

independIT Integrative Technologies GmbH  
Bergstraße 6  
D-86529 Schrobenhausen



**schedulix**

# **Installationshandbuch Release 2.5.1**

Dieter Stubler      Ronald Jeninga

July 18, 2013

Copyright © 2013 independIT GmbH

#### **Rechtlicher Hinweis**

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdrucks und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung der independIT GmbH in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>1</b>
<b>1 Voraussetzungen</b>	<b>1</b>
Compile Umgebung . . . . .	1
schedulix Server und Clients . . . . .	2
Zope Application Server . . . . .	2
<b>2 Compile des Systems</b>	<b>3</b>
Generelle Vorbereitung . . . . .	3
Compile . . . . .	3
<b>3 Installation in einer Linux Umgebung</b>	<b>5</b>
Installation des schedulix Servers . . . . .	5
Einleitung . . . . .	5
Installation . . . . .	5
Installation mit Postgres . . . . .	9
Einleitung . . . . .	9
Installation . . . . .	9
Installation mit mySQL . . . . .	11
Einleitung . . . . .	11
Installation . . . . .	11
Installation mit Ingres . . . . .	12
Einleitung . . . . .	12
Installation . . . . .	12
Installation des Zope Servers . . . . .	14
Einleitung . . . . .	14
Installation . . . . .	14

# 1 Voraussetzungen

## Compile Umgebung

Um aus dem Source Paket auf einem Linux System die benötigten Executables zu erstellen, wird folgende Software benötigt:

- Oracle(Sun) Java 1.6 JDK oder höher  
<http://www.oracle.com/technetwork/java/index.html>  
Alternativ dazu ein OpenJDK 1.6 oder höher  
<http://openjdk.java.net>
- gcc, gcc-c++  
<http://gcc.gnu.org>
- gnu make  
<http://www.gnu.org/software/make>
- jflex  
<http://jflex.de>
- jay  
Das jay Executable wird zwar im Paket mitgeliefert, aber einen Hinweis auf die Originalquellen bzw. -executables sollte hier nicht fehlen.  
<http://www.cs.rit.edu/~ats/projects/lp/doc/jay/package-summary.html>
- Eines der folgenden RDBMS Systeme mit zugehörigem JDBC Interface
  - PostgreSQL  
<http://www.postgresql.org>  
JDBC für 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

Im Paket sind einige Beispiele vorhanden, die ein installiertes SWT voraussetzen. Damit zum Einen der Compile nicht abbricht und zum Anderen die Examples funktionieren, wird ein Eclipse-SWT benötigt.

<http://www.eclipse.org/swt>

In vielen Fällen können die benötigte Software Pakete einfach über ein Package Manager wie yum, rpm oder dpkg installiert werden.

## schedulix Server und Clients

Zur Installation des schedulix Servers wird folgende Software benötigt:

- Oracle(Sun) Java 1.6 SE jre  
<http://www.oracle.com/technetwork/java/index.html>  
Alternativ dazu ein OpenJDK 1.6 oder höher  
<http://openjdk.java.net>
- Eines der folgenden RDBMS Systeme mit zugehörigem JDBC Interface
  - PostgreSQL  
<http://www.postgresql.org>  
JDBC für 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

Das Web Frontend wird von dem Zope Application Server bereitgestellt. Zur Installation des Zope Servers wird folgende Software benötigt:

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

## 2 Compile des Systems

### Generelle Vorbereitung

Es ist sinnvoll eine Software mit zentraler Bedeutung unter einem eigenen Account zu installieren. Dies vereinfacht die Administration und schützt gegen Missbrauch. Im nachfolgenden wird davon ausgegangen, dass die Umwandlung und Installation unter dem Account `schedulix` erfolgt. Als Home-Verzeichnis wird `/home/schedulix` angenommen. Selbstverständlich sind dies alles nur Vorschläge. Es gibt keine technische Notwendigkeit diese Vorschläge an zu nehmen. Allerdings muss die Anleitung bei Änderungen entsprechend interpretiert werden.

Wie ein Benutzer angelegt werden kann steht, im Installationskapitel auf Seite 5 dokumentiert.

