

independIT Integrative Technologies GmbH
Bergstraße 6
D-86529 Schrobenhausen



BICsuite

Installationshandbuch Release 2.10

Dieter Stubler

Ronald Jeninga

6. März 2024

Copyright © 2024 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

Inhaltsverzeichnis	1
1 Voraussetzungen	3
BICsuite Server und Clients	3
Zope Application Server	4
Zope https	4
2 Installation in einer Linux-Umgebung	5
Installation des BICsuite Servers	5
Installation eines BICsuite Clients	8
Beispiel Installation eines Jobservers	10
Szenario	10
Voraussetzungen	10
Installation	11
Installation mit Postgres	13
Einleitung	13
Installation	13
Installation mit MySQL	14
Einleitung	14
Installation	14
Installation mit Ingres	16
Einleitung	16
Installation	16
Installation mit Informix	17
Einleitung	17
Installation	17
Installation mit DB2	19
Einleitung	19
Installation	19
Installation mit Oracle	20
Einleitung	20
Installation	21
Konfiguration von TLS/SSL Verbindungen	21
Einleitung	21
Konfiguration von TLS/SSL	22
Installation des Zope Servers	25
Einleitung	25

Installation (Zope2)	25
HTTPS mit Hilfe eines vorgeschaltetem Apache Servers	31
Installation der Zope HTTPS Erweiterungen	32
Installation (Zope4+)	36
Migration einer bestehenden BICsuite!Web Zope2 Datenbank nach Zope4+	40
HTTPS mit Hilfe eines vorgeschaltetem Apache Servers	40
SSO für BICsuite mit Zope	42
Einleitung	42
Vorgehen	42
Kerberos Installation und Konfiguration	43
Apache Webserver und Module	45
Zope Erweiterung und Konfiguration	49
Konfiguration des BICsuite Servers	57
Einstellungen an Benutzerseite	58
Administration des Zope Servers	59
3 Installation in einer Windows Umgebung	61
Installation des BICsuite Servers	61
Einleitung	61
Installation	61
Installation mit Postgres	65
Einleitung	65
Installation	65
Installation mit MySQL	67
Einleitung	67
Installation	68
Installation mit Microsoft™ SQL Server	70
Einleitung	70
Installation	70
Installation mit Ingres	72
Einleitung	72
Installation	72
Installation mit DB2	74
Einleitung	74
Installation	74
Installation mit Oracle Express Edition	76
Einleitung	76
Installation	76
Installation des Zope Servers	77
Einleitung	77
Installation (Zope2)	78
Installation (Zope4+)	84

1 Voraussetzungen

BICsuite Server und Clients

Zur Installation des BICsuite Servers wird folgende Software benötigt:

- Oracle(Sun) Java 1.8 SE jre
<http://www.oracle.com/technetwork/java/index.html>
- Microsoft Visual C++ 2010 Redistributable package (x86)
(Nur für eine Installation unter Windows)
<http://www.microsoft.com>
- Einer der folgenden Datenbanksysteme
 - 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>
 - Oracle
<http://www.oracle.com>
 - Microsoft SQL Server
<http://www.microsoft.com>
Microsoft SQL Server JDBC Treiber
<http://www.microsoft.com>
- Java Native Access (JNA)
Um die Notwendigkeit einer JNI Library zu umgehen wird ab der Version 2.6 die JNA Bibliothek genutzt. Diese Bibliothek wird nur für den Java Jobexecutor benötigt.
<https://github.com/twall/jna>

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-virtualenv
<http://pypi.python.org>
- python-dev
<http://www.python.org>
- python setuptools (nur Windows)
<http://pypi.python.org/pypi/setuptools>
- pywin32 (optional, nur Windows)
<http://sourceforge.net/projects/pywin32>

Zope https

Um https in Kombination mit Zope benutzen zu können, wird zusätzlich folgende Software benötigt:

- swig
<http://www.swig.org>
- python M2Crypto package
<http://pypi.python.org>
- openssl
www.openssl.org
(Oder Vergleichbares)

2 Installation in einer Linux-Umgebung

Installation des BICsuite Servers

Die Installation des BICsuite Scheduling Servers ist einfach. Es bedarf nur einiger Handlungen die im Folgenden erläutert werden:

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

1. User `bicsuite` anlegen

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

Unter Ubuntu Linux kann ein User folgendermaßen angelegt werden:

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

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

2. Herunterladen und Installieren eines von BICsuite unterstützten Datenbank Management Systems.

BICsuite für Linux unterstützt derzeit die Systeme:

- Postgres (Seite [13](#))
- MySQL (Seite [14](#))
- Ingres (Seite [16](#))
- Oracle (Seite [20](#))

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

3. Software auspacken

tar-Archiv auspacken im Homeverzeichnis von `bicsuite`. Etwa:

```
$ tar xvfz bicsuite-2.10.tgz
```

Symbolic Link anlegen:

```
$ ln -s bicsuite-2.10 bicsuite
```

4. Konfiguration erstellen

a) Benutzerumgebung

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

```
BICSUITEHOME=/home/bicsuite/bicsuite  
BICSUITECONFIG=/home/bicsuite/etc  
PATH=$BICSUITEHOME/bin:$PATH  
SWTJAR=/usr/lib/java/swt.jar  
JNAJAR=/usr/share/java/jna.jar
```

Es hat sich in der Praxis als vorteilhaft 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 `$BICSUITEHOME/etc` liegen einige Vorlagen für Konfigurationsdateien, die als Basis für die Systemkonfiguration verwendet werden sollten. Diese müssen dazu ohne die Endung `".template"` ins Konfigurationsverzeichnis `$BICSUITECONFIG` kopiert werden.

Etwa

```
$ 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 Installationsverzeichnisses stattfinden zu lassen. In diesem Fall muss lediglich die Variable `BICSUITELOGDIR` entsprechend angepasst werden. Das in `BICSUITELOGDIR` gesetzte Verzeichnis muss vorhanden sein.

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

Die Datei `server.conf` enthält die Serverkonfiguration. Angepasst werden müssen hier die Einstellungen für die Verbindung des BICsuite Scheduling 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.

5. Datenbank einrichten

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

Für

- Ingres, siehe Seite 16,
- MySQL, siehe Seite 14,
- Oracle, siehe Seite 20, und für
- PostgreSQL, siehe Seite 13.

6. 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.

7. Anlegen der Datei `.sdmshrc`

Die Datei `.sdmshrc` wird, falls vorhanden, von allen BICsuite 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 -lG ~/.sdmshrc
-rw----- 1 bicsuite 73 2011-11-09 09:28 /home/bicsuite/.sdmshrc
```

8. Convenience Package installieren

Das Convenience Package installiert eine übliche Konfiguration eines Exit State-Modells.

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

9. Beispiele installieren (optional)

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_jobservers.sh
```

b) Einspielen der Ablaufdefinitionen

Zum Einspielen der Ablaufdefinitionen werden folgende Befehle eingegeben:

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

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

Installation eines BICsuite Clients

Die Installation eines BICsuite Scheduling Clients ist einfach. Es bedarf nur einiger Handlungen die im Folgenden erläutert werden.

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

1. User `bicsuite` anlegen

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

Unter Ubuntu Linux kann ein User folgendermaßen angelegt werden:

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

Alle nachfolgenden Aktionen werden unter User bicsuite ausgeführt, es sei denn es wird explizit anders angegeben.

2. Software auspacken

tar-Archiv auspacken im Homeverzeichnis von bicsuite. Etwa:

```
$ tar xvzf bicsuite-2.10.tgz
```

Symbolic Link anlegen:

```
$ ln -s bicsuite-2.10 bicsuite
```

3. Konfiguration erstellen

a) Benutzerumgebung

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

```
BICSUITEHOME=/home/bicsuite/bicsuite
BICSUITECONFIG=/home/bicsuite/etc
PATH=$BICSUITEHOME/bin:$PATH
SWTJAR=/usr/lib/java/swt.jar
JNAJAR=/usr/share/java/jna.jar
```

Es hat sich in der Praxis als vorteilhaft 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.

Für eine Client-Installation benötigen wir die Dateien `bicsuite.conf` und `java.conf`.

Diese müssen dazu ohne die Endung `".template"` ins Konfigurationsverzeichnis `$BICSUITECONFIG` kopiert werden.

```
$ cp $BICSUITEHOME/etc/bicsuite.conf.template \
    $BICSUITECONFIG/bicsuite.conf
$ cp $BICSUITEHOME/etc/java.conf.template \
    $BICSUITECONFIG/java.conf
```

Die Datei `bicsuite.conf` setzt einige Default-Einstellungen und muss im Allgemeinen nicht angepasst werden.

Die Datei `java.conf` beschreibt die zu verwendende Java-Umgebung und muss im Allgemeinen nicht weiter angepasst werden.

4. Anlegen der Datei `.sdmshrc`

Die Datei `.sdmshrc` wird, falls vorhanden, von allen BICsuite 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 -lG ~/.sdmshrc
-rw----- 1 bicsuite 73 2011-11-09 09:28 /home/bicsuite/.sdmshrc
```

Beispiel Installation eines Jobserver

Im Folgenden wird die Installation eines Jobserver anhand eines Beispieles durchgeführt.

Szenario

Auf dem Rechner `machine_42` soll ein Jobserver Prozesse als Benutzer `arthur` ausführen. Das HOME-Verzeichnis des Benutzers sei `/home/arthur`.

Der BICsuite Server sei auf dem Rechner `scheduling_server` installiert und hört auf den Port 2506. Das System-Passwort sei `G0H0ME`.

Vorraussetzungen

Auf dem Rechner `machine_42` wurde eine BICsuite Client-Installation im HOME-Verzeichnis `/home/bicsuite` durchgeführt.

Der Benutzer `arthur` hat folgende Zugriffsberechtigungen auf die Dateien der Client-Installation:

- Leserechte auf die Dateien `java.conf` und `bicsuite.conf` im Konfigurationsverzeichnis `$BICSUITECONFIG`. Auf alle Dateien unter `/home/bicsuite/lib` werden ebenfalls Leserechte benötigt.
- Lese- und Ausführungsrechte auf alle Dateien unter `/home/bicsuite/bin`.

Installation

Damit ein Jobserver sich am BICsuite Scheduling Server anmelden kann muss der Jobserver dem BICsuite Scheduling Server bekannt gemacht und konfiguriert werden. Im Folgenden führen wir die notwendigen Schritte dazu auf Kommandozeilenebene mit dem Werkzeug `sdmsh` durch. Dies kann jedoch alternativ auch über das Web GUI getan werden.

1. Anmelden als Benutzer `arthur` auf dem Rechner `machine_42`

2. Setzen der Umgebungsvariablen in der Shell und `.bashrc`.

```
export BICSUITEHOME=/home/bicsuite/bicsuite
export BICSUITECONFIG=/home/bicsuite/etc
export PATH=$BICSUITEHOME/bin:$PATH
```

3. Testen, ob die Umgebung korrekt ist

```
sdmsh --host localhost --port 2506 --user SYSTEM --pass G0H0ME
```

Ein `SDMS>` Prompt sollte erscheinen ('exit' beendet `sdmsh`).

4. Anlegen der Verzeichnisse

```
cd $HOME
mkdir etc
mkdir taskfiles
mkdir work
mkdir log
```

5. Erzeugen einen Scopes für die Maschine `machine_42` mit `sdmsh`

```
SDMS> CREATE OR ALTER SCOPE GLOBAL.'MACHINE_42'
WITH
CONFIG = (
  'JOBEXECUTOR' = '/home/bicsuite/bicsuite/bin/jobserver',
  'HTTPHOST' = 'machine_42'
);
```

Die Pfade müssen explizit angegeben werden, die Benutzung von Umgebungsvariablen ist hier nicht möglich.

6. Erzeugen des Jobserver mit `sdmsh`

```
SDMS> CREATE OR ALTER JOB SERVER GLOBAL.'MACHINE_42'.'ARTHUR'
WITH
  PASSWORD = 'dent',
  NODE = 'machine_42',
  CONFIG = (
    'JOBFILEPREFIX' = '/home/arthur/taskfiles/',
    'DEFAULTWORKDIR' = '/home/arthur/work',
    'HTTPPORT' = '8905',
    'NAME_PATTERN_LOGFILES' = '/home/arthur/work/.*\\.log'
  );
```

Der HTTPPORT ist für die Anzeige von Job Log-Dateien aus dem Browser notwendig und ist beliebig wählbar, muss aber für alle Jobserver auf derselben Maschine eindeutig sein.

Die Pfade müssen explizit angegeben werden, die Benutzung von Umgebungsvariablen ist hier nicht möglich.

7. Anlegen der Named Resource für den Jobserver mit sdmsh

```
SDMS> CREATE OR ALTER NAMED RESOURCE RESOURCE.'JOBSERVERS'
WITH USAGE = CATEGORY;
SDMS> CREATE NAMED RESOURCE RESOURCE.'JOBSERVERS'.'ARTHUR@MACHINE_42'
WITH USAGE = STATIC;
```

8. Anlegen eines Environments für den Jobserver mit sdmsh

```
SDMS> CREATE ENVIRONMENT 'ARTHUR@MACHINE_42'
WITH RESOURCE = (RESOURCE.'JOBSERVERS'.'ARTHUR@MACHINE_42');
```

9. Anlegen der Resource im Jobserver mit sdmsh

```
SDMS> CREATE RESOURCE RESOURCE.'JOBSERVERS'.'ARTHUR@MACHINE_42'
IN GLOBAL.'MACHINE_42'.'ARTHUR' WITH ONLINE;
```

10. Erzeugen der Konfigurationsdatei \$HOME/etc/jobserver.conf für den Jobserver mit folgendem Inhalt:

```
RepoHost= scheduling_server
RepoPort= 2506
RepoUser= "GLOBAL.'MACHINE_42'.'ARTHUR'"
RepoPass= dent
```

11. Starten des Jobserver

```
jobserver-run $HOME/etc/jobserver.conf $HOME/log/jobserver.out
```

Soll der Jobserver automatisch mit dem Start des Rechners gestartet werden, ist dies vom Systemadministrator entsprechend einzurichten.

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 dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

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 BICsuite Scheduling Server mit Benutzer und Passwort bei PostgreSQL authentifizieren kann, muss folgende Zeile in die Postgres-Konfigurationsdatei `pg_hba.conf` aufgenommen werden. Diese Datei liegt typischerweise im Verzeichnis `/var/lib/pgsql/<version>/data`.

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

PostgreSQL muss dann neu gestartet werden.

3. Anlegen des Postgres Users `bicsuite`

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

```
$ createuser -P bicsuite
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
```

beziehungsweise, für Version 9:

```
$ createuser -P -d bicsuite
```

Das eingegebene Passwort wird später noch benötigt.

4. Anlegen der Repository Datenbank `bicsuitedb`

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

```
$ createdb bicsuitedb
```

5. Anlegen und Initialisierung der Datenbanktabellen

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

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

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

Ändern Sie folgende Properties wie angegeben:

```
DbPasswd=bicsuite password
DbUrl=jdbc:postgresql:bicsuitedb
DbUser=bicsuite
JdbcDriver=org.postgresql.Driver
```

Die `DbUrl` ist etwas abhängig von der installierten PostgreSQL-Version. Unter Version 8 lautet sie

```
DbUrl=jdbc:postgresql:bicsuitedb
```

7. Konfigurieren Sie den BICsuite Java Class Path für Postgres JDBC

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

Etwa

```
BICSUITECLASSPATH=$BICSUITEJAR:/usr/share/java/postgresql-jdbc4-9.2.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 dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

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 Passwort für den MySQL root-User gefragt (nicht zu verwechseln mit dem Linux root-User). Dieses Passwort wird im nächsten Schritt wieder benötigt.

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

2. Anlegen des MySQL Users `bicsuite` und der Datenbank `bicsuitedb`

Starten Sie das Utility `mysql` und melden Sie sich als MySQL `root`-User an um den User `bicsuite` sowie die Datenbank `bicsuitedb` anzulegen:

```
$ 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 bicsuite identified by 'bicsuite_passwort';
Query OK, 0 rows affected (0.01 sec)
```

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

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

```
mysql> quit
Bye
```

3. Anlegen und Initialisierung der Datenbanktabellen

Führen Sie folgende Kommandos aus:

```
$ cd $BICSUITEHOME/sql
$ mysql --user=bicsuite --password=bicsuite_passwort
  --database=bicsuitedb --execute="source mysql/install.sql"
```

4. Konfigurieren der Datenbankverbindung in der BICsuite Server-Konfigurationsdatei

`$BICSUITECONFIG/server.conf`

Ändern Sie folgende Properties wie angegeben:

```
DbPasswd=bicsuite_passwort
DbUrl=jdbc:mysql:///bicsuitedb
DbUser=bicsuite
JdbcDriver=com.mysql.jdbc.Driver
```

5. Konfigurieren Sie den BICsuite Java Class Path für MySQL JDBC

In der Konfigurationsdatei `$BICSUITECONFIG/java.conf` muss nun nur noch der Pfad zum MySQL 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 dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

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 Standardwert entspricht, angenommen.

2. Anlegen des Users `bicsuite`

Um den Benutzer `bicsuite` im 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.

Die zweite Möglichkeit ist das Anlegen des Benutzers mittels SQL-Befehl. 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 bicsuite 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 bicsuitedb

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

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

Creation of database 'bicsuitedb' completed successfully.

4. Anlegen und Initialisierung der Datenbanktabellen

Zum Anlegen der benötigten Tabellen führen Sie folgende Befehle durch:

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

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

Ändern Sie folgende Properties wie angegeben:

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

6. Konfigurieren Sie den BICsuite Java Class Path für Ingres JDBC

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

Etwa

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

Installation mit Informix

Einleitung

Diese Anleitung erhebt nicht den Anspruch eine genaue Beschreibung der Installation des Datenbanksystems zu sein. Dazu wird auf die Informix-Dokumentation verwiesen. Im Normalfall sollte es mit dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

Installation

1. Installation von Informix

Wir gehen davon aus, dass das Informix-System unter User `informix` installiert wird. Als Installationsname (`INFORMIXSERVER`) benutzen wir hier `ol_informix1210`.

2. Anlegen der Repository Datenbank `bicsuitedb`

Als Benutzer `bicsuite` rufen Sie das Utility `dbaccess` auf, und führen die Befehle wie im untenstehenden Beispiel aus. (Mit Strg-D verlassen Sie das Utility `dbaccess`).

```
[bicsuite@puma ~]$ dbaccess - -  
> create database bicsuitedb with log;
```

Database created.

>

Database closed.

Die Angabe "with log" ist essentiell, da ansonsten keine Transaktionen möglich sind. Der Scheduling Server wird dann auch nicht starten können.

3. Anlegen und Initialisierung der Datenbanktabellen

Zum Anlegen der benötigten Tabellen führen Sie folgende Befehle durch:

```
$ cd $BICSUITEHOME/sql  
(cat informix/install.lst | while read script; do  
    cat $script;  
done;) | dbaccess bicsuitedb
```

4. Konfigurieren der Datenbankverbindung in der BICsuite Server Konfigurationsdatei `$BICSUITECONFIG/server.conf`

Ändern Sie folgende Properties wie angegeben. Dabei müssen die Werte für `<computername>` und `port` der Umgebung noch angepasst werden. Auch der Name des `INFORMIXSERVER` wird angepasst werden müssen.

```
DbPasswd=<bicsuite OS password>  
DbUrl=jdbc:Informix-sqli://<computername>:<port>/bicsuitedb:\  
    INFORMIXSERVER=ol_informix1210  
DbUser=bicsuite  
JdbcDriver=com.informix.jdbc.IfxDriver
```

5. Konfigurieren Sie den BICsuite Java Class Path für Informix JDBC

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

Etwa

```
BICSUITEJDBC=/home/informix/product/jdbc/lib/ixjdbc.jar  
BICSUITECLASSPATH=$BICSUITEJAR:$BICSUITEJDBC
```

Installation mit DB2

Einleitung

Diese Anleitung erhebt nicht den Anspruch eine genaue Beschreibung der Installation des Datenbanksystems zu sein. Dazu wird auf die DB2-Dokumentation verwiesen. Im Normalfall sollte es mit dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

Installation

1. Installation von DB2

Wir gehen davon aus, dass eine DB2-Instanz unter User `db2inst` installiert wird. Als Listen-Port wird in diesem Text von der 50000 ausgegangen. Sollte er in der Zielumgebung davon abweichen, muss dem in der Konfiguration der `DbUrl` Rechnung getragen werden.

2. Anlegen der Repository Datenbank `bicsuitedb`

Als Instanz-Owner `db2inst` wird die Repository Datenbank mit Namen `bicsuite` angelegt und alle benötigten Rechte dem Benutzer `bicsuite` zugeteilt. Der Datenbankname kann selbstverständlich beliebig gewählt werden, darf aber nur 8 Zeichen lang sein.

