic is the default setting if *type* option is omitted. *Basic* mode returns only the first tier of items. It returns all the selected attributes for the base item and a display name from each child item. Display name is a special attribute defined for each configuration item and can be used to further query the child items. See [2.2 Select](#) for instructions on how to select different attributes and [Appendix 2: Supported attributes per item](#) for a list of available attributes.

This is the most human readable format that Miradore web service provides. *Basic* formatted responses are useful when caller end doesn't need to know the attributes of child items. Like in [Example 2-10](#), the asset's responsible person is not divided into separate *FirstName* and *LastName* elements which is done with other formatting options.

Usage: *options=Type=Basic*

Example 2-10: Response when option type is set to basic

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Basic" ExportDate="26.11.2012 15:41:15">
  <Items item="Asset" count="1">
    <Asset id="1190">
      <AssetTag>VM0001190</AssetTag>
      <Location id="8">Helsinki</Location>
      <Model id="21">Microsoft Virtual Machine </Model>
      <OS id="7">Microsoft Windows 7 Enterprise x64 English</OS>
      <ResponsiblePerson id="2">Doe John</ResponsiblePerson>
      <SerialNumber>111-222-333-444</SerialNumber>
      <Status>Active</Status>
    </Asset>
  </Items>
</Response>
```

Embedded

Embedded mode extends the response by embedding additional attributes for selected child items. With *Embedded* formatting also attributes of child items can be selected. See [2.2 Select](#) for instructions how to select different attributes.

In *Embedded* formatted responses, all selected child item attributes are shown in separate XML elements, like responsible person's *FirstName* and *LastName* in [Example 2-11](#). This format is useful when calling end needs to separate those attributes.

- **Usage:** `options=Type=Embedded`

Example 2-11: Response when option type is set to embedded

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Embedded" ExportDate="26.11.2012 15:45:35">
  <Items item="Asset" count="1">
    <Asset id="1190">
      <AssetTag>VM0001190</AssetTag>
      <Location id="8">
        <FullName>EMEA &gt; Finland &gt; Helsinki</FullName>
      </Location>
      <Model id="21">
        <Manufacturer id="1">
          <Name>Microsoft</Name>
        </Manufacturer>
        <Name>Virtual Machine</Name>
      </Model>
      <OS id="7">
        <Manufacturer>Microsoft</Manufacturer>
        <Name>Windows</Name>
        <Version>7 Enterprise</Version>
      </OS>
      <ResponsiblePerson id="2">
        <FirstName>John</FirstName>
        <LastName>Doe</LastName>
      </ResponsiblePerson>
      <SerialNumber>111-222-333-444</SerialNumber>
      <Status>Active</Status>
    </Asset>
  </Items>
</Response>
```

Referred

Similarly to *embedded* mode, *referred* mode adds child items' attributes to the response. However, the XML format is considerably different. Instead of embedding the child item's attributes under the item element, only a reference is generated. By doing this, the size of the response can be optimized considerably. See [2.2 Select](#) for instructions on how to select different attributes.

Response document contains one *Items* element for each type of configuration item that the response contains. The definition of each separate configuration item is added to those elements only once. This format is useful when the amount of network traffic must be optimized. Drawback of this format compared to *embedded* is that parsing of the response in calling end may be more difficult.

[Example 2-](#) contains results of a query which has returned one asset with some child items. Asset's child elements reference to those defined in the *Items* elements. For example, *Asset* element contains *ResponsiblePerson* with attribute *id=2*. This links to *User* element with attribute *id=2* within the last *Items* element.

- **Usage:** `options=Type=Referred`

Example 2-12: Response when option type is set to referred

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Referred" ExportDate="26.11.2012 15:49:06">
  <Items item="Asset" count="1">
    <Asset root="1" id="1190">
      <AssetTag>VM0001190</AssetTag>
      <Location id="8" />
      <Model id="2104" />
      <OS id="7" />
      <ResponsiblePerson id="2" />
      <SerialNumber>111-222-333-444</SerialNumber>
      <Status>Active</Status>
    </Asset>
  </Items>
  <Items item="Location" count="1">
    <Location id="8">
      <FullName>EMEA &gt; Finland &gt; Helsinki</FullName>
    </Location>
  </Item>
  <Items item="Model" count="1">
    <Model id="21">
      <Manufacturer id="1" />
      <Name>Virtual Machine</Name>
    </Model>
  </Item>
  <Items item="HWManufacturer" count="1">
    <HWManufacturer id="1">
      <Name>Microsoft</Name>
    </HWManufacturer>
  </Items>
  <Items item="OS" count="1">
    <OS id="7">
      <Manufacturer>Microsoft</Manufacturer>
      <Name>Windows</Name>
      <Version>7 Enterprise</Version>
    </OS>
  </Items>
  <Items item="User" count="1">
    <User id="2">
      <FirstName>John</FirstName>
      <LastName>Doe</LastName>
    </User>
  </Items>
</Response>
```

2.2 Select

Select query setting is used to define which item attributes are included in the response. In *embedded* and *referred* response formats, it can also be used to define returned attributes of child items.

Each exportable item has a predefined list of selected attributes. These default attributes are returned if nothing else is selected. Best practice is to always define explicitly the attributes required by the caller end.

The syntax for the expression is the following:

```
/Item?select=Item1[Attribute1,Attribute2,ChildItem2],Item2[AttributeA, AttributeB]
```

Following examples explain the usage of the *select* setting.

Example 2-13: Simple example with Basic output format

Select only *AssetTag* and *SerialNumber* attributes and *Location* child item from each asset.

- **Usage:** `select=Asset[AssetTag,SerialNumber,Location]`
- **Example output:**

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Basic" ExportDate="27.11.2012 11:25:05">
  <Items item="Asset" count="2">
    <Asset id="1">
      <AssetTag>0000001</AssetTag>
      <Location id="55">Helsinki</Location>
      <SerialNumber>abc123</SerialNumber>
    </Asset>
    <Asset id="2">
      <AssetTag>0000002</AssetTag>
      <Location id="70">Lappeenranta</Location>
      <SerialNumber>ABC0001</SerialNumber>
    </Asset>
  </Items>
</Response>
```

Example 2-74: Child item with embedded output format

This example demonstrates how *select* is used to define attributes for child items. *Asset*'s *select* list contains *ResponsiblePerson* which is child item of type *User*. For *User* item there is separate *select* list which selects *FirstName*, *LastName* and *Email*.

- **Usage:** `select=Asset[DeviceName,ResponsiblePerson],User[FirstName,LastName,Email]&Options=type=embedded`
- **Example output:**

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.2" StatusCode="200"
Type="Embedded" Time="16.04.2013 19:02:12">
  <Items item="Asset" count="1">
    <Asset id="1852">
      <DeviceName>L0001852</DeviceName>
      <ResponsiblePerson id="50220">
        <FirstName>John</FirstName>
        <LastName>Doe</LastName>
        <Email>john.doe@miradore.com</Email>
      </ResponsiblePerson>
    </Asset>
  </Items>
</Response>
```

Example 2-15: Select with filtering

The *select* setting can be used in conjunction with *filters* setting (see [2.3 Filters](#)). Excluding an attribute with the *select* does not prevent filtering with the attribute.

- **Usage:** `select=Asset[AssetTag, SerialNumber]&filters=Asset.Location.Name eq Helsinki`
- **Example output:**

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Basic" ExportDate="27.11.2012 11:25:05">
  <Items item="Asset" count="1">
    <Asset id="1">
      <AssetTag>0000001</AssetTag>
      <SerialNumber>abc123</SerialNumber>
    </Asset>
  </Items>
</Response>
```

Example 2-16: "Select all" statement

By default, Miradore web service returns only default attributes for an item. *Select all* statement can be used to select all attributes, except extensions, from an item. Extensions are read-only attributes like asset's hardware inventories.

- **Usage:** `select=Asset[AssetTag,Location],Location[*]&options=type=embedded`
- **Example output:**

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Embedded" ExportDate="27.11.2012 11:56:34">
  <Items item="Asset" count="1">
    <Asset id="1">
      <AssetTag>0000001</AssetTag>
      <Location id="61">
        <Administrator>Administrator</Administrator>
        <DHCP>True</DHCP>
        <FullName>EMEA &gt; Finland &gt; Helsinki</FullName>
        <LocationID>61</LocationID>
        <ModifyDate>21.11.2012 05:08:27</ModifyDate>
        <Name>Helsinki</Name>
        <Parent id="16">Finland</Parent>
        <ShortName>HEL</ShortName>
        <Status>Active</Status>
      </Location>
    </Asset>
  </Items>
</Response>
```

Example 2-17: "Select nothing" statement

In contrast to *select all* statement, *select nothing* can be used to exclude all attributes (including default attributes) from an item. This can be used for performance reasons if only child item's *id* is needed (see *Location* element in following example output).

- **Usage:** `select=asset[AssetTag,Location],Location[]`
- **Example output:**

```
<?xml version="1.0" encoding="utf-8"?>
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.0" StatusCode="200"
Type="Embedded" ExportDate="27.11.2012 11:56:34">
  <Items item="Asset" count="1">
    <Asset id="1">
      <AssetTag>0000001</AssetTag>
      <Location id="5305" />
    </Asset>
  </Items>
</Response>
```

2.3 Filters

Web service queries can be filtered by adding *filters* parameter in request URL's query string. *Filters* specify conditions that must be met by a resource for it to be returned in the response.

Filter must contain at least item's attribute, comparison operator and value. [Table 4](#) lists supported comparison operators and shows examples of their usage. The syntax for the single value filter expression is as follows:

`/Item?filters=<Item.Attribute1> <comparison_operator> <value>`

Table 4 A: Supported comparison operators

Operator	Description	Usage
<i>eq</i>	Equals	Asset.Status eq active
<i>ne</i>	Not equals	Asset.Status ne removed
<i>gt</i>	Greater than	Asset.AssetId gt 78
<i>ge</i>	Greater than or equal	Asset.AssetId ge 78
<i>lt</i>	Less than	Asset.AssetId lt 78
<i>le</i>	Less than or equal	Asset.AssetId le 78
<i>contains</i>	Value contains a part of a string	Asset.AssetTag contains pc000
<i>not contains</i>	Value does not contain part of the string	Asset.AssetTag not contains pc000
<i>not null</i>	Value in database is not null or empty	Asset.LocationID not null
<i>is null</i>	Value in database is null or empty	Asset.DeviceUsage.DeviceUsageID is null

Table 5 B: Additional comparison operators for **DateTime** attributes only

Operator	Description	Usage
<i>more than</i>	The more than operator enables to define a dynamic filter which includes all items that are <u>outside</u> the defined time range from the present time, either before or after. The range can be defined in days (d) or in hours (h). Use positive values to define a time range to the future, and negative values for the past.	Asset.AgentStatus.LastSeenTime more than -12h
<i>within</i>	The within operator allows to define a dynamic filter which includes items that are <u>inside</u> the defined range of time from the current moment. The range can be defined either in days (d) or in hours (h). Use positive values to define a time range to the future, and negative values for the past.	Asset.WarrantyEnd within 30d

Examples:

/Connectors/rest.svc/asset?filters=Asset.AgentStatus.LastSeenTime within -3&select=Asset[AssetID,AgentStatus],AgentStatus[LastSeenTime]&options=type=embedded

/Connectors/rest.svc/asset?filters=Asset.AgentStatus.LastSeenTime within -3h&select=Asset[AssetID,AgentStatus],AgentStatus[LastSeenTime]&options=type=embedded

/Connectors/rest.svc/asset?filters=Asset.WarrantyEnd within -30d or Asset.WarrantyEnd within 30d&select=Asset[AssetID,WarrantyEnd]&options=type=embedded

/Connectors/rest.svc/asset?filters=Asset.AgentStatus.LastSeenTime more than -30d&select=Asset[AssetID,AgentStatus],AgentStatus[LastSeenTime]&options=type=embedded

Miradore web service supports also combining of multiple attribute comparisons with logical operators. [Table 6](#) lists supported logical operators and shows examples of their usage. The syntax for filtering expression with logical operators is as follows:

```
/Item?filters=<Item.Attribute1> <comparison_operator1> <value1> <logical_operator> <Item.Attribute2>
<comparison_operator2> <value2>
```

Table 6: Supported logical operators

Operator	Usage
and	Asset.Status eq active and Asset.AssetId eq pc000001
or	Asset.Status eq active or Asset.AssetId eq pc000001

Precedence of *and* and *or* operators is equal and expressions are evaluated from left to right. Brackets can be used to change the order of evaluation of expressions.

Following examples demonstrates the usage of *filters* setting.

Example 2-18: Filtering by one attribute

- **Usage:** filters=Asset.Status eq active

Example 2-19: Filtering with brackets

Brackets can be used to concatenate *AND* and *OR* filters:

- **Usage:** filters=asset.status eq active and (asset.model.manufacturer.name eq nokia or asset.model.manufacturer.name eq apple)

Example 2-20: Filtering by child item's attributes

Every attribute on an item can be used as a filter, including attributes on child items. Use the following expressions to request assets located in Finland and made by Nokia:

- **Usage:** filters= Asset.Location.Name eq Finland and Asset.Model.Manufacturer.Name eq Nokia

Example 2-21: Filtering by inventories

Assets can be filtered using inventory data. The following filter expressions returns assets with more than 2TB free space on drive C.

- **Usage:** filters=Asset.InvlogicalDisk.Caption eq 'c:' and Asset.InvlogicalDisk.Freespace gt 200000000000

Example 2-22: Filtering by values with empty spaces

Filters and filter operators are separated by empty spaces, and therefore, apostrophes are required if the filtered value contains empty spaces:

- **Usage:** filters=Asset.ResponsiblePerson.Organisation.Name eq 'Research and Development'

2.4 OrderBy

Order by query setting can be used to sort the result set by an attribute. Sorting can be done based on any plain text field on an item. Sorting can be done based on one or more attributes. *Child*, *extension* or *binding* attributes cannot be used in order by clause. Sorting can be done in either ascending (*asc*) (default) or descending (*desc*) order.

- **Default value:** -
- **Usage:** `orderby=Asset.ModifyDate desc`
- **Example output:** Asset list sorted by asset's last modification time in descending order

2.5 ChildFilters

ChildFilters query setting can be used to filter the returned extension child items. Child item filters can be added separately for each selected child item type. Notice that this can be used only for child items which type is extension and its meaningful only if filtered attribute is in attribute selection list.

- **Default value:** -
- **Usage:** `childFilters=InvManagedSoftware[SWName contains Office]`

Example 2-23: ChildFilters query setting

Query assets which are located in Lappeenranta and has managed software which name contains *Microsoft Office*. Select *DeviceName* and *InvManagedSoftware* attributes. Limit query to return only *InvManagedSoftware* items which *SWName* contains *Microsoft Office*. If *childFilters* query setting hasn't been specified, query would return all *InvManagedSoftware* items.

GET:

```
/Connectors/rest.svc/Asset?select=Asset[DeviceName,InvManagedSoftware]&options=type=embedded&filters=Asset.Location.Name eq Lappeenranta and Asset.InvManagedSoftware.SWName contains 'Microsoft Office'&childFilters=InvManagedSoftware[SWName contains 'Microsoft Office']
```

2.6 Aggregate

Aggregate function can be used to calculate a single value over the values of multiple rows returned by the query. For example, aggregate function *Min* returns the lowest value of selected attribute on selected items.

- **Default value:** -
- **Usage:** Aggregate=Min(QixPercent)

It is possible to give optional “display name” for the aggregate:

- Aggregate=Min(QixPercent) as MinQixPercent

Supported aggregate operations:

Operation	Parameters	Supported attribute types	Example
Count	-	-	
DistinctCount	Attribute	-	DistinctCount(AssetID)
Min	Attribute	int, bigint, datetime	Min(AgentStatus.LastSeenTime)
Max	Attribute	int, bigint, datetime	Max(AgentStatus.LastSeenTime)
Avg	Attribute	int, bigint	Avg(InvComputerSystem.TotalPhysicalMemory)
Sum	Attribute	int, bigint	Sum(InvDiskDrive.Size)

Notice: Null values are eliminated by: Min, Max, Avg and Sum.

2.7 GroupBy

GroupBy query setting can be used for collecting data across multiple records by grouping the result set by one or more attributes. The *GroupBy* operation creates a single subset for each group. This process is also known as aggregation.

Filters and Childfilters can be used to limit the results returned by the query.

Example 2-24: The basic syntax for a query with the groupby query setting

```
/Item?groupby=[Attribute1],aggregate[Count]
```

Example 2-25: Additional syntax examples about queries with the groupby query setting

```
/Item?groupby= [Attribute1 as SomeName],aggregate[Count as TotalCount]
```

```
/Item?groupby= [Attribute1 as SomeName asc],aggregate[Count as TotalCount desc]
```

```
/Item?groupby= [Attribute1,Attribute2],aggregate[max(DateTimeAttributeA)]  
,aggregate[min(DateTimeAttributeA)]
```

Notices:

1. Grouping can be done by one or more attributes and there can be 0 or more aggregates.
2. Order (asc/desc) can be set for the group by attributes and aggregates individually. Default: no ordering.
3. It is possible to give optional “display name” for each group by attribute and aggregate.
 - Format:
 - `groupby=[<Attribute1>as DisplayName1],aggregate(<Operation>as DisplayName2)`
 - Default value: `Attributename`
 - “`Location.Name`” from “`/Asset?groupby=[Location.Name]`”
 - Example:
 - `/Asset?groupby=[Location.name as Location],aggregate[Count as Total]`
4. Grouping by child items and binding items
 - It is possible to use child/binding item’s attributes for grouping:
 - `/Asset?groupby=[Location.Name]`
 - `/Asset?groupby=[DeviceUsage.Name]`
 - `/Asset?groupby=[DeviceUsage.DeviceRole.Name]`
 - It is however, recommended to filter by child/binding attribute itself because in that way the performance of the grouping operation is a lot better. In this case grouping is done based on the child/binding item’s primary key and the child/binding item’s display name is returned.
 - `/Asset?groupby=[Location]`
 - `/Asset?groupby=[DeviceUsage]`
 - `/Asset?groupby=[DeviceUsage.DeviceRole]`
 - Notice that Extensions must be filtered using extension item’s attributes.
 - `/Asset?groupby=[InvFileScan.FileName]`
 - **Not supported:** `/Asset?groupby=[InvFileScan]`
5. For supported aggregate operations check [chapter 2.6](#).
6. Limit results with “rows” option and use “[showother](#)” option when needed. In large environments, “showother” may have relatively large impact on the performance of the queries.

3. Authentication and security

Miradore web service uses HTTP basic authentication with Miradore user groups. HTTP basic authentication sends user credentials over the network. To ensure security the use of HTTPS is recommended. Otherwise, user credentials are sent unencrypted.

There is two built-in user groups for web service, *Web service readers* and *Web service writers*. Only users in those groups has access to web service. Users in *Web service writers* group can use all web service operations and users in *Web service readers* group can use only *get* operation. See *User management* in Miradore product guide for more information about Miradore user groups.

4. Available operations

This chapter explains operations available through Miradore web service. It supports five different operations: *get*, *create*, *update*, *remove* and *execute*. Each operation is explained more comprehensively below in operation specific sections.

4.1 Get

Get operation is used to get list of configuration items with selected attributes from CMDB.

Request

Get can be used to get single item with item ID. In this case item ID is defined in the path partition of URL. Another option is to define filters in query string and query returns all items which to filter matches. Refer [2 URL and query string](#) for information about query options.

Example 4-1: Get operation with item id

GET:

`/connectors/rest.svc/Asset/12?select=Asset[AssetTag,Location],Location[Name,ShortName]&options=type=embedded`

RESPONSE:

```
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.4" StatusCode="200"
Type="Embedded" Time="10.05.2013 15:24:48">
  <Items item="Asset" count="1">
    <Asset id="12">
      <AssetTag>A0000012</AssetTag>
      <Location id="5">
        <Name>Ruoholahti</Name>
      </Location>
    </Asset>
  </Items>
</Response>
```

Query returns asset which *AssetID* is 12. *AssetTag* and *Location* attributes are selected for *Asset* and *Location* contains *Name* and *ShortName* attributes. Query string contains also *type* option which states that response must use embedded formatting.

Example 4-2: Get operation with filters

GET: `/connectors/rest.svc/asset?filters=Asset.Location.Name eq Helsinki.`

RESPONSE:

```
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.4" StatusCode="200"
Type="Basic" Time="10.05.2013 15:26:59">
  <Items item="Asset" count="17">
    <Asset id="6">
      <AssetTag>A0000006</AssetTag>
      <DeviceName>L0000006</DeviceName>
      <Location id="3">Helsinki</Location>
      <ResponsiblePerson id="22">Steele Eric K</ResponsiblePerson>
    </Asset>
    <Asset id="7">
      <AssetTag>A0000007</AssetTag>
      <DeviceName>L0000007</DeviceName>
      <Location id="3">Helsinki</Location>
      <ResponsiblePerson id="38">Walton Beth G</ResponsiblePerson>
    </Asset>
    <!--More assets-->
  </Items>
</Response>
```

Query returns assets which location is Helsinki.

Response

Get operation returns HTTP status codes defined in [Table 7](#). Response contains found items in XML document. Format of XML do can be altered by query options, see [2.1 Options](#) for more information. See examples from [Request](#) section above.

Table 7: Get specific http status codes

Status code	Description
200	Get operation executed successfully
400	Bad request, check response's /Response/Error element
500	Internal server error has been occurred

4.2 Create

Create is used to create a configuration item to Miradore CMDB.

Only one item can be created with a single call. This makes error handling and identification of created items easier.

If an item already exists i.e. some of the item's attributes, that should be unique is already reserved by some existing item, an error is returned.

Request

Create uses HTTP POST method and created item is send in requests body. HTTP header *Content-Type* must be set to *text/raw*.

Created item must be defined in request XML document within *Items* element with embedded formatting. See following examples and [5 XML payload structure](#) for more information about structure of XML document.

Child items must already exists and they are only linked to parent item. Children are identified with attributes defined in request. If item which matches these attributes is not found, error is returned. Defined attributes may also match multiple child items. In this case, first matching item is selected by default. This can be prevented with *suppressMultiChildError* query option. Refer [2.1 Options](#) for more information about query options.

Query string setting *select* can be used to define which created item's attributes are returned. This can be useful for example when creating an asset and Miradore generates asset tag which can be returned to calling system in response.

Example 4-3: Create asset

POST: /connectors/rest.svc/asset?select=Asset[AssetTag]

BODY:

```

<Request>
  <Items>
    <Asset>
      <Location>
        <Name>Lappeenranta</Name>
      </Location>
      <Model>
        <Manufacturer>
          <Name>HP</Name>
        </Manufacturer>
        <Name>ProBook 6560b</Name>
        <Type>A0001D02</Type>
      </Model>
      <ResponsiblePerson>
        <FirstName>John</FirstName>
        <LastName>Doe</LastName>
      </ResponsiblePerson>
      <SerialNumber>4C11917R</SerialNumber>
      <Status>New</Status>
    </Asset>
  </Items>
</Request>

```

Response

Create operation returns HTTP status codes defined in [Table 8](#). It also returns XML document which contains by default created item's primary key attribute or attributes defined in request's URL. This XML document is formatted same way as responses of *get* operation.

Table 8: Create specific HTTP status codes

<i>Status code</i>	<i>Description</i>
200	All items are created successfully
202	Item has been created but there is some warnings, check response's Warnings element
400	Bad request, check response's /Response/Error element
500	Internal server error has occurred

4.3 Update

Update is used to update one or more attributes of specified item. Item must be specified with item name and id.

Request

Update operation is used with HTTP PUT method by sending request XML document to URL which identifies the item to be updated. HTTP header *Content-Type* must be set to *text/raw. Embedded* formatted request XML document must contain the updated values of item's attributes.

Child items must already exist and they are only linked to parent item. Children are identified with attributes defined in request. If item which matches these attributes is not found, error is returned. Defined attributes may also match multiple child items. In this case, first matching item is selected by default. This can be prevented with *suppressMultiChildError* query option. Refer [2.1 Options](#) for more information about query options.

Attribute can be updated to empty value by adding empty attribute element to request.

Query string setting *select* can be used to define which created item's attributes are returned.

Example 4-4: Update asset which AssetID is 123. Set Location to Helsinki and clear Notices.

PUT: /connectors/rest.svc/asset/123

BODY:

```
<Request>
  <Items>
    <Asset>
      <Location>
        <Name>Helsinki</Name>
      </Location>
      <Notices/>
    </Asset>
  </Items>
</Request>
```

Example 4-5: Add two device usages to asset which id is 123

Configuration items can contain multiple values for attributes which type is *binding*. In this example two device usages are added to target asset. Existing device usages are not cleared with *update* operation. They can be removed with [Remove](#) operation. For more information about attribute types, refer [Appendix 2: Supported attributes per item](#).

PUT: /connectors/rest.svc/asset/123

BODY:

```
<Request>
  <Items>
    <Asset>
      <DeviceUsage>
        <Name>Personal</Name>
      </DeviceUsage>
      <DeviceUsage>
        <Name>Developer</Name>
      </DeviceUsage>
    </Asset>
  </Items>
</Request>
```

Response

Update operation HTTP status codes defined in [Table 9](#). It also returns XML document which contains by default updated item's primary key. This XML document is formatted same way as responses of *get* operation.

Table 9: Update specific HTTP status codes

Status code	Description
200	Item updated successfully
400	Bad request, check response's /Response/Error element
500	Internal server error has been occurred

4.4 Remove

Remove is used to remove configuration item from CMDB. Only one item can be removed with a single call and removed item must be identified with item ID.

Request

Remove operation is used with HTTP DELETE method by sending request to URL which identifies the item to be removed.

Example 4-6: Remove asset which AssetID is 123

```
DELETE /Connectors/rest.svc/asset/123
```

Remove operation is also used to remove item bindings (i.e. item attributes which can have multiple values like asset's device usage). In this case URL must also define binding item type and optionally binding item identifier. If only binding type is defined, all bindings of that type are removed. See examples below for more information. Notice that this removes only link between item and binding item and actual binding configuration item remains unchanged.

Example 4-7: Remove all device usages from asset which AssetID is 123

```
DELETE /Connectors/rest.svc/asset/123/deviceusage
```

Example 4-8: Remove device usage which DeviceUsageID is 1 from asset which AssetID is 123