### Compile

Um nach der Installation der benötigten Pakete das System erfolgreich zu übersetzen, müssen noch einige Umgebungsvariablen gesetzt werden bevor "make" die eigentlichen Arbeit machen kann.

Da weder für das Umwandeln als auch für die Installation an sich keine besondere Rechte benötigt werden, wird unter dem User `schedulix` gearbeitet.

#### 1. Auspacken des Pakets

So weit das Paket noch nicht ausgepackt war, sollte dies noch passieren. Etwa

```
tar xvzf schedulix-2.5.1.tgz
```

Dies erstellt im aktuellen Verzeichnis ein Verzeichnis `schedulix-2.5.1`. Um auch nach Upgrades weiterhin wie gewohnt arbeiten zu können, empfiehlt es sich einen symbolischen Link auf dieses Verzeichnis an zu legen.

```
ln -s schedulix-2.5.1 schedulix
```

Im folgenden wird davon ausgegangen, dass der Link vorhanden ist. Sollte dies nicht der Fall sein, muss in den Code-Beispielen der Name `schedulix` entsprechend durch `schedulix-2.5.1` ersetzt werden. Weiterhin wird davon ausgegangen, dass die Installation im Home-Verzeichnis des Benutzers `schedulix` erfolgt.

## Compile

### 2. Setzen der Umgebungsvariablen

Nun müssen einige Umgebungsvariablen gesetzt werden. Die folgende Kommandos stammen von einer Installation auf einem CentOS (<http://www.centos.org>) Linux. In vielen Fällen können die Befehle eins-zu-eins übernommen werden, aber sie sind abhängig von der genauen Linux Distribution und installierten 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
```

Es empfiehlt sich diese Einstellungen in die `.bashrc` auf zu nehmen, zumindest so lange das Umwandeln des Systems noch nicht erledigt ist.

### 3. make

Nun bleibt nur noch das tatsächliche Umwandeln des Systems übrig.

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

Als letzte Aktion wird ein jar-File erzeugt und unter `~/schedulix/lib` abgelegt.

Wenn der Compile erfolgreich durchgelaufen ist, kann mit der normale Installation der Software fortgefahren werden.

# 3 Installation in einer Linux Umgebung

## Installation des schedulix Servers

### Einleitung

Manche benötigte Software wird in dieser Anleitung lokal installiert, obwohl auch eine globale Installation möglich ist. Eine lokale Installation hat zum Vorteil, dass Versionskonflikte vermieden werden. Selbstverständlich obliegt es der Verantwortung des Systemadministrators diese Entscheidung für sich zu treffen.

### Installation

Die Installation des schedulix Scheduling Servers ist einfach. Es bedarf einige einfache Handlungen die im Folgenden erläutert werden.

Wenn (Beispiel)Kommandos vorgestellt werden, wird als Prompt inormalerweise ein `$` gezeigt. Diese Kommandos werden dann unter dem neu anzulegenden Account `schedulix` ausgeführt. In einigen Fällen wird der privilegierte Account `root` benötigt. Dies wird dadurch gekennzeichnet, dass als Prompt ein `#` gezeigt wird.

#### 1. User `schedulix` anlegen

Es gibt keine Notwendigkeit den User `schedulix` zu nennen. Damit kann der Name auch einer beliebigen Kovention angepasst werden. In diesem Dokument wird davon ausgegangen, dass der User `schedulix` heisst.

Unter Ubuntu Linux kann ein User folgendmaßen angelegt werden:

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

Alle nachfolgende Aktionen werden unter User `schedulix` ausgeführt, es sei denn es wird explizit anders angegeben.

#### 2. Java installieren

Es wird davon ausgegangen, dass die Java-Installation sich im Verzeichnis `/home/schedulix/software/java` befindet.



### 3. Herunterladen und Installieren eines von schedulix unterstützten Datenbank Management Systems.

schedulix für Linux unterstützt derzeit die Systeme:

- Postgres (Seite 9)
- Mysql (Seite 11)
- Ingres (Seite 12)

Für die Installation des gewählten Datenbanksystems sowie die Anpassung der Konfiguration des schedulix Enterprise Scheduling Systems wird auf die entsprechenden nachfolgenden Abschnitte verwiesen.

### 4. Software auspacken

tar-Archiv auspacken im Homeverzeichnis von schedulix. Etwa:

```
$ tar xvzf schedulix-2.5.1.tgz
```

Symbolic link anlegen:

```
$ ln -s schedulix-2.5.1 schedulix
```

### 5. Konfiguration erstellen

#### a) Benutzerumgebung

Um mit dem schedulix System arbeiten zu können, müssen folgende Variablen gesetzt werden:

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

Es hat sich als gute Praxis erwiesen die Konfiguration des Systems außerhalb des Installationsverzeichnisses zu legen. Damit werden spätere Upgrades wesentlich erleichtert. Da die Variablen von allen Benutzern des Systems gesetzt werden müssen, kann es sinnvoll sein die Zuweisungen (und exports) in einer eigenen Datei zu schreiben, und diese dann im .profile oder .bashrc zu sourcen.

#### b) Softwareumgebung

Unter \$BICSUITEHOME/etc liegen einige Vorlagen für Konfigurationsdateien, die als Basis für die Systemkonfiguration verwendet werden sollten. Diese müssen dazu ohne die Endung ".template" ins Verzeichnis \$BICSUITECONFIG kopiert werden.

Etwa

## Installation des schedulix Servers

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

Anschliessend müssen die Dateien natürlich der Umgebung angepasst werden.

Die Datei `bicsuite.conf` setzt einige default Einstellungen und muss im Allgemeinen nicht angepasst werden. Allerdings kann man sich überlegen das Logging des Systems außerhalb des Installationsverzeichnis stattfinden zu lassen. In dem Fall muss lediglich die Variable `BICSUITELOGDIR` entsprechend angepasst werden. Das in `BICSUITELOGDIR` gesetzte Verzeichnis muss vorhanden sein.

Die Datei `java.conf` beschreibt die zu verwendenden Java-Umgebung. Insbesondere muss der Pfad zum JDBC-Treiber eingegeben werden. Weiterhin wird die Speicherkonfiguration des Servers geregelt. Auch in großen Umgebungen muss dazu normalerweise nur die Variable `BICSUITEMEM` angepasst werden.

Die Datei `server.conf` enthält die Serverkonfiguration. Angepasst werden müssen hier die Einstellungen für die Verbindung des schedulixScheduling Serves zu seinem RDBMS Repository. Mehr dazu finden Sie im jeweiligen Kapitel zum eingesetzten RDBMS.

Weiterhin muss in dieser Datei das Property `hostname` auf den Hostnamen oder die IP-Adresse des Servers gesetzt werden.

Die Datei `jobserver.conf` wird hier nicht benötigt, dient aber als Vorlage für die Jobserver Konfiguration.

### 6. Datenbank Einrichten

Abhängig davon welches Datenbanksystem Sie nutzen möchten, befolgen Sie die Anleitung zur Einrichtung der Datenbank.

Für

- ingres, siehe Seite 12,
- mySQL, siehe Seite 11, und für
- postgresql, siehe Seite 9.

### 7. Server hochfahren

Die Installation ist nun im Wesentlichen abgeschlossen. Was noch bleibt ist das Starten des Servers und, bei Bedarf, das Einspielen der Beispiele.

Der Server kann mittels