```
[db2inst@cheetah bin]$ db2
(c) Copyright IBM Corporation 1993,2007
Command Line Processor for DB2 Client 10.5.5
```

You can issue database manager commands and SQL statements from the command prompt. For example:

```
db2 => connect to sample
db2 => bind sample.bnd
```

For general help, type: `?`.

For command help, type: `? command`, where command can be the first few keywords of a database manager command. For example:

```
? CATALOG DATABASE for help on the CATALOG DATABASE command
? CATALOG           for help on all of the CATALOG commands.
```

To exit db2 interactive mode, type `QUIT` at the command prompt. Outside interactive mode, all commands must be prefixed with `'db2'`.

To list the current command option settings, type `LIST COMMAND OPTIONS`.

For more detailed help, refer to the Online Reference Manual.

```
db2 => create database bicsuite restrictive
DB20000I  The CREATE DATABASE command completed successfully.
db2 => connect to bicsuite
```

Database Connection Information

```
Database server      = DB2/LINUX8664 10.5.5
SQL authorization ID = DB2INST
Local database alias = BICSUITE
```

```
db2 => grant dbadm, dataaccess, accessctrl, secadm on database to
      user bicsuite
DB20000I The SQL command completed successfully.
```

3. Anlegen und Initialisierung der Datenbanktabellen

Zum Anlegen der benötigten Tabellen führen Sie folgende Befehle als Benutzer **bicsuite** durch:

```
[bicsuite@ocelot ~]$ cd $BICSUITEHOME/sql
[bicsuite@ocelot sql]$ . /home/db2inst/.ibm/db2/desktop/env
[bicsuite@ocelot sql]$ export PATH=/home/db2inst/sqllib/bin:$PATH
[bicsuite@ocelot sql]$ clpplus -nw \
      bicsuite/'<bicsuite OS password>'@localhost/bicsuite \
      @db2/install.sql > /tmp/install.log
```

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

Ändern Sie folgende Properties wie angegeben:

```
DbPasswd=<bicsuite OS password>
DbUrl=jdbc:db2://localhost:50000/bicsuite
DbUser=bicsuite
JdbcDriver=com.ibm.db2.jcc.DB2Driver
```

5. Konfigurieren Sie den BICsuite Java Class Path für DB2 JDBC

In der Konfigurationsdatei \$BICSUITECONFIG/java.conf muss nun nur noch der Pfad zum IBM DB2 JDBC-Treiber an dem CLASSPATH angehängt werden.

Etwa

```
DB2JAVA=/home/db2inst/sqllib/java
BICSUITEJDBC=$DB2JAVA/db2jcc4.jar
BICSUITEJDBC=$BICSUITEJDBC:$DB2JAVA/db2jcc.jar
BICSUITEJDBC=$BICSUITEJDBC:$DB2JAVA/db2jcc_license_cu.jar
BICSUITECLASSPATH=$BICSUITEJAR:$BICSUITEJDBC
unset DB2JAVA
```

Installation mit Oracle

Einleitung

Diese Anleitung erhebt nicht den Anspruch eine genaue Beschreibung der Installation des Datenbanksystems zu sein. Dazu wird auf die Oracle-Dokumentation verwiesen. Im Normalfall sollte es mit dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

Installation

1. Herunterladen und Installieren der aktuellen Oracle Express Edition

Merken Sie sich Ihr Oracle System Passwort

2. Anlegen des Oracle Users bicsuite

```
$ sqlplus SYSTEM/oracle_system_passwort
```

Anlegen des Benutzers bicsuite mit:

```
SQL> create user bicsuite identified by bicsuite_passwort;
```

Vergeben Sie die notwendigen Zugriffsrechte:

```
SQL> grant CONNECT, RESOURCE, CREATE VIEW, CREATE PROCEDURE  
      TO bicsuite;
```

3. Anlegen und Initialisierung der Datenbanktabellen

```
$ cd $BICSUITEHOME/sql  
$ sqlplus bicsuite/bicsuite_passwort @ora/install.sql
```

4. Konfigurieren der Datenbankverbindung in der BICsuite Server-Konfigurationsdatei

Editieren `$BICSUITECONFIG/server.conf` und ändern Sie folgende Properties:

```
DbPasswd=bicsuite_passwort  
DbUrl=jdbc:oracle:thin:@hostname:1521:XE  
DbUser=bicsuite  
JdbcDriver=oracle.jdbc.OracleDriver
```

Weiterhin muss das Property `Hostname` auf den Hostnamen oder die IP-Adresse des Servers gesetzt werden.

5. Konfigurieren Sie den BICsuite Java Class Path für Oracle-JDBC

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

Etwa

```
BICSUITECLASSPATH=$BICSUITEJAR:$ORACLE_HOME/jdbc/lib/ojdbc14.jar
```

Konfiguration von TLS/SSL Verbindungen

Einleitung

Das Aufsetzen einer verschlüsselten Kommunikation innerhalb des BICsuite Systems ist relativ einfach. Allerdings wird der Aufwand, je nach Anforderungen, variieren.

Es gibt folgende Möglichkeiten:

1. BICsuite ohne SSL/TLS Kommunikation
2. BICsuite mit und ohne SSL/TLS Kommunikation
3. BICsuite mit ausschließlich SSL/TLS Kommunikation

Wenn mittels TLS/SSL kommuniziert wird, gibt es wiederum zwei Möglichkeiten:

1. Nur serverseitige Authentifizierung: Das bedeutet, dass Clients überprüfen können, ob der Server mit dem sie kommunizieren, auch tatsächlich vertrauenswürdig ist. Diese Einstellung erfordert nur ein Keypair für den Server. Jegliche Kommunikation erfolgt verschlüsselt.
2. Server- und clientseitige Authentifizierung Bei dieser Konfiguration wird von beiden Seiten geprüft, ob die Identität des Kommunikationspartners bekannt ist. Diese Einstellung ist sehr sicher, aber aufwendig da für jeden Client und natürlich für den Server ein Keypair erzeugt werden muss. Selbstverständlich erfolgt die Kommunikation verschlüsselt.

Konfiguration von TLS/SSL

Das Aufsetzen der SSL/TLS Kommunikation erfordert im Wesentlichen einige wenige Aktionen.

1. Generieren des Keypairs für den Server

Zum Generieren der Keypairs wird das Utility `keytool`, was Bestandteil von Java (SE) ist, genutzt. Nähere Information zu diesem Utility gibt es z.B. unter

<http://download.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html>

Folgendes Vorgehen sollte funktionieren (für "nicht"-Spezialisten):

- a) Legen Sie ein Verzeichnis an, z.B.

```
$ mkdir $BICSUITECONFIG/certs
```

- b) Das Verzeichnis wird nachher den private Key des Servers enthalten und sollte somit entsprechend geschützt werden, z.B.

```
$ chmod 700 $BICSUITECONFIG/certs
```

- c) Generieren Sie ein Keypair mit Hilfe von `keytool`, z.B.

```
$ keytool -genkeypair -alias bicsuite -keypass secret \
> -dname "cn=servername, ou=BICsuite, o=independIT, c=DE" \
> -keystore $BICSUITECONFIG/certs/svrkeystore \
> -storepass secret -validity 365
```

WICHTIG: Die beiden Passwörter (`keypass` und `storepass`) *müssen* gleich sein!

- d) Der public Key vom Server muss später noch den Clients mitgeteilt werden. Dazu muss dieser zuerst aus dem Keyfile extrahiert werden.

```
$ keytool -export -alias bicsuite -file svrkey \  
> -keystore $BICSUITECONFIG/certs/svrkeystore
```

Es wird dazu das storepass (secret) benötigt.

2. Anpassen der Serverkonfiguration

Die Serverkonfiguration muss nun angepasst werden um dem Server die Information über seine Keys zukommen zu lassen. Weiterhin muss dabei entschieden werden, ob nur eine serverseitige Authentifizierung oder eine beidseitige Authentifizierung stattfinden soll. Auch muss eine Entscheidung bezüglich der unverschlüsselten Kommunikation getroffen werden.

```
#  
# SSLPort: der Port fuer die verschluesselte Kommunikation  
# (inaktiv = 0)  
#  
SSLPort=2507  
#  
# KeyStore definiert welche Datei mit Keys genutzt werden soll  
# $BICSUITEHOME/etc/certs/svrkeystore  
#  
KeyStore=/home/bicsuite/etc/certs/svrkeystore  
#  
# Das KeyStorePassword wird benoetigt um den Keystore auszulesen  
#  
KeyStorePassword=secret  
#  
# TrustStore definiert welche Datei die (public) Keys der  
# gueltigen Kommunikationspartner enthaelt  
#  
TrustStore=/home/bicsuite/etc/certs/svrkeystore  
#  
# Das TrustStorePassword wird benoetigt um den Truststore  
# auszulesen  
#  
TrustStorePassword=secret  
#  
# ClientAuthentication besagt ob Clients sich ebenfalls  
# autorisieren muessen (true), oder nicht (false)  
#  
ClientAuthentication=true  
#  
# Port definiert den Port fuer die unverschlüsselte  
# Kommunikation (inaktiv = 0)  
# Wenn alle Ports als inaktiv konfiguriert sind,  
# wird Port 2506 fuer unverschlüsselte Kommunikation geoeffnet  
#  
Port=2506  
#  
# ServicePort definiert den Port fuer System Zugang in Notfaellen
```

```
# (inaktiv = 0)
#
ServicePort=2505
```

Wenn Client Authentication=true ist, werden vom Server sowohl der Keystore als auch der Truststore benötigt. Wenn eine Client-Authentifizierung nicht erforderlich ist, wird nur der Keystore benötigt.

3. Client Konfiguration

Wenn keine Client-Authentifizierung erforderlich ist, benötigen Clients nur den Zugriff auf ein Truststore um die Identität des Servers überprüfen zu können. Muss auch die Identität der Clients überprüft werden, muss für jeden Client auch ein Keystore erzeugt werden. Dies erfolgt analog zu der Erzeugung des Keystores für den Server.

4. Anpassen von .sdmshrc

Wenn sdmsh bzw. die Standard Utilities über TLS/SSL mit dem Server kommunizieren soll ist die Benutzung eines "ini"-Files zwingend erforderlich. Es gibt drei Möglichkeiten:

- a) \$BICSUITECONFIG/sdmshrc
- b) \$HOME/.sdmshrc
- c) Eine beim Aufruf des Utilities spezifizierte Datei

Entscheidend dabei ist die Angabe der für den Aufbau der sicheren Verbindung benötigten Information. Etwa

```
User=donald
Password=duck
Host=localhost
Port=2507
SSL=true
KeyStore=/home/bicsuite/etc/certs/clntkeystore
TrustStore=/home/bicsuite/etc/certs/clnttruststore
KeyStorePassword=secret
TrustStorePassword=secret
Timeout=0
```

was eine symmetrische Authentifizierung erlaubt. Ist nur eine serverseitige Authentifizierung erforderlich, können die Keystore betreffenden Zeilen entfernt werden.

5. Jobserver Konfiguration

Die Jobserver haben vier neue Konfigurationsparameter entsprechend den Serverparametern bekommen:

```
KEYSTORE
TRUSTSTORE
```

```
KEYSTOREPASSWORD
TRUSTSTOREPASSWORD
```

Dazu kommt noch ein Parameter der angibt, ob eine verschlüsselte Kommunikation erwünscht ist:

```
USE_SSL
```

Wenn Client Authentication erwünscht ist, muss für jeden Jobserver ein Schlüsselpaar angefertigt werden. Dies geht genauso wie für den BICsuite Server beschrieben wurde.

Natürlich muss darauf geachtet werden, dass insbesondere der Konfigurationsparameter `RepoPort` richtig gesetzt ist.

6. Getting it all together

Last but not least müssen nun die public Keys zwischen den einzelnen Kommunikationspartnern ausgetauscht werden. Auch dies erfolgt mit dem Utility `keytool`. Um etwa den public Key des Servers als vertrauenswürdig zu definieren, wird er in den Truststore des Clients aufgenommen. z.B.:

```
$ keytool -import -keystore clntkeystore -alias bicsuite -file svrkey
Enter keystore password:
Owner: CN=servername, OU=BICsuite, O=independIT, C=DE
Issuer: CN=servername, OU=BICsuite, O=independIT, C=DE
Serial number: 4dc28814
Valid from: Thu May 05 13:20:52 2011 until: Fri May 04 13:20:52 2012
Certificate fingerprints:
    MD5: D3:83:F2:2B:93:2B:65:7A:41:3E:CE:2E:C8:EC:40:62
    SHA1: AF:B1:18:95:B2:2A:BB:1D:08:BD:A6:87:68:64:6B:FC:0D:A8:30:DA
Signature algorithm name: SHA1withDSA
Version: 3
Trust this certificate? [no]: yes
Certificate was added to keystore
```

Dazu wird natürlich wiederum ein Storepass benötigt.

Wird auch die Client-Authentifizierung verlangt, müssen auch die Client Certificates in den Truststore des Servers eingetragen werden.

Installation des Zope Servers

Einleitung

Um die BICsuite!Web User Interface-Oberfläche nutzen zu können, muss ein Zope Application Server aufgesetzt werden.

Installation (Zope2)

1. Installation virtualenv

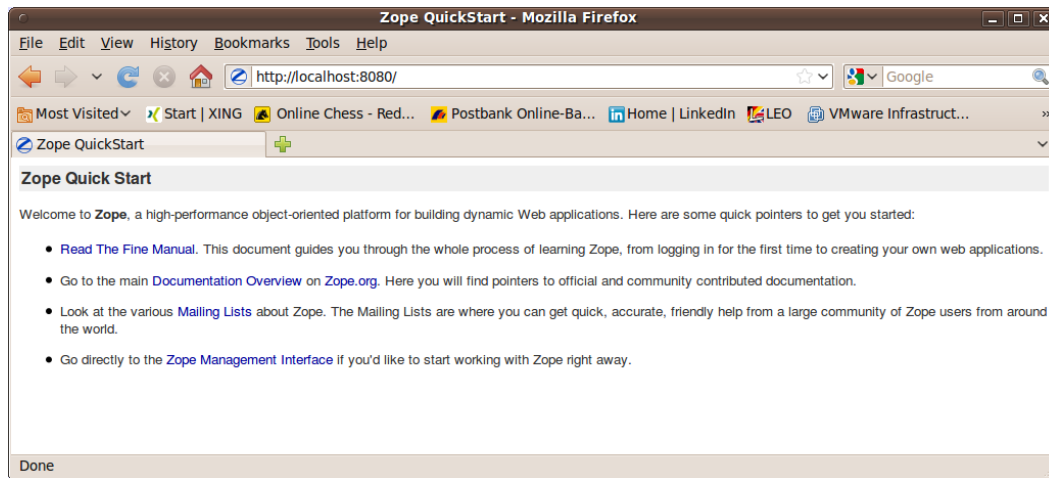


Abbildung 2.1: Zope Quick Start Seite

```
$ easy_install virtualenv
```

2. Erzeugen der virtuellen Python-Umgebung für die Zope-Installation

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

3. Installieren der Zope2 Software

Installieren Sie die neueste Release von Zope2. Zum Zeitpunkt der Erstellung dieses Dokumentes war die Release 2.13.29 die Aktuellste.

```
$ cd $HOME/software/Zope
$ bin/pip install -r \
https://raw.githubusercontent.com/zopefoundation/Zope/2.13.29/requirements.txt
```

Kann bei der Installation nicht auf das Internet zugegriffen werden, kann Zope auch offline installiert werden. Dazu ist wie folgt vorzugehen:

a) Download python packages

Auf einem möglichst identischen System mit Internetzugang führen Sie folgende Kommandos aus:

```
$ wget \
https://raw.githubusercontent.com/zopefoundation/Zope/2.13.29/requirements.txt
$ pip download -r requirements.txt -d packages
```

b) Dateien auf das Zielsystem übertragen

Die Datei requirements.txt und das Verzeichnis packages müssen nun auf das Zielsystem ohne Internetzugang übertragen werden. Legen Sie die Dateien in \$HOME/software ab.

c) Installation auf dem Zielsystem

Folgendes Kommando installiert Zope aus den heruntergeladenen Dateien:

```
$ cd $HOME/software
$ Zope/bin/pip install --no-index --use-wheel \
--find-links=./packages -r requirements.txt
```

4. Erzeugen einer Zope-Instanz für BICsuite!Web

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

Das Passwort kann beliebig gewählt werden und wird später wieder benötigt. Der Benutzer muss aber `sdmsadm` heißen.

Zum Testen wird der Zope Server kurz gestartet

```
$ $HOME/bicsuiteweb/bin/zopectl start
```

Im Internet Browser sollte die URL

`http://localhost:8080`

nun die Zope Quick Start Seite wie im Bild [2.1](#) anzeigen.

Die Zope-Instanz wird nun wieder angehalten.

```
$ $HOME/bicsuiteweb/bin/zopectl stop
```

5. Installieren der BICsuite!Web-Komponenten

Um die BICsuite!Web-Komponenten zu installieren, muss die Zope-Installation um einige Module erweitert werden:

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

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

```
$ $HOME/bicsuiteweb/bin/zopectl start
```

Die Zope Management Oberfläche wird nun unter der Adresse

`http://localhost:8080/manage`

mit Hilfe eines Browsers geöffnet (siehe Bild [2.2](#)). Dazu wird der Benutzer `sdmsadm` mit dem von Ihnen vergebenen Passwort benutzt.

Es wird jetzt die Frontend Software in Zope geladen (Import Button, siehe Bild [2.3](#))

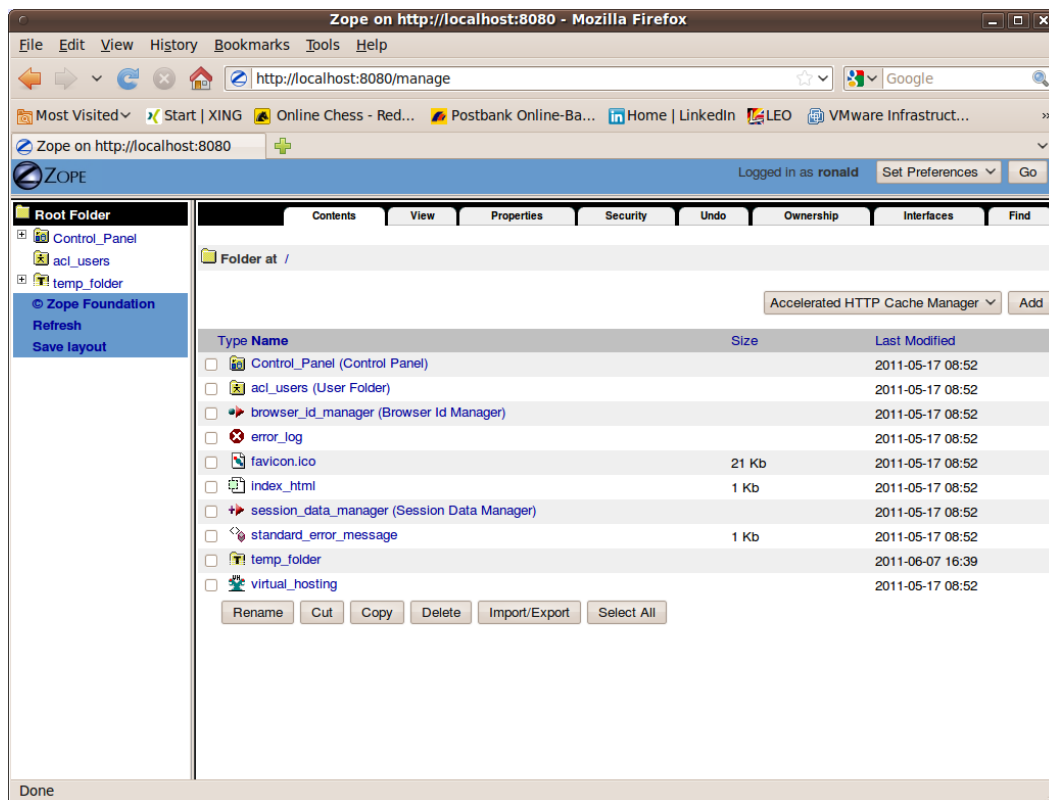


Abbildung 2.2: Zope Management Oberfläche

- a) im Folder / SDMS.zexp importieren
- b) im Folder /SDMS/Install die Folder User und Custom anwählen und mit Copy kopieren
- c) im Folder / mit Paste die Folder User und Custom erzeugen

Wenn nun alles fehlerfrei durchgeführt werden konnte, sieht die Oberfläche wie auf dem Bild [2.4](#) aus.

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 BICsuiteServer, der von dieser BIC-suite!Web Installation angesprochen werden soll, einen Eintrag der Form

