

independIT Integrative Technologies GmbH
Bergstraße 6
D-86529 Schrobenhausen



schedulix

Installation Guide Release 2.5.1

Dieter Stubler Ronald Jeninga

August 2, 2013

Copyright © 2013 independIT GmbH

Legal notice

This work is copyright protected

Copyright © 2013 independIT Integrative Technologies GmbH

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronically or mechanically, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner.

Contents

Table of contents	1
1 Requirements	1
Compile environment	1
schedulix Server and Clients	2
Zope Application Server	2
2 Compiling the system	3
General preparation	3
Compile	3
3 Installation in a Linux environment	5
Installing the schedulix Server	5
Introduction	5
Installation	5
Installation with Postgres	8
Introduction	8
Installation	8
Installation with MySQL	10
Introduction	10
Installation	10
Installation with Ingres	12
Introduction	12
Installation	12
Installing the Zope server	13
Introduction	13
Installation	13

1 Requirements

Compile environment

The following software is required to create the executables from the source package that are needed on a Linux system:

- Oracle(Sun) Java 1.6 JDK or later
<http://www.oracle.com/technetwork/java/index.html>
Alternatively including an OpenJDK 1.6 or later
<http://openjdk.java.net>
- gcc, gcc-c++
<http://gcc.gnu.org>
- gnu make
<http://www.gnu.org/software/make>
- jflex
<http://jflex.de>
- jay
The jay executable is provided in the package. The original sources and executables can be found here.
<http://www.cs.rit.edu/~ats/projects/lp/doc/jay/package-summary.html>
Important: the jay executable requires 32-bit libraries. They will have to be installed additionally on 64-bit systems.
- One of the following RDBMS systems with the corresponding JDBC interface:
 - PostgreSQL
<http://www.postgresql.org>
JDBC for PostgreSQL:
<http://jdbc.postgresql.org>
 - MySQL
<http://www.mysql.com>
MySQL (Connector/J) JDBC Driver
<http://www.mysql.com>
 - Ingres
<http://www.ingres.com>

Zope Application Server

- Eclipse SWT
The package includes several examples for which SWT has to be installed. To make sure that the compile does not crash and the examples, function properly requires an Eclipse SWT.
<http://www.eclipse.org/swt>

In many cases, the required software packages can be easily installed using a Package Manager such as yum, rpm or dpkg.

schedulix Server and Clients

The following software is required to install the schedulix Server:

- Oracle(Sun) Java 1.6 SE JRE
<http://www.oracle.com/technetwork/java/index.html>
Alternatively including an OpenJDK 1.6 or later
<http://openjdk.java.net>
- One of the following RDBMS systems with the corresponding JDBC interface:
 - PostgreSQL
<http://www.postgresql.org>
JDBC for PostgreSQL:
<http://jdbc.postgresql.org>
 - MySQL
<http://www.mysql.com>
MySQL (Connector/J) JDBC Driver
<http://www.mysql.com>
 - Ingres
<http://www.ingres.com>

Zope Application Server

The web front end is provided by the Zope Application Server. The following software is required to install the Zope server:

- Python 2.7
<http://www.python.org>
- Python development package (python-devel or python-dev)
<http://www.python.org>
- Python setuptools
<http://pypi.python.org>

2 Compiling the system

General preparation

Sensitive software should be installed under a separate account. This simplifies the administration and protects against abuse. In this guide it is assumed that the conversion and installation take place using the account `schedulix`. The home directory is assumed to be `/home/schedulix`. These are naturally just suggestions. It is not technically necessary to use them, although the guide will have to be interpreted accordingly if different parameters are being used.

How to create a user is described in the installation chapter on page 5.

Compile

To successfully translate the system after the required packages have been installed, some environment variables have to be set before "make" can do the actual work. Because no special privileges are required either for the conversion or the installation these are done under the user `schedulix`.