```
DELETE /Connectors/rest.svc/asset/123/deviceusage/1
```

Response

Remove operation returns an empty HTTP response with status code 200 if removing succeeds. Otherwise, it returns response XML document with error element.

Table 10: Remove specific HTTP status codes

Status code	Description
200	Removed successfully, response has no content
400	Bad request, check response's /Response/Error element
500	Internal server error has been occurred

4.5 Execute

Execute operation is used to run actions for configuration items. All the supported actions are listed in the table after this paragraph. The actions can be executed for one item at a time.

<i>Action name</i>	<i>Target item</i>	<i>Description</i>
<i>ClearInventory</i>	Asset	Clears the specified asset's inventory data from either all inventory data categories or from the specified inventory category. See examples below.
<i>RunScheduledTask</i>	Asset	Runs either one specified scheduled task or all scheduled tasks for the specified asset. See examples below.
<i>UseAssetData</i>	Asset	Sets the asset attribute's mismatch status to <i>acknowledged</i> . The target attribute's value remains unchanged in the asset configuration item, which means that the inventory value is not updated to the asset. See examples below.
<i>UseInventoryData</i>	Asset	Sets the asset attribute's mismatch status to <i>closed</i> . The target attribute is updated to use the inventory value in the asset configuration item. See examples below.
<i>ResetMismatch</i>	Asset	Resets the asset attribute's mismatch status back to its default value, which is <i>New</i> . See examples below.
<i>ResetPatchRetryCounts</i>	Asset	Resets patch (un)installation retry count value. See example below.
<i>UpdateDynamicGroupMembers</i>	AssetGroup	Updates the member assets of the asset group according to the asset filters defined at the asset group settings. See example below.
<i>DownloadNow</i>	Patch	Commands the Miradore Management Suite server to download the patch installation files to the media master installation point immediately. See example below.

About asset data mismatch feature

Asset data mismatch feature compares manually entered asset data with automatically collected inventory information and helps to detect the attributes which are conflicting or mismatching. For example, if asset's manually entered location is London, but the inventory data indicates that the asset is actually in Paris, there is a mismatch in the data.

Asset data mismatch attributes

Data mismatch feature supports the following asset attributes: *IMEI*, *Location*, *Phone number*, *Responsible person*, *Serial number*, and *UDID*.

Request

Execute operation is used with HTTP POST method by sending a request URL which identifies the asset configuration item where the action should be run and also the desired action with the possible parameter.

Example 4-9:

Clear all inventory data for asset whose AssetID is 123

`/Connectors/rest.svc/Asset/123/ClearInventory`

*Example 4-10:**Clear specific inventory data type for asset whose AssetID is 123*`/Connectors/rest.svc/Asset/123/ClearInventory/<TypeValue>`

If you want to execute the clear inventory information action to some specific inventory categories only, you can use the following queries to get a list of all inventory types:

Computer:

`/Connectors/rest.svc/Types?filters=Types.Group eq 24&select=Types[Description,GroupDescription,Value]`

Printer/SNMP:

`/Connectors/rest.svc/Types?filters=Types.Group eq 25&select=Types[Description,GroupDescription,Value]`

Mobile:

`/Connectors/rest.svc/Types?filters=Types.Group eq 26&select=Types[Description,GroupDescription,Value]`*Example 4-11:**Run all scheduled tasks for asset whose AssetID is 123*`/Connectors/rest.svc/Asset/123/RunScheduledTask`*Example 4-12:**Run a specific scheduled task for asset whose AssetID is 123*`/Connectors/rest.svc/Asset/123/RunScheduledTask/<InternalName>`

If you want to execute some specific scheduled task, you can get a list of all scheduled tasks per operating system category with the following query:

`/Connectors/rest.svc/ScheduledTask?select=ScheduledTask[InternalName,OSCategory]`*Example 4-13:**Acknowledge data mismatch report about the location attribute for the asset whose ID is 123 and use the asset value*`/Connectors/rest.svc/Asset/123/UseAssetData/Location`*Example 4-14:**Close data mismatch report about the serial number attribute for asset whose ID is 123 and use the inventory value*`/Connectors/rest.svc/Asset/123/UseInventoryData/Serial number`*Example 4-15:**Reset data mismatch report status of the Responsible person attribute for the asset whose ID is 123*`/Connectors/rest.svc/Asset/123/ResetMismatch/Responsible person`*Example 4-16:**Reset patch (un)installation retry count on asset which ID is 123*`/Connectors/rest.svc/Asset/123/ResetPatchRetryCounts`*Example 4-17:**Update the members of the asset group whose AssetGroupID is 123*`/Connectors/rest.svc/AssetGroup/123/UpdateDynamicGroupMembers`

Example 4-18:

Command Miradore server to download patch installation files for the patch whose PatchID is 123

`/Connectors/rest.svc/Patch/123/DownloadNow`

Response

Execute operation returns an empty HTTP response with status code 200 if the action succeeds. Otherwise, it returns response XML document with Error element.

5. XML payload structure

This chapter describes the structure of web service XML payload.

5.1 Root elements

Web service has two possible root elements, one for requests and one for responses.

Request

This is the root element of XML document consumed by web service's *create* and *update* operations. Web service ignores all attributes defined in the elements of consumed XML documents. Only exception is *xmlns* attribute which must not be included. For example, it's allowed to include *id* attributes returned in web service response to request document. This is demonstrated in [Example 5-1](#), *ResponsiblePerson* element contains *id* attribute which doesn't have any effect.

Example 5-1: Request element

```
<Request>
  <Items>
    <Asset>
      <ResponsiblePerson id="1">
        <FirstName>John</FirstName>
        <LastName>Doe</LastName>
      </ResponsiblePerson>
    </Asset>
  </Items>
</Request>
```

Response

This is root element of web service responses. *Get*, *Create* and *update* operations always returns it but *remove* operation contains response body only if some error occurs. Following table lists attributes which Response element always contains.

Table 11: List of attributes in Response element

Attribute	Description
<i>xmlns</i>	Contains XML namespace declaration <i>http://www.miradore.com/xmlns/ws/1.0</i> .
<i>MDVersion</i>	Miradore server version
<i>StatusCode</i>	HTTP status code
<i>Type</i>	Defines used response format. Possible values: <i>Basic</i> , <i>Embedded</i> , <i>Referred</i>
<i>Time</i>	Response time, server's time zone is used. Format of time string can be altered with <i>df</i> query option.

xmlns

This contains namespace declaration for web service responses if usage of namespace is enabled with

UseNamespace_option. Web service uses default namespace *http://www.miradore.com/xmlns/ws/1.0*.

MDVersion

Contains version string of Miradore server.

StatusCode

HTTP status code returned by the web service call.

Type

This defines the structure of configuration item elements in the response document. It can contain three different values *Basic*, *Embedded* or *Referred*. For more information, refer [Type](#) query option.

Time

Time when response has been sent from server. Server's time zone is used and format of this value can be altered with [Df](#) query option.

5.2 Child elements

Items

Both web service requests and responses can contain *Items* element. This element contains the actual data of the query i.e. a list of configuration items of certain type. Response document's *Items* element can contain zero or more configuration items as child elements. In request document, there must be one child element. Refer [Appendix 1: Supported items and operations](#) to see the list of supported configuration items.

Web service adds attributes defined in [Table 12](#) to response document's *Items* element. Request document can contain these attributes but they are ignored.

Table 12: Attributes of Items element

Attribute	Description
<i>item</i>	Configuration item name
<i>count</i>	Item count (count of child elements named as <i>items</i> attribute's value)

Items element's child elements are named with current configuration item's name, like *Asset* element in [Example 5-2](#). All configuration items has attributes which are represented as child elements under configuration item element, like *AssetTag*, *Location* and *ResponsiblePerson* elements in the example.

Refer to

[Appendix 2](#): Supported attributes per item to see the list of available attributes.

Attributes can also be configuration items (so called child items). Typically these child item elements are named with child configuration item name like asset's *Location* but names may also differ like asset's *ResponsiblePerson* which is child item of type *User*.

Get operation can be configured to add child item's type as *item* attribute's value, see *Location* and *ResponsiblePerson* elements in the following example. This is done by adding [AddChildType](#) query option.

Miradore web service has multiple formats for the structure of configuration item elements, see [Type](#) query option for more information.

Example 5-2: Items element in response document

```
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.4" Type="Embedded"
Time="23.05.2013 13:10:53" StatusCode="200">
  <Items item="Asset" count="1">
    <Asset id="1852">
      <AssetTag>A0001852</AssetTag>
    <Location id="70" item="Location">
      <Name>Lappeenranta</Name>
    </Location>
    <ResponsiblePerson id="50220" item="User">
      <FirstName>John</FirstName>
      <LastName>Doe</LastName>
    </ResponsiblePerson>
  </Asset>
</Items>
</Response>
```

Warnings

This element contains a list of warnings that has occurred during the requested operation. If warnings exists, it means that operation has been completed but something unexpected has occurred. In this case HTTP response code starts with 2 but is not 200. *Warnings* element contains one or more *Warning* child elements which contains human readable description. It also contains *count* attribute which defines the number of *Warning* elements. Following example contains *Warnings* element.

Example 5-3: Response which contains warnings

```
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.4" Type="Embedded"
Time="17.05.2013 15:58:27" StatusCode="202">
  <Items>
    <Asset id="1976">
      <AssetID>1976</AssetID>
    </Asset>
  </Items>
  <Warnings count="1">
    <Warning>Failed to attach 'FinancialTransaction' to 'Asset': 'Type' child item not found
with filter: Types.Description eq 'Purchase12' AND Types.Group eq '18'</Warning>
  </Warnings>
</Response>
```

Error

This element contains description of occurred error. It has following child elements *Status Code* (HTTP status code), *Description* (HTTP status description) and *Details* (human readable error description). If error occurs, processing of web service call is aborted immediately. Following example contains *Error* element.

Example 5-4: Response which contains error