```
# Servername unter dem der Server in der BICsuite!Web Oberflaeche
# sichtbar ist
```

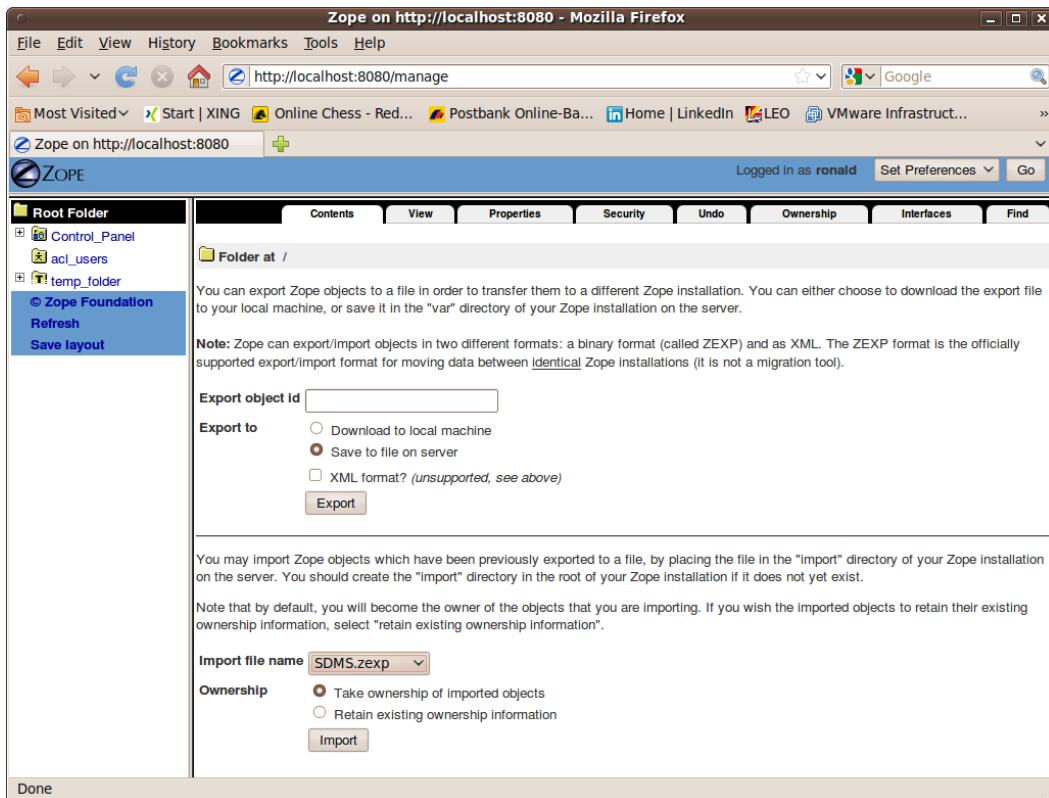


Abbildung 2.3: Zope Import Dialog

```
'servername' : {
    # IP Adresse oder Hostname auf dem der BICsuite Server laeuft
    'HOST'      : 'hostname',

    # Port unter dem der BICsuite Server angesprochen wird
    'PORT'      : '2506',

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

    # optionales Property, ob Zope Serververbindungen cachen soll
    'CACHE'     : 'Y'

    # optionales Property, wie lange gecachete BICsuite!web
    # Serververbindungen gueltig sein sollen
    # default ist 60 sekunden, nur von Bedeutung falls 'CACHE' : 'Y'
    'TIMEOUT'   : '60'
}
```

enthalten muss. Fürs Bootstrapping muss ein Eintrag mit Namen DEFAULT

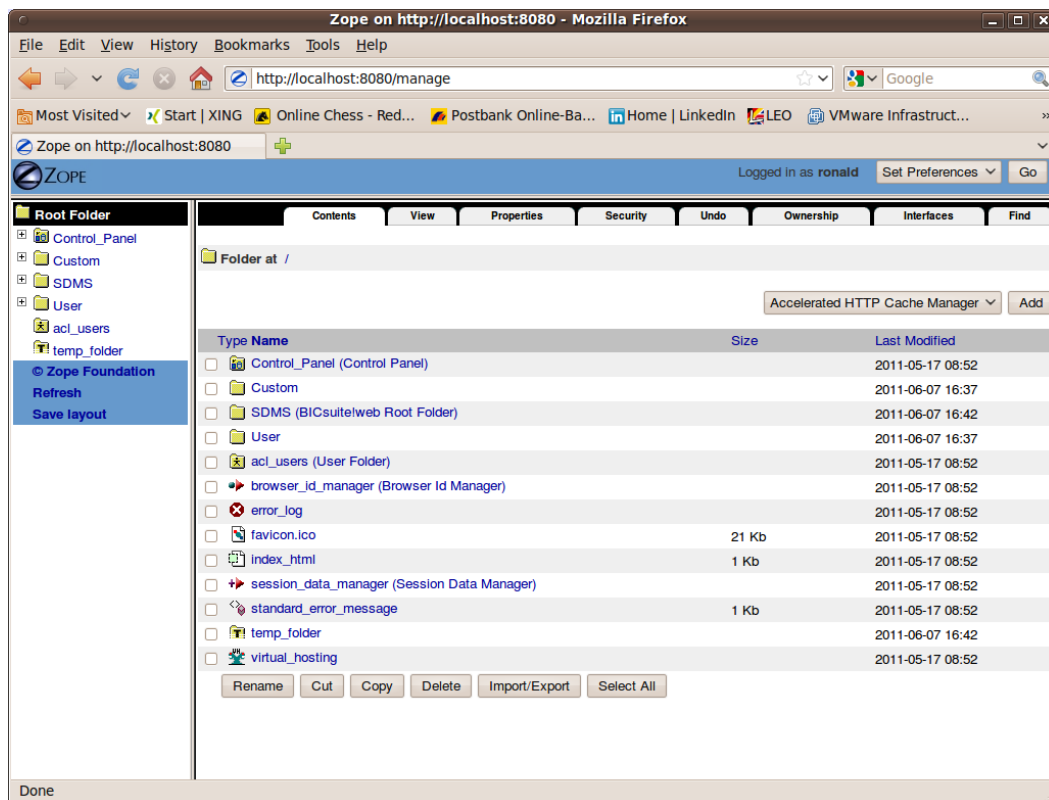


Abbildung 2.4: Zope Resultat Ansicht

vorhanden sein. Dieser Eintrag kann nach dem Einrichten der Benutzer (die dann diese Connection natürlich nicht benutzen sollten) entfernt werden.

Soll ein Server über eine sichere SSL-Verbindung angesprochen werden, dann müssen folgende weitere Eigenschaften definiert werden:

```
# Verbindung wird ueber Secure Socket Layer aufgebaut
'SSL' : 'true',

# falls angegeben, wird die Identitaet des BICSuite Servers
# ueberprueft. Die angegebene Datei muss das Server Certificate
# des BICSuite Servers enthalten
'TRUSTSTORE' : 'truststore.pem',

# falls der BICSuite Server eine Client Authentication fordert,
# muss dieses Property definiert sein und die angegebene Datei
# muss das Certificate und den Private Key des Clients enthalten.
# Das Certificate muss dem Server in seinem Truststore bekannt sein.
'KEYSTORE' : 'keystore.pem'
```

Anmerkung:

Bei Verwendung von SSL wird aus Performancegründen die Verwendung von cached Serververbindungen empfohlen, da der Aufbau einer gesicherten Verbindung eine rechenintensive Operation ist.

7. Die BICsuite!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 dazugehörige Dokumentation verwiesen.

HTTPS mit Hilfe eines vorgeschaltetem Apache Servers

Die einfachste Möglichkeit das HTTPS Protokoll anstelle des HTTP Protokolls zu nutzen ist das Vorschalten eines Apache Servers. Alternativ dazu kann auch der Zope Server dazu gebracht werden mittels HTTPS zu kommunizieren, aber die Konfiguration ist etwas komplizierter und wird im nachfolgenden Abschnitt behandelt.

In diesem Abschnitt wird nicht tief auf die Konfigurationsmöglichkeiten von Apache eingegangen. Es wird vielmehr beschrieben, wie das gesetzte Ziel, die Kommunikation über HTTPS, auf einfache Weise erreicht werden kann.

Als erste muss natürlich das Apache System sowie die `mod_ssl` Erweiterung installiert werden. Das genaue Vorgehen ist dabei Betriebssystems- sowie Distributionsabhängig. In einer RHEL oder CentOS Umgebung ist dazu lediglich ein

```
# yum install httpd mod_ssl
```

beziehungsweise in einer Ubuntu Umgebung ein

```
# apt install apache2  
# a2enmod ssl
```

notwendig.

Jetzt muss die ssl Engine von Apache konfiguriert werden. Häufig erledigen die Paketmanager schon einiges an Arbeit und erzeugen ein self-signed Certificate. Dieses kann übernommen, oder mit einem offiziellen Zertifikat ersetzt werden. Weiterhin muss in der ssl Konfiguration für den entsprechenden Virtual Host eine Proxy-Regel aufgenommen werden. Diese sieht in Prinzip folgendermaßen aus:

```
#  
# Redirect to Zope  
#  
<IfModule mod_proxy.c>
```

```

ProxyRequests Off
ProxyPreserveHost On
ProxyPass / http://127.0.0.1:8080/VirtualHostBase/\
          https/myvirtualhost:443/VirtualHostRoot/
ProxyPassReverse / http://127.0.0.1:8080/VirtualHostBase/\
                  http/https/myvirtualhost:443/VirtualHostRoot/
</IfModule>

<IfModule mod_headers.c>
    RequestHeader set X-Forwarded-Proto "https"
</IfModule>

```

Bitte beachten: Aus Gründe der Darstellung wurde die Zeile umgebrochen. Dies wird hier, wie so üblich, mit einem Backslash gekennzeichnet. In der tatsächliche Konfiguration sollten die beide Zeilen wieder, ohne Backslash und Whitespace, aneinandergehängt werden.

Da der Zope Server auf denselben Host läuft als der vorgeschaltete Apache Server, wird die zu bearbeitenden Anfrage an die 127.0.0.1:8080 weitergeleitet. Als der eigene Hostname wurde myvirtualhost eingesetzt.

Im Header der Anfrage wird das Kommunikationsprotokoll auf https gesetzt. Jetzt ist es nur noch wichtig jede Versuch mittles HTTP auf den Server zuzugreifen nach HTTPS umzulenken. Dies passiert in der globalen Konfiguration. Es wird in der Konfiguration des entsprechenden Virtual Hosts

```
Redirect permanent / https://myvirtualhost/
```

aufgenommen.

Installation der Zope HTTPS Erweiterungen

Die Installation von der Zope HTTPS Erweiterung erfordert einige Arbeit. Es ist nicht besonders schwierig. Es ist aber wichtig die Anweisungen genau zu befolgen und idealerweise auch zu verstehen.

1. Abholen des M2Crypto Moduls

Das M2Crypto Modul findet man derzeit unter

<http://pypi.python.org/packages/source/M/M2Crypto/M2Crypto-0.21.1.tar.gz>

Packen Sie es unter \$HOME aus.

2. Installieren von M2Crypto in Ihr Virtual Environment

```
$ cd $HOME/M2Crypto-0.21.1
$ $HOME/software/Zope/bin/python setup.py install
```

Zum Testen, ob die Installation erfolgreich war, kann schlichweg versucht werden das Modul in Python zu laden:

```
$ $HOME/software/Zope/bin/python
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
[GCC 4.5.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import M2Crypto
>>>
```

Bei einer Fehlermeldung ist evtl. ein `ldconfig` oder das Schließen und Neuöffnen des Terminals notwendig.

3. Zope Installation patchen

Der nächste Schritt sorgt dafür, dass Zope, je nach Konfiguration, auch das nun installierte M2Crypto Modul benutzen wird.

```
$ $BICSUITEHOME/zope/https/patch.sh
```

4. Generieren der von Zope HTTPS benötigten Dateien

Die von Zope benötigten Dateien können, müssen aber nicht, ins "etc"-Verzeichnis der Zope-Instanz abgelegt werden. Im Folgenden wird davon ausgegangen, dass dies auch passiert. Eine Modifizierung des Vorgehens setzt voraus, dass ein solides Verständnis von dem Funktionieren von TLS/SSL sowie HTTPS vorhanden ist.

Zuerst wird zum "etc"-Verzeichnis gewechselt:

```
$ cd $HOME/bicsuiteweb/etc
```

Dann wird ein SSL Certificate Authority erstellt:

```
$ openssl req -new -x509 -newkey rsa:2048 -keyout cakey.pem \
> -out cacert.pem -days 3650
```

Als Beispiel können dabei folgende Eingaben gemacht werden:

```
Enter PEM pass phrase: super_geheim
Verifying - Enter PEM pass phrase: super_geheim
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Bayern
Locality Name (eg, city) []:Schrobenhausen
Organization Name (eg, company) [Internet Widgits Pty]:independIT
Organizational Unit Name (eg, section) []:Development
Common Name (eg, YOUR name) []:Dieter Stubler
Email Address []:dieter.stubler@independit.de
```

Nun kann ein SSL Server Certificate erstellt werden

- Erzeugen des Keys für Server Certificate:

```
$ openssl genrsa -out serverkey.pem -aes128 2048 -days 3650
```

```
Enter pass phrase for serverkey.pem: dummy
```

```
Verifying - Enter pass phrase for serverkey.pem: dummy
```

- Entfernen des Passworts aus serverkey.pem

```
$ openssl rsa -in serverkey.pem -out serverkey.pem
```

```
Enter pass phrase for serverkey.pem: dummy
```

- **Certificate Signing Request erzeugen**

```
$ openssl req -new -key serverkey.pem -out req.pem -nodes
```

Als Beispiel können dabei folgende Eingaben gemacht werden:

```
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Bayern
Locality Name (eg, city) []:Schrobenhausen
Organization Name (eg, company) [Internet Widgits Pty]:independIT
Organizational Unit Name (eg, section) []:Development
Common Name (eg, YOUR name) []:my_hostname
Email Address []:dieter.stubler@independit.de
A challenge password []:
An optional company name []:
```

Achtung !!!

my_hostname muss durch den Namen ersetzt werden, unter dem der Browser den HTTPS Host anspricht !!!

- **Certificate signieren**

- a) **Sichern openssl.cnf (benötigt meistens root-Rechte)**

```
# cp /etc/ssl/openssl.cnf /etc/ssl/openssl.cnf_save
```

- b) **Editieren openssl.cnf**

```
# sudo vi /etc/ssl/openssl.cnf
```

Dabei werden folgende Einträge geändert:

```
dir                = .
private_key        = $dir/cakey.pem
RANDFILE           = $dir/.rand
default_days       = 3650
new_certs_dir      = $dir
```

- c) **Prepare files**

```
$ touch index.txt
$ echo 01 > serial
```

- d) **Signieren**

```
$ openssl ca -in req.pem -notext -out servercert.pem
```

Beispiel:

```
Enter pass phrase for ./cakey.pem: super_geheim
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
```

- e) **Wiederherstellen der alten openssl.cnf**

```
# mv /etc/ssl/openssl.cnf_save /etc/ssl/openssl.cnf
```

- SSL Server Certificate schreiben

```
$ cat servercert.pem serverkey.pem > server_cert.pem
```

- Erzeugen des SSL Entropy Pools

```
$ dd if=/dev/random of=ssl_entropy_pool.dat bs=1024 count=1
```

- Erzeugen des SSL DH Init:

```
$ openssl dhparam -out ssl_dh_init.pem 1024
```

5. Zope HTTPS konfigurieren

Falls HTTPS auf Port 8085 hören soll und

```
$HOME == /home/bicsuite
```

ist, soll in die Datei `$HOME/bicsuiteweb/etc/zope.conf` folgendes nach dem Endtag `</http-server>` eingefügt werden:

```
<https-server>
# valid keys are "address", "force-connection-close",
# required keys are
#     "x509_remote_users",
#     "ssl_certificate_authority",
#     "ssl_server_certificate"
address 8085
x509_remote_users off
ssl_certificate_authority /home/bicsuite/bicsuiteweb/etc/cacert.pem
ssl_server_certificate /home/bicsuite/bicsuiteweb/etc/server_cert.pem
ssl_dh_init /home/bicsuite/bicsuiteweb/etc/ssl_dh_init.pem
ssl_entropy_pool /home/bicsuite/bicsuiteweb/etc/ssl_entropy_pool.dat
# force-connection-close off
</https-server>
```

Soll nur über HTTPS zugegriffen werden können, so ist der `<http-server>` Bereich bis `</http-server>` auszukommentieren.

6. Zope neu starten

```
$ $HOME/bicsuiteweb/bin/zopectl stop
```

Wir benutzen `runzope`, damit wir Fehler gleich sehen.

```
$ $HOME/bicsuiteweb/bin/runzope
```

Für Problemanalyse ist auch `$HOME/bicsuiteweb/log/event.log` hilfreich.

7. Nachdem es funktioniert hat, `runzope` abbrechen (Ctrl-C) und Zope neu starten.

```
$ $HOME/bicsuiteweb/bin/zopectl start
```

Aufruf von BICsuite!Web:

Im Browser

`http://hostname:8080/SDMS`

oder

`https://hostname:8085/SDMS`

Beim ersten Aufruf muss das Server Certificate dem Browser als vertrauenswürdig bestätigt werden. Bei Firefox reicht dies aus. Unter Internet Explorer muss das

`/home/bicsuite/bicsuiteweb/etc/cacert.pem`

unter Internetoptionen → Inhalte → Herausgeber → Vertrauenswürdige Stammzertifizierungsstellen importiert werden, damit ohne andauernde Warnungen gearbeitet werden kann.

Installation (Zope4+)

1. Voraussetzungen

Um Zope4 bzw. Zope5 zu installieren müssen folgende Pakete installiert werden:

a) python3

```
$ sudo yum install python3
```

b) python3 development headers

```
$ sudo yum install python3-devel
```

2. Erzeugen der virtuellen Python-Umgebung für die Zope-Installation

```
$ export INSTALLDIR=$HOME/software
$ export ZOPE5ENV=Zope5
$ export ZOPE5DIR=$INSTALLDIR/$ZOPE5ENV
$ mkdir -p $INSTALLDIR
$ cd $INSTALLDIR
$ python3 -m venv $ZOPE5ENV
```

3. Installieren der Zope5 Software

Installieren Sie die neueste Release von Zope5. Zum Zeitpunkt der Erstellung dieser Dokumentation war Zope 5.1.2 die aktuellen Version.

