

KIX Professional InventorySync Backend "VMWare" Installation and configuration

c.a.p.e. IT® GmbH

Version: 17.0.0

Imprint

Title:	KIX Professional InventorySync Backend "VMWare"
Topic:	Installation and configuration
Author:	c.a.p.e. IT® GmbH
Keywords:	KIX, Inventory, VMWare, vSphere, KIX Professional
Comments:	
Filename:	InventorySyncVMWareBackend_installation_and_configuration_en.odt
Version:	17.0.0
Set and Layout:	LibreOffice.org Version 5.2.x

Table of contents

1 General advices.....	4
1.1 Document information.....	4
1.1.1 Objectives.....	4
1.2 Changes.....	4
1.2.1 Change history.....	4
1.2.2 Authors.....	4
1.2.3 Document owner.....	4
1.2.4 Authorized staff.....	4
1.3 List of abbreviations.....	4
2 Installation.....	5
2.1 Requirements.....	5
2.2 Package Installation.....	5
3 Basics.....	6
4 Configuration.....	7
4.1 SysConfig Options.....	7
4.1.1 Sources.....	7
4.1.2 Source-Backend-Mapping.....	7
4.1.3 Source Parameters.....	7
4.1.4 Alternative ObjectType mapping.....	8
4.1.5 ObjectType fallback.....	8
4.1.6 SSL verification.....	8
4.2 InventoryContentXPath for backend "VMWare".....	8
4.3 Example CI classes for backend "VMWare".....	13

1 General advices

1.1 Document information

1.1.1 Objectives

This document contains all the information to install and configure the KIX Professional InventorySync backend "VMWare".

1.2 Changes

1.2.1 Change history

Version	Date	Changed chapters	Short description	Changed by
1.0.0	26.11.15	all	document creation	Rene Boehm
1.0.0	03.12.15	all	fixed some typos and added some info	Rene Boehm
1.0.0	18.01.16	4.2	Added some more attributes	Rene Boehm
1.0.0	01.02.16	2.1	Some more information	Rene Boehm
1.0.0	03.02.16	2.1	Some more information	Rene Boehm
1.0.0	08.02.16	4.2	Added attribute isTemplate	Rene Boehm
1.0.1	08.06.16	4.2	Added attributes writeThrough and split	Rene Boehm
1.0.2	30.11.16	4.1	Fixed typos	Rene Boehm
17.0.0	05.02.17	All	Updated for KIX Professional 17.0.0	Rene Boehm

1.2.2 Authors

Surname, first name	Organisation	Function
Boehm, Rene	c.a.p.e. IT® GmbH	Head of Development

1.2.3 Document owner

Organisation	Surname, first name	Address	Contacts
c.a.p.e. IT® GmbH		Schönherrstr. 8 09113 Chemnitz	info@cape-it.de +49 371 27095 620

1.2.4 Authorized staff

Surname, first name	Organisation	Function	Auth.
Boehm, Rene	c.a.p.e. IT® GmbH	Head of Development	rw
Boehm, Ralf	c.a.p.e. IT® GmbH	Developer	rw

1.3 List of abbreviations

CI	Config Item
CLI	Command Line Interface
GUI	Graphical User Interface
SDK	Software Development Kit

2 Installation

2.1 Requirements

To install and use this data backend for the KIX Professional function "InventorySync" the following is required:

- a KIX Professional 17.0.x installation
- installed vSphere Perl SDK for vSphere 5.5+ on the KIX server
 - <https://developercenter.vmware.com/web/sdk/55/vsphere-perl>
- a vCenter user with read-only permission (usually a user with default "Read Only" role assigned at Datacenter level will be sufficient)

2.2 Package Installation

Just install the „VMWare“ backend module via your operating systems package management. The package name is „kixpro-inventorysync-vmware“. Please make sure you have fulfilled the necessary requirements (see 2.1).

3 Basics

The VMWare backend works based on ObjectTypes. Each ObjectType has its own set of attributes and provides the data that will be mapped to the CI class attributes in the CI class definition.

It is always necessary to assign an ObjectType, either on the level of the datasource itself (see 4.1.3) or on the level of CI class name mapping (see 4.1.4) or based on the names of the CI classes itself.

The following ObjectTypes are possible:

- Datacenter
- Datastore
- Cluster
- Host
- VM

Example:

If you have CI class named "VMWare-Servers", you have to assign the ObjectType "Host", since this provides the data for a VMWare host system.