```
<Response xmlns="http://www.miradore.com/xmlns/ws/1.0" MDVersion="3.5.4" Type="Embedded"
Time="17.05.2013 15:56:51" StatusCode="400">
  <Error>
    <StatusCode>400</StatusCode>
    <Description>Bad Request</Description>
    <Details>'Location' child item not found with filter: Location.Name eq 'Turku'</Details>
  </Error>
</Response>
```

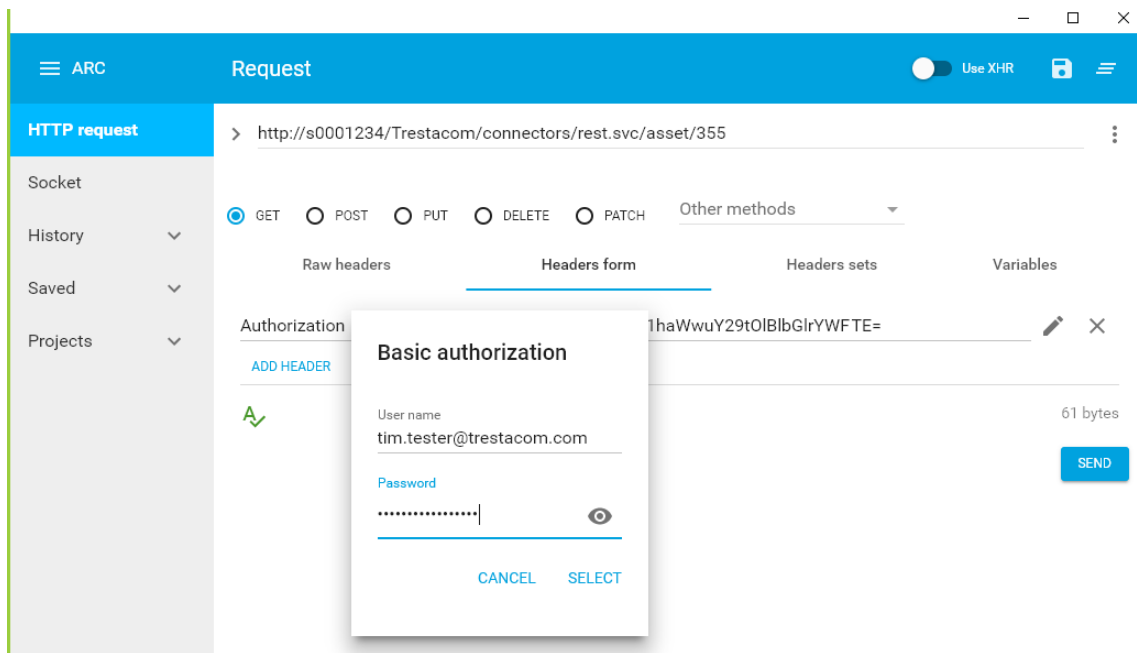
There may also be higher level errors like web service call's URL path is not defined correctly or unsupported HTTP method or content type is used. In this case Miradore server returns more generic error message which status code and contents depends on the type of error.

6. Testing the web service manually

Testing API with your custom HTTP requests requires an application to interact with the API. It is recommended to test the web service using a REST client which makes it possible to add the authorization headers to the request, because without that the access will be denied, and you cannot really test your queries. All your requests should be carefully tested before using them in production.

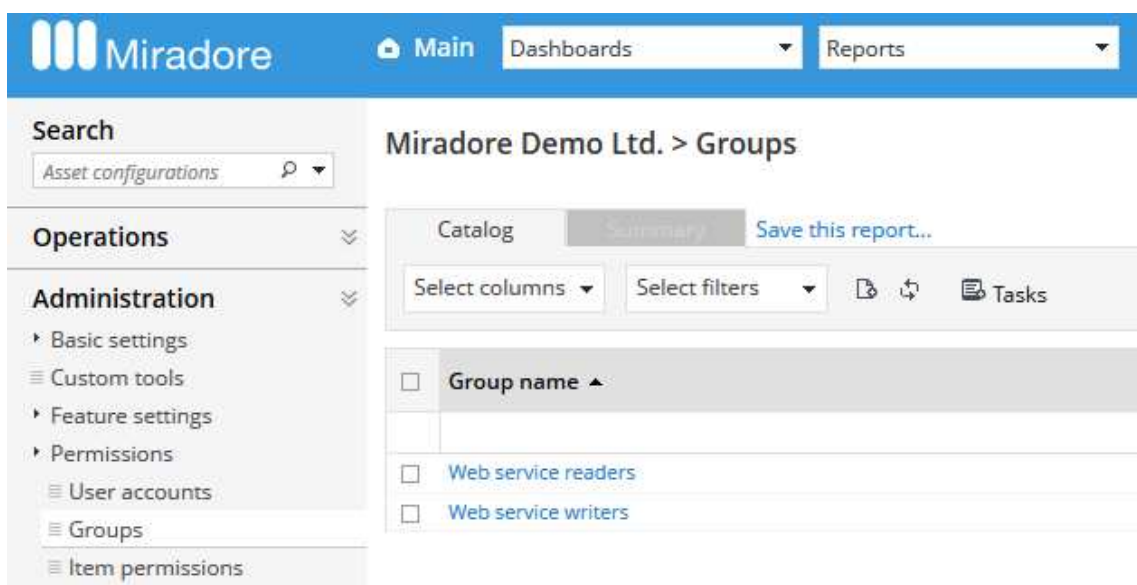
You can use whatever tool you prefer, but here is a short list of tools you might find useful:

- Advanced REST client for Chrome
- RESTClient for Firefox
- RestMan for Opera



Picture: Adding the authorization header in Advanced REST client for Chrome

The HTTP authorization request header contains the credentials to authenticate the user agent. The user must belong to either *Web service readers* or *Web service writers* user group in Miradore Management Suite.



Appendix 1: Supported items and operations

<i>Item</i>	<i>Get</i>	<i>Create</i>	<i>Update</i>	<i>Remove</i>	<i>Execute</i>
<i>AgentStatus</i>	X				
<i>Asset</i>	X	X	X	X	X
<i>AssetConfirmationRequest</i>	X	X			
<i>AssetGroup</i>	X	X	X	X	X
<i>AssetServiceContract</i>	X				
<i>AssetSpecificPackage</i>	X	X	X	X	
<i>Comment</i>	X				
<i>CompanyName</i>	X	X	X	X	
<i>Connector</i>	X				
<i>Contract</i>	X				
<i>ContractServiceLevel</i>	X				
<i>CostCenter</i>	X				
<i>DetailedStatus</i>	X				
<i>DeviceRole</i>	X				
<i>DeviceUsage</i>	X				
<i>Distribution</i>	X	X		X	
<i>EventLogEvent</i>	X				
<i>EventMonitoringEvent</i>	X				
<i>FinancialTransaction</i>	X				
<i>GroupDistribution</i>	X	X	X	X	
<i>GroupDistributionTarget</i>	X				
<i>HistorySeries</i>	X				
<i>HWCategory</i>	X				
<i>HWInventoryChangeLog</i>	X				
<i>HWManufacturer</i>	X				
<i>Incident</i>	X	X	X	X	
<i>IncidentAssignment</i>	X	X			
<i>IncidentCategory</i>	X				
<i>InitialInstall</i>	X	X		X	
<i>LicenseAllocation</i>	X				
<i>LicenseAutoAllocRule</i>	X				
<i>LicensePool</i>	X	X	X	X	
<i>LicenseTransaction</i>	X				
<i>Location</i>	X	X	X	X	
<i>ManagedSoftware</i>	X	X	X	X	
<i>Model</i>	X				
<i>Organisation</i>	X				
<i>OS</i>	X				
<i>OSCategory</i>	X				
<i>Package</i>	X				
<i>Patch</i>	X		X		X

<i>PatchApproval</i>	x	x	x	x
<i>PatchCategory</i>	x		x	
<i>PatchInstallationStatus</i>	x			
<i>PatchProduct</i>	x			
<i>PatchStatusSummary</i>	x			
<i>PatchVendor</i>	x		x	
<i>Proxy</i>	x			
<i>QixReport</i>	x			
<i>RegionalSetting</i>	x			
<i>ScheduledTask</i>	x			
<i>ServerNetworkConfiguration</i>	x			
<i>SoftwareCatalog</i>	x			
<i>SSPEnrollment</i>	x	x	x	x
<i>Status</i>	x			
<i>StatusChangeReason</i>	x			
<i>Subnet</i>	x			
<i>SubnetPublicIP</i>	x			
<i>Supplier</i>	x			
<i>SWCategory</i>	x			
<i>SWInventoryChangeLog</i>	x			
<i>SWManufacturer</i>	x			
<i>SystemTask</i>	x			
<i>Timezone</i>	x			
<i>Types</i>	x			
<i>User</i>	x	x	x	x
<i>Vendor</i>	x			

Appendix 2: Supported attributes per item

Following tables list all available items and their attributes.

Type column contains the type of attribute. This may contain SQL data type or some of the abbreviations listed in the following table.

<i>Abbreviation</i>	<i>Meaning</i>	<i>Description</i>
<i>B</i>	Binding	Bindings are like child items but one configuration item can have many of them like asset can have many device usages. Target item column defines the actual type of binding item.
<i>C</i>	Child item	Child items are another configuration items which are linked to parent item. Target item column defines the actual type of child item.
<i>CA</i>	Custom attribute	Custom attributes are instance-specific custom fields for storing asset information. Administrators can define custom attributes through the system settings of Miradore Management Suite.
<i>DN</i>	Display name	This is read-only attribute which contains human-readable identifier. Other attributes are used internally to compose this.
<i>E</i>	Extension	Typically, extension attributes contain extended information about a configuration item. The extension attributes may contain child attributes which can be selected with select statement and used in filtering as any other attribute. By default, all extension attributes are selected.
<i>EL</i>	Extension list	Extensions that are wrapped inside one element in the response document. Typically <i>extension list</i> attributes contain inventory data which can consist of huge number of records. For example, asset's <i>InvFileScan</i> may contain thousands of records. Wrapper XML element contains <i>count</i> attribute which tells the number of child elements.
<i>PK</i>	Primary key	Primary key uniquely identifies the item. This is needed with update and remove operations and can be used with get operation to identify an item in web service call URL.
<i>S</i>	Selection	Available values of selection attributes are predefined and can be found by querying item defined in <i>Target item</i> column.
<i>T</i>	Tree	Tree attributes are used with items which can be arranged as tree like structures like locations. Typically <i>tree</i> attributes contains full name of an item ordered from parent to child.

Target item column contains the exact configuration item type of binding or child item.

G (get), *C* (create), *U* (update) and *R* (remove) tell which operations are available for current attribute.

Notice that removing of attributes of type *binding* means the removing of link between the target item of operation and binding item. See documentation of [Remove](#) operation for more information.

AgentStatus

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Asset</i>	C	Asset	x			
<i>AssetID</i>	PK		x			
<i>ClientVersion</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>LastSeenLocation</i>	C	Location	x			
<i>LastSeenTime</i>	datetime		x			
<i>OnlineStatus</i>	nvarchar		x			

Asset

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AgentStatus</i>	C	AgentStatus	x			
<i>AssetConfirmationRequest</i>	B	AssetConfirmationRequest	x			
<i>AssetGroup</i>	B	AssetGroup	x	x	x	x
<i>AssetGUID</i>	guid		x			
<i>AssetID</i>	PK		x			
<i>AssetTag</i>	nvarchar		x	x	x	
<i>AssetTag2</i>	nvarchar		x	x	x	
<i>AutomatedInstallation</i>	bit		x	x	x	
<i>ChildAsset</i>	B	Asset	x	x	x	x
<i>Comment</i>	B	Comment	x	x		x
<i>CompanyName</i>	C	CompanyName	x	x	x	x
<i>CostCenter</i>	C	CostCenter	x	x	x	
<i>CostCenter2</i>	C	CostCenter	x	x	x	
<i>CreateDate</i>	datetime		x			
<i>DetailedLocation</i>	nvarchar		x	x	x	
<i>DetailedStatus</i>	S	DetailedStatus	x	x	x	
<i>DeviceName</i>	nvarchar		x	x	x	
<i>DeviceRole</i>	C	DeviceRole	x	x	x	
<i>DeviceUsage</i>	B	DeviceUsage	x	x	x	x
<i>DHCP</i>	bit		x	x	x	
<i>DisplayName</i>	DN		x			
<i>ExternalID</i>	nvarchar		x	x	x	
<i>FinancialTransaction</i>	B	FinancialTransaction	x	x	x	x
<i>IMEI</i>	nvarchar		x	x	x	
<i>InitialInstall</i>	B		x			
<i>InitialInstallOS</i>	C	OS	x	x	x	
<i>IP</i>	nvarchar		x	x	x	
<i>Location</i>	C	Location	x	x	x	
<i>MAC</i>	nvarchar		x	x	x	
<i>Model</i>	C	Model	x	x	x	
<i>ModifyDate</i>	datetime		x			
<i>NetworkConnectivity</i>	int		x	x	x	
<i>Notices</i>	nvarchar		x	x	x	
<i>OS</i>	C	OS	x			
<i>OSLicense</i>	nvarchar		x	x	x	
<i>ParentAsset</i>	B	Asset	x			
<i>Password</i>	nvarchar		x	x	x	
<i>PhoneNumber</i>	nvarchar		x	x	x	
<i>PlannedRemovalDate</i>	datetime		x	x	x	
<i>RegionalSettings</i>	C	RegionalSetting	x	x	x	
<i>ResponsiblePerson</i>	C	User	x	x	x	
<i>SerialNumber</i>	nvarchar		x	x	x	
<i>ServiceContract</i>	B	AssetServiceContract	x	x	x	x
<i>SharedUser</i>	nvarchar		x	x	x	
<i>SNMPCommunity</i>	nvarchar		x	x	x	
<i>SNMPScan</i>	bit		x	x	x	
<i>Status</i>	S	Status	x	x	x	
<i>StatusChangeReason</i>	S	StatusChangeReason	x	x	x	

<i>Subnet</i>	C	Subnet	x	x	x
<i>SupplyCritical</i>	nvarchar		x	x	x
<i>SupplyWarning</i>	nvarchar		x	x	x
<i>Timezone</i>	C	TimeZone	x	x	x
<i>TonerCritical</i>	nvarchar		x	x	x
<i>TonerWarning</i>	nvarchar		x	x	x
<i>UDID</i>	nvarchar		x	x	x
<i>WarrantyEnd</i>	datetime		x	x	x
<i>WarrantyStart</i>	datetime		x	x	x
<i>DistributionStatus</i>	E	DistributionStatus	x		
<i>EndpointBackupJobConfiguration</i>	E	EndpointBackupJobConfiguration	x		
<i>EndpointBackupReport</i>	E	EndpointBackupReport	x		
<i>InvADComputer</i>	E	InvADComputer	x		
<i>InvADComputerGroup</i>	E	InvADComputerGroup	x		
<i>InvAntivirus</i>	E	InvAntivirus	x		
<i>InvARPScan</i>	E	InvARPScan	x		
<i>InvBIOS</i>	E	InvBIOS	x		
<i>InvBootDuration</i>	E	InvBootDuration	x		
<i>InvComputerSystem</i>	E	InvComputerSystem	x		
<i>InvComputerSystemProduct</i>	E	InvComputerSystemProduct	x		
<i>InvCustomInventory</i>	E	InvCustomInventory	x		
<i>InvDataMismatch</i>	E	InvDataMismatch	x		
<i>InvDevices</i>	E	InvDevices	x		
<i>InvDiskDrive</i>	E	InvDiskDrive	x		
<i>InvDisplayConfiguration</i>	E	InvDisplayConfiguration	x		
<i>InvFileScan</i>	E	InvFileScan	x		
<i>InvFinancialData</i>	E	InvFinancialData	x		
<i>InvFirewall</i>	E	InvFirewall	x		
<i>InvKeyboard</i>	E	InvKeyboard	x		
<i>InvLastLoggedInUser</i>	E	InvLastLoggedInUser	x		
<i>InvLocalAdmin</i>	E	InvLocalAdmin	x		
<i>InvLogicalDisk</i>	E	InvLogicalDisk	x		
<i>InvLoginHistory</i>	E	InvLoginHistory	x		
<i>InvManagedSoftware</i>	E	InvManagedSoftware	x		
<i>InvMonitor</i>	E	InvMonitor	x		
<i>InvNetworkAdapter</i>	E	InvNetworkAdapter	x		
<i>InvNetworkAdapterConfiguration</i>	E	InvNetworkAdapterConfiguration	x		
<i>InvOperatingSystem</i>	E	InvOperatingSystem	x		
<i>InvPackageManager</i>	E	InvPackageManager	x		
<i>InvPatchInstalled</i>	E	InvPatchInstalled	x		
<i>InvPhysicalMemory</i>	E	InvPhysicalMemory	x		
<i>InvPortableBattery</i>	E	InvPortableBattery	x		
<i>InvPowerUsage</i>	E	InvPowerUsage	x		
<i>InvPrinter</i>	E	InvPrinter	x		
<i>InvProcessor</i>	E	InvProcessor	x		
<i>InvSecurityPatch</i>	E	InvSecurityPatch	x		
<i>InvSecurityProduct</i>	E	InvSecurityProduct	x		
<i>InvSoftwareCatalog</i>	E	InvSoftwareCatalog	x		
<i>InvSystemProfiler</i>	E	InvSystemProfiler	x		
<i>InvTPM</i>	E	InvTPM	x		

<i>InvVideoController</i>	E	InvVideoController	x	
<i>InvWindowsEventLog</i>	E	InvWindowsEventLog	x	
<i>InvWindowsStabilityScore</i>	E	InvWindowsStabilityScore	x	
<i>MiradoreLicense</i>	E	MiradoreLicense	x	
<i>MiradoreOnlineDevice</i>	E	MiradoreOnlineDevice	x	
0-25 custom fields e.g. “_Name”	CA	Custom attributes with ‘_’ prefix	x	x

Removing of a single ServiceContract must be made by using ContractID instead of AssetServiceContractID, for example: <http://server/instance/connectors/rest.svc/Asset/ServiceContract/<ContractID>>

Asset extensions

DistributionStatus

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>ClientVersion</i>	nvarchar
<i>CreateTime</i>	datetime
