

Systems - VEMA: Synchronization

<https://wiki.czechidm.com/>

2020/01/21 18:34

Table of Contents

Systems - VEMA: Synchronization	1
VEMA cloud	1
Local VEMA installation	6
Common part	6

Systems - VEMA: Synchronization

You may have VEMA installed locally or in the cloud. The basic principals of integration of VEMA and VEMA Cloud with CzechIdM are very similar or the same. In both cases we will use an external MS-SQL database "Interchange" to store the data from VEMA which will be read by CzechIdM. We will also need a server with the VEMA thick client installed.

VEMA cloud

1. Prepare the Windows server with VEMA thick client
 - It's necessary to install **v4data** package. Start VEMA Administrator, go to the "Products" tab, invoke context menu by clicking the right mouse button and select "Install". Then choose the latest version of in the dialog.
1. Prepare MS-SQL database for the "Interchange".
2. Install the scripts:
 1. Create an user in VEMA with a permission to switch to the admin role and with access to the personal data (recommended role name = "IDM export"). Emit a certificate for this user and install it to the server with the VEMA client. You can check the installation correctness with running following command:

```
C:\Program Files (x86)\Vema\Vema.exe" /app=DBV HR PAM  
/server=your_id.cloud.vema.cz /username=your_username  
/world=your_ID
```

2. Create a database user in MS-SQL database "Interchange" with permission to read&write to the database and schema. It needs to be a local user, domain authentication doesn't work within the scripts.
3. Create the MS-SQL "Interchange" database structure. Using following script is recommended to get the default structure.

```
-- create table skript pro vytvoreni tabulek v MS SQL pro export z  
Vema  
-- if 'nchar' doesn't work, change it for 'char' and vice-versa -  
the reason is that the Python module from Vema can't compare  
varchar -> comparison always ends up with false so all records are  
marked as "changed"  
  
CREATE TABLE "dbo"."tpzr"  
(  
    pzar INT PRIMARY KEY NOT NULL,  
    pzar NCHAR(255)  
);  
CREATE UNIQUE INDEX PK_TPZR ON "dbo"."tpzr"(pzar);  
  
CREATE TABLE "dbo"."todp"  
(  
    povfc INT PRIMARY KEY NOT NULL,
```

```
povfct NCHAR(255)
);
CREATE UNIQUE INDEX PK_TODP ON "dbo"."todp"(povfc);

CREATE TABLE "dbo"."tkat"
(
    kat INT PRIMARY KEY NOT NULL,
    katt NCHAR(255)
);
CREATE UNIQUE INDEX PK_TKAT ON "dbo"."tkat"(kat);

CREATE TABLE "dbo"."todz"
(
    zames INT PRIMARY KEY NOT NULL,
    czisco NCHAR(10),
    kateg INT,
    skjkz INT,
    povfc INT,
    funkc INT,
    povolani NCHAR(255),
    pocrd INT,
    platnost INT,
    zamest NCHAR(255),
    zamest1 NCHAR(255),
    zamest2 NCHAR(255)
)
;

CREATE TABLE "dbo"."tros"
(
    ros INT PRIMARY KEY NOT NULL,
    rost NCHAR(255)
)

CREATE TABLE "dbo"."osobyx"
(
    oscis INT PRIMARY KEY NOT NULL,
    jmenozd NCHAR(250) DEFAULT (NULL),
    prijmsz NCHAR(250) DEFAULT (NULL),
    tituly NCHAR(250) DEFAULT (NULL),
    upljmeno NCHAR(250) DEFAULT (NULL),
    full_update_timestamp datetime DEFAULT (getdate()) NOT NULL,
    greatest_timestamp datetime DEFAULT (getdate()) NOT NULL,
    rocis NCHAR(15),
    cpoj NCHAR(15),
    ICL NCHAR(6),
    druhlp INT,
    datlp datetime,
    mistolp NCHAR(255),
    vysllp NCHAR(10),
```

```
    kopllp datetime,  
    rosu INT  
);  
  
CREATE TABLE "dbo"."pzarazx"  
(  
    oscis INT NOT NULL,  
    cicin INT NOT NULL,  
    povfc INT DEFAULT (NULL),  
    kateg INT DEFAULT (NULL),  
    pracvz INT DEFAULT (NULL),  
    pzar INT DEFAULT (NULL),  
    utvar NCHAR(6) DEFAULT (NULL),  
    zauc datetime NOT NULL,  
    zapl datetime DEFAULT (NULL),  
    ppvdr INT DEFAULT (NULL),  
    povfct1 INT DEFAULT (NULL),  
    zames INT DEFAULT (NULL),  
    full_update_timestamp datetime DEFAULT (getdate()) NOT NULL,  
    CONSTRAINT PK__pzarazx__D560F4C128ED12D1 PRIMARY KEY  
(oscis,cicin,zauc)  
);  
  
CREATE TABLE "dbo"."smlouvax"  
(  
    oscis INT NOT NULL,  
    cicin INT NOT NULL,  
    ppvza datetime NOT NULL,  
    ppvko datetime DEFAULT (NULL),  
    full_update_timestamp datetime DEFAULT (getdate()) NOT NULL,  
    CONSTRAINT PK__smlouvax__3E3BBE5213F1F5EB PRIMARY KEY  
(oscis,cicin,ppvza)  
);  
  
CREATE TABLE "dbo"."vynetix"  
(  
    oscis INT NOT NULL,  
    vynz datetime NOT NULL,  
    vynk datetime DEFAULT (NULL),  
    vynd INT NOT NULL,  
    full_update_timestamp datetime DEFAULT (getdate()) NOT NULL,  
    CONSTRAINT PK__vynetix__8758F81519AACF41 PRIMARY KEY  
(oscis,vynz,vynd)  
);  
  
CREATE TABLE tutv (  
    utvar NCHAR(50) NOT NULL,  
    utvart NCHAR(255) DEFAULT NULL,  
    zapl datetime DEFAULT NULL,  
    kopl datetime DEFAULT NULL,  
    zkratka NCHAR(20) DEFAULT NULL,
```

```
        CONSTRAINT utvar_pkey PRIMARY KEY (utvar)
    );

CREATE TABLE czisco (
    czisco NCHAR(10) NOT NULL,
    praczar NCHAR(255) NOT NULL,
    praczarzk NCHAR(10) NOT NULL,
    CONSTRAINT czisco_pkey PRIMARY KEY (czisco)
);
```

