

KIX Professional InventorySync Backend "DB" Installation and configuration

c.a.p.e. IT® GmbH

Version: 17.0.0

Imprint

Title:	KIX Professional InventorySync Backend "DB"
Topic:	Installation and configuration
Author:	c.a.p.e. IT® GmbH
Keywords:	KIX, Inventory, DB, KIX Professional
Comments:	
Filename:	InventorySyncDBBackend_installation_and_configurati on_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 Configuration.....	6
3.1 SysConfig Options.....	6
3.1.1 Sources.....	6
3.1.2 Source-Backend-Mapping.....	6
3.2 Configuration in Config.pm.....	6
3.3 InventoryContentXPath for backend "DB".....	8
4 Appendix.....	9
4.1 Example CI class "Computer".....	9

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 "DB".

1.2 Changes

1.2.1 Change history

Version	Date	Changed chapters	Short description	Changed by
1.0.0	16.01.13	all	document creation	Rene Boehm
1.0.1	04.07.13	Changes	Format fixes	Rene Boehm
1.0.2	28.10.13	2.2	Minor fix	Rene Boehm
1.0.3	30.10.13	1.1.1, 3.1	Minor Fix	Rene Boehm
1.0.4	21.05.15	2.1	Upgrade for OTRS 4.0	Ricky Kaiser
1.0.5	24.02.16	all	Upgrade for OTRS 5.0	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		Schoenherrstr. 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
Kaiser, Ricky	c.a.p.e. IT® GmbH	Developer	rw

1.3 List of abbreviations

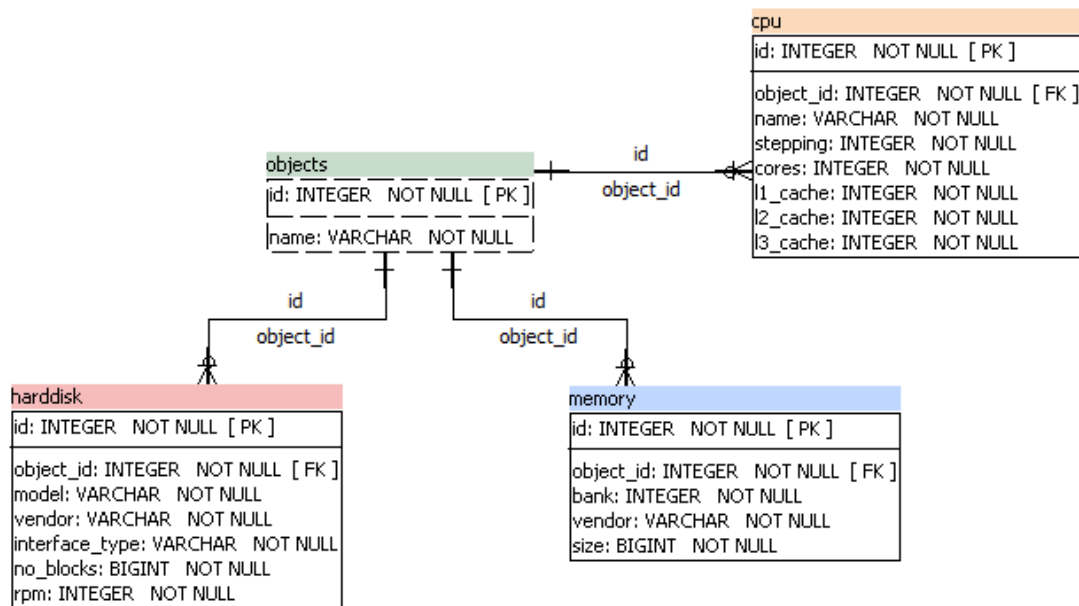
CI	Config Item
CLI	Command Line Interface
DB	Database
GUI	Graphical User Interface

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
- an external database with inventory data organized in a table matrix (see picture below)



This picture is just a schema example to show you how the data has to be organized in order to work with this "DB" backend. You can have as many tables (or views) with as many attributes as you like, but you always have to have a foreign key to the table (or view) containing all your objects (i.e. servers or mobile devices or whatever). This so called "object table" is the primary object list for InventorySync, and has to contain only one row for each object. All the other tables contain the data of the object.

2.2 Package Installation

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

3 Configuration

3.1 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 DBMS.

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.

3.1.1 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. "inventory1". The hash-value is the display name of this source, i.e. "inventory server 1".

You can add more sources for each external inventory server.

3.1.2 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. "inventory1" in our example) and the hash-value has to be "DB" to tell InventorySync to use the DB backend for this data source.

3.2 Configuration in Config.pm

Just the name (and internal key) of a DB datasource and its mapping to the DB backend is configured in the SysConfig. All the other "complex" configuration for the configured DB data sources is done in the Config.pm in a way known from the configuration of CustomerUser backends.