<i>DistributionID</i>	int
<i>DistributionType</i>	nvarchar
<i>EndTime</i>	datetime
<i>ErrorCode</i>	int
<i>InstPoint</i>	nvarchar
<i>PackageName</i>	nvarchar
<i>PackageRunAs</i>	nvarchar
<i>PackageStatus</i>	nvarchar
<i>PackageType</i>	varchar
<i>Reboot</i>	bit
<i>DistributedBy</i>	nvarchar
<i>StartTime</i>	datetime
<i>StatusText</i>	nvarchar

EndpointBackupJobConfiguration

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>BackupJobID</i>	int
<i>BackupJobName</i>	nvarchar
<i>LastRunTime</i>	datetime

 EndpointBackupReport

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>EndpointBackupClientVersion</i>	nvarchar
<i>EndpointBackupReportID</i>	int
<i>EndTime</i>	datetime
<i>Errors</i>	int
<i>BackupJobID</i>	int
<i>BackupJobName</i>	nvarchar
<i>Operation</i>	nvarchar
<i>Server</i>	nvarchar
<i>SourceDataMB</i>	double
<i>SourceFiles</i>	int
<i>StartTime</i>	datetime
<i>TransferDuration</i>	int
<i>TransferredDataMB</i>	double
<i>TransferredFiles</i>	nvarchar
<i>TransferSpeedKBs</i>	double
<i>UserName</i>	nvarchar

InvADComputer

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>CreateDate</i>	datetime
<i>Description</i>	nvarchar
<i>Domain</i>	nvarchar
<i>InvDate</i>	datetime
<i>LDAPPath</i>	nvarchar
<i>Location</i>	nvarchar
<i>LogonDate</i>	datetime
<i>LogonServer</i>	nvarchar
<i>ModifyDate</i>	datetime
<i>OS</i>	nvarchar
<i>OSVersion</i>	nvarchar
<i>ServicePack</i>	nvarchar

InvADComputerGroup

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Description</i>	nvarchar
<i>Domain</i>	nvarchar
<i>Email</i>	nvarchar
<i>InvDate</i>	datetime
<i>LDAPPath</i>	nvarchar
<i>Name</i>	nvarchar
<i>Scope</i>	int
<i>Type</i>	int

InvAntivirus

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>ClientGroup</i>	nvarchar
<i>DefinitionDate</i>	datetime
<i>DefinitionVersion</i>	nvarchar
<i>InvDate</i>	datetime
<i>IPAddress</i>	nvarchar
<i>LastCheckInTime</i>	datetime
<i>ProductName</i>	nvarchar
<i>ProductVersion</i>	nvarchar
<i>Server</i>	nvarchar
<i>ServerGroup</i>	nvarchar
<i>LastScanTime</i>	datetime

InvARPScan

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>DisplayName</i>	nvarchar
<i>DisplayVersion</i>	nvarchar
<i>EstimatedSize</i>	bigint
<i>EstimatedSizeInMB</i>	double
<i>InstallDate</i>	nvarchar
<i>InstallLocation</i>	nvarchar
<i>InstallSource</i>	nvarchar
<i>InvDate</i>	datetime
<i>ProductCode</i>	nvarchar
<i>Publisher</i>	nvarchar
<i>UninstallString</i>	nvarchar
<i>Version</i>	nvarchar

InvBIOS

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>InvDate</i>	datetime
<i>Manufacturer</i>	nvarchar
<i>ReleaseDate</i>	nvarchar
<i>SerialNumber</i>	nvarchar
<i>SMBIOSVersion</i>	nvarchar
<i>SoftwareElementID</i>	nvarchar
<i>Version</i>	nvarchar
<i>UEFIVersion</i>	nvarchar
<i>UEFIArchitecture</i>	nvarchar
<i>UEFISecureBoot</i>	int

InvBootDuration

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>BootDurationAvg</i>	int
<i>BootDurationFastest</i>	int
<i>BootDurationSlowest</i>	int
<i>BootFrequency</i>	double
<i>InvDate</i>	datetime
<i>MainBootDurationAvg</i>	int
<i>LastBootDate</i>	datetime
<i>LastBootDuration</i>	int
<i>OldestBootDate</i>	datetime
<i>PostBootDurationAvg</i>	int

InvComputerSystem

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Caption</i>	nvarchar
<i>Domain</i>	nvarchar
<i>InvDate</i>	datetime
<i>Manufacturer</i>	nvarchar
<i>Model</i>	nvarchar
<i>Name</i>	nvarchar
<i>SystemType</i>	nvarchar
<i>TotalPhysicalMemory</i>	bigint
<i>TotalPhysicalMemoryInGB</i>	double
<i>UserName</i>	nvarchar

InvComputerSystemProduct

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>IdentifyingNumber</i>	nvarchar
<i>InvDate</i>	datetime
<i>Name</i>	nvarchar
<i>UUID</i>	nvarchar
<i>Vendor</i>	nvarchar
<i>Version</i>	nvarchar

InvCustomInventory

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>AttributeName</i>	nvarchar
<i>ClassIndex</i>	int
<i>ClassName</i>	nvarchar
<i>ClassNameIndex</i>	nvarchar
<i>InvDate</i>	datetime
<i>Value</i>	nvarchar
<i>ValueAsDateTime</i>	datetime
<i>ValueAsNumber</i>	double
<i>ValueType</i>	int

InvDataMismatch

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>AssetValue</i>	nvarchar
<i>Attribute</i>	nvarchar
<i>InventoryDate</i>	datetime
<i>InventoryValue</i>	nvarchar
<i>Status</i>	nvarchar

InvDevices

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>ClassName</i>	nvarchar
<i>Manufacturer</i>	nvarchar
<i>Name</i>	nvarchar
<i>BuiltinDriver</i>	nvarchar
<i>DriverDate</i>	datetime
<i>DriverVersion</i>	nvarchar
<i>DriverProvider</i>	nvarchar
<i>InstalledDrvStatus</i>	nvarchar
<i>HardwareID</i>	nvarchar

InvDiskDrive

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Description</i>	nvarchar
<i>FirmwareRevision</i>	nvarchar
<i>HealthStatus</i>	nvarchar
<i>InterfaceType</i>	nvarchar
<i>InvDate</i>	datetime
<i>Model</i>	nvarchar
<i>Name</i>	nvarchar
<i>Partitions</i>	nvarchar
<i>ReallocatedSectorCount</i>	int
<i>SerialNumber</i>	nvarchar
<i>Size</i>	bigint
<i>SizeInGB</i>	double
<i>Temperature</i>	int
<i>UncorrectableErrorCount</i>	int
<i>WearLevel</i>	int

InvDisplayConfiguration

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>BitsPerPel</i>	int
<i>InvDate</i>	datetime
<i>PelsHeight</i>	int
<i>PelsWidth</i>	int

InvFileScan

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Company</i>	nvarchar
<i>FileName</i>	nvarchar
<i>FilePath</i>	nvarchar
<i>FileSize</i>	int
<i>FileSizeInMB</i>	double
<i>FileVersion</i>	nvarchar
<i>FingerPrint</i>	nvarchar
<i>InvDate</i>	datetime
<i>ProductName</i>	nvarchar

InvFinancialData

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>ChangeDate</i>	datetime
<i>ContractNumber</i>	nvarchar
<i>CostCenter</i>	nvarchar
<i>DeviceModel</i>	nvarchar
<i>DeviceRent</i>	nvarchar
<i>EndDate</i>	datetime
<i>EndingOption</i>	nvarchar
<i>ExtraInfo</i>	nvarchar
<i>FinanceCompany</i>	nvarchar
<i>FinancialID</i>	int
<i>FinancialType</i>	nvarchar
<i>HardDisk</i>	nvarchar
<i>InvDate</i>	datetime
<i>LeaseLength</i>	int
<i>Location</i>	nvarchar
<i>Memory</i>	nvarchar
<i>OrderReference</i>	nvarchar
<i>Other</i>	nvarchar
<i>PickupLocation</i>	nvarchar
<i>Processor</i>	nvarchar
<i>ProductGroup</i>	nvarchar
<i>ReceivedDate</i>	datetime
<i>SerialNumber</i>	nvarchar
<i>StartDate</i>	datetime
<i>UserName</i>	nvarchar

InvFirewall

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>InvDate</i>	datetime
<i>ProductName</i>	nvarchar
<i>ProductVersion</i>	nvarchar
<i>Server</i>	nvarchar
<i>SettingsName</i>	nvarchar
<i>SettingUpdateTime</i>	datetime

InvKeyboard

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>InvDate</i>	datetime
<i>Layout</i>	nvarchar

InvLastLoggedInUser

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>InvDate</i>	datetime
<i>UserName</i>	nvarchar

InvLocalAdmin

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>InvDate</i>	datetime
<i>Name</i>	nvarchar

InvLogicalDisk

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Caption</i>	nvarchar
<i>Description</i>	nvarchar
<i>DriveType</i>	int
<i>FreeSpace</i>	bigint
<i>FreeSpaceInGB</i>	double
<i>InvDate</i>	datetime
<i>Size</i>	bigint
<i>SizeInGB</i>	double
<i>BitLockerStatus</i>	nvarchar
<i>BitLockerEncryptionMethod</i>	nvarchar
<i>BitLockerLockStatus</i>	nvarchar
<i>BitLockerKeyProtectors</i>	nvarchar
<i>BitLockerHasRecoveryKey</i>	nvarchar

InvLoginHistory

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>UserName</i>	nvarchar
<i>InventoryDate</i>	datetime
<i>InventoryCount</i>	int

InvManagedSoftware

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Description</i>	nvarchar
<i>IdentifiedByARPScan</i>	varchar
<i>IdentifiedByFileScan</i>	varchar
<i>IdentifiedBySWCatalog</i>	varchar
<i>IdentifiedDate</i>	datetime
<i>ARPScanDate</i>	datetime
<i>FileScanDate</i>	datetime
<i>LicenseStatus</i>	varchar
<i>ManagedSoftwareID</i>	int
<i>PackageNames</i>	nvarchar
<i>QixRule</i>	varchar
<i>SWCategory</i>	nvarchar
<i>SWName</i>	nvarchar
<i>UsageFrequency</i>	varchar

InvMonitor

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Description</i>	nvarchar
<i>HardwareID</i>	nvarchar
<i>InvDate</i>	datetime
<i>Model</i>	nvarchar
<i>SerialNumber</i>	nvarchar

InvNetworkAdapter

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Idx</i>	int
<i>InvDate</i>	datetime
<i>MACAddress</i>	nchar
<i>Manufacturer</i>	nchar
<i>Name</i>	nchar
<i>PNPDeviceID</i>	nvarchar
<i>TimeOfLastReset</i>	datetime

 InvNetworkAdapterConfiguration

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>DefaultIPGateway</i>	nchar
<i>DHCPEnabled</i>	nchar
<i>DHCPLeaseExpires</i>	datetime
<i>DHCPLeaseObtained</i>	datetime
<i>DHCPServer</i>	nchar
<i>DNSDomain</i>	nchar
<i>DNSDomainSuffixSearchOrder</i>	nchar
<i>DNSServerSearchOrder</i>	nchar
<i>FullDNSRegistrationEnabled</i>	nchar
<i>Idx</i>	int
<i>InvDate</i>	datetime
<i>IPAddress</i>	nvarchar
<i>IPConnectionMetric</i>	int
<i>IPEnabled</i>	nchar
<i>IPSubnet</i>	nvarchar
<i>SubnetAddressInt</i>	bigint
<i>WINSPrimaryServer</i>	nchar
<i>WINSSecondaryServer</i>	nchar

 InvOperatingSystem

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Caption</i>	nvarchar
<i>ComputerDescription</i>	nvarchar
<i>CSDVersion</i>	nvarchar
<i>FullVersion</i>	nvarchar
<i>InstallDate</i>	datetime
<i>InvDate</i>	datetime
<i>Kernel</i>	nvarchar
<i>LastBootUpTime</i>	datetime
<i>Locale</i>	nvarchar
<i>OperatingSystemSKU</i>	int
<i>OSArchitecture</i>	nvarchar
<i>OSLanguage</i>	int
<i>OtherTypeDescription</i>	nvarchar
<i>ProductType</i>	int
<i>RegisteredUser</i>	nvarchar
<i>ReleaseId</i>	nvarchar
<i>SerialNumber</i>	nvarchar
<i>ServicePackMajorVersion</i>	int
<i>Version</i>	nvarchar
<i>WindowsDirectory</i>	nvarchar

InvPackageManager

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>CategoryName</i>	nvarchar
<i>Description</i>	nvarchar
<i>InvDate</i>	datetime
<i>Name</i>	nvarchar
<i>Version</i>	nvarchar

InvPatchInstalled

<i>Attribute</i>	<i>Type</i>
<i>Approved</i>	bit
<i>AssetID</i>	int
<i>Bulletin</i>	nvarchar
<i>Description</i>	nvarchar
<i>IncludeInQIX</i>	bit
<i>ExtendedStatus</i>	int
<i>InvDate</i>	datetime
<i>KBID</i>	int
<i>LastReportTime</i>	datetime
<i>PatchID</i>	nvarchar
<i>InstalledStatus</i>	int
<i>ReleaseDate</i>	datetime
<i>Severity</i>	int
<i>Title</i>	nvarchar

InvPhysicalMemory

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Capacity</i>	bigint
<i>CapacityInGB</i>	double
<i>DataWidth</i>	int
<i>Description</i>	nvarchar
<i>DeviceLocator</i>	nvarchar
<i>FormFactor</i>	nvarchar
<i>InvDate</i>	datetime
<i>Manufacturer</i>	nvarchar
<i>PartNumber</i>	nvarchar
<i>SerialNumber</i>	nvarchar
<i>Speed</i>	int
<i>TotalWidth</i>	int

InvPortableBattery

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>BatteryStatus</i>	nvarchar
<i>Chemistry</i>	nvarchar
<i>Description</i>	nvarchar
<i>DesignCapacity</i>	double
<i>DesignVoltage</i>	double
<i>DeviceID</i>	nvarchar
<i>ErrorDescription</i>	nvarchar
<i>EstimatedChargeRemaining</i>	int
<i>EstimatedRemainingCapacity</i>	double
<i>EstimatedRunTime</i>	bigint
<i>FullChargeCapacity</i>	double
<i>HealthStatus</i>	nvarchar
<i>InvDate</i>	datetime
<i>LastErrorCode</i>	int
<i>Location</i>	nvarchar
<i>Manufacturer</i>	nvarchar
<i>Name</i>	nvarchar
<i>SerialNumber</i>	nvarchar

InvPowerUsage

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>AvgPowerConsumption</i>	int