1. Download of the `schedulix` source distribution from github
All files needed for compilation and installing the system are available in github and can be retrieved using the following command:

```
cd $HOME
git clone https://github.com/schedulix/schedulix.git
```

Afterwards all the files of the `schedulix` source distribution will be stored in the subdirectory

```
$HOME/schedulix
```

2. Set the environment variables
You now have to set some environment variables. The following commands are from an installation using a CentOS (<http://www.centos.org>) Linux distribution. In many cases the commands can be taken over as given here, but they are dependent upon the specific Linux distribution and the installed software.

```
export SDMSHOME=/home/schedulix/schedulix
export CLASSPATH=$CLASSPATH:/usr/share/java/jflex.jar
export JAVAHOME=/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0
export SWTJAR=/usr/lib/java/swt.jar
```

Compile

It is advisable to add these settings to `.bashrc` at least until the conversion of the system has been completed.

3. make

All that remains to be done now is the actual conversion of the system.

```
cd ~/schedulix/src  
make
```

In the last operation a jar file is created and saved under `~/schedulix/lib`.

Once the compile has been successfully completed, you can continue with the normal installation of the software.

3 Installation in a Linux environment

Installing the schedulix Server

Introduction

In this guide, some of the required software is installed locally, although a global installation is also possible. The advantage of a local installation is that this avoids version conflicts. It is, of course, the responsibility of the system administrator to decide which installation method is to be used.

Installation

The installation of the schedulix Scheduling Server is uncomplicated and only requires a few simple actions which are explained in the following.

Where (example) commands are shown, the prompt is usually indicated with `$`. These commands are then executed under the `schedulix` account, which needs to be created. In some cases, the privileged `root` account is required. This is indicated by a `#` as prompt.

1. Creating the user `schedulix`

It is not necessary to name the user `schedulix`. This means that the name can also be modified for any convention. In this document it is assumed that the user is called `schedulix`.

Under the Ubuntu distribution of Linux, a user can be created as follows:

```
# useradd -d /home/schedulix -m -s /bin/bash -U schedulix
# passwd schedulix
```

All the following operations are executed under the user `schedulix` except where a different user is explicitly stated.

2. Installing

It is assumed that Java has been installed in the directory

```
/home/schedulix/software/java
```

3. Downloading and installing a database management system supported by schedulix.

`schedulix` for Linux currently supports the following systems:

- Postgres (page 8)
- MySQL (page 10)
- Ingres (page 12)

Reference is made to the appropriate sections regarding the installation of the chosen database system as well as how to modify the configuration of the schedulix Enterprise Scheduling System.

4. Create the configuration

a) User environment

The following variables have to be set to be able to work with the schedulix system:

```
BICSUITEHOME=/home/schedulix/schedulix
BICSUITECONFIG=/home/schedulix/etc
PATH=$BICSUITEHOME/bin:$PATH
SWTJAR=/usr/lib/java/swt.jar
```

It has proved to be good practice to save the system configuration in a different folder to the installation directory. This will make it substantially easier to upgrade the system later. Since the variables of all the systems users have to be set, it may be sensible to write the assignments (and exports) to a separate file and then to source this in `.profile` or `.bashrc`.

b) Software environment

Several templates for the configuration files that should be used as a basis for the system configuration can be found under `$BICSUITEHOME/etc`. These have to be copied without the `".template"` extension to the directory `$BICSUITECONFIG`.

For instance:

```
$ cd $BICSUITEHOME/etc; for fff in *.template; do
> TRG=`basename $fff .template`;
> cp $fff $BICSUITECONFIG/$TRG;
> done
```

Afterwards, the files obviously have to be modified to accommodate the environment.

The file `bicsuite.conf` configures some default settings and does not usually have to be modified. However, it may be worth considering running the system logging in a different folder to the installation directory. In this case it is only necessary to change the variable `BICSUITELOGDIR` accordingly. The directory set in `BICSUITELOGDIR` must exist.