Option 1: you can do this on the datasource level

Define a datasource and add ";ObjectType=Host" to the end of the parameter line (see 4.1.3).

Option 2: you can alternatively define a class mapping

Create a CI class mapping where the key is "VMWare-Servers" and the value defines the ObjectType "Host" (see 4.1.4).

Option 3: you can rename the CI class

If you don't want to use option 1 and option 2 you only can do one more thing to make sure the correct ObjectType will be used to communicate with the vCenter server. You have to rename your CI class "VMWare-Servers" to "Host". In this case the name of the CI class itself represents the ObjectType to use (see 4.1.5).

4 Configuration

4.1 SysConfig Options

After you have installed all the needed packages, you have to configure all the parameters necessary for this backend to communicate with one or more vSphere servers.

To configure the extension just open the SysConfig in the admin area. Select the SysConfig group "KIX Professional". After the page reload select the subgroup "ITSMConfigItem::InventorySync".

An example configuration is already included in the package. You should change it to your needs.

The following section lists all the important options needed to get the InventorySync function working.

4.1.1 Sources

SysConfig Option: ITSMConfigItem::InventorySync###Sources

A source represents a named communication point. A SysConfig hash option is used for its configuration. The hash-key is the internal name of the source, i.e. "VMWare". The hash-value is the display name of this source, i.e. "my VMWare environment".

You can add more sources for each external inventory server.

4.1.2 Source-Backend-Mapping

SysConfig Option: ITSMConfigItem::InventorySync###Backend

The system needs to know which backend should be used for a source to communicate with the external inventory tool. The hash-key is the Source's identifier (i.e. "VMWare" in our example) and the hash-value has to be "VMWare" to tell InventorySync to use the VMWare backend for this data source.

4.1.3 Source Parameters

SysConfig Option: ITSMConfigItem::InventorySync###Parameters

All the parameters have to be entered in the following form:

<parameter>=<value>, <parameter>=<value>, ...

You have to configure the following parameters to use the "VMWare" backend:

Parameter	Required	Description
URL	X	the URL to the Webservice interface of the vCenter server (usually https://<host>:443/sdk/webService)
User	X	username to authenticate to the vCenter server
Password	X	password to authenticate to the vCenter server
ObjectType		optional parameter to restrict the datasource to a specific object type. Possible values are: Datacenter, Datastore, Cluster, Host, VM

4.1.4 Alternative ObjectType mapping

SysConfig Option: InventorySync::Backend::VMWare###CIClassObjectTypeMapping

This option can be found in the subgroup "ITSMConfigItem::InventorySync::VMWare". If the datasource itself is not configured to be restricted to a specific ObjectType, here you can assign the ObjectType to be used for each CI class. The Key of the mapping hash represents the name of the CI class and the value represents the ObjectType.

The following ObjectTypes are possible: Datacenter, Datastore, Cluster, Host, VM.

4.1.5 ObjectType fallback

If neither the datasource has been restricted to a specific ObjectType nor a class-based mapping exists, the name of the CI class will be used by the system to determine the ObjectType.

4.1.6 SSL verification

SysConfig Option: InventorySync::Backend::VMWare###UserAgentVerifyHostname

This option can be found in the subgroup "ITSMConfigItem::InventorySync::VMWare" and allows you to disable the SSL certificate validation if necessary.

4.2 InventoryContentXPath for backend "VMWare"

The basic extension of CI classes and the usage of the InventoryContentXPath parameter are described in the documentation of the KIX Professional function „InventorySync“.

The Xpath for VMWare has the following structure:

Hardware Xpath:

HW/<Container>/<Attribute>

Software Xpath:

no software Xpath exists at the moment

Which attributes and container are available depends on the permission of the user you use to connect to the vCenter server and the status of the corresponding objects. Not all of these attributes will always be available.

The following attributes are usually available:

ObjectType	Container	Attribute
Datacenter	Info	configStatus numVMDKs numHosts numClusters numDatastores numVMs overallStatus
	Clusters	name
	Datastores	name
	Hosts	name