<i>Date</i>	datetime
<i>ElectricityCost</i>	money
<i>ElectricityPrice</i>	money
<i>LowPowerSeconds</i>	int
<i>PowerOffSeconds</i>	int
<i>PowerOnLoggedInSeconds</i>	int
<i>PowerOnLoggedOutSeconds</i>	int
<i>PowerUsed</i>	double

InvPrinter

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Color</i>	int
<i>Comment</i>	nvarchar
<i>DriverName</i>	nvarchar
<i>Duplex</i>	bit
<i>FormName</i>	nvarchar
<i>InvDate</i>	datetime
<i>Location</i>	nvarchar
<i>Name</i>	nvarchar
<i>PaperSize</i>	nvarchar
<i>PortName</i>	nvarchar
<i>ServerName</i>	nvarchar
<i>ShareName</i>	nvarchar

InvProcessor

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Architecture</i>	nvarchar
<i>CurrentClockSpeed</i>	int
<i>DataWidth</i>	int
<i>Description</i>	nvarchar
<i>InvDate</i>	datetime
<i>L2CacheSize</i>	int
<i>Manufacturer</i>	nvarchar
<i>MaxClockSpeed</i>	int
<i>Name</i>	nvarchar
<i>NumberOfCores</i>	int
<i>NumberOfLogicalProcessors</i>	int

InvSecurityPatch

<i>Attribute</i>	<i>Type</i>
<i>ApprovalType</i>	nvarchar
<i>AssetGroup</i>	nvarchar
<i>AssetGroupID</i>	int
<i>AssetID</i>	int
<i>InstallInfo</i>	nvarchar
<i>InvDate</i>	datetime
<i>LastStatusChange</i>	datetime
<i>Name</i>	nvarchar
<i>PatchCategory</i>	nvarchar
<i>PatchID</i>	int
<i>PatchProduct</i>	nvarchar
<i>PatchVendor</i>	nvarchar
<i>ReleaseDate</i>	datetime
<i>Status</i>	int
<i>StatusDescription</i>	nvarchar

InvSecurityProduct

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Enabled</i>	bit
<i>InvDate</i>	datetime
<i>Name</i>	nvarchar
<i>Timestamp</i>	datetime
<i>Type</i>	nvarchar
<i>UpToDate</i>	bit

InvSoftwareCatalog

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>CatalogID</i>	int
<i>Category</i>	nvarchar
<i>IsSuite</i>	bit
<i>Manufacturer</i>	nvarchar
<i>Name</i>	nvarchar
<i>Version</i>	nvarchar
<i>LicenseNeeded</i>	nvarchar

InvSystemProfiler

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>GetInfoString</i>	nvarchar
<i>InvDate</i>	datetime
<i>Is64Bit</i>	bit
<i>Kind</i>	nvarchar
<i>Location</i>	nvarchar
<i>ModifyDate</i>	datetime
<i>Name</i>	nvarchar
<i>Version</i>	nvarchar

InvTPM

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>InvDate</i>	datetime
<i>ManufacturerId</i>	bigint
<i>ManufacturerName</i>	nvarchar
<i>SpecVersion</i>	nvarchar
<i>ManufacturerVersion</i>	nvarchar
<i>ManufacturerVersionInfo</i>	nvarchar
<i>PhysicalPresenceVersionInfo</i>	nvarchar
<i>IsEnabled</i>	bit
<i>IsActivated</i>	bit
<i>IsOwned</i>	bit

InvVideoController

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>AdapterRAM</i>	bigint
<i>CurrentBitsPerPixel</i>	int
<i>CurrentHorizontalResolution</i>	int
<i>CurrentNumberOfColors</i>	bigint
<i>CurrentRefreshRate</i>	int
<i>CurrentVerticalResolution</i>	int
<i>Description</i>	nvarchar
<i>DriverDate</i>	nvarchar
<i>DriverVersion</i>	nvarchar
<i>InvDate</i>	datetime
<i>Name</i>	nvarchar

InvWindowsEventLog

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>Category</i>	int
<i>CategoryString</i>	nvarchar
<i>EventCode</i>	int
<i>EventType</i>	nvarchar
<i>InvDate</i>	datetime
<i>Logfile</i>	nvarchar
<i>Message</i>	nvarchar
<i>Param1</i>	nvarchar
<i>Param2</i>	nvarchar
<i>Param3</i>	nvarchar
<i>Param4</i>	nvarchar
<i>RecordNumber</i>	bigint
<i>SourceName</i>	nvarchar
<i>TimeWritten</i>	datetime

InvWindowsStabilityScore

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>InvDate</i>	datetime
<i>Score</i>	double
<i>TimeGenerated</i>	datetime

MiradoreLicense

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>CalculateDate</i>	datetime
<i>RequiresLicense</i>	bit

MiradoreOnlineDevice

<i>Attribute</i>	<i>Type</i>
<i>AssetID</i>	int
<i>DeviceID</i>	int
<i>OnlineSite</i>	nvarchar

AssetConfirmationRequest

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>ApprovedBy</i>	nvarchar		x			
<i>ApprovalDate</i>	datetime		x			
<i>Asset</i>	C	Asset	x	x		
<i>AssetConfirmationRequestID</i>	PK		x			
<i>CreateDate</i>	datetime		x			
<i>DisplayName</i>	DN		x			
<i>Status</i>	S		x			

AssetGroup

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Asset</i>	B	Asset	x	x	x	x
<i>AssetGroupID</i>	PK		x			
<i>Description</i>	nvarchar		x	x	x	x
<i>DisplayName</i>	DN		x			
<i>IsDynamicGroup</i>	bit		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x	x	x	x
<i>Parent</i>	C	AssetGroup	x	x	x	x
<i>Status</i>	S	Status	x	x	x	x

AssetServiceContract

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AdditionalInfo</i>	nvarchar		x	x	x	
<i>Asset</i>	C	Asset	x	x	x	
<i>AssetServiceContractID</i>	PK		x			
<i>Contract</i>	C	Contract	x	x	x	
<i>ContractServiceLevel</i>	S	ContractServiceLevel	x	x	x	
<i>DateAdded</i>	datetime		x			
<i>DisplayName</i>	DN		x			
<i>EndDate</i>	datetime		x	x	x	
<i>IdentificationNumber</i>	nvarchar		x	x	x	
<i>PlannedEndDate</i>	datetime		x	x	x	
<i>StartDate</i>	datetime		x	x	x	
<i>Status</i>	S	Status	x	x	x	

AssetSpecificPackage

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Asset</i>	C	Asset	x	x		
<i>AssetSpecificPackageID</i>	PK		x			
<i>DisplayName</i>	nvarchar		x			
<i>Licensee</i>	nvarchar		x	x	x	
<i>LicenseKey</i>	nvarchar		x	x	x	
<i>Package</i>	C	Package	x	x		
<i>Priority</i>	int		x	x	x	
<i>Reboot</i>	bit		x	x	x	

Comment

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CommentID</i>	PK		x			
<i>CommentText</i>	nvarchar		x			
<i>CommentTime</i>	datetime		x			
<i>DisplayName</i>	DN		x			
<i>ItemID</i>	int		x			
<i>ItemName</i>	nvarchar		x			
<i>Subject</i>	nvarchar		x			
<i>User</i>	nvarchar		x			

CompanyName

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CompanyID</i>	nvarchar		x			
<i>CompanyNameID</i>	PK		x			
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			
<i>TaxID</i>	nvarchar		x			

Connector

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>ConnectorID</i>	PK		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>Identifier</i>	nvarchar		x			
<i>LastStartTime</i>	datetime		x			
<i>LastEndTime</i>	datetime		x			
<i>LastResult</i>	nvarchar		x			
<i>Name</i>	nvarchar		x			
<i>Status</i>	nvarchar		x			
<i>Type</i>	nvarchar		x			
<i>Version</i>	nvarchar		x			

Contract

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>ContractID</i>	PK		x			
<i>ContractNumber</i>	nvarchar		x			
<i>Customer</i>	C	Organisation	x			
<i>CustomerContact</i>	C	User	x			
<i>CreateDate</i>	datetime		x			
<i>Default</i>	bit		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>EndDate</i>	datetime		x			
<i>Manager</i>	C	User	x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			
<i>Parent</i>	C	Contract	x			
<i>StartDate</i>	datetime		x			
<i>Status</i>	int		x			
<i>SubType</i>	int		x			
<i>Supplier</i>	C	Supplier	x			
<i>SupplierContact</i>	nvarchar		x			
<i>Type</i>	nvarchar		x			

ContractServiceLevel

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Contract</i>	C	Contract	x			
<i>ContractServiceLevelID</i>	PK		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>IsDefault</i>	bit		x			
<i>Name</i>	nvarchar		x			
<i>OrderNumber</i>	int		x			

CostCenter

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CostCenterID</i>	PK		x			
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>FullName</i>	T		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			
<i>Parent</i>	C	CostCenter	x			
<i>ResponsiblePerson</i>	C	User	x			
<i>Status</i>	nvarchar		x			

DetailedStatus

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Description</i>	nvarchar		x			
<i>DetailedStatus</i>	nvarchar		x			
<i>DetailedStatusID</i>	PK		x			
<i>DisplayName</i>	DN		x			
<i>IsDefault</i>	bit		x			
<i>StatusGroup</i>	int		x			
<i>StatusGroupDescription</i>	nvarchar		x			
<i>StatusValue</i>	int		x			
<i>StatusDescription</i>	nvarchar		x			

DeviceRole

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AutomatedInstallation</i>	bit		x			
<i>CreateDate</i>	datetime		x			
<i>DefaultOS</i>	C	OS	x			
<i>Description</i>	nvarchar		x			
<i>DeviceRoleID</i>	PK		x			
<i>DisplayName</i>	DN		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			
<i>ShortName</i>	nvarchar		x			

DeviceUsage

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x			
<i>DeviceRole</i>	C	DeviceRole	x			
<i>DeviceUsageID</i>	PK		x			
<i>DisplayName</i>	DN		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			

Distribution

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Asset</i>	C	Asset	x	x		
<i>ClientVersion</i>	nvarchar		x			
<i>CreateTime</i>	datetime		x			
<i>DisplayName</i>	DN		x			
<i>DistributionID</i>	PK		x			
<i>EndTime</i>	datetime		x			
<i>ErrorCode</i>	int		x			
<i>GroupDistribution</i>	C	GroupDistribution	x			
<i>InactivateTime</i>	datetime		x			
<i>IsPreviousInstallation</i>	bit		x			
<i>Package</i>	C	Package	x	x		
<i>PackageRunAs</i>	nvarchar		x			
<i>PackageStatus</i>	nvarchar		x			
<i>PackageType</i>	S	Types	x	x		
<i>Reboot</i>	bit		x	x		
<i>RetryCount</i>	int		x			
<i>StartTime</i>	datetime		x			
<i>Status</i>	nvarchar		x			
<i>User</i>	nvarchar		x	x		
<i>Type</i>	nvarchar		x			
<i>WakeUpClient</i>	bit				x	

EventLogEvent

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>EventID</i>	PK		x			
<i>EventAction</i>	nvarchar		x			
<i>EventLevel</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>Description</i>	nvarchar		x			
<i>UserID</i>	nvarchar		x			
<i>UserName</i>	nvarchar		x			
<i>Source</i>	nvarchar		x			
<i>EventTime</i>	datetime		x			
<i>Category</i>	nvarchar		x			
<i>EventTarget</i>	nvarchar		x			
<i>EventTargetID</i>	nvarchar		x			
<i>EventTarget2</i>	nvarchar		x			
<i>EventTargetID2</i>	nvarchar		x			
<i>EventTarget3</i>	nvarchar		x			
<i>EventTargetID3</i>	nvarchar		x			
<i>EventTargetName</i>	nvarchar		x			

EventMonitoringEvent

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Created</i>	datetime		x			
<i>EventClass</i>	nvarchar		x			
<i>EventCount</i>	int		x			
<i>EventID</i>	int, PK		x			
<i>EventName</i>	nvarchar		x			
<i>ItemID</i>	int		x			
<i>ItemType</i>	nvarchar		x			
<i>Severity</i>	nvarchar		x			
<i>Source</i>	nvarchar		x			
<i>SourceHost</i>	nvarchar		x			
<i>Status</i>	nvarchar		x			
<i>StatusUpdatedBy</i>	nvarchar		x			
<i>Subject</i>	nvarchar		x			
<i>Updated</i>	datetime		x			
<i>Value</i>	nvarchar		x			

FinancialTransaction

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AdditionalInfo</i>	nvarchar		x			
<i>Asset</i>	C	Asset	x			
<i>BillToParty</i>	nvarchar		x			
<i>CreateDate</i>	datetime		x			
<i>Currency</i>	nvarchar		x			
<i>Date</i>	datetime		x			
<i>DispatchNumber</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>EndDate</i>	datetime		x			
<i>EndingOption</i>	S	Types	x			
<i>FinancialTransactionID</i>	PK		x			
<i>InvoiceNumber</i>	nvarchar		x			
<i>ItemPrice</i>	money		x			
<i>LeaseContract</i>	C	Contract	x			
<i>LeaseTime</i>	nvarchar		x			
<i>ModifyDate</i>	datetime		x			
<i>OrderNumber</i>	nvarchar		x			
<i>PickupLocation</i>	nvarchar		x			
<i>StartDate</i>	datetime		x			
<i>SupplyContract</i>	C	Contract	x			
<i>Type</i>	S	Types	x			

GroupDistribution

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>GroupDistributionID</i>	PK		x			
<i>Package</i>	C	Package	x	x		
<i>DisplayName</i>	DN		x			
<i>StatusText</i>	nvarchar		x			
<i>Status</i>	int		x			
<i>DistributionType</i>	S	Types	x	x		
<i>CreatedBy</i>	nvarchar		x	x		
<i>Approver</i>	nvarchar		x			
<i>Description</i>	nvarchar		x	x		