The file `java.conf` describes the Java environment that is to be used. In particular, the path to the JDBC driver must be entered. The memory

configuration of the server is also regulated here. Even in large-scale environments it is usually only necessary to adapt the variable `BICSUITEMEM`. The file `server.conf` contains the server configuration. The settings for connecting the schedulix Scheduling Server to its RDBMS repository have to be modified. More details about this can be found in the respective chapter on the RDBMS that is being used.

In this file it is also necessary to change the `hostname` property to the servers hostname or IP address.

The file `jobserver.conf` is not required here, but serves as a template for the job server configuration.

5. Set up the database

Follow the instructions for setting up the database dependent upon which database system you want to use.

For

- Ingres, see page 12,
- MySQL, see page 10, and for
- PostgreSQL, see page 8.

6. Boot the server

The installation is now more or less complete. You just have to start the server and load the examples as required.

The server can be started with

```
$ server-start
```

7. Create the file `.sdmshrc`

The file `.sdmshrc`, if present, is read by all the schedulix command line tools for populating the command line parameters. In the following it is assumed that this file exists and that the correct values have been set for the users, password, host and port. The file `.sdmshrc` is created in the Linux user's home directory.

Here is an example of the contents:

```
$ cat ~/.sdmshrc
User=SYSTEM
Password=G0H0ME
Host=localhost
Port=2506
Timeout=0
```

Important: Since the file contains the data for accessing the Scheduling Server, the file rights should be set so that only its owner can read the file.

Installation with Postgres

```
$ chmod 600 ~/.sdmshrc
$ ls -l ~/.sdmshrc
-rw----- 1 schedulix schedulix 73 2011-11-09 09:28 /home/schedulix/.sdmshrc
```

8. Install the convenience package

The convenience package installs a commonly used configuration for an exit state model.

```
$ sdmsh < $BICSUITEHOME/install/convenience.sdms
```

9. Install the examples

The installation routine for the examples comprises two parts. First of all, three so-called job servers that are required for the subsequent workflow definitions are created. Sample workflow definitions are then loaded onto the server.

a) Create the job servers

It just takes one script to create the job servers:

```
$ cd $BICSUITEHOME/install
$ setup_example_jobserver.sh
```

b) Load the workflow definitions

The following commands are entered to load the workflow definitions:

```
$ cd $BICSUITEHOME/install
$ sdmsh < setup_examples.sdms
```

Because the examples assume that the job servers have already been created, the above sequence is mandatory.

Installation with Postgres

Introduction

This guide does not claim to precisely describe the installation of the database system. Details about this can be found in the Postgres documentation. Normally, though, this guide should be adequate for performing a "standard" installation.

Installation

1. Download and install the latest version of Postgres

A Postgres package is normally provided for every Linux distribution. This package, as well as a package for the JDBC driver for Postgres, should usually be easy to install.

Installation with Postgres

2. Configure the file `pg_hba.conf`

To enable the schedulix Scheduling Server to identify itself to PostgreSQL with a user and password, the following line has to be added to the Postgres configuration file `pg_hba.conf` (this is usually found under `/var/lib/pgsql/<version>/data`):

```
host all all 127.0.0.1/32 md5
```

PostgreSQL then has to be restarted.

3. Create the Postgres user schedulix

Log on as the Postgres user and run the command `createuser` as shown in the example (version 8):

```
$ createuser -P schedulix
Enter password for new role:
Enter it again:
Shall the new role be a superuser? (y/n): n
Shall the new role be allowed to create databases? (y/n): y
Shall the new role be allowed to create more new roles? (y/n): n
```

or, in the case of version 9:

```
$ createuser -P -d schedulix
```

The entered password will be required again later.