```
$ cd $ZOPE5DIR
$ bin/pip install -U pip wheel
$ bin/pip install Zope[wsgi]==5.0 \
-c https://zopefoundation.github.io/Zope/releases/5.1.2/constraints.txt
$ bin/pip install Products.ExternalMethod
$ bin/pip install Products.Sessions
```

```
$ bin/pip install Products.SiteErrorLog
$ bin/pip install Products.PythonScripts
```

Kann bei der Installation nicht auf das Internet zugegriffen werden, kann Zope auch offline installiert werden. Dazu ist wie folgt vorzugehen:

a) Download python packages

Auf einem möglichst identischen System mit Internetzugang führen Sie folgende Kommandos aus:

```
$ wget \
https://zopefoundation.github.io/Zope/releases/5.1.2/constraints.txt
$ pip download -r requirements.txt -d packages
```

b) Dateien auf das Zielsystem übertragen

Die Datei requirements.txt und das Verzeichnis packages müssen nun auf das Zielsystem ohne Internetzugang übertragen werden. Legen Sie die Dateien in \$HOME/software ab.

c) Installation auf dem Zielsystem

Folgendes Kommando installiert Zope aus den heruntergeladenen Dateien:

```
$ cd $HOME/software
$ Zope/bin/pip install --no-index --use-wheel \
--find-links=./packages -r requirements.txt
```

4. Erzeugen einer Zope-Instanz für BICsuite!Web

```
$ cd $HOME
$ export ZOPE5INSTANCE=$HOME/Zope5Instance
$ $ZOPE5DIR/bin/mkwsgiinstance -d $ZOPE5INSTANCE \
-u sdmsadm:sdmsadm_passwort
```

Das Passwort kann beliebig gewählt werden und wird später wieder benötigt. Der Benutzer muss aber sdmsadm heißen.

Test:

Starten Sie die Zope5 Instanz mit folgendem Befehl:

```
$ $ZOPE5DIR/bin/runwsgi -v $ZOPE5INSTANCE/etc/zope.ini
```

Im Browser sollte die URL <http://localhost:8080> nun eine 'Zope Auto-generated default page' zeigen.

Zope5 kann nun mit Strg-C oder durch Schließen der Windows-Eingabeaufforderung wieder beendet werden.

5. Installieren der BICsuite!Web-Komponenten

Um die BICsuite!Web-Komponenten zu installieren, muss die Zope-Installation um einige Module erweitert werden:

```

$ cd $ZOE5INSTANCE
$ mkdir Extensions
$ cd Extensions
$ ln -s $BICSUITEHOME/zope4/Extensions/*.py .
$ cd ..
$ ln -s $BICSUITEHOME/zope4/BICsuiteSubmitMemory \
$ZOE5DIR/lib64/python3*/site-packages/Products
$ ln -s $BICSUITEHOME/zope4/StringFixer \
$ZOE5DIR/lib64/python3*/site-packages/Products
$ mkdir import
$ cd import
$ ln -s $BICSUITEHOME/zope4/import/SDMS.zexp .

```

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

```
$ $ZOE5DIR/bin/runwsgi -v $ZOE5INSTANCE/etc/zope.ini
```

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 benutzt.

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

- a) im Folder / `SDMS.zexp` importieren
- b) im Folder `/SDMS/Install` die Folder `User` und `Custom` anwählen und mit `Copy` kopieren
- c) im Folder / mit `Paste` die Folder `User` und `Custom` erzeugen

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 `BICsuiteServer`, der von dieser `BICsuite!Web` Installation angesprochen werden soll, einen Eintrag der Form

```

# Servername unter dem der Server in der BICsuite!Web Oberflaeche
# sichtbar ist
'servername' : {
    # IP Adresse oder Hostname auf dem der BICsuite Server laeuft
    'HOST'      : 'hostname',

    # Port unter dem der BICsuite Server angesprochen wird
    'PORT'      : '2506',

    # BASIC, PROFESSIONAL, ENTERPRISE

```



```

    'VERSION' : 'BASIC',

    # optionales Property, ob Zope Serververbindungen cachen soll
    'CACHE'   : 'Y'

    # optionales Property, wie lange gecachte BICsuite!web
    # Serververbindungen gueltig sein sollen
    # default ist 60 sekunden, nur von Bedeutung falls 'CACHE' : 'Y'
    'TIMEOUT' : '60'
}

```

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

Soll ein Server über eine sichere SSL-Verbindung angesprochen werden, dann müssen folgende weitere Eigenschaften definiert werden:

```

# Verbindung wird ueber Secure Socket Layer aufgebaut
'SSL'           : 'true',

# falls angegeben, wird die Identitaet des BICsuite Servers
# ueberprueft. Die angegebene Datei muss das Server Certificate
# des BICsuite Servers enthalten
'TRUSTSTORE'    : 'truststore.pem',

# falls der BICsuite Server eine Client Authentication fordert,
# muss dieses Property definiert sein und die angegebene Datei
# muss das Certificate und den Private Key des Clients enthalten.
# Das Certificate muss dem Server in seinem Truststore bekannt sein.
'KEYSTORE'      : 'keystore.pem'

```

Anmerkung:

Bei Verwendung von SSL wird aus Performancegründen die Verwendung von cached Serververbindungen empfohlen, da der Aufbau einer gesicherten Verbindung eine rechenintensive Operation ist.

7. Die BICsuite!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 dazugehörige Dokumentation verwiesen.

Migration einer bestehenden BICsuite!Web Zope2 Datenbank nach Zope4+

Um die Datenbank einer bestehenden BICsuite!Web Zope2 Installation in eine neue BICsuite!Web Zope4+ Installation zu übernehmen, sind folgende Schritte durchzuführen:

1. Führen Sie alle Schritte der vorangegangenen Anleitung zur "Installation (Zope4+)" durch.
2. Stoppen Sie die installierte BICsuite!web Zope4+ Instanz falls diese aktiv ist
3. Installieren zodbupdate

Für die Migration der bestehenden BICsuite!web Zope2 Datenbank nach Zope4+ muss das tool zodbupdate installiert werden.

```
$ cd $ZOPE5DIR  
$ bin/pip install zodbupdate
```

4. Stoppen Sie Ihre BICsuite!web Zope2 Instanz falls diese aktiv ist
5. Übernehmen Sie das Data.fs aus Ihrer BICsuite!web Zope2 Instanz in Ihre Zope4+ Instanz

Setzen dafür die Umgebungsvariablen ZOPE5DIR, ZOPE2INSTANCE und ZOPE5INSTANCE auf die für Sie gültigen Verzeichnisse.

```
$ rm $ZOPE5INSTANCE/var/Data.fs*  
$ cp $ZOPE2INSTANCE/var/Data.fs $ZOPE5INSTANCE/var  
$ $ZOPE5DIR/bin/zodbupdate -v -f var/Data.fs --convert-py3 --encoding utf8
```

6. Starten Sie Ihre BICsuite!web Zope4+ Instanz
7. Ersetzen des SDMS Folders
 - a) Öffnen Sie die Management Oberfläche Ihrer Zope4+ Instanz
 - b) Lösche Sie den temp_folder aus dem Root (/) Folder
 - c) Navigieren Sie zu dem Folder, welcher den Folder SDMS enthält
 - d) Löschen Sie den Folder SDMS
 - e) Importieren Sie das Zope4+ SDMS.zexp

HTTPS mit Hilfe eines vorgeschaltetem Apache Servers

Die einfachste Möglichkeit das HTTPS Protokoll anstelle des HTTP Protokolls zu nutzen ist das Vorschalten eines Apache Servers. Alternativ dazu kann auch der Zope Server dazu gebracht werden mittels HTTPS zu kommunizieren, aber die Konfiguration ist etwas komplizierter und wird im nachfolgenden Abschnitt behandelt.

In diesem Abschnitt wird nicht tief auf die Konfigurationsmöglichkeiten von Apache eingegangen. Es wird vielmehr beschrieben, wie das gesetzte Ziel, die Kommunikation über HTTPS, auf einfache Weise erreicht werden kann.

Als erste muss natürlich das Apache System sowie die `mod_ssl` Erweiterung installiert werden. Das genaue Vorgehen ist dabei Betriebssystems- sowie Distributionsabhängig. In einer RHEL oder CentOS Umgebung ist dazu lediglich ein

```
# yum install httpd mod_ssl
```

beziehungsweise in einer Ubuntu Umgebung ein

```
# apt install apache2
# a2enmod ssl
```

notwendig.

Jetzt muss die ssl Engine von Apache konfiguriert werden. Häufig erledigen die Paketmanager schon einiges an Arbeit und erzeugen ein self-signed Certificate. Dieses kann übernommen, oder mit einem offiziellen Zertifikat ersetzt werden. Weiterhin muss in der ssl Konfiguration für den entsprechenden Virtual Host eine Proxy-Regel aufgenommen werden. Diese sieht in Prinzip folgendermaßen aus:

```
#
# Redirect to Zope
#
<IfModule mod_proxy.c>
    ProxyRequests Off
    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:8080/VirtualHostBase/\
        https/myvirtualhost:443/VirtualHostRoot/
    ProxyPassReverse / http://127.0.0.1:8080/VirtualHostBase/\
        http/https/myvirtualhost:443/VirtualHostRoot/
</IfModule>

<IfModule mod_headers.c>
    RequestHeader set X-Forwarded-Proto "https"
</IfModule>
```

Bitte beachten: Aus Gründe der Darstellung wurde die Zeile umgebrochen. Dies wird hier, wie so üblich, mit einem Backslash gekennzeichnet. In der tatsächliche Konfiguration sollten die beide Zeilen wieder, ohne Backslash und Whitespace, aneinandergehängt werden.

Da der Zope Server auf denselben Host läuft als der vorgeschaltete Apache Server, wird die zu bearbeitenden Anfrage an die `127.0.0.1:8080` weitergeleitet. Als der eigene Hostname wurde `myvirtualhost` eingesetzt.

Im Header der Anfrage wird das Kommunikationsprotokoll auf `https` gesetzt.

Jetzt ist es nur noch wichtig jede Versuch mittles HTTP auf den Server zuzugreifen nach HTTPS umzulenken. Dies passiert in der globalen Konfiguration. Es wird in der Konfiguration des entsprechenden Virtual Hosts

Redirect permanent / <https://myvirtualhost/>

aufgenommen.

SSO für BICsuite mit Zope

Einleitung

In kleinen Umgebungen ist die in BICsuite integrierte Benutzerverwaltung sinnvoll und erlaubt auf einfacher Weise eine Trennung von Systemen. Dies ändert sich jedoch drastisch, wenn die Umgebungen größer werden. Um die Vielzahl der unterschiedlichen Systeme, deren Benutzer und einhergehenden Rechten im Überblick behalten zu können, ist eine zentrale Verwaltung unabdingbar.

In vielen Fällen wird eine solche Verwaltung mit Hilfe von Microsoft's Active Directory implementiert. Eine interessante Funktionalität ist dabei das Single Sign-On (SSO) Prinzip. Dabei wird eine Authentifizierung eines Benutzers bei der Anmeldung an seinem Arbeitsrechner durchgeführt. Bei einem Zugriff auf ein System welches SSO unterstützt, wird keine erneute Password-Abfrage durchgeführt. Vielmehr wird mit Hilfe eines Tokens festgestellt, dass die Authentifizierung bereits erfolgt ist.

Abgesehen davon, dass dies eine Erleichterung für den Benutzer darstellt, werden in dem Prozess auch keine sensiblen Daten ausgetauscht, was zu einer deutlichen Erhöhung der Sicherheit führt.

Anhand einer Beispielumgebung wird in diesem Abschnitt gezeigt, wie eine Anbindung von BICsuite an das Active Directory gemacht werden kann. Es wird einfach sein, anhand dieser Beschreibung die Abbildung in die eigene Umgebung zu machen.

In der Beispielumgebung ist der BICsuite Server sowie der Zope Server auf ein CentOS 7 Linux System installiert. Der Rechner heißt `centos7sso`. Im Netzwerk befindet sich auch ein Active Directory Server mit Namen `adserver`, sowie ein Windows Client. Beide Windows Maschinen befinden sich in der Windows Domäne `INDEPENDIT.DE`.

Vorgehen

Zuerst wird ein Apache Webserver installiert, mit den benötigten Modulen versehen, und konfiguriert. Ziel ist dabei nicht nur die SSO Funktionalität, sondern auch die Kommunikation über https. Der Apache Server wird über eine `proxy_html` Schnittstelle mit dem Zope Server kommunizieren. Da letztendlich auch der Scheduling Server eine Authentifizierung vornimmt, muss auch dieser von der SSO Situation in Kenntnis gesetzt werden.

Kerberos Installation und Konfiguration

Das SSO Protokoll baut intern auf Kerberos auf. Daher muss die dazu benötigte Software installiert und konfiguriert werden.

Die Installation ist am Einfachsten. In einer RHEL oder CentOS Umgebung reicht ein einfaches

```
yum install krb5-libs
yum install krb5-workstation
yum install sssd-krb5
yum install sssd-krb5-common
```

aus.

Die Datei `/etc/krb5.conf` enthält die Konfiguration der Kerberos Installation. In unserer Beispielumgebung sieht sie folgendermaßen aus:

```
includedir /etc/krb5.conf.d/

[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
dns_lookup_realm = false
default_keytab_name = /etc/httpd/krb5.keytab
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
pkinit_anchors = FILE:/etc/pki/tls/certs/ca-bundle.crt
default_realm = INDEPENDIT.DE
default_ccache_name = KEYRING:persistent:%{uid}

[realms]
INDEPENDIT.DE = {
kdc = ADSERVER.INDEPENDIT.DE
master_kdc = ADSERVER.INDEPENDIT.DE
admin_server = ADSERVER.INDEPENDIT.DE
default_domain = INDEPENDIT.DE
}

[login]
krb4_convert = true
krb4_get_tickets = false

[domain_realm]
.independit.de = INDEPENDIT.DE
independit.de = INDEPENDIT.DE
```

Die genannte Datei `/etc/httpd/krb5.keytab` kann auf dem Active Directory Server erzeugt und nach dem rüberkopieren dort gelöscht werden:

```

ktpass -princ HOST/centos7sso.independit.de@INDEPENDIT.DE -mapuser \
bicsuite@INDEPENDIT.DE -crypto RC4-HMAC-NT \
-ptype KRB5_NT_PRINCIPAL -pass "VerySecret" \
-out c:\temp\krb5.keytab
ktpass -princ HTTP/centos7sso.independit.de@INDEPENDIT.DE -mapuser \
bicsuite@INDEPENDIT.DE -crypto RC4-HMAC-NT \
-ptype KRB5_NT_PRINCIPAL -pass "VerySecret" \
-out c:\temp\krb5.keytab -in c:\temp\krb5.keytab

```

Es kann sein, dass anstelle von RC4-HMAC-NT eine andere Verschlüsselung, wie z.B. AES256-SHA1 gewählt werden muss. Aus Sicherheitsgründen ist die Benutzung moderner Verschlüsselungsverfahren empfehlenswert.

Um später die Authorisierung durchführen zu können, benötigt der Apache Server einen Zugang zum Active Directory Server. Im Beispiel wurde dazu der Account `bicsuite@INDEPENDIT.DE` genutzt und hat keine besonderen Privilegien. Der Name an sich ist unwichtig. Als Password wurde `VerySecret` eingestellt. Ein Password, was leicht erraten wird, auch wenn's "SehrGeheim" ist.

Ob die Kerberos Konfiguration so weit richtig ist, kann mittels

```

KRB5_TRACE=/dev/stdout kinit -k -t krb5.keytab \
-p HTTP/centos7sso.independit.de

```

überprüft werden:

```

[root@centos7sso httpd]# KRB5_TRACE=/dev/stdout kinit -k \
-t krb5.keytab -p HTTP/centos7sso.independit.de
....: Getting initial credentials for \
HTTP/centos7sso.independit.de@INDEPENDIT.DE
....: Looked up etypes in keytab: aes256-cts
....: Sending unauthenticated request
....: Sending request (225 bytes) to INDEPENDIT.DE
....: Resolving hostname ADSERVER.INDEPENDIT.DE
....: Sending initial UDP request to dgram 192.168.123.45:88
....: Received answer (204 bytes) from dgram 192.168.123.45:88
....: Response was from master KDC
....: Received error from KDC: -1765328359/Additional \
pre-authentication required
....: Preauthenticating using KDC method data
....: Processing preauth types: PA-PK-AS-REQ (16), \
PA-PK-AS-REP_OLD (15), PA-ETYPE-INFO2 (19), \
PA-ENC-TIMESTAMP (2)
....: Selected etype info: etype aes256-cts, salt \
"INDEPENDIT.DEHTTPcentos8sso.independit.de", params ""
....: Retrieving HTTP/centos8sso.independit.de@INDEPENDIT.DE \
from FILE:krb5.keytab (vno 0, enctype aes256-cts) with \
result: 0/Success
....: AS key obtained for encrypted timestamp: aes256-cts/8EB8
....: Encrypted timestamp (for 1619424732.825845): plain ...
....: Preauth module encrypted_timestamp (2) (real) returned: \
0/Success
....: Produced preauth for next request: PA-ENC-TIMESTAMP (2)

```

```

....: Sending request (305 bytes) to INDEPENDIT.DE
....: Resolving hostname ADSERVER.INDEPENDIT.DE
....: Sending initial UDP request to dgram 192.168.123.45:88
....: Received answer (98 bytes) from dgram 192.168.123.45:88
....: Response was from master KDC
....: Received error from KDC: -1765328332/Response too big for \
      UDP, retry with TCP
....: Request or response is too big for UDP; retrying with TCP
....: Sending request (305 bytes) to INDEPENDIT.DE (tcp only)
....: Resolving hostname ADSERVER.INDEPENDIT.DE
....: Initiating TCP connection to stream 192.168.123.45:88
....: Sending TCP request to stream 192.168.25.3:88
....: Received answer (1706 bytes) from stream 192.168.123.45:88
....: Terminating TCP connection to stream 192.168.123.45:88
....: Response was from master KDC
....: Processing preauth types: PA-ETYPE-INFO2 (19)
....: Selected etype info: etype aes256-cts, salt \
      "INDEPENDIT.DEHTTPcentos8sso.independit.de", params ""
....: Produced preauth for next request: (empty)
....: AS key determined by preauth: aes256-cts/8EB8
....: Decrypted AS reply; session key is: aes256-cts/9218
....: FAST negotiation: unavailable
....: Initializing KCM:0:99729 with default princ \
      HTTP/centos8sso.independit.de@INDEPENDIT.DE
....: Storing HTTP/centos8sso.independit.de@INDEPENDIT.DE -> \
      krbtgt/INDEPENDIT.DE@INDEPENDIT.DE in KCM:0:99729
....: Storing config in KCM:0:99729 for \
      krbtgt/INDEPENDIT.DE@INDEPENDIT.DE: pa_type: 2
....: Storing HTTP/centos8sso.independit.de@INDEPENDIT.DE -> \
      krb5_ccache_conf_data/pa_type/krbtgt\INDEPENDIT.DE\
      @INDEPENDIT.DE@X-CACHECONF: in KCM:0:99729

```

(Aus Darstellungsgründen wurde der Output etwas gekürzt und umformatiert).

Apache Webserver und Module

Installation

Als nächste können jetzt Apache (httpd) sowie einige benötigte Module installiert werden. Unter RHEL/CentOS 7 kann entweder das Modul `mod_auth_kerb` oder das Modul `mod_auth_gssapi` benutzt werden, ab RHEL/CentOS 8 gibt es allerdings kein Support mehr für das Kerberos Modul, so dass hier zwingend die `gssapi` Schnittstelle benutzt werden muss.

Daher entweder

```

yum install httpd
yum install mod_ssl
yum install mod_ldap
yum install mod_proxy_html
yum install mod_auth_kerb

```

für das Kerberos Modul, oder

```
yum install httpd
yum install mod_ssl
yum install mod_ldap
yum install mod_proxy_html
yum install mod_auth_gssapi
```

für das gssapi Modul.

Konfiguration

Die Konfiguration von Apache liegt bei RedHat basierenden Systemen unter `/etc/httpd` sowie in einigen Unterverzeichnissen. Das kann bei anderen Distributionen anders sein, jedoch bleibt das Prinzip gleich.

Zuerst wird dafür gesorgt, dass ausschließlich über https mit dem Apache Webserver kommuniziert wird. Dazu müssen in der Datei `/etc/httpd/conf.d/ssl.conf` einige Einträge vorgenommen werden. In der vorliegenden Beispielumgebung sind dies:

```
ServerName centos7sso.independit.de:443
SSLCertificateFile /etc/pki/tls/certs/centos7sso.crt
SSLCertificateKeyFile /etc/pki/tls/private/centos7sso.key
```

Je nach Umgebung kann es natürlich sein, dass auch die Intermediate Certificates noch eine Rolle spielen.

In der Datei `etc/httpd/conf/httpd.conf` wird dann noch dafür gesorgt, dass eventuelle Anfragen auf dem Standard http Port auf den https Port umgeleitet werden:

```
<VirtualHost _default_:80>
    ServerName centos7sso.independit.de:80
    #
    # force the use of https
    #
    Redirect permanent / https://centos7sso.independit.de/
</VirtualHost>
```

Selbstverständlich sollte auch die Firewall informiert werden, zum Beispiel:

```
firewall-cmd --zone=public --add-service=https
firewall-cmd --zone=public --permanent --add-service=https
```

Die Authentifizierung des Benutzers sieht etwas komplizierter aus und muss mit Information aus der Umgebung angereichert werden. Dabei gibt es, abhängig von dem benutzten Modul, zwei ähnliche, dennoch unterschiedliche Konfigurationen. Im Falle von `mod_auth_kerb` funktioniert folgende Konfiguration in der Beispielumgebung:


```
# Wenn AD User vielen Gruppen zugehörigen sind kann die
# FieldSize schnell ausgereizt werden und man rennt in einen Fehler
# Deshalb wird diese hier vergrößert
LimitRequestFieldSize 32768
<Location "/bicsuite">
    AuthType          Kerberos
    KrbAuthRealms      INDEPENDIT.DE
    KrbServiceName      HTTP
    Krb5Keytab          /etc/httpd/krb5.keytab
    KrbMethodNegotiate On
    KrbMethodK5Passwd  Off
    require valid-user