```
$ server-start
```

gestartet werden.

### 8. Anlegen der Datei `.sdmshrc`

Die Datei `.sdmshrc` wird, falls vorhanden, von allen schedulix Kommandozeilen Werkzeugen gelesen um Kommandozeilen Parameter vorzubelegen. Im folgenden wird davon ausgegangen, dass diese Datei existiert und für User, Passwort, Host und Port die korrekten Werte gesetzt enthält. Die Datei `.sdmshrc` wird im Home-Verzeichnis des Linux Benutzers angelegt.

Ein Beispiel für den Inhalt ist:

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

Wichtig: Da die Datei die Daten für den Zugang zum Scheduling Server enthält, sollten die Datei-Rechte so gesetzt sein, dass nur der Owner die Datei lesen kann.

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

### 9. Convenience Package installieren

Das convenience Package installiert eine übliche Konfiguration eines exit state Models.

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

### 10. Beispiele installieren

Das Installieren der Beispiele besteht aus zwei Teilen. Zum Einen werden drei sogenannte Jobserver angelegt, welche für die nachfolgenden Ablaufdefinitionen benötigt werden. Zum Anderen werden Beispiele für Ablaufdefinitionen in den Server geladen.

#### a) Anlegen der Jobserver

Um die Jobserver anzulegen, muss nur ein Skript ausgeführt werden

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

#### b) Einspielen der Ablaufdefinitionen

Zum Einspielen der Ablaufdefinitionen, werden folgende Befehle eingegeben:

## Installation mit Postgres

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

Da die Beispiele davon ausgehen, dass die Jobserver bereits angelegt wurden, ist die obige Reihenfolge zwingend.

## Installation mit Postgres

### Einleitung

Diese Anleitung erhebt nicht den Anspruch eine genaue Beschreibung der Installation des Datenbanksystems zu sein. Dazu wird auf die Postgres Dokumentation verwiesen. Im Normalfall sollte es mit diese Anleitung allerdings möglich sein eine "standard" Installation durch zu führen.

### Installation

1. Herunterladen und installieren der aktuellen postgres Version

Normalerweise wird für jede Linux Distribution ein Postgres Package angeboten. Dieses Package, sowie ein Package für den JDBC Treiber für Postgres sollte problemlos installiert werden können.

2. Konfiguration der Datei `pg_hba.conf`

Damit sich der schedulix Scheduling Server mit Benutzer und Passwort bei postgresql authentifizieren kann, muss folgende Zeile in die Postgres Konfigurationsdatei `pg_hba.conf` (diese liegt typischerweise unter `/var/lib/pgsql/<version>/conf`) aufgenommen werden:

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

Postgresql muss dann neu gestartet werden.

3. Anlegen des postgres users schedulix

Führen Sie als User postgres den Befehl `createuser` wie im Beispiel (Version 8) aus:

```
$ 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
```

## Installation mit Postgres

beziehungsweise, für Version 9:

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

Das eingegebene Passwort wird später noch benötigt.

### 4. Anlegen der Repository Datenbank schedulixdb

Legen Sie nun als Benutzer schedulix die Datenbank für das Repository wie im untenstehenden Beispiel an:

```
$ createdb schedulixdb
```

### 5. Anlegen und Initialisierung der Datenbanktabellen

Um das benötigte Datenbankschema anzulegen, wechseln Sie in das schedulix sql Verzeichnis und rufen Sie das Postgres Utility `psql` wie im untenstehenden Beispiel auf:

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

### 6. Konfigurieren der Datenbankverbindung in der schedulix Server Konfigurationsdatei `$BICSUITECONFIG/server.conf`

Ändern Sie folgende Properties wie angegeben:

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

Die `DbUrl` ist etwas abhängig von der installierten postgresql Version. Unter Version 9.2 lautet sie

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

### 7. Konfigurieren sie den schedulix Java Class Path für postgres JDBC

In der Konfigurationsdatei `$BICSUITECONFIG/java.conf` muss nun nur noch den Pfad zum Postgres JDBC Treiber an dem `CLASSPATH` angehängt werden.

Etwa

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

## Installation mit MySQL

### Einleitung

Diese Anleitung erhebt nicht den Anspruch eine genaue Beschreibung der Installation des Datenbanksystems zu sein. Dazu wird auf die MySQL Dokumentation verwiesen. Im Normalfall sollte es mit diese Anleitung allerdings möglich sein eine "standard" Installation durch zu führen.

### Installation

1. Herunterladen und Installieren der aktuellen MySQL Version.

Für die meisten Linux Distributionen gibt es fertige MySQL Packages. Diese können mit den entsprechenden Tools einfach installiert werden.

Im Rahmen dieser Installation wird nach einem Password für den MySQL root-User gefragt (nicht zu verwechseln mit dem Linux root-User). Dieses Password wird im nächsten Schritt wieder benötigt.

Da schedulix für den Zugriff auf die Datenbank eine JDBC Connection aufbaut, muss auch der MySQL JDBC Treiber installiert werden.

2. Anlegen des mysql Users schedulix und der Datenbank schedulixdb

Starten Sie das Utility `mysql` und melden Sie sich als MySQL root-User an um den User schedulix sowie die Datenbank schedulixdb an zu legen:

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

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 5.1.54-lubuntu4 (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_passwort';
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
```