4. Create the repository database

When you are logged on as the user schedulix, create the database for the repository as shown in the following example:

```
$ createdb schedulixdb
```

5. Create and initialise the database tables

To create the required database schema, switch to the schedulix sql directory and call the Postgres utility `psql` as shown in the following example:

```
$ cd $BICSUITEHOME/sql
$ psql -f pg/install.sql schedulixdb
```

6. Configure the database connection in the schedulix Server configuration file `$BICSUITECONFIG/server.conf`

Change the following properties as shown here:

Installation with MySQL

```
DbPasswd=schedulix password
DbUrl=jdbc:postgresql:schedulixdb
DbUser=schedulix
JdbcDriver=org.postgresql.Driver
```

The `DbUrl` is to a certain degree dependent upon which version of PostgreSQL is installed. Under Version 9.2 this is

```
DbUrl=jdbc:postgresql:schedulixdb
```

7. Configure the schedulix Java Class Path for the Postgres JDBC

Now all you have to do is to append the path to the Postgres JDBC to `CLASSPATH` in the configuration file `$BICSUITECONFIG/java.conf`.

For instance:

```
BICSUITECLASSPATH=$BICSUITEJAR:/usr/share/java/postgresql-jdbc3-8.4.jar
```

Installation with MySQL

Introduction

This guide does not claim to precisely describe the installation of the database system. Details about this can be found in the MySQL documentation. Normally, though, this guide should be adequate for performing a "standard" installation.

Installation

1. Download and install the latest version of MySQL.

Ready-to-use MySQL packages are available for most Linux distributions. These can be simply installed using the appropriate tools.

During this installation, you will be prompted to enter a password for the MySQL root user (not to be confused with the Linux root user). This password is required again in the next step.

Because schedulix sets up a JDBC connection to access the database, the MySQL JDBC driver has to be installed as well.

2. Create the MySQL user schedulix and the database schedulixdb

Start the Utility `mysql` utility and log on as the MySQL root user to create the user schedulix and the database schedulixdb:

```
$ mysql --user=root --password=mysql-root-password
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
```

Installation with MySQL

Server version: 5.1.54-1ubuntu4 (Ubuntu)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> create user schedulix identified by 'schedulix_password';  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> create database schedulixdb;  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> grant all on schedulixdb.* to schedulix;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> quit  
Bye
```

3. Create and initialise the database tables

Run the following commands:

```
$ cd $BICSUITEHOME/sql  
$ mysql --user=schedulix --password=schedulix_password  
    --database=schedulixdb --execute="source mysql/install.sql"
```

4. Configure the database connection in the \$BICSUITECONFIG/server.conf configuration file

Change the following properties as shown here:

```
DbPasswd=schedulix_password  
DbUrl=jdbc:mysql:///schedulixdb  
DbUser=schedulix  
JdbcDriver=com.mysql.jdbc.Driver
```

5. Configure the schedulix Java Class Path for the MySQL JDBC

Now all you have to do is to append the path to the MySQL JDBC to CLASSPATH in the configuration file \$BICSUITECONFIG/java.conf.

For instance

```
BICSUITECLASSPATH=$BICSUITEJAR:/usr/share/java/mysql-connector-java.jar
```

Installation with Ingres

Introduction

This guide does not claim to precisely describe the installation of the database system. Details about this can be found in the Ingres documentation. Normally, though, this guide should be adequate for performing a “standard” installation.

Installation

1. Install Ingres

We assume that the Ingres system will be installed under the user `ingres`. The installation identifier is taken here as being `II`, which is the default value.

2. Create the user `schedulix`

There are two ways of registering the user `schedulix` in the Ingres system. The first method involves creating the user with the help of the tool `accessdb`. This method is not explained here any further.

The second method is to create the user using SQL commands. To do this, start the SQL Terminal Monitor as the user `ingres`:

```
$ su - ingres
Password:
ingres@cheetah:~$ sql iidbdb
INGRES TERMINAL MONITOR Copyright 2008 Ingres Corporation
Ingres Linux Version II 9.2.1 (a64.lnx/103)NPTL login
Mon Jun 13 10:05:19 2011

continue
* create user schedulix with privileges = (createdb);
* \g
Executing . . .