    RewriteEngine on
    RewriteCond %{REMOTE_USER} (.*?)
    RewriteRule .* - [E=X_REMOTE_USER:%1]
    RequestHeader set REMOTE_USER %{X_REMOTE_USER}e
    RequestHeader set X-Remote-User %{REMOTE_USER}s

    SetHandler "proxy:http://127.0.0.1:8080"
    SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
</Location>
```

Die Konfiguration für mod_auth_gssapi sieht ähnlich, sogar einfacher aus:

```
<Location "/bicsuite">
    AuthType          GSSAPI
    AuthName          "INDEPENDIT.DE"
    GssapiCredStore    keytab:/etc/httpd/krb5.keytab
    GssapiSSLOnly      On
    GssapiLocalname    Off

    Require valid-user

    RewriteEngine on
    RewriteCond %{REMOTE_USER} (.*?)
    RewriteRule .* - [E=X_REMOTE_USER:%1]
    RequestHeader set REMOTE_USER %{X_REMOTE_USER}e
    RequestHeader set X-Remote-User %{REMOTE_USER}s

    SetHandler "proxy:http://127.0.0.1:8080/"
    SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
</Location>
```

Wenn der Webbenutzer die Location bicsuite, oder natürlich eine Ressource unterhalb des bicsuite Folders anfordert, dann wird mittels Kerberos geprüft, ob er dazu berechtigt ist. Dazu wird die Domäne, in unserem Fall INDEPENDIT.DE benötigt, sowie auch eine Datei krb5.keytab, die im Konfigurationsverzeichnis von Apache abgelegt wurde.

Wenn die Authentifizierung erfolgreich ist, wird die Anfrage mittels http an den Zope Server weitergeleitet. Eventuelle Authorization Headers werden dabei ebenfalls mitgegeben.

selinux

Wenn selinux aktiviert ist, ist es dem Apache Server untersagt selbst Socket-Verbindungen zu öffnen. Da dies aber für die Kommunikation zwischen Apache und Zope benötigt wird, muss dies erlaubt werden:

```
/usr/sbin/setsebool -P httpd_can_network_connect 1
```

Ergebnistest

Ob alles so weit gut gegangen ist, kann einfach getestet werden. Dazu wird im DocumentRoot, in diesem Fall `/var/www/html`, ein Verzeichnis, z.B. `ssotest`, angelegt. In dem Verzeichnis wird eine Datei `index.html` angelegt, mit z.B. folgendem Inhalt:

```
<DOCTYPE html>
<html>
  <body>

    <h1>My First Heading</h1>

    <p>Hello World</p>

  </body>
</html>
```

Für den Test muss jetzt noch vorübergehend die Apache Konfiguration angepasst werden. Dazu wird der `location` Abschnitt wie folgt modifiziert:

```
<Location "/ssotest">
    AuthType            Kerberos
    KrbAuthRealms        INDEPENDIT.DE
    KrbServiceName       HTTP
    Krb5Keytab            /etc/httpd/krb5.keytab
    KrbMethodNegotiate    On
    KrbMethodK5Passwd     Off
    require valid-user

    # SetHandler "proxy:http://127.0.0.1:8080"
    # SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
</Location>
```

Statt `Location bicsuite` wird das neu erzeugte Verzeichnis `ssotest` angesprochen. Weiterhin wird verhindert, dass Apache die Anfrage an Zope weiterleitet. Wenn nach einem Neustart des Apache Servers von einer Windows Workstation auf `https://centos7sso/ssotest` zugegriffen wird, sollte das "Hello World" angezeigt werden. Ein Aufruf von `wget` vom Server selbst, wie zum Beispiel `wget https://localhost/ssotest`, sollte ein "401 Unauthorized" zur Folge haben.

Sofern der Test erfolgreich war, sollten die Änderungen jetzt wieder zurückgenommen werden.

Zope Erweiterung und Konfiguration

Der zweite Mitspieler ist der Zope Server, der natürlich auch von seinem Glück erfahren soll. Auch Zope muss in der Lage sein, mit dem Active Directory Server zu reden. Dazu benötigt er das `python_ldap` Paket, dessen Installation wiederum das `openldap-devel` Paket benötigt:

```
yum install openldap-devel
cd $BICSUITEHOME/./software/Zope
bin/pip install python-ldap
```

Es empfiehlt sich die Vorbereitung von Zope auf die Verwendung von SSO vor dem Einspielen vom `SDMS.zexp` zu machen. Allerdings ist das Aufsetzen von SSO auch nachträglich möglich, bedarf dann aber ein paar extra Schritten. Insbesondere ist zu beachten, dass die URL, mit der die GUI angesprochen wird, sich im Vergleich mit einer normalen Installation ändert.

Product Installation

Damit Zope Anfragen im richtigen Kontext ausführen kann, wird eine Benutzerkennung benötigt. Ohne SSO erfolgt eine Anmeldung innerhalb von Zope und damit ist die Information vorhanden. Im SSO Fall bekommt Zope die Benutzerinformation von dem Apache Server übermittelt. Damit Zope mit der Information umgehen kann, wird eine Erweiterung benötigt.

Das Installieren geht einfach:

```
cd $BICSUITEHOME/./bicsuiteweb/Products
ln -s $BICSUITEHOME/bicsuite/zope/RemoteUserFolder .
```

Es ist wichtig den Port 8080 zwar für Apache, aber nicht von außen zugänglich zu machen. Mit Hilfe von `firewalld` kann das wie folgt aussehen:

```
firewall-cmd --permanent --zone=public --add-rich-rule='
rule family="ipv4"
source address="127.0.0.1/32"
port protocol="tcp" port="8080" accept'
```

Installieren von SDMS.zexp

Die Installation der GUI Anwendung läuft analog zu der Installation ohne SSO. Wichtig jedoch ist, dass die Installation diesmal nicht im Root-Folder sondern in einem Unterfolder erfolgt. Wie der Folder heißt, ist weniger wichtig, so fern die Namensgebung überall korrekt beibehalten wird.

Aus technischen Gründen werden zwei leere Folder, sowie ein Folder für die Installation der Software im Root-Folder von Zope benötigt. Dazu wird die Management Oberfläche von Zope, unter Umgehung des Apache Servers aufgerufen. In der Beispielumgebung wird dazu also die URL

`http://centos7sso.independit.de:8080/manage`

in den Browser angegeben. Wenn eine Login Aufforderung erscheint, müssen jetzt die Daten des Benutzers, der bei der Installation von Zope angegeben wurde, benutzt werden. Typischerweise ist das der Benutzer `sdmsadm`. Zuerst wird jetzt ein Folder `bicsuite`, ein Folder `web`, sowie ein Folder `GUI` angelegt. Und es wird ein Property `SDMSROOT` auf den Wert `/bicsuite/GUI` erzeugt. Später wird die GUI dann unter der URL

`https://centos7sso.independit.de/bicsuite/GUI/SDMS`

aufgerufen werden können.

Jetzt folgen alle Schritte wie bei der normalen Installation, allerdings wird `SDMS.zexp` in den Folder `GUI` importiert. Auch das Anlegen der Folders `Custom` und `User` erfolgt im Folder `GUI`.

Im `Custom` Folder gibt es ein Konfigurationsscript namens `SDMSServers`, in dem festgehalten wird, welche Scheduling Servers von der Zope Instanz aus bedient werden können. Zope muss wissen, dass SSO benutzt werden soll:

```
#
# define all accessible SDMS Servers here
#
return {
  'DEFAULT' : {
    'HOST'    : 'localhost',
    'PORT'    : '2506',
    'VERSION' : 'BASIC',
    'CACHE'   : 'Y',
    'TIMEOUT' : '60',
    'SSO'     : True, # <-- Benutze SSO
    'SSL'     : 'N',
  }
  #
  # Edit following Options if SSL set to 'Y'
  #
  # 'TRUSTSTORE' : 'truststore.pem',
  # 'KEYSTORE'   : 'keystore.pem'
}
}
```

Zum Schluss wird noch ein Objekt vom Typ `RemoteUserFolder` im Folder `GUI` angelegt.

Konfiguration der SDMS Anwendung

Unter `$BICSUITEHOME/etc` liegt eine Datei namens `ZopeSSO.conf.template`. Diese wird nach dem Konfigurationsverzeichnis, dies ist üblicherweise das Verzeichnis `$BICSUITEHOME/./etc` kopiert und nach `ZopeSSO.conf` umbenannt. Die Datei ist ziemlich selbsterklärend. Es geht los mit allgemeinen Einstellungen, die dann pro Domäne, oder Server, überschrieben werden können.

In der Beispielumgebung wurde mit folgenden Einstellungen gearbeitet:

```

# =====
# ZopeSSO.conf.template
#
# Copy this file to the BICSUITECONFIG directory and edit it according
# to your needs
# At least all properties set to <TO_BE_CONFIGURED> have to be set.
# =====
# Configurations for handling SSO for the BICsuite web frontend
#
# WARNING:
# This file contains credentials for LDAP and BICsuite ADMIN access.
# Make this file only readable for the user running the Zope
# application server
{
    #=====
    # General configurations which can be (partially) overridden by
    # domain-specific or server-specific configurations
    #-----
    # Defaults for domain specific settings if not set in the DOMAINS
    # section
    #- - - - -
    # WebNameCase defines how names for Zope authenticated user names
    # are converted
    # 'UPPER' convert to upper case
    # 'LOWER' convert to lower case
    # 'MIXED' no conversion (default)
    'WebNameCase' : 'MIXED',
    #- - - - -
    # WebAutoCreateUsers indicates if AD users should be created
    # automatically as BICsuite frontend users. If WebUseLdapGroups,
    # is set to True, only AD users who are members of the UserGroup
    # and/or ManagerGroup below will be allowed.
    # True
    # False (default)
    'WebAutoCreateUsers' : True,
    #- - - - -
    # WebUseLdapGroups indicates if ldap groups should be used to
    # detect whether an AD user is allowed to log in to the BICsuite
    # web frontend
    # True
    # False (default)
    # 'WebUseLdapGroups' : True,
    #- - - - -
    # WebIncludeDomainNames indicates if Domain Names should be part
    # of web user identifiers
    # True or False
    # defaults to False
    'WebIncludeDomainNames' : False,
    #- - - - -
    # WebUserGroup allowed to log in in via SSO
    # defaults to 'BICSUITE_WEB_USER'
    'WebUserGroup' : 'bicsuite',
    #- - - - -

```

```

# manager group granting manage privilege on Zope website
# defaults to 'BICSUITE_WEB_MANAGER'
'WebManagerGroup' : 'bicsuite_admin',
#-----
# WebGroupCheckIntervall is the time in minutes after which ldap
# group assignments for a BICSuite web server are checked again
# defaults to 60 (1 hour)
# 'WebGroupCheckIntervall' : 60
#-----
# Defaults for server-specific settings if not set in the SERVERS
# section
#-----
# ServerIncludeUserDomainNames indicates if domain names should be
# part of user identifiers
# True or False
# defaults to False
'ServerIncludeUserDomainNames' : False,
#-----
# ServerIncludeGroupDomainNames indicates if domain names should
# part of group identifiers
# True or False
# defaults to False
'ServerIncludeGroupDomainNames' : False,
#-----
# ServerUserNameCase defines how names for BICSuite are converted
# UPPER case
# LOWER case
# MIXED case (don't convert them)
# defaults to 'UPPER'
'ServerUserNameCase' : 'UPPER',
#-----
# ServerGroupNameCase defines how names for BICSuite groups are
# converted
# UPPER case
# LOWER case
# MIXED case (don't convert them)
# defaults to 'UPPER'
'ServerGroupNameCase' : 'UPPER',
#-----
# ServerAutoCreateUsers indicates if AD users should be created
# automatically. If ServerUseLdapGroups is True, only AD users
# who are a member of any AD group named
# <ServerBicsuitePrefix>_<ServerName>_<groupname> are allowed
# True or False
# defaults to False
'ServerAutoCreateUsers' : True,
#-----
# ServerUseLdapGroups indicates if AD groups should be used
# True or False
# defaults to False
'ServerUseLdapGroups' : True,
#-----
# ServerBicsuitePrefix is the prefix used for AD groups. Groups

```

```

# called other than <ServerBicsuitePrefix>_<ServerName>_<groupname>
# are ignored
# defaults to 'BICSUITE'
'ServerBicsuitePrefix' : 'bicsuite',
#-----
# ServerName is the server name used for AD groups. Groups called
# other than <ServerBicsuitePrefix>_<ServerName>_<groupname>
# are ignored
# defaults to 'DEFAULT'
'ServerName' : 'centos7sso',
#-----
# ServerDefaultGroupSuffix
# Suffix used to decide whether a AD group should be the default
# group defaults to '_ISDEFAULT'
'ServerDefaultGroupSuffix' : '_ISDEFAULT',
#=====
# Domain-specific configurations independent of the BICsuite
# server. Accessing users from domains not configured here will not
# be able to log on to the BICsuite web frontend via SSO
#-----
'DOMAINS' : {
    # domain name as in <DOMAIN_NAME>\UserName
    'INDEPENDIT.DE' : {
        #-----
        # Domain-specific settings or one BICsuite server
        #-----
        # ldap server and base to get group membership from
        # Example:
        'LdapServer' : 'ldap://192.168.0.1',
        'LdapServer' : 'ldap://adserver.independit.de',
        # Example:
        'LdapBaseDn' : 'DC=INDEPENDIT,DC=dieter,DC=de',
        'LdapBaseDn' : 'DC=INDEPENDIT,DC=de',
        #-----
        # ldap credentials to use for group membership retrieval
        # Example
        'LdapUsername' : 'Administrator@INDEPENDIT.DIETER.DE',
        'LdapUsername' : 'bicsuite_admin',
        'LdapPassword' : 'G0H0ME-123',
        #-----
        # WebNameCase defines if names for Zope authenticated user
        # names have to be converted to
        # 'UPPER' convert to upper case (default)
        # 'LOWER' convert to lower case
        # 'MIXED' no conversion
        'WebNameCase' : 'UPPER',
        #-----
        # WebAutoCreateUsers indicates if AD users should be created
        # automatically as BICsuite frontend users. If
        # UseLdapWebGroups is set to True, only AD users who are
        # members of the UserGroup and/or ManagerGroup below will
        # be allowed.
        # True
    }
}

```

```

# False (default)
'WebAutoCreateUsers' : True,
#-----
# WebUseLdapGroups indicates if Ldap groups should be used
# to detect whether AD user is allowed to log in to the
# BICsuite web frontend
# True
# False (default)
# 'WebUseWebGroups' : False,
#-----
# WebIncludeDomainNames indicates if Domain Names should be
# part of web user identifiers
# True or False
# defaults to False
'WebIncludeDomainNames' : False,
#-----
# WebUserGroup allowed to log in in via SSO
# defaults to 'BICSUITE_WEB_USER'
'WebUserGroup' : 'bicsuite',
#-----
# manager group granting manage privilege on Zope website
# defaults to 'BICSUITE_WEB_MANAGER'
'WebManagerGroup' : 'bicsuite_admin',
#-----
# WebGroupCheckIntervall is the time in minutes after which ldap
# group assignments for a BICsuite web server are
# checked again
# defaults to 60 (1 hour)
'WebGroupCheckIntervall' : 60
#-----
}

},
#=====
# BICsuite server-specific configurations
#-----
'SERVERS' : {
#-----
# For every BICsuite server to be accessed via SSO, the following
# section must be created with hostname:port
# Example: localhost:2506
'localhost:2506' : {
#-----
# General configuration for a BICsuite server independent
# of the login domain
#-----
# login credentials for a BICsuite admin user who is allowed
# to manage users and group. Used also to connect to BICsuite
# before sending the 'alter session set user' command when
# executing statements vis-a-vis BICsuite for a user
'AdminUser'      : 'SYSTEM',
'AdminPassword'  : 'GOHOME',
#-----
# Defaults for server-specific settings if not set in the

```



```

# DOMAINS section
#-----
# ServerIncludeUserDomainNames indicates if domain names
# should be part of user identifiers
# True or False
# defaults to False
'ServerIncludeUserDomainNames' : False,
#-----
# ServerIncludeGroupDomainNames indicates if domain names
# should be part of group identifiers
# True or False
# defaults to False
'ServerIncludeGroupDomainNames' : False,
#-----
# ServerUserNameCase defines how names for BICSuite users
# are converted
# UPPER case
# LOWER case
# MIXED case (don't convert them)
# defaults to 'UPPER'
'ServerUserNameCase' : 'UPPER',
#-----
# ServerGroupNameCase defines how names for BICSuite groups
# are converted
# UPPER case
# LOWER case
# MIXED case (don't convert them)
# defaults to 'UPPER'
'ServerGroupNameCase' : 'UPPER',
#-----
# ServerAutoCreateUsers indicates if AD users should be
# created automatically. If ServerUseLdapGroups is True
# only AD users who are a member of any AD group named
# <ServerBicsuitePrefix>_<ServerName>_<groupname> are allowed
# True or False
# defaults to False
'ServerAutoCreateUsers' : True,
#-----
# ServerUseLdapGroups indicates if AD groups should be used
# True or False
# defaults to False
'ServerUseLdapGroups' : True,
#-----
# ServerBicsuitePrefix is the prefix used for AD groups.
# Groups called other than
# <ServerBicsuitePrefix>_<ServerName>_<groupname> are ignored
# defaults to 'BICSUITE'
'ServerBicsuitePrefix' : 'BICSUITE',
#-----
# ServerName is the server name used for AD groups. Groups
# called other than
# <ServerBicsuitePrefix>_<ServerName>_<groupname> are ignored
# defaults to 'DEFAULT'

```

```

'ServerName' : 'centos7sso',
#-----
# ServerDefaultGroupSuffix
# Suffix used to decide whether a AD group should be the
# default group
# defaults to '_ISDEFAULT'
# 'ServerDefaultGroupSuffix' : '_ISDEFAULT',
#-----
# Domain-specific configurations for this BICsuite server
#-----
'DOMAINS' : {
    # optional server and domain-specific configuration
    # overriding server and base defaults
    'INDEPENDIT.DE' : {
        #-----
        # Server-specific settings for this domain
        #-----
        # ServerIncludeUserDomainNames indicates if
        # domain names should be part of user
        # identifiers
        # True or False
        # defaults to False
        # 'ServerIncludeUserDomainNames' : False,
        #-----
        # ServerIncludeGroupDomainNames indicates if
        # domain names should be part of group
        # identifiers
        # True or False
        # defaults to False
        # 'ServerIncludeGroupDomainNames' : False,
        #-----
        # ServerUserNameCase defines how names for
        # BICsuite users
        # are converted
        # UPPER case
        # LOWER case
        # MIXED case (don't convert them)
        # defaults to 'UPPER'
        # 'ServerUserNameCase' : 'UPPER',
        #-----
        # ServerGroupNameCase defines how names for
        # BICsuite groups are converted
        # UPPER case
        # LOWER case
        # MIXED case (don't convert them)
        # defaults to 'UPPER'
        # 'ServerUserNameCase' : 'UPPER',
        #-----
        # ServerAutoCreateUsers indicates if AD users
        # should be created automatically.
        # If ServerUseLdapGroups is True, only AD users
        # who are a member of any AD group named
        # <ServerBicsuitePrefix>_<ServerName>_<groupname>

```

```

# are allowed
# True or False
# defaults to False
# 'ServerAutoCreateUsers' : False,
#-----
# ServerUseLdapGroups indicates if AD groups
# should be used
# True or False
# defaults to False
# 'ServerUseLdapGroups' : False,
#-----
# ServerBicsuitePrefix is the prefix used for AD
# groups. Groups called other than
# <ServerBicsuitePrefix>_<ServerName>_<groupname>
# are ignored
# defaults to 'BICSUITE'
# 'ServerBicsuitePrefix' : 'BICSUITE',
#-----
# ServerName is the server name used for AD
# groups. Groups called other than
# <ServerBicsuitePrefix>_<ServerName>_<groupname>
# are ignored
# defaults to 'DEFAULT'
# 'ServerName' : 'DEFAULT',
#-----
# ServerDefaultGroupSuffix
# Suffix used to decide whether a AD group
# should be the default group
# defaults to '_ISDEFAULT'
# 'ServerDefaultGroupSuffix' : '_ISDEFAULT',
#-----
    }
}
}
}
}

```

Konfiguration des BICsuite Servers

Auch der BICsuite Server braucht noch etwas Konfiguration, damit bekannt ist, wie er sich im Falle von SSO Anmeldungen verhalten soll. Wichtig dabei ist, dass die Einstellungen zu den Einstellungen des Zope Servers passen. In der Beispielumgebung wurden die relevanten Parameter wie folgt gesetzt:

```

#
# SSOincludeDomainNames indicates if Domain Names should
# be part of Group/User Identifiers
# default = false
#
SSOincludeDomainNames=false

```

```

#
# SSOautoCreateUsers indicates if AD users should be
# created automatically
# default = false
#
SSOautoCreateUsers=true

#
# SSOautoCreateGroups indicates if AD groups should be
# created automatically
# default = false
#
SSOautoCreateGroups=true

#
# SSOuseADGroups indicates if AD groups should be used
# or not
# default = false
#
SSOuseADGroups=true

#
# SSOserverName is the name of the server (within AD);
# this is used to filter out groups
#
SSOserverName=centos7sso

#
# SSObicsuitePrefix is the prefix used for AD groups.
# Groups called otherwise than
#   <SSObicsuitePrefix>_<SSOserverName>_<groupname>
# are ignored
#
SSObicsuitePrefix=bicsuite

#
# SSOnameCase defines if names have to be converted to
# UPPER case (= default value; make identifier case
#             insensitive if they adhere to BICsuite
#             naming standards)
# LOWER case
# MIXED case (= don't convert them)
#
SSOnameCase=UPPER

```

Einstellungen an Benutzerseite

Damit SSO funktioniert, muss der verwendete Browser wissen, dass er die Credentials dem Apache Server melden soll. Je nach Browser ist jedoch die Konfiguration etwas unterschiedlich. In den nächsten Abschnitten wird für einige häufig benutzte Browser erklärt welche Einstellungen gesetzt werden müssen. Die Reihenfolge ist

rein alphabetisch.

Chrome und Microsoft Edge

Bei Chrome und Edge ist die Prozedur identisch. In Windows Systemsteuerung → Netzwerk und Internet → Internetoptionen → Tab Sicherheit wählen Sie das lokale Intranet. Nach einem Klick auf den Sites und anschließend den Erweitert Button, tragen Sie den Hostname des Apache Servers ein.

Firefox

In Firefox geben Sie die URL `about:config` ein. Dann suchen Sie nach dem Wort `negotiate`. Einer der Suchergebnisse sollte

```
network.negotiate-auth.trusted-uris
```

sein.

Ein Doppelklick erlaubt die Eingabe eines Wertes. Dabei wird als Wert der Hostname des Apache Servers eingetragen. In der Beispielumgebung ist das `centos7sso.independit.de`.

Nach dem Speichern und einem Neustart des Browsers sollte die GUI ohne Password-Eingabe erreichbar sein.

Administration des Zope Servers

Für die Administration des Zope Servers wird ein von der SSO Logik getrennter Zugang benötigt. Dies liegt darin begründet, dass die Benutzer die mittels SSO einen Zugang bekommen anders behandelt werden als "normale" Zope Benutzer. Dennoch ist es wünschenswert, auch die Administration über eine gesicherte Verbindung abzuwickeln. Dies ist, mit einigen wenigen Einschränkungen, auch problemlos möglich. Prinzipiell wird, wie bereits beschrieben, der Apache Server als Reverse Proxy aufgesetzt. Allerdings dürfen Anfragen die `/bicsuite` referenzieren nicht übersetzt werden, da hier die SSO Logik greifen soll.

Damit ergibt sich folgende Ergänzung der Apache Konfiguration:

```
ProxyRequests Off
ProxyPreserveHost On
ProxyPass / http://127.0.0.1:8080/VirtualHostBase/https/\
centos7sso.independit.de:443/VirtualHostRoot/
ProxyPassReverse / http://127.0.0.1:8080/VirtualHostBase/\
http/https/centos7sso.independit.de:443/VirtualHostRoot/
RequestHeader set X-Forwarded-Proto "https"
```

Bitte beachten: Aus Gründen der Darstellung wurde die Zeile umgebrochen. Dies wird hier, wie so üblich, mit einem Backslash gekennzeichnet. In der tatsächlichen

Konfiguration sollten die beiden Zeilen wieder, ohne Backslash und Whitespace, aneinandergelinkt werden.

Diese Ergänzung kann direkt hinter den zuvor zugefügten Anweisungen in `ssl.conf` vorgenommen werden.

Es ist ratsam für administrative Aufgaben einen anderen als den Standard Browser zu benutzen, sonst entstehen leicht unerwünschte Nebeneffekte. Man kann nun mal nicht gleichzeitig als zwei verschiedene Benutzer in einem Browser an einer Zope Instanz angemeldet sein.

3 Installation in einer Windows Umgebung

Installation des BICsuite Servers

Einleitung

Diese Anleitung geht davon aus, dass Sie die BICsuite Distribution in einem lokalen Verzeichnis entpackt haben. Wir empfehlen als Verzeichnis `C:\Programme`. Nach dem Entpacken befindet sich die Distribution dann in dem Verzeichnis

`C:\Programme\bicsuite`.

Für einige Installationsschritte wird die Eingabeaufforderung benötigt. In diesem Dokument gehen viele Anweisungen, beziehungsweise Beispiele, davon aus, dass mit der Eingabeaufforderung gearbeitet wird. In den Beispielen wird in dem Fall ein `C:>` als Prompt gezeigt. Dieser soll natürlich nicht mit eingegeben werden. Vielfach können Aktionen auch (einfacher) über die Windows-Oberfläche erledigt werden. Es ist dem Anwender selbstverständlich freigestellt den für sich einfachsten Weg zu wählen.

Weiterhin wurden in den Beispielen aus satztechnischen Gründen einige Zeilen umgebrochen. Diese kann man daran erkennen, dass die Zeile eingerückt ist und kein Prompt am Anfang der Zeile vorhanden ist. Beim Abschreiben der Befehle darf dann *kein* Return an der Stelle des Zeilenumbruchs eingegeben werden.

Installation

1. Anlegen Installationsverzeichnis

Legen Sie einen Ordner an, in welchem die BICsuite Software installiert werden soll. Im folgenden Text verweisen wir auf dieses Verzeichnis mit `INSTALLDIR`.

Beispiel:

```
C:> cd \Programme
C:> mkdir bicsuite
```

2. Entpacken Sie die BICsuite Distribution unter dem Verzeichnis `INSTALLDIR`

Im Verzeichnis `INSTALLDIR` befindet sich nun ein Unterverzeichnis mit Namen `bicsuite-release`.

Beispiel:

```
C:\Programme\bicsuite\bicsuite-2.10
```

3. Setzen der Umgebungsvariablen BICSUITEHOME

Schließen Sie alle Windows-Eingabeaufforderungen. Über Systemsteuerung → System → Erweitert → Umgebungsvariablen die Variable `BICSUITEHOME` als Systemvariable auf das BICSuite Distributionsverzeichnis setzen

Beispiel:

```
BICSUITEHOME=C:\Programme\bicsuite\bicsuite-2.10
```

4. Setzen der optionalen Umgebungsvariablen BICSUITECONFIG

Wir empfehlen die BICSuite Konfigurationsdateien nicht unter `%BICSUITEHOME%` abzulegen, sondern dies in einem separaten Verzeichnis zu tun.

Schließen Sie alle Windows-Eingabeaufforderungen. Über Systemsteuerung → System → Erweitert → Umgebungsvariablen die Variable `BICSUITECONFIG` als Systemvariable auf das BICSuite Konfigurationsverzeichnis setzen.

Beispiel:

```
BICSUITECONFIG=C:\Programme\bicsuite\etc
```

Ein späteres Upgrade auf eine neuere BICSuite Release wird dadurch vereinfacht.

5. Anlegen des Konfigurationsfolders

Soweit die Konfigurationsdateien nicht unter `%BICSUITEHOME%` abgelegt werden, muss der Konfigurationsfolder eventuell noch angelegt werden. Etwa:

```
C:> mkdir "%BICSUITECONFIG%"
```

Unabhängig davon, ob ein getrennter Folder für die Konfiguration verwendet wird, oder nicht, werden Konfigurationsdateien benötigt, die aus den mitgelieferten Templates erzeugt werden können. Diese werden dann im späteren Verlauf der Installation noch angepasst.

Öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable `BICSUITECONFIG` gesetzt haben verwenden Sie statt `%BICSUITECONFIG%` `%BICSUITEHOME%\etc`.

Ein reines Kopieren der Template-Konfiguration genügt vorerst:

```
C:> cd /d "%BICSUITEHOME%\etc"
C:> copy server_template.conf "%BICSUITECONFIG%\server.conf"
C:> copy BICSUITE_CONF_TEMPLATE.BAT "%BICSUITECONFIG%\BICSUITE_CONF.BAT"
C:> copy JAVA_CONF_TEMPLATE.BAT "%BICSUITECONFIG%\JAVA_CONF.BAT"
```


In der Datei `BICSUITE_CONF.BAT` kann der Wert der Variablen `BICSUITELOGDIR` bei Bedarf so geändert werden, dass das Logging des Systems außerhalb des Installationsverzeichnisses stattfindet. Natürlich muss in diesem Fall dafür gesorgt werden, dass das Verzeichnis auch existiert.

Wenn das Java Executable nicht im Pfad liegt, sollte in der Datei `JAVA_CONF.BAT` die Variable `BICSUITEJAVA` auf den vollqualifizierten Pfad gesetzt werden. Zum Beispiel:

```
SET BICSUITEJAVA=C:\Program Files (x86)\Java\bin\java
```

6. Herunterladen und Installieren eines von BICsuite unterstützten Datenbank Management Systems

BICsuite für Windows unterstützt derzeit die Systeme:

- Postgres (Seite [65](#))
- MySQL (Seite [67](#))
- Microsoft™ SQL Server (Seite [70](#))
- Oracle (Seite [76](#))
- Ingres (Seite [72](#))

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

7. Ändern der Umgebungsvariablen PATH

Schließen Sie alle Windows-Eingabeaufforderungen. Über Systemsteuerung → System → Erweitert → Umgebungsvariablen die Systemvariable PATH erweitern mit:

```
%BICSUITEHOME%\bin
```

Beispiel:

```
PATH= ... ;C:\Programme\bicsuite\bicsuite-2.10\bin
```

8. Falls nicht vorhanden, muss ein aktuelles Java Runtime Environment installiert werden.

Ein jre kann von

<http://www.oracle.com/technetwork/java/index.html>

heruntergeladen und installiert werden.

9. Starten des BICsuite Servers

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Führen Sie das BICsuite Commandline Utility `RUN_SERVER` aus.

```
C:> RUN_SERVER
```

Lassen Sie dieses Fenster für das weitere Vorgehen geöffnet.

10. Optionale Datei `.sdmshrc` im Home Verzeichnis des Windows Benutzers anlegen

Wird diese Datei erzeugt, so wird diese von den meisten BICsuite Commandline Utilities gelesen, um fehlende Parameter zu ergänzen.

So müssen zum Beispiel beim Aufruf von `sdmsh` nicht jedesmal Host Port User und Password angegeben werden.

Beispieldatei:

```
User=system
Password=G0H0ME
Host=localhost
Port=2506
```

Wichtig ist, dass in dieser Datei eventuell das Passwort für den Zugang zum Scheduling System steht und diese Datei daher gegen Zugriff, auch lesenden Zugriff von anderen Benutzern geschützt werden sollte.

11. Test

Öffnen Sie eine Windows-Eingabeaufforderung.

Führen Sie das BICsuite Commandline Utility `sdmsh` aus:

```
C:> sdmsh
```

Oder, falls Sie keine Datei `.sdmsrc` erzeugt haben:

```
C:> sdmsh -u SYSTEM -w G0H0ME -h localhost -p 2506
```

Wenn die Anmeldung am Server erfolgreich war, wird ein entsprechender Prompt gezeigt. Es gibt dann die Möglichkeit Befehle an den Scheduling Server abzusetzen.

```
[SYSTEM@localhost:2506] SDMS> show system;
```

Jetzt sollten Systeminformationen des laufenden BICsuite Servers angezeigt werden.

Das Utility kann folgendermaßen wieder verlassen werden:

```
[SYSTEM@localhost:2506] SDMS> exit
```

12. Convenience Package installieren

Das Convenience Package installiert eine übliche Konfiguration eines Exit State Models. Es wird in der folgenden Befehlszeile davon ausgegangen, dass die `.sdmshrc` Datei vorhanden ist. Sollte dies nicht der Fall sein, muss die Kommandozeile entsprechend des vorherigen Schritts erweitert werden.

```
C:> type "%BICSUITEHOME%\install\convenience.sdms | sdmsh
```

13. Beispiele installieren

Um die mitgelieferten Beispiele von Ablaufdefinitionen zu installieren, müssen nur einige wenige Befehle ausgeführt werden:

```
C:> cd /d "%BICSUITEHOME%\install  
C:> SETUP_EXAMPLE_JOBSERVERS.BAT  
C:> type setup_examples.sdms | sdmsh
```

14. Automatischer Start des BICsuiteServers

Um den BICsuiteServer automatisch bei der Anmeldung zu starten, muss nur eine Verknüpfung vom Startskript in den Autostart-Folder angelegt werden.

Das Startskript heißt:

```
%BICSUITEHOME%\bin\RUN_SERVER.BAT
```

15. Automatischer Start der Jobserver

Um die Jobserver automatisch bei der Anmeldung zu starten, muss nur eine Verknüpfung vom Startskript in den Autostart-Folder angelegt werden.

Die Skripte sind:

```
%BICSUITEHOME%\bin\RUN_JOBSERVER_LOCALHOST.BAT  
%BICSUITEHOME%\bin\RUN_JOBSERVER_HOST_1.BAT  
%BICSUITEHOME%\bin\RUN_JOBSERVER_HOST_2.BAT
```

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 dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

Es wurden in den Beispielen aus satztechnischen Gründen einige Zeilen umgebrochen. Diese kann man daran erkennen, dass die Zeile eingerückt ist und kein Prompt am Anfang der Zeile vorhanden ist. Beim Abschreiben der Befehle darf dann *kein* Return an der Stelle des Zeilenumbruchs eingegeben werden.

Installation

1. Herunterladen und Installieren der aktuellen Postgres-Version

Am Ende der Installation mit Stackbuilder auch den Postgres JDBC Driver (pgJDBC) installieren.

Das von Ihnen während der Installation vergebene Postgres Passwort wird später noch benötigt.

Hinweis:

Falls der Postgres-Service nicht startet, kann das an einer Firewall oder Antivirensoftware liegen. Den Postgres-Service als Administrator laufen zu lassen kann dann evtl. Abhilfe schaffen.

2. Anlegen des Postgres Users `bicsuite`

Öffnen Sie eine Windows-Eingabeaufforderung und wechseln Sie ins bin-Verzeichnis Ihrer Postgres-Installation.

Beispiel:

```
C:> cd /d C:\Programme\PostgreSQL\9.2\bin
```

Führen Sie das Postgres `Createuser`-Kommando aus um einen Benutzer `bicsuite` anzulegen, welcher auch Datenbanken anlegen darf:

Beispiel:

```
C:> cd /d C:\Programme\PostgreSQL\9.2\bin
C:> .\createuser -U postgres -W -P -d bicsuite
```

Benutzen Sie hierbei das Postgres Passwort.

Merken Sie sich das von Ihnen vergebene `bicsuite` Passwort.

3. Anlegen der Repository Datenbank `bicsuitedb`

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung und wechseln Sie ins bin-Verzeichnis Ihrer Postgres-Installation. Führen Sie das Postgres `createdb`-Kommando aus, um eine Datenbank mit Namen `bicsuitedb` zu erzeugen.

Beispiel:

```
C:> cd /d C:\Programme\PostgreSQL\9.2\bin
C:> .\createdb -U bicsuite -W -O bicsuite bicsuitedb
```

Benutzen Sie hierbei das von Ihnen vergebene `bicsuite` Passwort.

4. Anlegen und Initialisierung der Datenbanktabellen

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Wechseln Sie ins BICsuite SQL-Verzeichnis.

Beispiel:

```
C:> cd /d "%BICSUITEHOME%\sql
```

Führen Sie das Postgres `PSQL`-Kommando aus um das BICsuite Repository zu initialisieren.

Beispiel:

```
C:> C:\Programme\PostgreSQL\9.2\bin\psql -U bicsuite
      -f pg\install.sql bicsuitedb
```

Benutzen Sie hierbei das von Ihnen vergebene bicsuite Passwort.

5. Konfigurieren der Datenbankverbindung in der BICsuite Server Konfigurationsdatei

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable BICSUITECONFIG gesetzt haben verwenden Sie statt %BICSUITECONFIG% %BICSUITEHOME%\etc.

Editieren Sie %BICSUITECONFIG%\server.conf und ändern Sie folgende Properties wie folgt:

```
DbPasswd=bicsuite password
DbUrl=jdbc:postgresql:bicsuitedb
DbUser=bicsuite
JdbcDriver=org.postgresql.Driver
```

Hostname muss auf Hostnamen des Rechners gesetzt werden: In der Windows-Eingabeaufforderung liefert echo %USERDOMAIN% diesen Namen

Hostname=<hier den hostname einsetzen>

6. Konfigurieren Sie den BICsuite Java Class Path für Postgres JDBC

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable BICSUITECONFIG gesetzt haben verwenden Sie statt %BICSUITECONFIG% %BICSUITEHOME%\etc.

Bearbeiten Sie %BICSUITECONFIG%\JAVA_CONF.BAT so, dass die Umgebungsvariable BICSUITEJDBC auf das Postgres JDBC jar gesetzt wird.

Beispiel:

```
SET DBMSHOME=C:\Programme\PostgreSQL
SET BICSUITEJDBC=%DBMSHOME%\pgJDBC\postgresql-9.2-1004.jdbc4.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 dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

Es wurden in den Beispielen aus satztechnischen Gründen einige Zeilen unterbrochen. Diese kann man daran erkennen, dass die Zeile eingerückt ist und kein Prompt am Anfang der Zeile vorhanden ist. Beim Abschreiben der Befehle darf dann *kein* Return an der Stelle des Zeilenumbruchs eingegeben werden.

Installation

1. Herunterladen und Installieren der aktuellen MySQL Community Version

Nach dem Herunterladen der Software wird der Instance Configuration Wizard aufgerufen.

Auswählen:

- a) Detailed Configuration
→ Next
- b) Developer Machine
→ Next
- c) Transactional Database Only
→ Next
- d) InnoDB tablespace Settings belassen
→ Next
- e) Manual Setting (10 Connections sind mehr als genug)
→ Next
- f) Add firewall exception für port setzen
Port 3306, Enable TCP und Enable Strict Mode belassen
→ Next
- g) Standard Character set belassen
→ Next
- h) Install AS Windows Service belassen
Include bin directory in windows path setzen
→ Next
- i) root_passwort eingeben (bitte merken!)
→ Next
→ Execute

2. Anlegen des MySQL Users `bicsuite` und der Datenbank `bicsuitedb`

Öffnen Sie eine Windows-Eingabeaufforderung und rufen Sie das MySQL Command Line Utility auf.

```
C:> mysql --user=root --password=root_passwort
```

Legen Sie den Benutzer `bicsuite` an und merken Sie sich das Passwort.

```
mysql> create user bicsuite identified by 'bicsuite_passwort';
```

Legen Sie die Datenbank an.

```
mysql> create database bicsuitedb;
```

Geben Sie dem Benutzer `bicsuite` alle Rechte auf diese Datenbank.

```
mysql> grant all on bicsuitedb.* to bicsuite;
```

3. Anlegen und Initialisierung der Datenbanktabellen

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung.

Führen Sie folgende Kommandos aus:

```
C:> cd /d %BICSUITEHOME%/sql
C:> mysql --user=bicsuite --password=bicsuite_passwort
      --database=bicsuitedb --execute="source mysql\install.sql"
```

4. Herunterladen und Installieren des MySQL (Connector/J) JDBC Driver

Entpacken z.B. unter C:\Programme\MySQL

Das MySQL JDBC jar Archiv liegt dann z.B. unter:

```
C:\Programme\MySQL\mysql-connector-java-5.1.16
```

5. Konfigurieren der Datenbankverbindung in der BICsuite Server Konfigurationsdatei

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung.

Falls Sie keine Umgebungsvariable BICSUITECONFIG gesetzt haben verwenden Sie statt %BICSUITECONFIG% %BICSUITEHOME%\etc.