4. Install test scripts to a folder and change the log file path according to your environment. Default is C:\VemaV4\idmlogs\exportvymeniklog.txt

1. test3.py:

```
# -*- coding: utf-8 -*-

from __future__ import unicode_literals
from vmsys import RunAppFile
from v4data.dbv import DBVDatabase, DBVData, DBVApp
from v4core.meta import Property, PNumber

def LogInfo(message):
    log = open('C:\\VemaV4\\idmlogs\\export_vymenik_log.txt',
              'a')
    log.write(u' INFO: ' + message + '\n')
    log.close()

class PersInfoTable(DBVData):
    fileName = 'osobyt'

class PAM(DBVDatabase):
    data = [PersInfoTable]

class MyApp(DBVApp):
    db = PAM
    def run(self):
        for row in self.db.PersInfoTable():
            LogInfo(u'TEST - KOMUNIKACE S VEMA DB' +
                  str(row.oscis) + u' ' + row.prijmzd);
            break;

main = MyApp
```

2. test6.py:

```
# -*- coding: utf-8 -*-

from __future__ import unicode_literals
from vmsys import RunAppFile
from v4data.dbv import DBVDatabase, DBVData, DBVApp
from v4core.meta import Property, PNumber
import PyVRDA, pyodbc
```



```

def LogInfo(message):
    log = open('C:\\VemaV4\\idmlogs\\export_vyменik_log.txt',
'a')
    log.write(u' INFO: ' + message + '\n')
    log.close()

class MyApp(DBVApp):
    def run(self):

        conn = pyodbc.connect('DRIVER={ODBC Driver 13 for SQL
Server};SERVER=127.0.0.1;DATABASE=001;UID=vema;PWD=*****')
        cursor = conn.cursor()
        cursor.execute('INSERT INTO osobyt (oscis) VALUES
(123)')
        conn.commit()
        LogInfo(u'connectDB: COMMIT')
        conn.close()

        LogInfo(u'Export DB connected')

main = MyApp

```

5. Run the test scripts

```

C:\Program Files (x86)\Vema\Vema.exe" /app=pyscript test6.py
/database=DB_NAME /server=cloud.vema.cz /world=your_ID
/databasePER=DB_PER_NAME /serverPER=your_ID.cloud.vema.cz
/application=pam /roleName="IDM export" /username=login

```

```

C:\Program Files (x86)\Vema\Vema.exe" /app=pyscript test3.py
/database=DB_NAME /server=cloud.vema.cz /world=your_ID
/databasePER=DB_PER_NAME /serverPER=your_ID.cloud.vema.cz
/application=pam /roleName="IDM export" /username=login

```

1. the path to the Vema.exe might differ according to a certain installation

6. Check the results

- test3.py - tests the communication with VEMA. Note: You might need to change the name of data file you want to read from. The default is "osobyt" in database PAM. In other cases it might be "osobyx" in database PER.
- test6.py - tests ability to write into the "Interchange" database. Fill the following values (SERVER=...,DATABASE=...,UID=...,PWD=...), you might also need to change the table from "osobyt" to "osobyx".
- **Expected results:**
 1. No errors while running the scripts
 2. test3.py: Writest a list of family names from VEMA's data file to the log file C:\\VemaV4\\idmlogs\\exportvyменiklog.txt
 3. test6.py: Inserts a record into the table osobyt in the "Interchange" database. The record will have value of "oscis" 123 and a info message is written into the log file C:\\VemaV4\\idmlogs\\exportvyменiklog.txt

Local VEMA installation

BOTH local and cloud solutions pleas. TODO

Common part

Then you need to implement a certain script which will read the data you want from VEMA and write them to the "Interchange" database. Also you might need to alter the tables according the data you read. Scripts above are just an examples.

The script must be run automatically e.g. once a day.

In IDM you will create a System with the Database Table Connector or Scripted SQL Connector according to the certain use-case. The first case is covered tutorials [Systems - DB: Users' contracts synchronization](#) and [Systems - DB: Source of identities](#).

From:
<https://wiki.czechidm.com/> - **CzechIdM Identity Manager**

Permanent link:
https://wiki.czechidm.com/tutorial/adm/vema_cloud

Last update: **2019/08/21 14:03**