<i>StartTime</i>	datetime		x	x	x	
<i>EndTime</i>	datetime		x	x	x	
<i>Reboot</i>	bit		x	x	x	
<i>AddToDevice</i>	bit		x			
<i>PackageTimeout</i>	int		x	x	x	
<i>KnownLocationsOnly</i>	bit		x	x	x	
<i>UseWakeOnLAN</i>	int		x	x	x	
<i>DefineAllowedTimeFrame</i>	bit		x	x	x	
<i>RunWeekDays</i>	nvarchar		x	x	x	
<i>AllowedTimeStart</i>	int		x	x	x	
<i>AllowedTimeEnd</i>	int		x	x	x	
<i>AllowedTimeTZOption</i>	int		x	x	x	
<i>EnableRetries</i>	bit		x	x	x	
<i>NumberOfRetries</i>	int		x	x	x	
<i>RetryDelay</i>	int		x	x	x	
<i>RetryExceptionCodes</i>	nvarchar		x	x	x	
<i>EnablePostpone</i>	bit		x	x	x	
<i>PostponeTimeout</i>	int		x	x	x	
<i>NumberOfPostpones</i>	int		x	x	x	
<i>GroupDistributionTarget</i>	B	GroupDistributionTarget	x	x		

GroupDistributionTarget

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>GroupDistributionTargetID</i>	PK		x			
<i>DisplayName</i>	DN		x			
<i>StatusText</i>	nvarchar		x			
<i>Status</i>	int		x			
<i>TimeAdded</i>	datetime		x			
<i>Asset</i>	C	Asset	x			
<i>GroupDistribution</i>	C	GroupDistribution	x			
<i>Distribution</i>	C	Distribution	x			

HistorySeries

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AggregateOrder</i>	nvarchar		x			
<i>Attribute</i>	nvarchar		x			
<i>DisplayFilters</i>	nvarchar		x			
<i>DisplayName</i>	nvarchar		x			
<i>Filters</i>	nvarchar		x			
<i>Function</i>	nvarchar		x			
<i>GroupByAttribute</i>	nvarchar		x			
<i>GroybByOrder</i>	nvarchar		x			
<i>HistorySeriesID</i>	int, PK		x			
<i>IsBuiltIn</i>	bit		x			
<i>Item</i>	nvarchar		x			
<i>Name</i>	nvarchar		x			
<i>NumerOfSeries</i>	int		x			
<i>ShowRestAsOther</i>	bit		x			
<i>Type</i>	nvarchar		x			
<i>HistoryData</i>	E	HistoryData	x			

HistorySeries extensions

HistoryData

<i>Attribute</i>	<i>Type</i>
<i>Date</i>	datetime
<i>DayOfWeek</i>	int
<i>DayOfMonth</i>	int
<i>HistorySeriesID</i>	int
<i>MonthName</i>	nvarchar
<i>Month</i>	int
<i>Quarter</i>	int
<i>Timestamp</i>	datetime
<i>Value</i>	double
<i>Week</i>	int

HWCategory

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>ActivationRequired</i>	bit		x			
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>HWCategory</i>	PK		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			
<i>PowerConsumption</i>	int		x			
<i>ShortName</i>	nvarchar		x			
<i>SNMPScan</i>	bit		x			
<i>Type</i>	nvarchar		x			

HWInventoryChangeLog

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AssetID</i>	int		x			
<i>AssetName</i>	nvarchar		x			
<i>DetailedLocation</i>	nvarchar		x			
<i>DeviceName</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>HWCategory</i>	nvarchar		x			
<i>HWName</i>	nvarchar		x			
<i>InventoryChangeLogID</i>	PK		x			
<i>LastSeen</i>	datetime		x			
<i>Location</i>	nvarchar		x			
<i>NewValue</i>	nvarchar		x			
<i>OldValue</i>	nvarchar		x			
<i>OnlineStatus</i>	nvarchar		x			
<i>Organisation</i>	nvarchar		x			
<i>Role</i>	nvarchar		x			
<i>Status</i>	nvarchar		x			
<i>TableDisplayName</i>	nvarchar		x			
<i>Type</i>	varchar		x			
<i>UpdateDate</i>	datetime		x			
<i>UpdateMonth</i>	nvarchar		x			
<i>UserFullName</i>	nvarchar		x			

HWManufacturer

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>ID</i>	nvarchar		x			
<i>HWManufacturerID</i>	PK		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			

Incident

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Asset</i>	C	Asset	x	x	x	
<i>Assignment</i>	B	IncidentAssignment	x	x	x	
<i>ClosureDate</i>	datetime		x			
<i>ClosureCode</i>	int		x	x	x	
<i>ClosureBriefDesc</i>	nvarchar		x	x	x	
<i>ClosureLongDesc</i>	nvarchar		x	x	x	
<i>Contact</i>	C	User	x	x	x	
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x	x	x	
<i>DetailedStatus</i>	S	DetailedStatus	x	x	x	
<i>DisplayName</i>	DN		x			
<i>Impact</i>	S	Types	x	x	x	
<i>IncidentCategory</i>	C	IncidentCategory	x	x	x	
<i>IncidentGUID</i>	guid		x			
<i>IncidentID</i>	PK		x			
<i>ID</i>	nvarchar		x			
<i>IncidentType</i>	S	Types	x	x	x	
<i>Location</i>	C	Location	x	x	x	
<i>Owner</i>	C	User	x	x	x	
<i>ReportedBy</i>	C	User	x	x	x	
<i>SendEmail</i>	bit			x	x	
<i>SourceType</i>	S	Types	x	x	x	
<i>Status</i>	S	Status	x	x	x	
<i>StatusChangeReason</i>	S	StatusChangeReason	x	x	x	
<i>Subject</i>	nvarchar		x	x	x	
<i>SupportMethod</i>	int		x	x	x	
<i>UpdateDate</i>	datetime		x			
<i>Urgency</i>	S	Types	x	x	x	
<i>User</i>	C	User	x	x	x	

IncidentAssignment

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Assignee</i>	C	User	x	x		
<i>CreateDate</i>	datetime		x			
<i>DisplayName</i>	DN		x			
<i>IncidentAssignmentCompany</i>	int		x	x		
<i>IncidentAssignmentID</i>	PK		x			
<i>Incident</i>	C	Incident	x	x		
<i>Supplier</i>	C	Supplier	x	x		

IncidentCategory

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>IncidentCategoryID</i>	PK		x			
<i>IsBuiltin</i>	bit		x			
<i>Name</i>	nvarchar		x			
<i>Parent</i>	C	IncidentCategory	x			
<i>SubCategoryRequired</i>	int		x			

InitialInstall

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Asset</i>	C	Asset	x	x		
<i>ActivatedBy</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>EndTime</i>	datetime		x			
<i>InitialInstallID</i>	PK		x			
<i>StartTime</i>	datetime		x			
<i>Status</i>	nvarchar		x			

LicenseAllocation

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AllocationDate</i>	datetime		x			
<i>AllocationType</i>	S	Types	x			
<i>AllocatedBy</i>	C	User	x			
<i>Amount</i>	int		x	x	x	
<i>Asset</i>	C	Asset	x	x	x	
<i>Description</i>	nvarchar		x	x	x	
<i>DisplayName</i>	DN		x			
<i>LicenseAllocationID</i>	PK		x			
<i>LicensePool</i>	C	LicensePool	x	x	x	
<i>Status</i>	S	Status	x	x	x	
<i>UnallocationDate</i>	datetime		x			
<i>UnallocationType</i>	int		x			
<i>UnallocatedBy</i>	C	User	x			
<i>User</i>	C	User	x	x	x	

LicenseAutoAllocRule

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>LicenseAutoAllocRuleID</i>	PK		x			
<i>Name</i>	nvarchar		x			

LicensePool

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AutomaticAllocation</i>	bit		x			
<i>AutoAllocPriority</i>	int		x			
<i>AutoAllocRule</i>	C	LicenseAutoAllocRule	x			
<i>Contract</i>	B	Contract	x	x	x	x
<i>DisplayName</i>	DN		x			
<i>FormAllocEnabled</i>	bit		x			
<i>ID</i>	nvarchar		x			
<i>Info</i>	nvarchar		x	x	x	
<i>LicenseAgreement</i>	C	Contract	x	x	x	
<i>LicenseAllocation</i>	B	LicenseAllocation	x	x	x	x
<i>LicensedBy</i>	S	Types	x	x	x	
<i>LicenseGUID</i>	guid		x			
<i>LicenseModel</i>	S	Types	x	x	x	
<i>LicensePoolID</i>	PK		x			
<i>LicenseTransaction</i>	B	LicenseTransaction	x	x	x	x
<i>ManagedSoftware</i>	B	ManagedSoftware	x	x	x	
<i>MinAllocDays</i>	int		x	x	x	
<i>Name</i>	nvarchar		x	x	x	
<i>Owner</i>	C	User	x	x	x	
<i>Purchaser</i>	C	User	x	x	x	
<i>Status</i>	S	Status	x	x	x	
<i>TechnicalResponsible</i>	C	User	x	x	x	

LicenseTransaction

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Action</i>	S	Types	x	x	x	
<i>AdditionalInfo</i>	nvarchar		x	x	x	
<i>Amount</i>	int		x	x	x	
<i>CreateDate</i>	datetime		x			
<i>Currency</i>	nvarchar		x	x	x	
<i>DisplayName</i>	DN		x			
<i>EndDate</i>	datetime		x	x	x	
<i>IsUnlimited</i>	bit		x	x	x	
<i>License</i>	C	LicensePool	x	x	x	
<i>LicenseTransactionID</i>	PK		x			
<i>ID</i>	nvarchar		x			
<i>Name</i>	nvarchar		x	x	x	
<i>OrderNumber</i>	int		x	x	x	
<i>Organisation</i>	C	Organisation	x	x	x	
<i>Price</i>	money		x	x	x	
<i>StartDate</i>	datetime		x	x	x	
<i>Status</i>	S	Status	x			
<i>ReassignTarget</i>	C	LicensePool	x	x	x	
<i>User</i>	C	User	x	x	x	
<i>Vendor</i>	C	Vendor	x	x	x	

Location

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Administrator</i>	nvarchar		x	x	x	x
<i>BandwidthLimit</i>	int		x	x	x	x
<i>ConcurrentInstallationsLimit</i>	int		x	x	x	x
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x	x	x	x
<i>DHCP</i>	bit		x	x	x	
<i>DisplayName</i>	DN		x			
<i>ElectricityPrice</i>	money		x			
<i>FullName</i>	T		x			
<i>LocationID</i>	PK		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x	x	x	
<i>Parent</i>	C	Location	x	x	x	x
<i>SelectableForAsset</i>	bit (true/false)		x	x	x	
<i>SelectableForUser</i>	bit (true/false)		x	x	x	
<i>ShortName</i>	nvarchar		x	x	x	x
<i>Status</i>	Status		x	x	x	
<i>Address</i>	E	Address	x			

Location extensions

Address

<i>Attribute</i>	<i>Type</i>
<i>LocationID</i>	int
<i>Address1</i>	nvarchar
<i>Address2</i>	nvarchar
<i>AddressID</i>	PK
<i>Type</i>	nvarchar
<i>City</i>	nvarchar
<i>Country</i>	nvarchar
<i>Description</i>	nvarchar
<i>State</i>	nvarchar
<i>ZipCode</i>	nvarchar

ManagedSoftware

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Asset</i>	C	Asset configuration item	x			
<i>DefInstFolder</i>	nvarchar		x			
<i>Description</i>	nvarchar		x	x	x	
<i>DisplayName</i>	DN		x			
<i>DistributionLicenseRule</i>	int		x	x	x	
<i>EnablePackageBuilding</i>	bit		x	x	x	
<i>Language</i>	nvarchar		x	x	x	
<i>LicenseKey</i>	nvarchar		x	x	x	
<i>LicenseMessage</i>	nvarchar		x	x	x	
<i>LicenseNeeded</i>	bit		x	x	x	
<i>ManagedSoftwareID</i>	PK		x			
<i>Manufacturer</i>	C	SWManufacturer	x	x	x	
<i>MeasureUsage</i>	bit		x			
<i>MediaFolder</i>	nvarchar		x	x	x	
<i>Name</i>	nvarchar		x	x	x	
<i>OSCategory</i>	S	OSCategory	x	x	x	
<i>Owner</i>	C	User	x	x	x	
<i>QixRule</i>	int		x	x	x	
<i>Status</i>	S	Status	x	x	x	
<i>SWCategory</i>	C	SWCategory	x	x	x	
<i>Version</i>	nvarchar		x	x	x	

Possible values for DistributionLicenseRule: 0 = Allowed, 1 = Allow with message and 2 Deny with message.

Model

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>HWCATEGORY</i>	C	HWCATEGORY	x			
<i>Manufacturer</i>	C	HWManufacturer	x			
<i>ModelID</i>	PK		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			
<i>PowerUsage</i>	int		x			
<i>Status</i>	nvarchar		x			
<i>Type</i>	nvarchar		x			
<i>Warranty</i>	int		x			

Organisation

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>FullName</i>	T		x			
<i>LDAP</i>	nvarchar		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			
<i>OfflineDomainJoinConnector</i>	C		x			
<i>OrganisationID</i>	PK		x			
<i>Parent</i>	C	Organisation	x			
<i>ShortName</i>	nvarchar		x			
<i>Status</i>	nvarchar		x			

OS

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>InitialInstallationEnabled</i>	bit		x			
<i>Language</i>	nvarchar		x			
<i>Manufacturer</i>	nvarchar		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			
<i>OSCategory</i>	nvarchar		x			
<i>OSID</i>	PK		x			
<i>Platform</i>	nvarchar		x			
<i>Status</i>	nvarchar		x			
<i>Version</i>	nvarchar		x			

OSCategory

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>DisplayName</i>	DN		x			
<i>Name</i>	nvarchar		x			
<i>OSCategoryID</i>	PK		x			

Package

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AllowHelpdesk</i>	<i>bit</i>		x			
<i>Approver</i>	<i>nvarchar</i>		x			
<i>BindToAsset</i>	<i>bit</i>		x			
<i>Builder</i>	<i>nvarchar</i>		x			
<i>Description</i>	<i>nvarchar</i>		x			
<i>DisplayName</i>	<i>DN</i>		x			
<i>FreeSpace</i>	<i>int</i>		x			
<i>FullName</i>	<i>nvarchar</i>		x			
<i>GroupDistributable</i>	<i>bit</i>		x			
<i>HideMessages</i>	<i>bit</i>		x			
<i>InitialInstallOnly</i>	<i>bit</i>		x			
<i>InstallationDrive</i>	<i>nvarchar</i>		x			
<i>InteractWithDesktop</i>	<i>bit</i>		x			
<i>ManagedSoftware</i>	<i>C</i>	<i>ManagedSoftware</i>	x			
<i>MapInstPoint</i>	<i>bit</i>		x			
<i>MediaFolder</i>	<i>nvarchar</i>		x			
<i>OptionalSpecification</i>	<i>nvarchar</i>		x			
<i>Owner</i>	<i>nvarchar</i>		x			
<i>PackageID</i>	<i>PK</i>		x			