Editieren %BICSUITECONFIG%\server.conf und ändern Sie folgende Properties wie folgt:

```
DbPasswd=bicsuite_passwort
DbUrl=jdbc:mysql:///bicsuitedb
DbUser=bicsuite
JdbcDriver=com.mysql.jdbc.Driver
```

Hostname muss auf Hostnamen des Rechners gesetzt werden: In der Windows-Eingabeaufforderung liefert echo %USERDOMAIN% diesen Namen

```
Hostname=<hier den hostname einsetzen>
```

6. Konfigurieren Sie den BICsuite Java Class Path für MySQL JDBC

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung.

Falls Sie keine Umgebungsvariable BICSUITECONFIG gesetzt haben verwenden Sie statt %BICSUITECONFIG% %BICSUITEHOME%\etc

Bearbeiten Sie %BICSUITECONFIG%\JAVA_CONF.BAT so, dass die Umgebungsvariable BICSUITEJDBC auf das MySQL JDBC jar gesetzt wird.

Beispiel:

```
SET JARHOME=C:\Programme\MySQL\mysql-connector-java-5.1.16
SET BICSUITEJDBC=%JARHOME%\mysql-connector-java-5.1.16-bin.jar
```

Installation mit Microsoft™ SQL Server

Einleitung

Diese Anleitung erhebt nicht den Anspruch eine genaue Beschreibung der Installation des Datenbanksystems zu sein. Dazu wird auf die SQL Server-Dokumentation verwiesen. Im Normalfall sollte es mit dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

Es wurden in den Beispielen aus satztechnischen Gründen einige Zeilen umgebrochen. Diese kann man daran erkennen, dass die Zeile eingerückt ist und kein Prompt am Anfang der Zeile vorhanden ist. Beim Abschreiben der Befehle darf dann *kein* Return an der Stelle des Zeilenumbruchs eingegeben werden.

Installation

1. Herunterladen und Installieren der aktuellen SQL Server Express Version

Beim Installieren gemischte (SQL und Windows) Authentifikation konfigurieren

2. Anlegen des MySQL Users `bicsuite` und der Datenbank `bicsuitedb`

Öffnen Sie eine Windows-Eingabeaufforderung und rufen Sie das SQL Server Command Line Utility auf.

```
C:> sqlcmd -S %USERDOMAIN%\SQLEXPRESS
```

Anlegen des Benutzers `bicsuite` mit

```
1> sp_addlogin 'bicsuite','bicsuite_passwort'
2> go
1> EXEC master..sp_addsrvrolemember @loginame = N'bicsuite',
    @rolename = N'dbcreator'
2> go
```

Schließen Sie das SQL Server Command Line Utility.

```
1> exit
```

Starten Sie SQL Server Command Line Utility als Benutzer `bicsuite` und legen Sie die Datenbank an.

```
C:> sqlcmd -S %USERDOMAIN%\SQLEXPRESS -U bicsuite -P bicsuite_passwort
1>create database bicsuitedb
2>go
```

Schließen Sie das SQL Server Command Line Utility.

```
1> exit
```


3. Anlegen und Initialisierung der Datenbanktabellen

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Führen Sie folgende Kommandos aus:

```
C:> cd /d %BICSUITEHOME%\sql
C:> sqlcmd -S %USERDOMAIN%\SQLEXPRESS -U bicsuite -P bicsuite_passwort
      -d bicsuitedb -i mssql\install.sql
```

4. Herunterladen und Installieren des JDBC Treibers für SQL Server

Wie von Microsoft™ empfohlen unter C:\Programme installieren.

5. Konfigurieren der Datenbankverbindung in der BICsuite Server Konfigurationsdatei

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable BICSUITECONFIG gesetzt haben verwenden Sie statt %BICSUITECONFIG% %BICSUITEHOME%\etc.

Editieren Sie %BICSUITECONFIG%\server.conf und ändern Sie folgende Properties wie folgt:

```
DbPasswd=bicsuite_passwort
DbUrl=jdbc:sqlserver://localhost;database=bicsuitedb;instance=SQLEXPRESS
DbUser=bicsuite
JdbcDriver=com.microsoft.sqlserver.jdbc.SQLServerDriver
```

Hostname muss auf Hostnamen des Rechners gesetzt werden: In der Windows-Eingabeaufforderung liefert echo %USERDOMAIN% diesen Namen.

Hostname=<hier den hostname einsetzen>

6. Konfigurieren Sie den BICsuite Java Class Path für SQL Server JDBC

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable BICSUITECONFIG gesetzt haben verwenden Sie statt %BICSUITECONFIG% %BICSUITEHOME%\etc

Bearbeiten Sie %BICSUITECONFIG%\JAVA_CONF.BAT so, dass die Umgebungsvariable BICSUITEJDBC auf das SQL Server JDBC jar gesetzt wird.

Beispiel:

```
SET DBMSHOME=C:\Programme\Microsoft SQL Server JDBC Driver 3.0
SET BICSUITEJDBC=%DBMSHOME%\sqljdbc_3.0\enu\sqljdbc4.jar
```

7. Konfigurieren TCP/IP Zugang zum SQL Server

Öffnen Sie den SQL Server Configuration-Manager aus dem Start Menue:

Start → Programme → Microsoft SQL Server 2008 R2 → Konfigurationstools
→ SQL Server-Konfigurations-Manager

Nun führen Sie folgende Schritte durch:

- a) Unter SQL Server-Netzwerkconfiguration wählen Sie Protokolle für SQL-EXPRESS
- b) Doppelklicken Sie auf TCP/IP um die Eigenschaften der TCP/IP Verbindung zu editieren
- c) Stellen Sie aktiviert auf "Ja"
- d) Wechseln Sie zum Reiter IP-Adressen
- e) Unter IPALL das Feld Dynamische TCP-Ports leeren
- f) Unter IPALL das Feld TCP-Port auf 1433 setzen

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 dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

Es wurden in den Beispielen aus satztechnischen Gründen einige Zeilen umgebrochen. Diese kann man daran erkennen, dass die Zeile eingerückt ist und kein Prompt am Anfang der Zeile vorhanden ist. Beim Abschreiben der Befehle darf dann *kein* Return an der Stelle des Zeilenumbruchs eingegeben werden.

Installation

1. Herunterladen und Installieren der aktuellen Ingres Community Edition

Im Ingres Setup Wizard auswählen:

- a) Transactional System belassen
→ Weiter
- b) Express belassen
→ Weiter
- c) Typical Server belassen
→ Next
→ Install

2. Anlegen des Users `bicsuite`

Um den Benutzer `bicsuite` im 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.

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

```
C:> 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 bicsuite 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
```

3. Anlegen der Repository Datenbank bicsuitedb

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung und legen Sie als Benutzer `bicsuite` die Repository Datenbank an.

```
C:> createdb bicsuitedb
```

4. Anlegen und Initialisierung der Datenbanktabellen

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Wechseln Sie ins BICsuite SQL Verzeichnis.

```
C:> cd /d "%BICSUITEHOME%\sql
```

Führen Sie das Ingres SQL Kommando aus um das BICsuite Repository zu initialisieren.

```
C:> sql bicsuitedb < ing\install.sql
```

5. Konfigurieren der Datenbankverbindung in der BICsuite Server Konfigurationsdatei

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable `BICSUITECONFIG` gesetzt haben verwenden Sie statt `%BICSUITECONFIG%` `%BICSUITEHOME%\etc`

Für Ihren Windows User muss ein Passwort vergeben sein !

Editieren Sie `%BICSUITECONFIG%\server.conf` und ändern Sie folgende Properties wie folgt:

```
DbPasswd=<windows passwort>
DbUrl=jdbc:ingres://localhost:II7/bicsuitedb;
DbUser=bicsuite
JdbcDriver=com.ingres.jdbc.IngresDriver
```

Ersetzen Sie `<windows password>` mit dem Windows Passwort für `bicsuite`.

Hostname muss auf Hostnamen des Rechners gesetzt werden: In der Windows-Eingabeaufforderung liefert `echo %USERDOMAIN%` diesen Namen.

Hostname=`<hier den hostname einsetzen>`

6. Konfigurieren Sie den BICSuite Java Class Path für Ingres JDBC

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable `BICSUITECONFIG` gesetzt haben verwenden Sie statt `%BICSUITECONFIG% %BICSUITEHOME%\etc`.

Bearbeiten Sie `%BICSUITECONFIG%\JAVA_CONF.BAT` so, dass die Umgebungsvariable `BICSUITEJDBC` auf das Ingres JDBC jar gesetzt wird.

Beispiel:

```
SET BICSUITEJDBC=%II_SYSTEM%\ingres\lib\iijdbc.jar
```

Installation mit DB2

Einleitung

Diese Anleitung erhebt nicht den Anspruch eine genaue Beschreibung der Installation des Datenbanksystems zu sein. Dazu wird auf die IBM DB2-Dokumentation verwiesen. Im Normalfall sollte es mit dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

Es wurden in den Beispielen aus satztechnischen Gründen einige Zeilen umgebrochen. Diese kann man daran erkennen, dass die Zeile eingerückt ist und kein Prompt am Anfang der Zeile vorhanden ist. Beim Abschreiben der Befehle darf dann *kein* Return an der Stelle des Zeilenumbruchs eingegeben werden.

Installation

1. Herunterladen und Installieren der aktuellen IBM DB2 Express-C Version
2. Anlegen des `bicsuite` Users `bicsuite`

DB2 hat keine eigene Benutzerverwaltung. Aus diesem Grund ist es nötig einen Betriebssystem- Benutzer anzulegen, welcher zur Authentifizierung bei Datenbank-Connects dient. In diesem Beispiel wurde der Windows Benutzer `bicsuite` mit Passwort `bicsuitepw` verwendet.

3. Anlegen der Datenbank `bicsuitedb`

Öffnen Sie eine Windows-Eingabeaufforderung und öffnen Sie eine Subshell für den Setup der nötigen Umgebung auf

```
C:>db2cmd -i -w db2clpsetcp
```

Rufen Sie das DB2 Command Line Utility auf.

```
C:> db2
```

Führen Sie folgende Kommandos aus:

```
db2> create database bicsuite restrictive
db2> connect to bicsuite
db2> grant dbadm, dataaccess, accessctrl, secadm on database
      to user bicsuite
db2> terminate
```

Schließen Sie die mit DB2cmd geöffnete Subshell.

```
C:>exit
```

Anmerkungen:

Das Anlegen der Datenbank dauert sehr lange, also haben Sie etwas Geduld.

Der Datenbankname darf maximal 8 Zeichen lang sein.

4. Anlegen und Initialisierung der Datenbanktabellen

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Führen Sie folgende Kommandos aus:

```
C:> cd /d %BICSUITEHOME%\sql
C:> clpplus -nw bicsuite/bicsuitepw@localhost/bicsuite
      @db2/install.sql > install.log
```

Überprüfen Sie das install.log, ob Fehler aufgetreten sind.

5. Konfigurieren der Datenbankverbindung in der BICsuite Server Konfigurationsdatei

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable BICSUITECONFIG gesetzt haben verwenden Sie statt %BICSUITECONFIG% %BICSUITEHOME%\etc

Editieren Sie %BICSUITECONFIG%\server.conf und ändern Sie folgende Properties wie folgt:

```
DbPasswd=bicsuitepw
DbUrl=jdbc:db2://localhost:50000/bicsuite
DbUser=bicsuite
JdbcDriver=com.ibm.db2.jcc.DB2Driver
```

Hostname muss auf Hostnamen des Rechners gesetzt werden: In der Windows-Eingabeaufforderung liefert das Kommando `hostname` diesen Namen.

```
Hostname=<hier den hostname einsetzen>
```

6. Konfigurieren Sie den BICsuite Java Class Path für IBM DB2 JDBC

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable BICSUITECONFIG gesetzt haben verwenden Sie statt %BICSUITECONFIG% %BICSUITEHOME%\etc.

Bearbeiten Sie %BICSUITECONFIG%\JAVA_CONF.BAT so, dass die Umgebungsvariable BICSUITEJDBC auf das IBM DB2 JDBC jar gesetzt wird.

Beispiel:

```
SET BICSUITEJDBC=C:\Programme\IBM\SQLLIB\java\db2jcc4.jar
```

Installation mit Oracle Express Edition

Einleitung

Diese Anleitung erhebt nicht den Anspruch eine genaue Beschreibung der Installation des Datenbanksystems zu sein. Dazu wird auf die Oracle-Dokumentation verwiesen. Im Normalfall sollte es mit dieser Anleitung allerdings möglich sein eine "Standard"-Installation durchzuführen.

Es wurden in den Beispielen aus satztechnischen Gründen einige Zeilen umgebrochen. Diese kann man daran erkennen, dass die Zeile eingerückt ist und kein Prompt am Anfang der Zeile vorhanden ist. Beim Abschreiben der Befehle darf dann *kein* Return an der Stelle des Zeilenumbruchs eingegeben werden.

Installation

1. Herunterladen und Installieren der aktuellen Oracle Express Edition

Merken Sie sich Ihr Oracle System Passwort

2. Anlegen des Oracle Users bicsuite

Öffnen Sie eine Windows-Eingabeaufforderung und rufen Sie das Oracle sqlplus Server Command Line Utility auf.

```
C:> sqlplus SYSTEM/oracle_system_passwort
```

Anlegen des Benutzers bicsuite mit:

```
SQL> create user bicsuite identified by bicsuite_passwort;
```

Vergeben Sie die notwendigen Zugriffsrechte:

```
SQL> grant CONNECT, RESOURCE, CREATE VIEW, CREATE PROCEDURE  
      TO bicsuite;
```

3. Anlegen und Initialisierung der Datenbanktabellen

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Wechseln Sie ins BICsuite SQL-Verzeichnis und führen Sie folgende Kommandos aus:

```
C:> cd /d %BICSUITEHOME%\sql
C:> sqlplus bicsuite/bicsuite_passwort @ora/install.sql
```

4. Konfigurieren der Datenbankverbindung in der BICsuite Server Konfigurationsdatei

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable BICSUITECONFIG gesetzt haben verwenden Sie statt %BICSUITECONFIG% %BICSUITEHOME%\etc.

Editieren Sie %BICSUITECONFIG%\server.conf und ändern Sie folgende Properties wie folgt:

```
DbPasswd=bicsuite_passwort
DbUrl=jdbc:oracle:thin:@hostname:1521:XE
DbUser=bicsuite
JdbcDriver=oracle.jdbc.OracleDriver
```

Hostname muss sowohl im obigen Beispiel als auch in der Konfigurationsdatei auf Hostnamen des Rechners gesetzt werden: In der Windows-Eingabeaufforderung liefert `echo %USERDOMAIN%` diesen Namen

Hostname=<hier den hostname einsetzen>

5. Konfigurieren Sie den BICsuite Java Class Path für Oracle JDBC

Falls noch nicht geschehen, öffnen Sie eine Windows-Eingabeaufforderung. Falls Sie keine Umgebungsvariable BICSUITECONFIG gesetzt haben verwenden Sie statt %BICSUITECONFIG% %BICSUITEHOME%\etc.

Bearbeiten Sie %BICSUITECONFIG%\JAVA_CONF.BAT so, dass die Umgebungsvariable BICSUITEJDBC auf das Oracle JDBC jar gesetzt wird.

Beispiel:

```
SET BICSUITEJDBC=%ORACLE_HOME%\jdbc\lib\ojdbc14.jar
```

Installation des Zope Servers

Einleitung

Um die BICsuite!Web User Interface-Oberfläche nutzen zu können, muss ein Zope Application Server aufgesetzt werden.

Es wurden in den Beispielen aus satztechnischen Gründen einige Zeilen umgebrochen. Diese kann man daran erkennen, dass die Zeile eingerückt ist und kein Prompt am Anfang der Zeile vorhanden ist. Beim Abschreiben der Befehle darf dann *kein* Return an der Stelle des Zeilenumbruchs eingegeben werden.

Zur Installation der Benutzeroberfläche BICsuite!Web werden einige Dateien aus der Distribution benötigt. Bevor Zope installiert werden kann, müssen die ersten drei Schritte der BICsuite Server-Installation durchgeführt werden.

Mit der aktuellen BICsuiteRelease 2.10 wird auch Zope4+ basierend auf python 3 unterstützt. Die bisherige verwendete Zope2 Umgebung wird parallel dazu ebenfalls weiter unterstützt, ist jedoch deprecated und wird voraussichtlich ab Release 2.11 nicht mehr unterstützt.

Installation (Zope2)

1. Herunterladen und Installieren von Python 2.7 von www.python.org

Das Default-Verzeichnis in dem Python installiert wird ist `C:\Python27`. Im Beispiel verwenden wir als Installationsverzeichnis `C:\Programme\Python27`. Wir verweisen im Folgenden auf dieses Verzeichnis als `PYTHONDIR`.

2. Herunterladen und Installieren Python setuptools

Achtung: Auf korrekte Version für Python 2.7 achten !

3. Optionales Herunterladen und Installieren von pywin32

Achtung: Auf korrekte Version für Python 2.7 achten !

Dieser Schritt ist nur notwendig wenn der BICsuite!Web Zope-Server als Service gestartet werden soll.

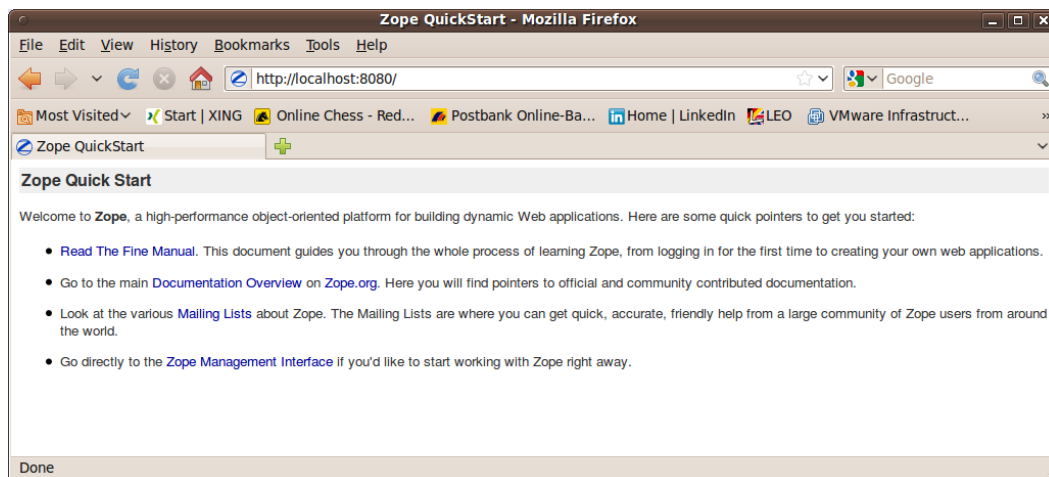


Abbildung 3.1: Zope Quicks Start Seite

4. Installieren der Zope2 Software

Öffnen Sie eine Windows-Eingabeaufforderung. Wechseln Sie in die virtuelle Python-Umgebung für die Zope-Installation und installieren Sie Zope2.

Installieren Sie die neueste Release von Zope2. Welche dies ist, kann mit einem Browser auf <http://download.zope.org/Zope2/index> ermittelt werden.

Zum Zeitpunkt der Erstellung dieses Dokumentes war die Release 2.13.29 die Aktuellste.

Beispiel:

```
C:> cd /d C:\Programme\Python27
C:> Scripts\pip install -r https://raw.githubusercontent.com/zopefoundation/Zope/2.13.
```

Anmerkung:

In manchen Varianten des Windows Betriebssystems, lieferte obiges Kommando beim ersten Aufruf einen Fehler (rmdir Verzeichnis nicht leer) und musste wiederholt werden.

5. Erzeugen einer Zope-Instanz für BICsuite!Web

Öffnen Sie eine Windows-Eingabeaufforderung. Wechseln Sie in die virtuelle Python-Umgebung für die Zope-Installation und erzeugen Sie die Zope-Instanz bicsuiteweb. Das hier benutzte Passwort ist frei wählbar, aber der Benutzer muss sdmsadm heißen.

Beispiel:

```
C:> cd /d C:\Programme\Python27
C:> Scripts\mkzopeinstance -d ..\bicsuiteweb -u sdmsadm:sdmsadm_passwort
```

Achtung:

Falls eine Oracle-Installation auf diesem Rechner vorhanden ist, wird der Zope Standard Port 8080 möglicherweise bereits benutzt. In diesem Fall muss in der Datei

```
bicsuiteweb\etc\zope.conf
```

der Port geändert werden. Im Folgenden wird davon ausgegangen, dass es keine Konflikte bei der Benutzung von Ports gibt und Zope auf Port 8080 erreichbar ist.

Test:

```
C:> cd /d C:\Programme\bicsuite
C:> bicsuiteweb\bin\runzope
```

Im Browser sollte die URL <http://localhost:8080> nun die Zope Quick Startseite, wie im Bild 3.1, sichtbar sein.