continue
* commit;\g
Executing . . .

continue
* \q
Ingres Version II 9.2.1 (a64.lnx/103)NPTL logout
Mon Jun 13 10:07:58 2011
ingres@cheetah:~$
```

3. Create the repository database `schedulixdb`

```
$ $II_SYSTEM/ingres/bin/createdb schedulixdb
Creating database 'schedulixdb' . . .
```

Installing the Zope server

```
Creating DBMS System Catalogs . . .  
Modifying DBMS System Catalogs . . .  
Creating Standard Catalog Interface . . .  
Creating Front-end System Catalogs . . .
```

Creation of database 'schedulixdb' completed successfully.

4. Create and initialise the database tables

Run the following commands to create the required tables:

```
$ cd $BICSUITEHOME/sql  
$ sql schedulixdb < ing\install.sql
```

5. Configure the database connection in the schedulix Server configuration file \$BICSUITECONFIG/server.conf

Change the following properties as shown here:

```
DbPasswd=<schedulix OS password>  
DbUrl=jdbc:ingres://localhost:II7/schedulixdb;  
DbUser=schedulix  
JdbcDriver=com.ingres.jdbc.IngresDriver
```

6. Configure the schedulix Java Class Path for the Ingres JDBC

Now all you have to do is to append the path to the Ingres JDBC to CLASSPATH in the configuration file \$BICSUITECONFIG/java.conf.

For instance:

```
BICSUITECLASSPATH=$BICSUITEJAR:$II_SYSTEM/ingres/lib/iijdbc.jar
```

Installing the Zope server

Introduction

A Zope Application Server has to be installed before you can use the schedulixinterface.

Installation

1. Install virtualenv

```
$ easy_install virtualenv
```


Installing the Zope server

2. Create a virtual Python environment for the Zope installation

```
$ mkdir $HOME/software
$ cd $HOME/software
$ virtualenv --no-site-packages Zope
```

3. Install the Zope2 software

```
$ cd $HOME/software/Zope
$ bin/easy_install -i http://download.zope.org/Zope2/index/2.13.6 Zope2
```

4. Create a Zope instance for schedulix user interface

```
$ cd $HOME/software/Zope
$ bin/mkzopeinstance -d $HOME/schedulixweb -u sdmsadm:sdmsadm_password
```

You can use any password of your choice. This will be required again later again.

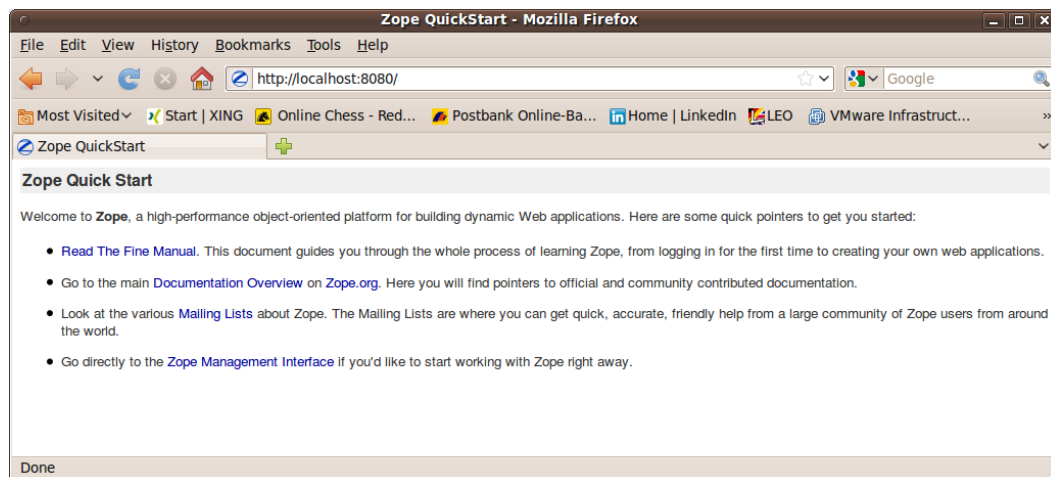
The Zope server is started briefly for testing:

```
$ $HOME/schedulixweb/bin/zopectl start
```

In the web browser, the URL

`http://localhost:8080`

should now be displayed on the Zope QuickStart page.



The Zope instance is now stopped again:

Installing the Zope server

```
$ $HOME/schedulixweb/bin/zopectl stop
```

5. Install the schedulix GUI components

The Zope installation has to be extended with several modules before the schedulix GUI components can be installed.

```
$ cd $HOME/schedulixweb
$ mkdir Extensions
$ cd Extensions
$ ln -s $HOME/schedulix/zope/*.py .
$ cd ../Products
$ ln -s $HOME/schedulix/zope/BICsuiteSubmitMemory .
$ cd ../import
$ ln -s $HOME/schedulix/zope/SDMS.zexp .
```

The Zope instance now has to be started again to register the changes on the Zope side.

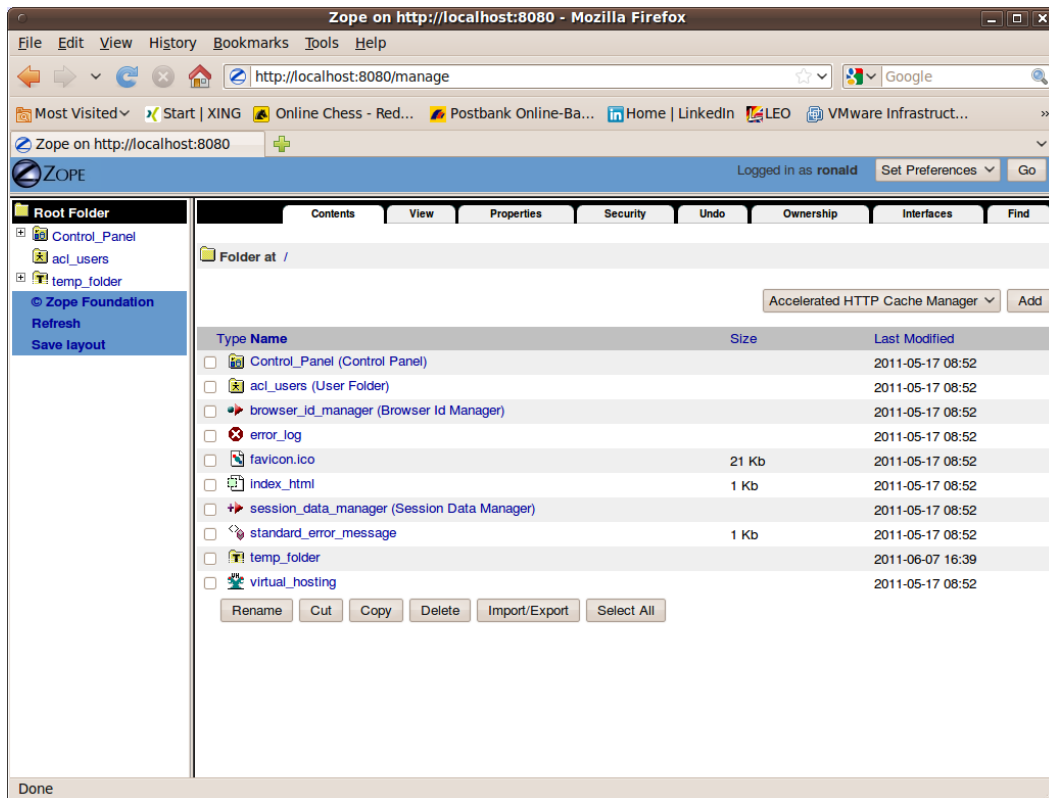
```
$ $HOME/schedulixweb/bin/zopectl start
```

The Zope Management user interface is now opened at the address