<i>PackageTimeout</i>	<i>int</i>		x			
<i>PackageType</i>	<i>int</i>		x			
<i>Platform</i>	<i>int</i>		x			
<i>RunAs</i>	<i>nvarchar</i>		x			
<i>ShowInSSP</i>	<i>bit</i>		x			
<i>SSPAproval</i>	<i>bit</i>		x			
<i>SSPAproverInfo</i>	<i>nvarchar</i>		x			
<i>SSPDisplayName</i>	<i>nvarchar</i>		x			
<i>SSPUserInfo</i>	<i>nvarchar</i>		x			
<i>Status</i>	<i>nvarchar</i>		x			
<i>Version</i>	<i>nvarchar</i>		x			

Patch

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Cached</i>	bit		x			
<i>ContentSize</i>	int		x			
<i>CVECode</i>	nvarchar		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	nvarchar		x			
<i>DownloadDate</i>	datetime		x			
<i>Hidden</i>	bit		x		x	
<i>Identifier</i>	nvarchar		x			
<i>ImportedDate</i>	datetime		x			
<i>Info</i>	nvarchar		x			
<i>IsCritical</i>	bit		x			
<i>IsSecurity</i>	bit		x			
<i>LastInstalled</i>	datetime		x			
<i>MD5</i>	nvarchar		x			
<i>Name</i>	nvarchar		x			
<i>PatchApproval</i>	B	PatchApproval	x			
<i>PatchCategory</i>	C	PatchCategory	x			
<i>PatchGUID</i>	uniqueid		x			
<i>PatchID</i>	PK		x			
<i>PatchProduct</i>	C	PatchProduct	x			
<i>PatchStatus</i>	C	PatchStatus	x			
<i>PatchVendor</i>	C	PatchVendor	x			
<i>Platform</i>	nvarchar		x			
<i>PlsFileName</i>	nvarchar		x			
<i>ReleaseDate</i>	datetime		x			
<i>Uninstall</i>	bit		x			

PatchApproval

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Action</i>	S	Types	x	x	x	
<i>ActivateTime</i>	datetime		x	x	x	x
<i>AssetGroup</i>	C	AssetGroup	x	x	x	
<i>CreateDate</i>	datetime		x			
<i>DisplayName</i>	nvarchar		x			
<i>ModifyDate</i>	datetime		x			
<i>Patch</i>	C	Patch	x	x	x	
<i>PatchApprovalID</i>	PK		x			

PatchCategory

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>DisplayName</i>	nvarchar		x			
<i>Hidden</i>	bit		x		x	
<i>Name</i>	nvarchar		x			
<i>PatchCategoryID</i>	int					

PatchInstallationStatus

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Asset</i>	C	Asset	x			
<i>Displayname</i>	dn		x			
<i>Installed</i>	bit		x			
<i>Patch</i>	C	Patch	x			
<i>PatchInstallationStatusID</i>	hexadecimal		x			
<i>PatchStatus</i>	nvarchar		x			
<i>RetryCount</i>	int		x			

PatchProduct

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Name</i>	nvarchar		x			
<i>PatchProductID</i>	int		x			

PatchStatusSummary

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>DisplayName</i>	nvarchar		x			
<i>IsSuperceded</i>	bit		x			
<i>Patch</i>	C	Patch	x			
<i>PatchID</i>	int		x			
<i>TotalAssets</i>	int		x			
<i>TotalFailed</i>	int		x			
<i>TotalInstalled</i>	int		x			
<i>TotalInstalledBySuperceding</i>	int		x			
<i>TotalInstalledPendingReboot</i>	int		x			
<i>TotalNotInstalled</i>	int		x			
<i>TotalUninstalledPendingReboot</i>	int		x			
<i>TotalUninstallfailed</i>	int		x			
<i>TotalUnknown</i>	int		x			

PatchVendor

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>DisplayName</i>	nvarchar		x			
<i>Hidden</i>	bit		x		x	
<i>Name</i>	nvarchar		x			
<i>PatchVendorID</i>	int		x			

Proxy

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AuthenticationMethod</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>HostName</i>	nvarchar		x			
<i>Port</i>	int		x			
<i>ProxyID</i>	int		x			

QixReport

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CalculationDate</i>	datetime		x			
<i>CurrentState</i>	nvarchar		x			
<i>Description</i>	nvarchar		x			
<i>IndicatorsTotal</i>	int		x			
<i>ItemCount</i>	int		x			
<i>Name</i>	nvarchar		x			
<i>QixID</i>	PK		x			
<i>QixPercent</i>	double		x			
<i>QualityManager</i>	C	User	x			
<i>ShowOnDashboard</i>	nvarchar		x			
<i>Status</i>	nvarchar		x			
<i>TechnicalResponsible</i>	C	User	x			
<i>ThresholdCritical</i>	double		x			
<i>ThresholdWarning</i>	double		x			
<i>UpdateDate</i>	datetime		x			

RegionalSetting

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>DisplayName</i>	DN		x			
<i>LanguageCode</i>	nvarchar		x			
<i>LanguageName</i>	nvarchar		x			
<i>RegLangID</i>	PK		x			

ScheduledTask

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>DisplayName</i>	DN		x			
<i>InternalName</i>	nvarchar		x			
<i>OSCategory</i>	nvarchar		x			
<i>ScheduledTaskID</i>	PK		x			

ServerNetworkConfiguration

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>ClientListenPort</i>	int		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>IgnoreSSErrors</i>	bit		x			
<i>IsDefault</i>	bit		x			
<i>Name</i>	nvarchar		x			
<i>Priority</i>	nvarchar		x			
<i>Proxy</i>	C	Proxy	x			
<i>ServerHostName</i>	nvarchar		x			
<i>ServerHttpPort</i>	int		x			
<i>ServerNetworkConfigurationID</i>	int		x			
<i>SocketTimeout</i>	int		x			
<i>Status</i>	nvarchar		x			
<i>UseHttps</i>	bit		x			

SoftwareCatalog

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Asset</i>	C	Asset configuration item	x			
<i>CatalogID</i>	PK		x			
<i>Category</i>	nvarchar		x			
<i>CategoryID</i>	int		x			
<i>DisplayName</i>	DN		x			
<i>IsSuite</i>	bit		x			
<i>LicenseNeeded</i>	nvarchar		x			
<i>Manufacturer</i>	nvarchar		x			
<i>Name</i>	nvarchar		x			
<i>Version</i>	nvarchar		x			

SSPEnrollment

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>ActivationTime</i>	datetime		x	x	x	
<i>Asset</i>	C	Asset	x			
<i>CreatedBy</i>	nvarchar		x			
<i>DeviceRole</i>	C	DeviceRole	x	x	x	
<i>DeviceUsage</i>	C	DeviceUsage	x	x	x	
<i>DisplayName</i>	DN		x			
<i>EnrollmentCode</i>	nvarchar		x			
<i>EnrollmentUrl</i>	nvarchar		x			
<i>OfflineDomainJoinConnector</i>	C	Connector	x	x	x	
<i>ServerNetworkConfiguration</i>	C	ServerNetworkConfiguration	x	x	x	
<i>SSPEnrollmentID</i>	int		x			
<i>Status</i>	nvarchar		x			
<i>User</i>	C	User	x	x	x	

Status

Status is a list of available status values of configuration items. This is used by items which have *selection* attribute which target item is *Status*. *Status.GroupDescription* defines for what configuration item current status applies. Notice that these group names are used internally by Miradore and doesn't always match the names used by web service. *Status.ValueDescription* contains the value that should be used with configuration items.

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>ChangeReasonRequired</i>	int		x			
<i>DetailedStatusRequired</i>	int		x			
<i>DisplayName</i>	DN		x			
<i>Group</i>	int		x			
<i>GroupDescription</i>	nvarchar		x			
<i>StatusID</i>	PK		x			
<i>Value</i>	int		x			
<i>ValueDescription</i>	nvarchar		x			

StatusChangeReason

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>IsDefault</i>	bit		x			
<i>StatusChangeReason</i>	nvarchar		x			
<i>StatusChangeReasonID</i>	PK		x			
<i>StatusGroup</i>	int		x			
<i>StatusGroupDescription</i>	nvarchar		x			
<i>StatusValue</i>	int		x			
<i>StatusDescription</i>	nvarchar		x			

Subnet

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Address</i>	nvarchar		x			
<i>ClientWakeUpIPAddress</i>	nvarchar		x			
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>Gateway</i>	nvarchar		x			
<i>Location</i>	C	Location	x			
<i>ModifyDate</i>	datetime		x			
<i>NetScannerEnabled</i>	bit		x			
<i>PublicIP</i>	B		x			
<i>Status</i>	nvarchar		x			
<i>SubnetID</i>	PK		x			
<i>SubnetMask</i>	nvarchar		x			

SubnetPublicIP

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>DisplayName</i>	DN		x			
<i>PublicIP</i>	nvarchar		x			
<i>PublicIPID</i>	PK		x			
<i>SubnetID</i>	int		x			

Supplier

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CreateDate</i>	datetime		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>ID</i>	nvarchar		x			
<i>ModifyDate</i>	datetime		x			
<i>Name</i>	nvarchar		x			
<i>SupplierID</i>	PK		x			

SWCategory

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>IsBuiltin</i>	bit		x			
<i>Name</i>	nvarchar		x			
<i>Parent</i>	C	SWCategory	x			
<i>SWCategoryID</i>	PK		x			

SWInventoryChangeLog

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>AssetID</i>	int		x			
<i>AssetName</i>	nvarchar		x			
<i>DetailedLocation</i>	nvarchar		x			
<i>DeviceName</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>HWCategory</i>	nvarchar		x			
<i>HWName</i>	nvarchar		x			
<i>InventoryChangeLogID</i>	PK		x			
<i>LastSeen</i>	datetime		x			
<i>Location</i>	nvarchar		x			
<i>NewValue</i>	nvarchar		x			
<i>OldValue</i>	nvarchar		x			
<i>OnlineStatus</i>	nvarchar		x			
<i>Organisation</i>	nvarchar		x			
<i>Role</i>	nvarchar		x			
<i>Status</i>	nvarchar		x			
<i>TableDisplayName</i>	nvarchar		x			
<i>Type</i>	varchar		x			
<i>UpdateDate</i>	datetime		x			
<i>UpdateMonth</i>	nvarchar		x			
<i>UserFullName</i>	nvarchar		x			

SWManufacturer

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	DN		x			
<i>Name</i>	nvarchar		x			
<i>SWManufacturerID</i>	PK		x			
<i>ID</i>	nvarchar		x			

SystemTask

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Category</i>	nvarchar		x			
<i>Description</i>	nvarchar		x			
<i>DisplayName</i>	nvarchar		x			
<i>Enabled</i>	bit		x			
<i>LastRunResult</i>	nvarchar		x			
<i>Name</i>	nvarchar		x			
<i>Schedule</i>	nvarchar		x			
<i>Status</i>	nvarchar		x			
<i>SystemTaskID</i>	int		x			
<i>SystemTaskResult</i>	E	SystemTaskResult	x			

System Task extension

SystemTaskResult

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Endtime</i>	datetime		x			
<i>Result</i>	nvarchar		x			
<i>ReturnCode</i>	int		x			
<i>StartTime</i>	datetime		x			
<i>SystemTaskID</i>	int		x			

Timezone

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>DisplayName</i>	DN		x			
<i>OSCategory</i>	S		x			
<i>Name</i>	nvarchar		x			
<i>TimezoneID</i>	PK		x			

Types

Types contains list of configuration item types. This is used by items which has *selection* attribute which target item is *Types*. *Types.GroupDescription* defines for what configuration item current type applies. Notice that these group names are used internally by Miradore and doesn't always match the names used by web service. *Types.Description* contains the value that should be used with configuration items.

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>DisplayName</i>	DN		x			
<i>Description</i>	nvarchar		x			
<i>Group</i>	int		x			
<i>GroupDescription</i>	nvarchar		x			
<i>TypeID</i>	PK		x			
<i>Value</i>	int		x			

User

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>CostCenter</i>	C	CostCenter	x	x	x	x
<i>CreateDate</i>	datetime		x			
<i>DisplayName</i>	DN		x			
<i>Email</i>	nvarchar		x	x	x	x
<i>EmployeeID</i>	nvarchar		x	x	x	x
<i>FirstName</i>	nvarchar		x	x	x	x
<i>LastName</i>	nvarchar		x	x	x	x
<i>Location</i>	C	Location	x	x	x	
<i>MiddleName</i>	nvarchar		x	x	x	x
<i>ModifyDate</i>	datetime		x			
<i>Organisation</i>	C	Organisation	x	x	x	
<i>PhoneNumber</i>	nvarchar		x	x	x	x
<i>Status</i>	nvarchar		x	x	x	
<i>Supervisor</i>	C	User	x	x	x	x
<i>Type</i>	S	Types	x			
<i>UserID</i>	PK		x			
<i>WindowsUID</i>	nvarchar		x	x	x	
<i>InvADUserGroup</i>	E	InvADUserGroup	x			

User extensions

InvADUserGroup

<i>Attribute</i>	<i>Type</i>
<i>UserID</i>	int
<i>Description</i>	nvarchar
<i>Domain</i>	nvarchar
<i>Email</i>	nvarchar
<i>InvDate</i>	datetime
<i>LDAPPath</i>	nvarchar
<i>Name</i>	nvarchar
<i>Scope</i>	int
<i>Type</i>	int

Vendor

<i>Attribute</i>	<i>Type</i>	<i>Target item</i>	<i>G</i>	<i>C</i>	<i>U</i>	<i>R</i>
<i>Description</i>	nvarchar		x	x	x	
<i>DisplayName</i>	DN		x			
<i>Name</i>	nvarchar		x	x	x	
<i>VendorID</i>	PK		x			
<i>ID</i>	nvarchar		x	x	x	