Zope kann nun mit Strg-C oder durch Schließen der Windows-Eingabeaufforderung wieder beendet werden.

6. Installieren der BICsuite!Web Komponenten

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

```
C:> cd /d C:\Programme\bicsuite\bicsuiteweb
C:> mkdir Extensions
C:> cd Extensions
```

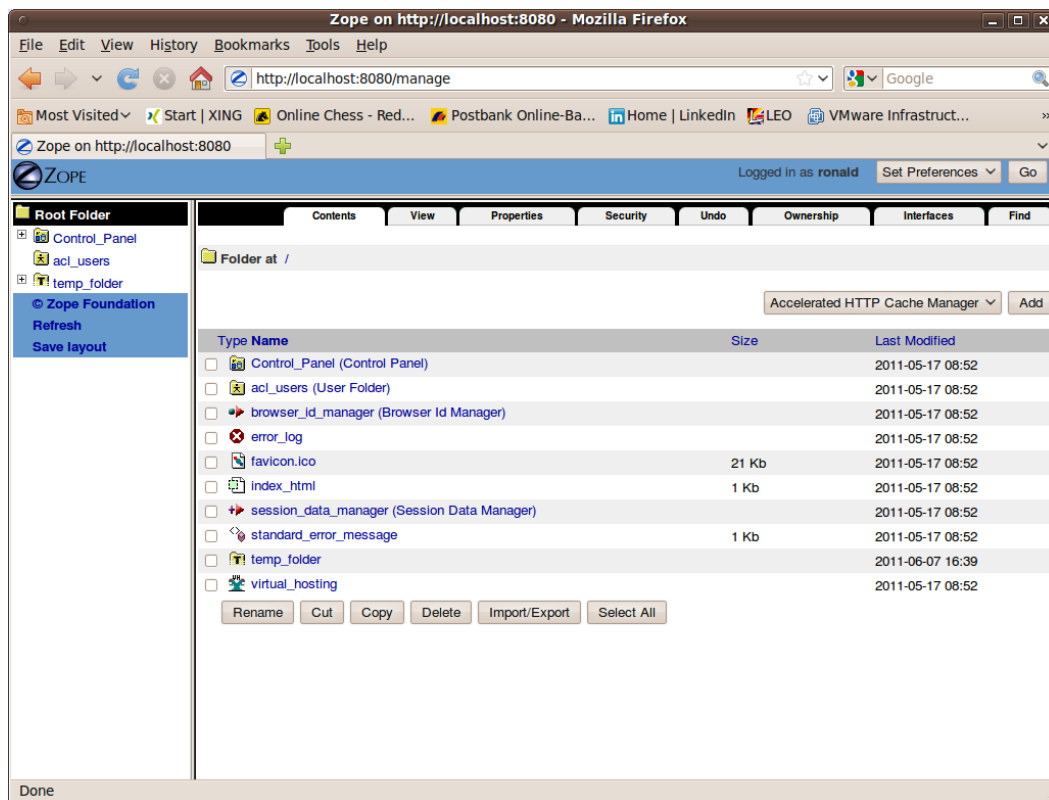


Abbildung 3.2: Zope Management Oberfläche

```
C:> copy "%BICSUITEHOME%\zope\*.py" .
C:> cd ..\Products
C:> mkdir BICsuiteSubmitMemory
C:> cd BICsuiteSubmitMemory
C:> copy "%BICSUITEHOME%\zope\BICsuiteSubmitMemory\*.*" .
C:> cd ..\..\import
C:> copy "%BICSUITEHOME%\zope\SDMS.zexp" .
```

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

Öffnen Sie eine Windows-Eingabeaufforderung.

```
C:> cd /d C:\Programme\bicsuite\bicsuiteweb\bin
C:> runzope
```

Die Zope Management-Oberfläche, wie im Bild 3.2, wird nun unter der Adresse

<http://localhost:8080/manage>

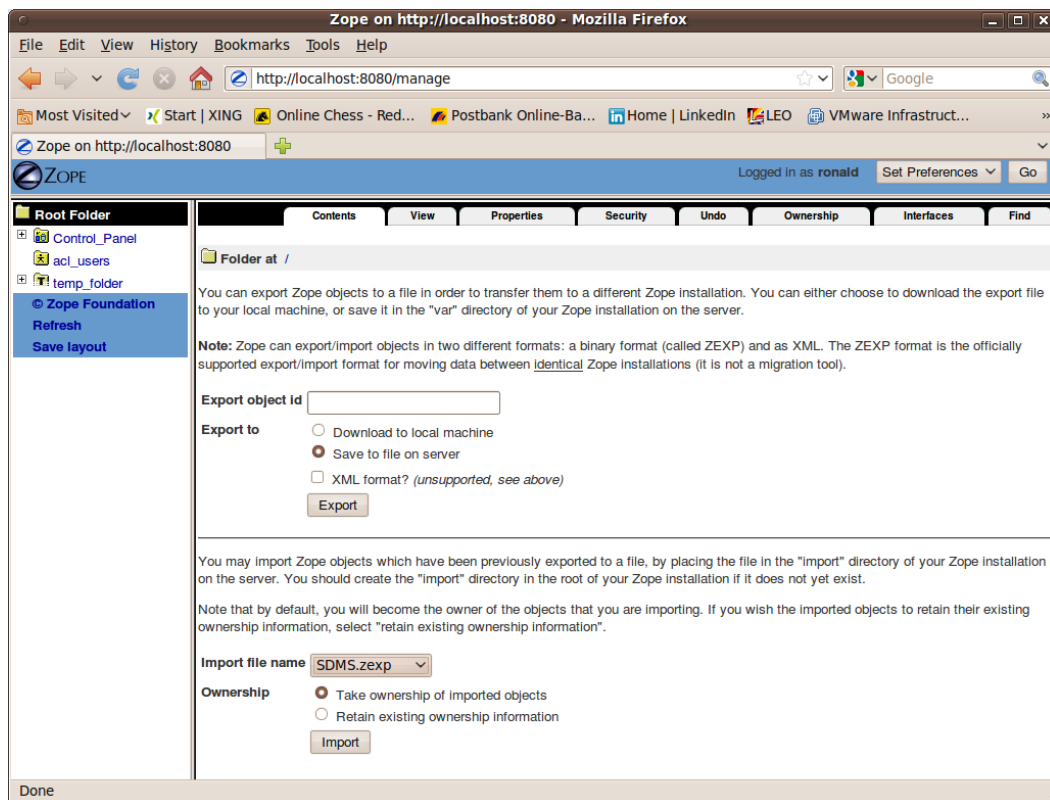


Abbildung 3.3: Zope Import Dialog

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, siehe dazu auch Bild 3.3):

- a) im Folder `/` `SDMS.zexp` importieren
- b) im Folder `/SDMS/Install` die Folder `User` und `Custom` anwählen und mit `Copy` kopieren
- c) im Folder `/` mit `Paste` die Folder `User` und `Custom` erzeugen

Wenn nun alles fehlerfrei durchgeführt werden konnte, sieht die Oberfläche wie auf Bild 3.4 aus.

7. Serververbindungen konfigurieren

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

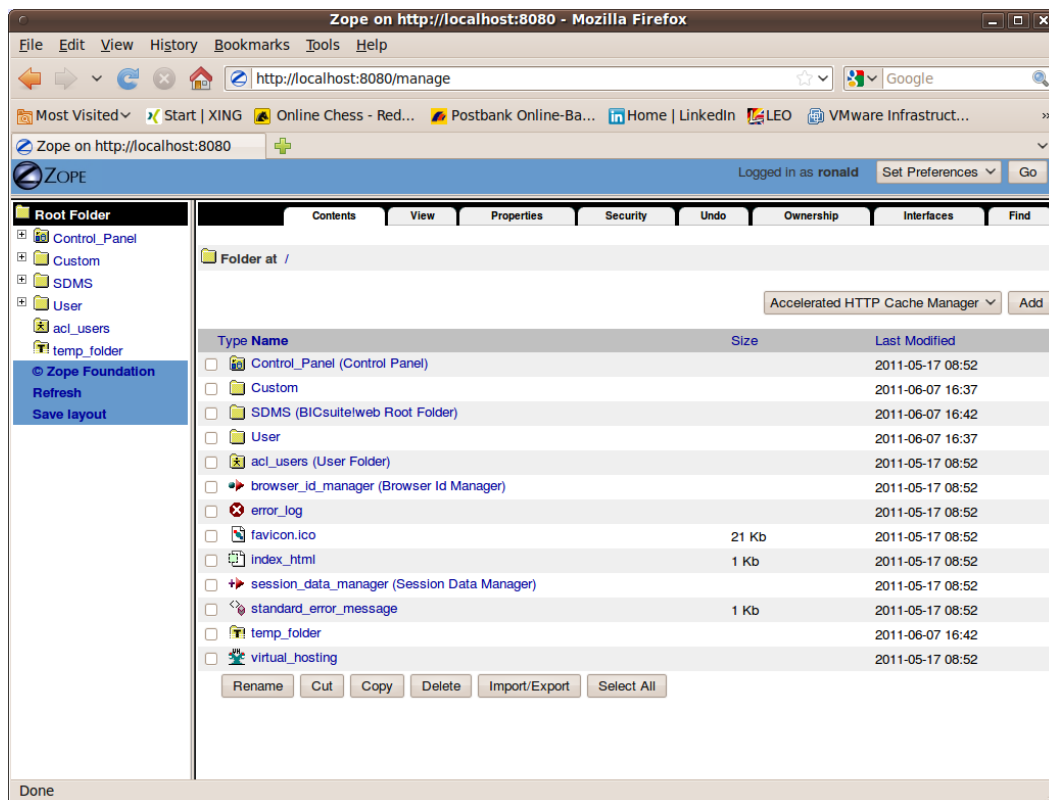


Abbildung 3.4: Zope Resultat Ansicht

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

```
# Servername unter dem der Server in der BICSuite!Web Oberflaeche
# sichtbar ist
'servername' : {
    # IP Adresse oder Hostname auf dem der BICSuite Server laeuft
    'HOST'      : 'hostname',

    # Port unter dem der BICSuite Server angesprochen wird
    'PORT'      : '2506',

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

    # optionales Property, ob Zope Serververbindungen cachen soll
    'CACHE'     : 'Y'

    # optionales Property, wie lange gecachete BICSuite!Web
```

```

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

```

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

Soll ein Server über eine sichere SSL-Verbindung angesprochen werden, müssen folgende weitere Eigenschaften definiert werden:

```

# Verbindung wird ueber Secure Socket Layer aufgebaut
'SSL' : 'true',

# falls angegeben, wird die Identitaet des BICsuite Server ueberprueft
# die angegebene Datei muss das Server Certificate des BICsuite Server
# enthalten
'TRUSTSTORE' : 'truststore.pem',

# falls der BICsuite Server eine Client Authentication fordert,
# muss dieses Property definiert sein und die angegebene Datei
# muss das Certificate und den Private Key des Clients enthalten.
# Das Certificate muss dem Server in seinem Truststore bekannt sein.
'KEYSTORE' : 'keystore.pem'

```

Anmerkung:

Bei Verwendung von SSL, wird aus Performancegründen die Verwendung von cached Serververbindungen empfohlen, da der Aufbau einer gesicherten Verbindung eine rechenintensive Operation ist.

8. Die BICsuite!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 dazugehörige Dokumentation verwiesen.

9. Automatischer Start von BICsuite!Web

Um BICsuite!Web automatisch nach der Anmeldung zur Verfügung zu haben, erzeugen Sie eine Verknüpfung von

```
C:\Programme\bicsuite\bicsuiteweb\bin\runzope.bat
```

nach Ihrem Autostart Folder.

Installation (Zope4+)

1. Herunterladen und Installieren von Python 3 von www.python.org

python3 in der zum Zeitpunkt der Erstellung dieser Dokumentation aktuellen Version kann über die URL:

```
https://www.python.org/ftp/python/3.9.2/python-3.9.2-amd64.exe
```

heruntergeladen werden. Im Beispiel verwenden wir als Installationsverzeichnis `C:\Programme\Python3`. Wir verweisen im Folgenden auf dieses Verzeichnis als `PYTHONDIR`.

2. Herunterladen und Installieren der Microsoft Visual C++ Build Tools

Die für die Installation notwendigen Microsoft Visual C++ Build Tools können über die URL

```
https://visualstudio.microsoft.com/de/thank-you-downloading-visual-studio/?sku=BuildTools&rel=16
```

heruntergeladen werden. Führen Sie das Executable aus, um die Build Tools zu installieren.

3. Erzeugen einer virtualenv Umgebung

Für diesen und die folgenden Installationsschritte öffnen Sie eine Windows Eingabeaufforderung (cmd) mit Administratorrechten. Die Zope5 Installation wird in dieser Installationsanweisung unter `C:\Users\BICsuite\Zope5` installiert. Es kann auch ein anderes Verzeichnis gewählt werden. Wir verweisen auf dieses Verzeichnis im folgenden als `ZOPE5DIR`,

```
C:> set INSTALLDIR=C:\Users\BICsuite
C:> set ZOPE5ENV=Zope5
C:> set ZOPE5DIR=%INSTALLDIR%\%ZOPE5ENV%
C:> cd %INSTALLDIR%
C:> python -m venv %ZOPE5ENV%
```

4. Installieren der Zope5 Software

Öffnen Sie eine Windows Eingabeaufforderung (cmd) mit Administratorrechten, falls noch nicht geschehen. Wechseln Sie in die virtuelle Python-Umgebung für die Zope-Installation und installieren Sie Zope5. Zum Zeitpunkt der Erstellung dieser Dokumentation war Zope 5.1.2 die aktuelle Version.

```
C:> set INSTALLDIR=C:\Users\BICsuite
C:> set ZOPE5ENV=Zope5
C:> set ZOPE5DIR=%INSTALLDIR%\%ZOPE5ENV%
C:> cd %ZOPE5DIR%
C:> Scripts\pip install -U pip wheel
C:> Scripts\pip install Zope[wsgi]==5.0 ^
-c https://zopefoundation.github.io/Zope/releases/5.1.2/constraints.txt
C:> Scripts\pip install Products.ExternalMethod
C:> Scripts\pip install Products.Sessions
```

```
C:> Scripts\pip install Products.SiteErrorLog
C:> Scripts\pip install Products.PythonScripts
```

5. Erzeugen einer Zope-Instanz für BICsuite!Web

Wir erzeugen in dieser Installationsanweisung die Zope5 unter:

```
C:\Users\BICsuite\Zope5Instance
```

Es kann auch ein anderes Verzeichnis gewählt werden. Wir verweisen auf dieses Verzeichnis im folgenden als ZOPE5INSTANCE.

```
C:> set INSTALLDIR=C:\Users\BICsuite
C:> set ZOPE5ENV=Zope5
C:> set ZOPE5DIR=%INSTALLDIR%\%ZOPE5ENV%
C:> set ZOPE5INSTANCE=%INSTALLDIR%\Zope5Instance
C:> cd %INSTALLDIR%
C:> %ZOPE5ENV%\Scripts\mkwsgiinstance -d %ZOPE5INSTANCE% ^
-u sdmsadm:sdmsadm_passwort
```

Test:

Starten Sie die Zope5 Instanz mit folgendem Befehl:

```
C> %ZOPE5DIR%\Scripts\runwsgi -v %ZOPE5INSTANCE%\etc\zope.ini
```

Im Browser sollte die URL <http://localhost:8080> nun eine 'Zope Auto-generated default page' zeigen.

Zope5 kann nun mit Strg-C oder durch Schließen der Windows-Eingabeaufforderung wieder beendet werden.

Anmerkung: In der hier verwendeten Zope5 Release wurde die zope.ini Datei fehlerhaft erstellt. Dies lässt sich Editieren der zope.ini Datei jedoch leicht beheben. Ersetzen Sie alle '\\' in Pfadnamen durch '/ '.

6. Installieren der BICsuite!Web Komponenten

Um die BICsuite!Web Komponenten zu installieren, muss die Zope5 Instanz um einige Komponenten erweitert werden. Wir gehen in dieser installationsanweisung davon aus, dass die Umgebungsvariable BICSUITEHOME korrekt gesetzt ist.

```
C:> set INSTALLDIR=C:\Users\BICsuite
C:> set ZOPE5ENV=Zope5
C:> set ZOPE5DIR=%INSTALLDIR%\%ZOPE5ENV%
C:> set ZOPE5INSTANCE=%INSTALLDIR%\Zope5Instance
C:> cd %INSTALLDIR%
C:> xcopy %BICSUITEHOME%\zope4\Extensions %ZOPE5INSTANCE%\Extensions\
C:> xcopy %BICSUITEHOME%\zope4\Products\BICsuiteSubmitMemory ^
%ZOPE5DIR%\Lib\site-packages\Products\BICsuiteSubmitMemory\
C:> xcopy %BICSUITEHOME%\zope4\Products\StringFixer ^
%ZOPE5DIR%\Lib\site-packages\Products\StringFixer\
C:> mkdir %ZOPE5INSRTANCE%\import
C:> copy %BICSUITEHOME%\zope4\import\SDMS.zexp %ZOPE5INSRTANCE%\import
```

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

```
C> %ZOPE5DIR%\Scripts\runwsgi -v %ZOPEINSTANCE%\etc\zope.ini
```

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):

- a) im Folder / SDMS.zexp importieren
- b) im Folder /SDMS/Install die Folder User und Custom anwählen und mit Copy kopieren
- c) im Folder / mit Paste die Folder User und Custom erzeugen

7. 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 BICsuite Server, der von dieser BICsuite!Web Installation angesprochen werden soll, einen Eintrag der Form:

```
# Servername unter dem der Server in der BICsuite!Web Oberfläche
# sichtbar ist
'servername' : {
    # IP Adresse oder Hostname auf dem der BICsuite Server laeuft
    'HOST'      : 'hostname',

    # Port unter dem der BICsuite Server angesprochen wird
    'PORT'      : '2506',

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

    # optionales Property, ob Zope Serververbindungen cachen soll
    'CACHE'     : 'Y'

    # optionales Property, wie lange gecachte BICsuite!Web
    # Serververbindungen gueltig sein sollen
    # default ist 60 sekunden, nur von Bedeutung falls 'CACHE' : 'Y'
    'TIMEOUT'   : '60'
}
```

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

Soll ein Server über eine sichere SSL-Verbindung angesprochen werden, müssen folgende weitere Eigenschaften definiert werden:

```
# Verbindung wird ueber Secure Socket Layer aufgebaut
'SSL' : 'true',

# falls angegeben, wird die Identitaet des BICsuite Server ueberprueft
# die angegebene Datei muss das Server Certificate des BICsuite Server
# enthalten
'TRUSTSTORE' : 'truststore.pem',

# falls der BICsuite Server eine Client Authentication fordert,
# muss dieses Property definiert sein und die angegebene Datei
# muss das Certificate und den Private Key des Clients enthalten.
# Das Certificate muss dem Server in seinem Truststore bekannt sein.
'KEYSTORE' : 'keystore.pem'
```

Anmerkung:

Bei Verwendung von SSL, wird aus Performancegründen die Verwendung von cached Serververbindungen empfohlen, da der Aufbau einer gesicherten Verbindung eine rechenintensive Operation ist.

8. Die BICsuite!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 dazugehörige Dokumentation verwiesen.

9. Zope5 als Windows Service installieren und starten

Um den Zope5 BICsuiteWeb Application Server als Windows Service zu installieren und zu starten führen Sie folgendes Kommandos in einer als Administrator geöffneten Eingabeaufforderung aus (Es wird davon ausgegangen, dass die BICSUITEHOME Umgebungsvariable korrekt gesetzt ist:

```
C:> set INSTALLDIR=C:\Users\BICsuite
C:> set ZOPE5VENV=Zope5
C:> set ZOPE5DIR=%INSTALLDIR%\%ZOPE5VENV%
C:> set ZOPE5INSTANCE=%INSTALLDIR%\Zope5Instance
C:> sc create Zope5 ^
start= auto ^
binPath= "%BICSUITEHOME%\bin\scrolllog -c Zope5 -w %ZOPE5INSTANCE% log\logfile
-e %ZOPE5DIR%\Scripts\runwsgi -v %ZOPE5INSTANCE%\etc\zope.ini" ^
displayName= "Zope5 BICsuite!Web Application Server"
C:> sc start Zope5
```

Anmerkung: Die Zeile mit der Option `binPath` wurde dem Layout geschuldet umgebrochen. Dieser Umbruch darf natürlich bei der Ausführung der Kommandos nicht stattfinden.