	Networks	name
Datastore	Info	LUN build capacity capacityGB capacityKB capacityMB freeSpace freeSpaceGB freeSpaceKB freeSpaceMB instanceUuid isSSD numHosts numVMs type url version vmfsVersion
	Hosts	name
	VMs	name
Cluster	Info	actionHistory alarmActionsEnabled apiType apiVersion build configStatus disabledMethod effectiveCpuGhz effectiveCpuMhz effectiveMemory effectiveMemoryGB effectiveMemoryKB effectiveMemoryMB fullName hBDatastoreCandidatePolicy instanceUuid licenseProductName licenseProductVersion localeBuild localeVersion name numCpuCores numCpuThreads numDatastores numEffectiveHosts numHosts numVMs osType overallStatus productLineId resourcePool totalCpuGhz

		totalCpuMhz totalMemory totalMemoryKB totalMemoryMB totalMemoryGB vendor version
	HA	admissionControl configuredFailoverCapacity dasConfig dasIsolationAddress defaultVMSettings.dasIsolationResponse defaultVMSettings.restartPriority defaultVmSettings.vmToolsMonitoringSettings defaultVmSettings.vmToolsMonitoringSettings.clusterSettings defaultVmSettings.vmToolsMonitoringSettings.failureInterval defaultVmSettings.vmToolsMonitoringSettings.maxFailureWindow defaultVmSettings.vmToolsMonitoringSettings.maxFailures defaultVmSettings.vmToolsMonitoringSettings.minUpTime defaultVmSettings.vmToolsMonitoringSettings.vmMonitoring failoverLevel hostMonitoring status vmMonitoring
	DRS	defaultVmBehavior status
	Datastores	extent<No>.diskName extent<No>.partition name url
	Hosts	name
	VMs	name
Host	Info	boofTime build cluster datacenter DNS.dhcp DNS.hostname DNS.nameserver DNS.searchDomains defaultGateway cpuMhz cpuGhz cpuModel datacenter fullName hostMaxVirtualDiskCapacity hostMaxVirtualDiskCapacityGB hostMaxVirtualDiskCapacityKB hostMaxVirtualDiskCapacityMB licenseProductName memorySize memorySizeGB memorySizeKB

		memorySizeMB model name numCpuCores numCpuPkgs numCpuThreads numDatastores numHBAs numNICs numVMs port powerState uptime uuid vendor version vmotion.ipAddress vmotion.subnetMask
	vSwitches	mtu name numPorts numPortsAvailable portgroup<No>.name portgroup<No>.usedPorts portgroup<No>.vlanId usedPorts
	NICs	autoNegotiateSupported dhcp driver duplex ipAddress macAddress name pci resourcePoolSchedulerAllowed speedMb subnetMask wakeOnLanSupported
	VMs	name
VM	Info	cluster faultToleranceState guestId guestMemoryUsage guestMemoryUsageKB guestMemoryUsageMB guestMemoryUsageGB guestOS host hostMemoryUsage hostMemoryUsageKB hostMemoryUsageMB hostMemoryUsageGB host instanceUuid ipAddress

		isTemplate memorySize memorySizeKB memorySizeMB memorySizeGB name numCPU numEthernetCards numVirtualDisks overallStatus resourcePool storageFree storageFreeKB storageFreeMB storageFreeGB storageTotal storageTotalKB storageTotalMB storageTotalGB storageUsed storageUsedKB storageUsedMB storageUsedGB toolsStatus uptime vmPathName vmState
	CDRoms	label summary
	Datastores	name
	vDisks	backingObjectId capacity capacityKB capacityMB capacityGB datastore diskMode filename label size split type uuid writeThrough
	vNICs	ipAddress label macAddress network
	Snapshots	createTime description name quiesced state

Please Note:

The vCenter API defines Virtual Machine Templates as Virtual Machine objects too. You can determine whether a VM is a template or not, by looking at the attribute "isTemplate".

4.3 Example CI classes for backend "VMWare"

The package includes example CI class definitions for all ObjectTypes. These examples can be found in the following directory:

```
/opt/kix/var/packagesetup/InventorySyncVMWareBackend
```

Please note that these are just examples and they contain ALL available attributes, including some attributes with usage values (i.e. current memory or CPU usage) or different metrics (KB, MB, GB, Mhz, Ghz, ...). You are absolutely free to adjust these definitions to your own needs.

IMPORTANT:

Please be aware that mapped usage values usually will create a new CI version since they are subject to frequent changes. You should remove them from your CI class definition if these values are not of any interest in your CMDB.

You also have to make sure that mandatory CI attributes will get some value during the sync process. Otherwise KIX Professional will show an error message, since the attribute is empty but needed.

These examples make use of the CI attribute "CIClassReference", which is already included in KIX. It is extremely helpful to automatically create links between the different CIs.

To fully utilize the advantages of these configuration examples, you should import at first and synchronize afterwards, so that all the objects are already available in the synchronization process and therefore can be linked.