## Installation mit Ingres

### 3. Anlegen und Initialisierung der Datenbanktabellen

Führen sie folgende Kommandos aus:

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

### 4. Konfigurieren der Datenbankverbindung in der schedulix Server Konfigurationsdatei \$BICSUITECONFIG/server.conf

Ändern Sie folgende Properties wie angegeben:

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

### 5. Konfigurieren sie den schedulix Java Class Path für mysql JDBC

In der Konfigurationsdatei \$BICSUITECONFIG/java.conf muss nun nur noch den Pfad zum Postgres JDBC Treiber an dem CLASSPATH angehängt werden.  
Etwa

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

## Installation mit Ingres

### Einleitung

Diese Anleitung erhebt nicht den Anspruch eine genaue Beschreibung der Installation des Datenbanksystems zu sein. Dazu wird auf die Ingres Dokumentation verwiesen. Im Normalfall sollte es mit diese Anleitung allerdings möglich sein eine "standard" Installation durch zu führen.

### Installation

#### 1. Installation von Ingres

Wir gehen davon aus, dass das ingres System unter User `ingres` installiert wird. Der Installations-Identifizier wird hier als `II`, was dem Standard Wert entspricht, angenommen.

#### 2. Anlegen des Benutzers schedulix

Um den Benutzer `schedulix` in das Ingres System bekannt zu machen, gibt es zwei Möglichkeiten. Als erste kann der Benutzer mit Hilfe des Tools `accessdb` angelegt werden. Diese Möglichkeit wird hier nicht weiter erläutert.

## Installation mit Ingres

Die zweite Möglichkeit ist das Anlegen des Benutzers mittels SQL Befehle. Dazu starten Sie als Ingres den SQL Terminal Monitor:

```
$ 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. Anlegen der Repository Datenbank schedulixdb

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

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

Creation of database 'schedulixdb' completed successfully.
```

### 4. Anlegen und Initialisierung der Datenbanktabellen

Zum Anlegen der benötigten Tabellen führen Sie folgende Befehle durch:

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

### 5. Konfigurieren der Datenbankverbindung in der schedulix Server Konfigurationsdatei \$BICSUITECONFIG/server.conf

Ändern Sie folgende Properties wie angegeben:



## Installation des Zope Servers

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

### 6. Konfigurieren sie den schedulix Java Class Path für ingres JDBC

In der Konfigurationsdatei `$BICSUITECONFIG/java.conf` muss nun nur noch den Pfad zum Ingres JDBC Treiber an dem CLASSPATH angehängt werden.

Etwa

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

## Installation des Zope Servers

### Einleitung

Um die schedulix!web User Interface Oberfläche nutzen zu können, muss ein Zope Application Server aufgesetzt werden.

### Installation

#### 1. Installation virtualenv

```
$ easy_install virtualenv
```

#### 2. Erzeugen virtuelle python Umgebung für die Zope installation

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

#### 3. Installieren Zope2 Software

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

#### 4. Erzeugen einer Zope Instanz für schedulix!web

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

## Installation des Zope Servers

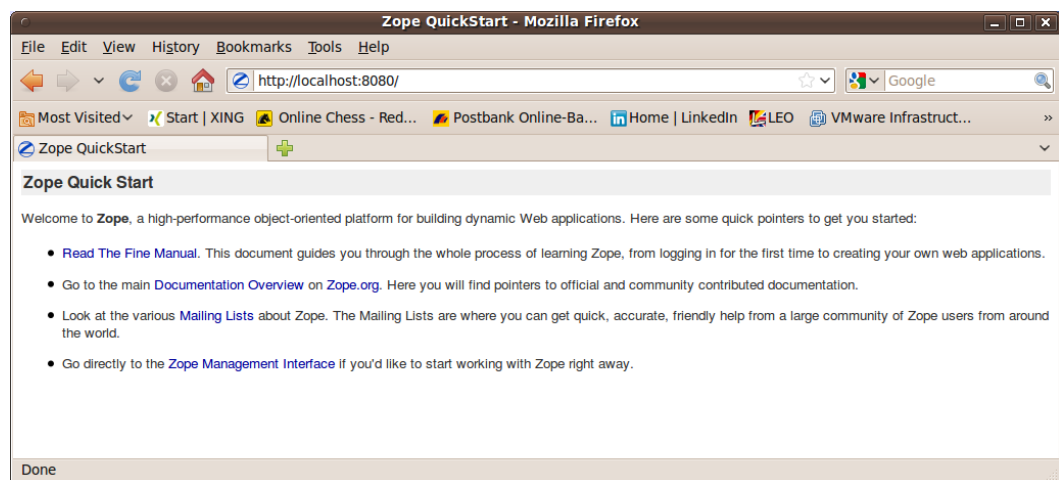
Das Passwort kann beliebig gewählt werden und wird später wieder benötigt.  
Zum Testen wird der Zope Server kurz gestartet:

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

Im Internet Browser sollte die Url

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

nun die Zope Quick Start Seite anzeigen.



Die Zope Instanz wird nun wieder angehalten:

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

### 5. Installieren der schedulix!Web Komponenten

Um die schedulix!Web Komponenten zu installieren, muss die Zope Installation um einige Module erweitert werden.

```
$ 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 .
```

Nun muss die Zope Instanz wieder gestartet werden um die Änderungen auch Zope-seitig bekannt zu machen.

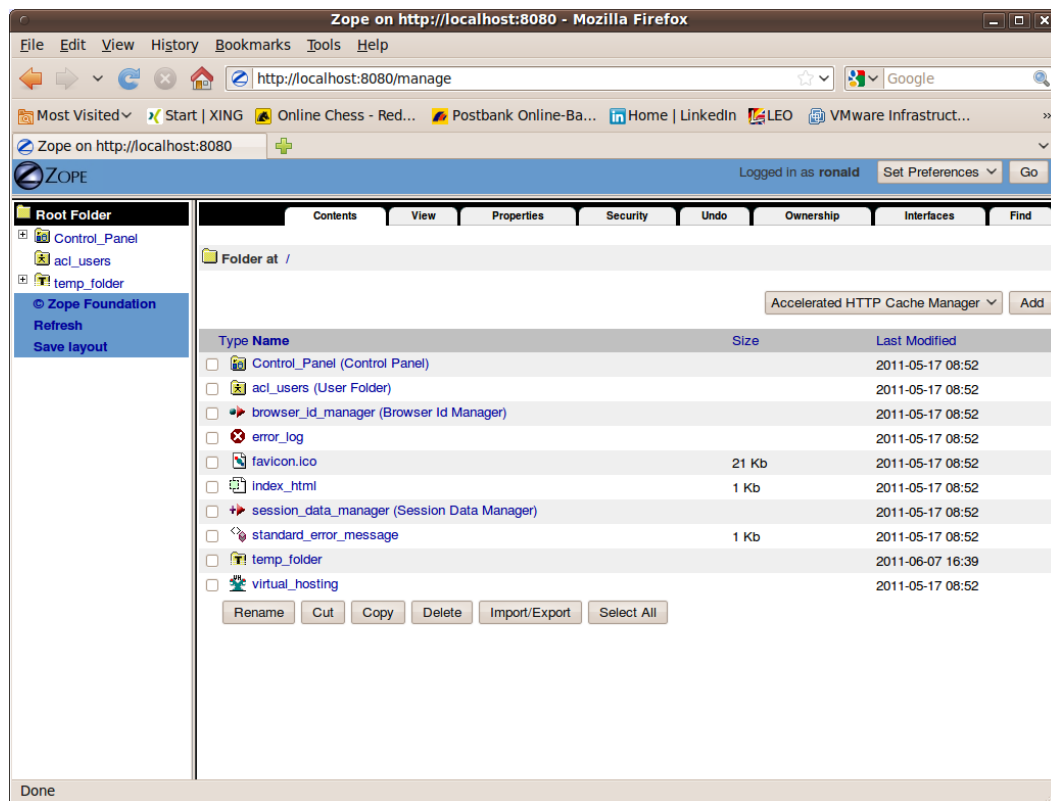
## Installation des Zope Servers

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

Die Zope Management Oberfläche wird nun unter der Adresse

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

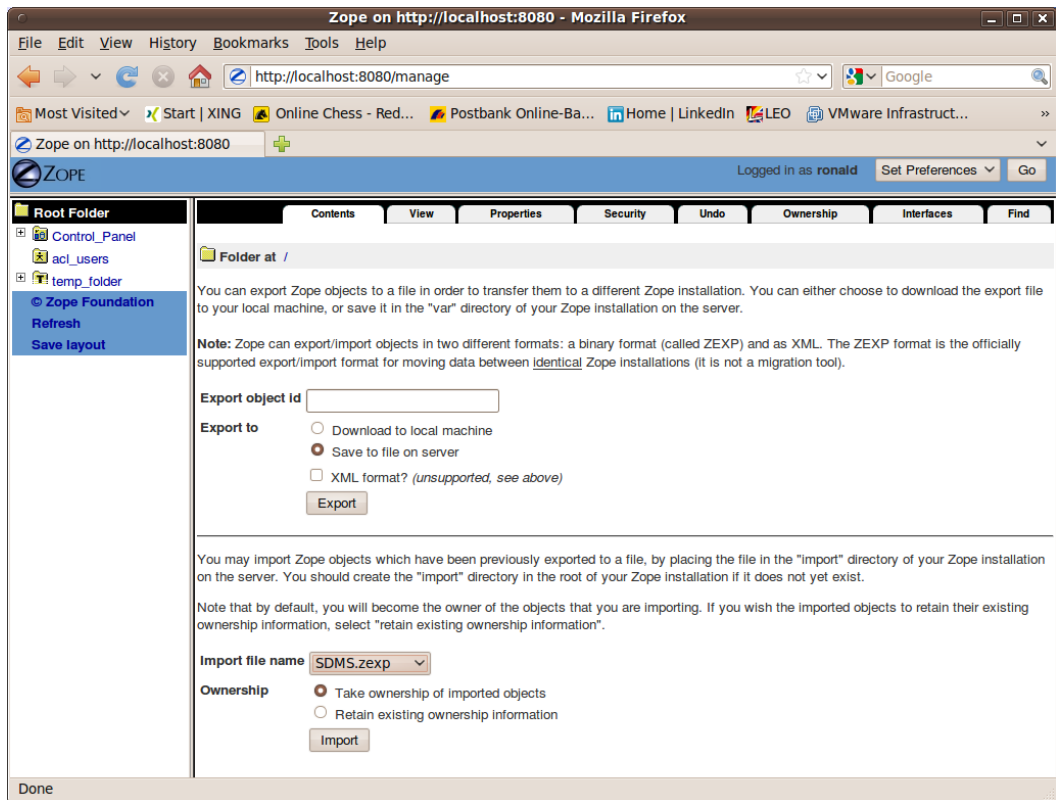
mit Hilfe eines Browsers geöffnet. Dazu wird der Benutzer `sdmsadm` mit dem von Ihnen vergebenen Passwort, in diesem Dokument `sdmsadm_passwort`, benutzt.



Es wird jetzt die Frontend Software in Zope geladen (import Button):