For each DB data source you have configured for InventorySync in the SysConfig, you have to create a configuration hash in the Config.pm.

Here is an example for a configuration of our example data source "inventory1". This one does the configuration for the CI class "Computer", which gets its data from another MySQL database "ocsweb" on localhost. All the objects (computers) are distinctively contained in the database table (or view) "all_clients" and the identifier for each client is the attribute "HARDWARE_ID" in this table. The name of the object is contained in the table attribute "NAME". The attribute named as object identifier (here "HARDWARE_ID") has to be available (Foreign Key) in all the other tables/views where the DB backend should get its data from.

Common tables for Software (SW) and Hardware (HW) information are configured, as well as two separate/special tables/views which should be used to get all the data relevant for the device types "PROCESSOR" and "CONTROLLER".

In the "Mapping" section the attributes from the configured tables/views are mapped into the data structure which will be created for each object. This section therefore defines the structure for the InventoryContentXPath attribute in the CI class configuration.

```
$Self->{InventorySyncDB}->{inventory1} = {
    Params => {
        # if you want to use an external database, add the
        # required settings
        # DSN => 'DBI:odbc:yourdsn',
        DSN => 'DBI:mysql:database=ocsweb;host=localhost',
        User => 'Test',
        Password => 'xxxxx',

        SourceCharset => 'utf-8',
        DestCharset => 'utf-8',
    },
    Classes => {
        Computer => {
            Params => {
                # client list and key value
                ObjectTable => 'all_clients',
                ObjectID => 'HARDWARE_ID',
                ObjectName => 'NAME',

                # common SW and HW tables/views
                # where our hardware/software info comes from
                HWTable => 'csweb.test_view2',
                SWTable => 'ocsweb.test_view1',

                # special tables to use for PROCESSOR/CONTROLLER mapping
                SpecialTables => {
                    HW => {
                        PROCESSOR => 'cpu_table',
                        CONTROLLER => 'controllers',
                    }
                }
            },
            # the following mapping defines the structure for
            # InventoryContentXPath
            Mapping => {
                HW => {
                    PROCESSOR => {
                        name => 'NAME',
                        type => 'TYPE',
                        version => 'VERSION',
                    },
                    serialNumber => {
                        serialnumber => 'SNR',
                    },
                    BIOS => {
                        description => 'DESCRIPTION',
                        name => 'NAME',
                    },
                    CONTROLLER => {
                        description => 'DESCRIPTION',
                        name => 'NAME',
                        type => 'CONTROLLER_TYPE',
                    },
                },
                SW => {
                    Software => {
                        name => 'NAME',
                        version => 'VERSION',
                        descriptions => 'DESCRIPTION',
                    },
                }
            }
        }
    }
}
```

```
}  
}  
}
```

3.3 InventoryContentXPath for backend "DB"

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

The structure and contents of the inventory data (hardware and software) returned by the DB backend is defined in the configuration for the DB data source done in the Config.pm (see 3.2).

The Xpath generally has the following structure:

Hardware Xpath:

HW/<some device key>/<attribute>

Software Xpath:

SW/<configured key>/<attribute>

Typical Xpaths may be:

HW/PROCESSOR/name

to get the name attribute of each processor

HW/HARDDISK_DRIVE/model

to get the model attribute of each harddisk drive

SW/*.?/displayVersion

to get the name of the software, regardless of a SW key

SW/Software/displayVersion

to get the name of the software in a configured SW key

4 Appendix

4.1 Example CI class "Computer"

The following example shows a CI class definition for class "Computer" with a couple of changes for InventorySync. Please note that these are no mandatory changes! You are absolutely free to make you own changes. You have just to make sure that mandatory CI attributes will get some value during the sync process. Otherwise KIX Professional will show an error message. The format of the values of `InventoryContentXPath` attribute is described in 3.2 and 3.3.

```
[
  {
    Key => 'Vendor',
    Name => 'Vendor',
    Searchable => 1,
    Input => {
      Type => 'Text',
      Size => 50,
      MaxLength => 50,
    },
  },
  {
    Key => 'Model',
    Name => 'Model',
    InventoryContentXPath => 'HW/COMPUTER_SYSTEM/model',
    Searchable => 1,
    Input => {
      Type => 'Text',
      Size => 50,
      MaxLength => 50,
    },
  },
  {
    Key => 'Description',
    Name => 'Description',
    Searchable => 1,
    Input => {
      Type => 'TextArea',
    },
  },
  {
    Key => 'Type',
    Name => 'Type',
    Searchable => 1,
    Input => {
      Type => 'GeneralCatalog',
      Class => 'ITSM::ConfigItem::Computer::Type',
      Translation => 1,
    },
  },
  {
    Key => 'Owner',
    Name => 'Owner',
    Searchable => 1,
    Input => {
      Type => 'Customer',
    },
  },
  {
    Key => 'SerialNumber',
    Name => 'SerialNumber',
    InventoryContentXPath => 'HW/BASE_BOARD/serialnumber',
    Searchable => 1,
    Input => {
```

```

    Type => 'Text',
    Size => 50,
    MaxLength => 100,
  },
},
{
  Key => 'Software',
  Name => 'Software',
  InventoryContentXPath => 'SW.*?/displayName',
  InventoryContentEvalString => '$Inventory{displayVersion}',
  Input => {
    Type => 'Text',
    Size => 50,
    MaxLength => 100,
  },
  CountMax => 1000,
  Sub => [
    {
      Key => 'Version',
      Name => 'Version',
      InventoryContentXPath => 'SW.*?/displayVersion',
      Input => {
        Type => 'Text',
        Size => 20,
        MaxLength => 20,
      },
    },
  ],
},
},
{
  Key => 'CPU',
  Name => 'CPU',
  InventoryContentXPath => 'HW/PROCESSOR/name',
  Input => {
    Type => 'Text',
    Size => 50,
    MaxLength => 100,
  },
  CountMax => 16,
},
},
{
  Key => 'Ram',
  Name => 'Ram',
  InventoryContentXPath => 'HW/MEMORY_MODULE/memorytype',
  Input => {
    Type => 'Text',
    Size => 50,
    MaxLength => 100,
  },
  CountMax => 10,
},
},
{
  Key => 'HardDisk',
  Name => 'Hard Disk',
  InventoryContentXPath => 'HW/HARDDISK_DRIVE/name',
  Input => {
    Type => 'Text',
    Size => 50,
    MaxLength => 100,
  },
  CountMax => 10,
  Sub => [
    {
      Key => 'Capacity',
      Name => 'Capacity',
      InventoryContentXPath => 'HW/HARDDISK_DRIVE/size',
      Input => {
        Type => 'Text',
        Size => 20,
        MaxLength => 10,
      },
    },
  ],
},
},
},
{

```

```

    Key => 'CD-Rom',
    Name => 'CD-Rom',
    Searchable => 1,
    InventoryContentXPath => 'HW/OPTICAL_DRIVE/name',
    Input => {
      Type => 'Text',
      Size => 50,
      MaxLength => 100,
    },
  },
  {
    Key => 'FQDN',
    Name => 'FQDN',
    Searchable => 1,
    InventoryContentXPath => 'HW/COMPUTER_SYSTEM/name',
    Input => {
      Type => 'Text',
      Size => 50,
      MaxLength => 100,
    },
  },
  {
    Key => 'NIC',
    Name => 'Network Adapter',
    InventoryContentXPath => 'HW/NETWORK_CONTROLLER/name',
    InventoryContentEvalString => '$Inventory{ipaddress}',
    Input => {
      Type => 'Text',
      Size => 50,
      MaxLength => 100,
    },
    CountMin => 0,
    CountMax => 10,
    CountDefault => 1,
    Sub => [
      {
        Key => 'IPoverDHCP',
        Name => 'IP over DHCP',
        Input => {
          Type => 'GeneralCatalog',
          Class => 'ITSM::ConfigItem::YesNo',
          Translation => 1,
        },
      },
      {
        Key => 'IPAddress',
        Name => 'IP Address',
        Searchable => 1,
        InventoryContentXPath => 'HW/NETWORK_CONTROLLER/ipaddress',
        Input => {
          Type => 'Text',
          Size => 40,
          MaxLength => 40,
        },
        CountMin => 0,
        CountMax => 20,
        CountDefault => 0,
      },
    ],
  },
  {
    Key => 'GraphicAdapter',
    Name => 'Graphic Adapter',
    InventoryContentXPath => 'HW/VIDEO_CONTROLLER/name',
    InventoryContentEvalString => '$Inventory{deviceId}',
    Input => {
      Type => 'Text',
      Size => 50,
      MaxLength => 100,
    },
  },
  {
    Key => 'OtherEquipment',
    Name => 'Other Equipment',
    Input => {

```

```
        Type => 'TextArea',
    },
    CountMin => 0,
    CountDefault => 0,
},
{
    Key => 'WarrantyExpirationDate',
    Name => 'Warranty Expiration Date',
    Searchable => 1,
    Input => {
        Type => 'Date',
    },
},
{
    Key => 'InstallDate',
    Name => 'Install Date',
    Searchable => 1,
    Input => {
        Type => 'Date',
    },
    CountMin => 0,
    CountDefault => 0,
},
{
    Key => 'Note',
    Name => 'Note',
    Searchable => 1,
    Input => {
        Type => 'TextArea',
    },
    CountMin => 0,
    CountDefault => 0,
},
];
```