```
http://localhost:8080/manage
```

in a browser. This is done with the user `sdmsadm` together with the password you have assigned (in this guide this is `sdmsadm_password`).

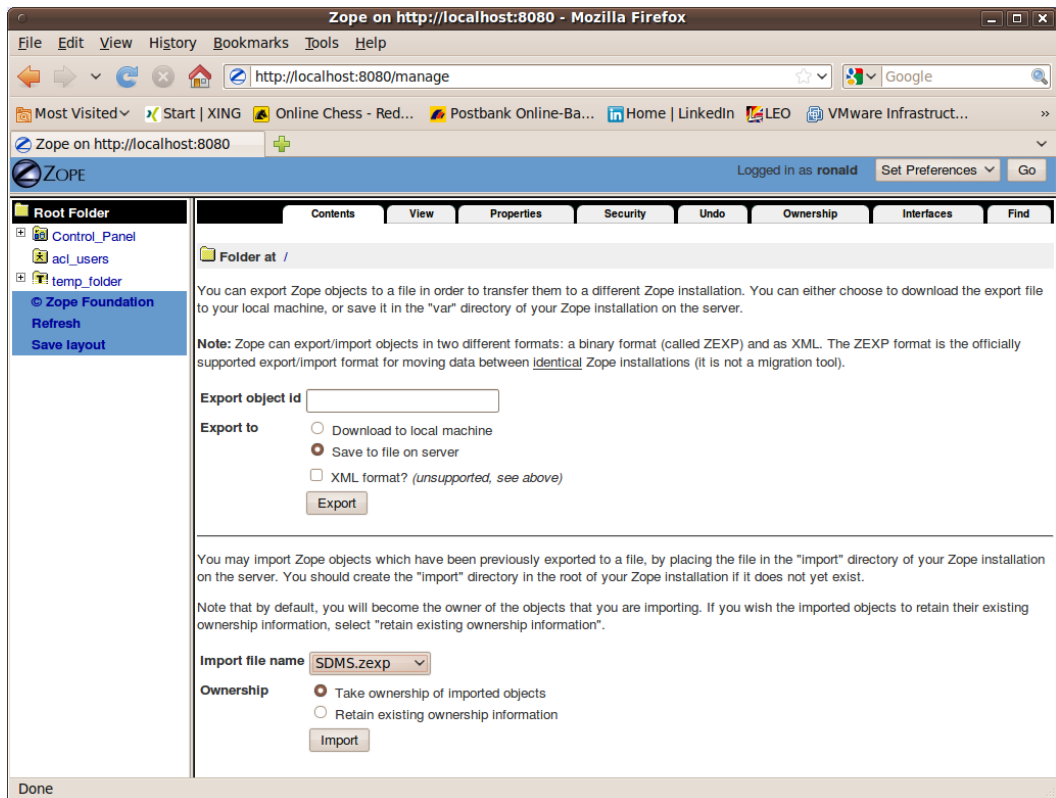
Installing the Zope server



The front end software is now loaded into Zope (Import button):

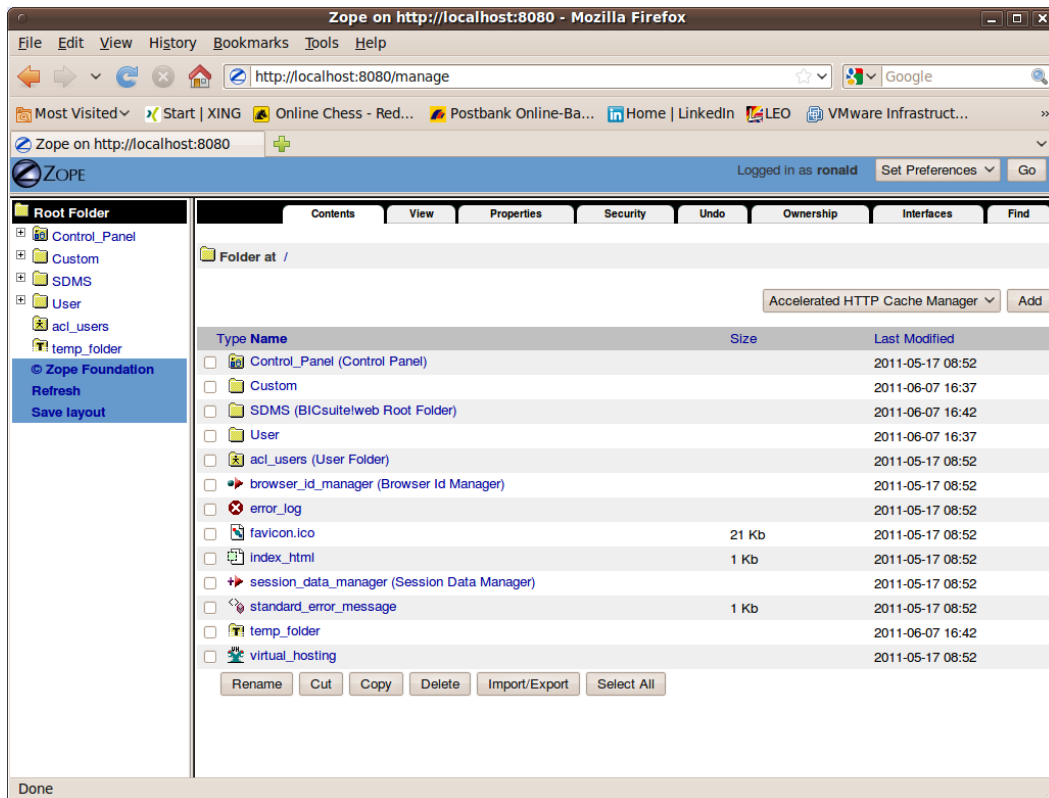
- Import into the folder / SDMS.zexp.
- In the folder /SDMS/Install, mark and copy the folders User and Custom.
- Create the folders User and Custom in the folder / by pasting them.

Installing the Zope server



If everything has been carried out without any errors, the interface will look like the screenshot below.

Installing the Zope server



6. Configure the server connections

The server connections are also configured in the Zope Management user interface. To do this, log on as the user `sdmsadm`.

The `SDMSServers` Python script is edited in the folder `Custom`. This script delivers a dictionary which has to contain an entry for every schedulix Server that is to be addressed by this schedulix GUI installation. The entry looks like this:

```
# Server name that identifies the server in the schedulix
# user interface
'servername' : {
    # TCP/IP hostname or address under which the schedulix Server is running
    'HOST'      : 'hostname',

    # Port at which the schedulix Server is addressed
    'PORT'      : '2506',

    # BASIC, PROFESSIONAL, ENTERPRISE
    'VERSION'   : 'BASIC',

    # Optional property stating whether the schedulix GUI should
    # cache the server connections
```

Installing the Zope server

```
'CACHE'      : 'Y'

# Optional property stating for how long cached schedulix GUI
# server connections should continue to be valid
# Default setting is 60 seconds, only significant if 'CACHE' : 'Y'
'TIMEOUT'    : '60'
}
```

An entry with the name `DEFAULT` must exist for bootstrapping. This entry can be deleted after the user has been set up (who obviously should not then use this connection).

7. Open the schedulix interface

The user interface is now available at the address

```
http://localhost:8080/SDMS
```

A logon prompt is displayed when this page is opened. When the user has logged on, the application is started by clicking the "Take Off" button.

How to work with the interface is described in the relevant documentation.