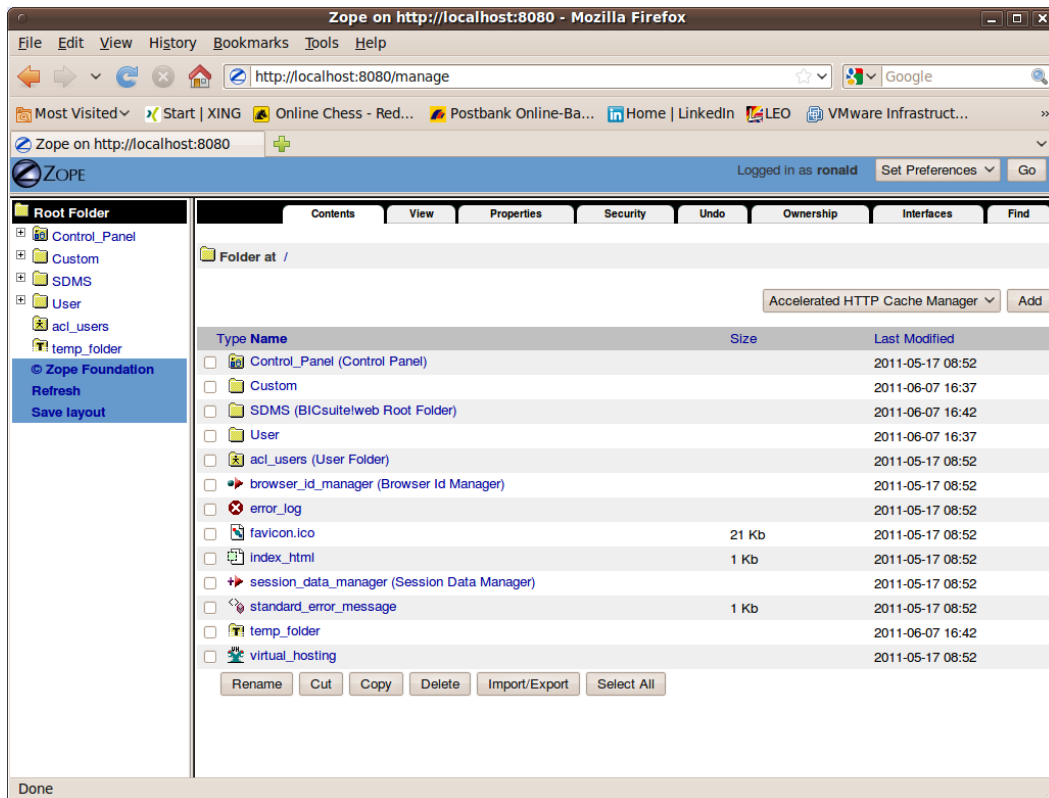
- Im Folder / SDMS.zexp importieren.
- Im Folder /SDMS/Install die Folder User und Custom anwählen und mit Copy kopieren
- Im Folder / mit Paste die Folder User und Custom erzeugen

## Installation des Zope Servers



Wenn nun alles fehlerfrei durchgeführt werden konnte, sieht die Oberfläche wie auf dem nachfolgenden Bild aus.

## Installation des Zope Servers



### 6. Serververbindungen konfigurieren

Das Konfigurieren der Serververbindungen erfolgt ebenfalls aus der Zope Management Oberfläche heraus. Dazu meldet man sich als Benutzer `sdmsadm` an.

Im Folder `Custom` wird das PythonScript `SDMSServers` editiert. Dieses Skript liefert ein Dictionary, welches für jeden `schedulix!server`, der von dieser `schedulix!web` Installation angesprochen werden soll, einen Eintrag der Form:

```
# Servername unter dem der Server in der schedulix!web Oberfläche
# sichtbar ist
'servername' : {
    # TCP/IP Hostname oder Adresse auf dem der schedulix!server laeuft
    'HOST'      : 'hostname',

    # Port unter dem der schedulix!server angesprochen wird
    'PORT'      : '2506',

    # BASIC
    'VERSION'   : 'BASIC',

    # optionales Property, ob schedulix!web Serververbindungen cachten soll
    'CACHE'     : 'Y'

    # optionales Property, wie lange gecachte schedulix!web
```

## Installation des Zope Servers

```
# Serververbindungen gueltig sein sollen
# default ist 60 sekunden, nur von Bedeutung falls 'CACHE' : 'Y'
'TIMEOUT' : '60'
}
```

enthalten muss. Fürs Bootstrapping muss einen Eintrag mit dem Namen `DEFAULT` vorhanden sein. Dieser Eintrag kann nach dem Einrichten der Benutzer (die dann diese Connection nicht benutzen sollten, natürlich) entfernt werden.

### 7. Die schedulix!Web Oberfläche öffnen

Die Benutzeroberfläche steht nun unter der Adresse

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

bereit. Nach dem Öffnen dieser Seite erscheint eine Aufforderung zur Anmeldung. Nach der Anmeldung wird die Applikation dann mit dem "Take Off" Button gestartet.

Für das weitere Arbeiten mit der Oberfläche sei nun auf die dazu gehörende Dokumentation verwiesen.