

independIT Integrative Technologies GmbH  
Bergstraße 6  
D-86529 Schrobenhausen



**BICsuite**

**Installation Guide  
Release 2.6.1**

Dieter Stubler

Ronald Jeninga

March 3, 2015

Copyright © 2015 independIT GmbH

**Legal notice**

This work is copyright protected

Copyright © 2015 independIT Integrative Technologies GmbH

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

# Contents

|  |           |
|--|-----------|
| <b>Table of contents</b>                           | <b>1</b>  |
| <b>1 Requirements</b>                              | <b>1</b>  |
| BICsuite Server and Clients . . . . .              | 1         |
| Zope Application Server . . . . .                  | 2         |
| Zope https . . . . .                               | 2         |
| <b>2 Installation in a Linux environment</b>       | <b>3</b>  |
| Installing the BICsuite Server . . . . .           | 3         |
| Installing a BICsuite Client . . . . .             | 6         |
| Jobserver Sample Installation . . . . .            | 8         |
| Scenario . . . . .                                 | 9         |
| Requirements . . . . .                             | 9         |
| Installation . . . . .                             | 9         |
| Installation with Postgres . . . . .               | 11        |
| Introduction . . . . .                             | 11        |
| Installation . . . . .                             | 11        |
| Installation with MySQL . . . . .                  | 13        |
| Introduction . . . . .                             | 13        |
| Installation . . . . .                             | 13        |
| Installation with Ingres . . . . .                 | 14        |
| Introduction . . . . .                             | 14        |
| Installation . . . . .                             | 14        |
| Installation with Oracle . . . . .                 | 16        |
| Introduction . . . . .                             | 16        |
| Installation . . . . .                             | 16        |
| Configuration of the TLS/SSL connections . . . . . | 17        |
| Introduction . . . . .                             | 17        |
| TLS/SSL configuration . . . . .                    | 18        |
| Installing the Zope server . . . . .               | 21        |
| Introduction . . . . .                             | 21        |
| Installation . . . . .                             | 22        |
| Installing the Https extensions . . . . .          | 26        |
| <b>3 Installation in a Windows environment</b>     | <b>31</b> |
| Installing the BICsuite Server . . . . .           | 31        |

|  |    |
|--|----|
| Introduction . . . . .                             | 31 |
| Installation . . . . .                             | 31 |
| Installation with Postgres . . . . .               | 35 |
| Introduction . . . . .                             | 35 |
| Installation . . . . .                             | 35 |
| Installation with MySQL . . . . .                  | 37 |
| Introduction . . . . .                             | 37 |
| Installation . . . . .                             | 38 |
| Installation with Microsoft™ SQL Server . . . . .  | 40 |
| Introduction . . . . .                             | 40 |
| Installation . . . . .                             | 40 |
| Installation with Ingres . . . . .                 | 42 |
| Introduction . . . . .                             | 42 |
| Installation . . . . .                             | 42 |
| Installation with Oracle Express Edition . . . . . | 44 |
| Introduction . . . . .                             | 44 |
| Installation . . . . .                             | 44 |
| Installing the Zope server . . . . .               | 46 |
| Introduction . . . . .                             | 46 |
| Installation . . . . .                             | 46 |

# 1 Requirements

## BICsuite Server and Clients

The following software is required to install the BICsuite Server:

- Oracle(Sun) Java 1.7 SE jre  
<http://www.oracle.com/technetwork/java/index.html>
- Microsoft Visual C++ 2010 Redistributable package (x86)  
(only for an installation under Windows)  
<http://www.microsoft.com>
- One of the following database systems:
  - PostgreSQL  
<http://www.postgresql.org>  
JDBC for PostgreSQL:  
<http://jdbc.postgresql.org>
  - MySQL  
<http://www.mysql.com>  
MySQL (Connector/J) JDBC Driver  
<http://www.mysql.com>
  - Ingres  
<http://www.ingres.com>
  - Oracle  
<http://www.oracle.com>
  - Microsoft SQL Server  
<http://www.microsoft.com>  
Microsoft SQL Server JDBC Driver  
<http://www.microsoft.com>
- Java Native Access (JNA)  
In order to avoid the use of a JNI library we use the JNA library from version 2.6 and later. This library is only required for the Java Jobexecutor.  
<https://github.com/twall/jna>

## Zope Application Server

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

- Python 2.7  
<http://www.python.org>
- python-virtualenv  
<http://pypi.python.org>
- python-dev  
<http://www.python.org>
- python setuptools (Windows only)  
<http://pypi.python.org/pypi/setuptools>
- pywin32 (optional, Windows only)  
<http://sourceforge.net/projects/pywin32>

## Zope https

The following software is also required for using https in combination with Zope:

- SWIG  
<http://www.swig.org>
- python M2Crypto package  
<http://pypi.python.org>
- OpenSSL  
[www.openssl.org](http://www.openssl.org)  
(Or comparable)

## 2 Installation in a Linux environment

### Installing the BICsuite Server

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

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

1. Create the user `bicsuite`

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

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

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

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

2. Download and install a database management system supported by BICsuite. BICsuite for Linux currently supports the following systems:

- Postgres (page [11](#))
- MySQL (page [13](#))
- Ingres (page [14](#))
- Oracle (page [16](#))

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

3. Unpack the software

Unpack the tar archive in the `bicsuite` home directory. For instance:

## Installing the BICsuite Server

```
$ tar xvzf bicsuite-2.6.1.tgz
```

Create a symbolic link:

```
$ ln -s bicsuite-2.6.1 bicsuite
```

### 4. Create the configuration

#### a) User environment

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

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

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

#### b) Software environment

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

For instance:

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

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

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



## Installing the BICsuite Server

The file `java.conf` describes the Java environment that is to be used. In particular, the path to the JDBC driver must be entered. The memory configuration of the server is also regulated here. Even in large-scale environments it is usually only necessary to adapt the variable `BICSUITEMEM`.

The file `server.conf` contains the server configuration. The settings for connecting the BICsuite Scheduling Server to its RDBMS repository have to be modified. More details about this can be found in the respective chapter on the RDBMS that is being used.

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

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

### 5. Set up the database

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

For

- Ingres, see page 14,
- MySQL, see page 13,
- Oracle, see page 16, and for
- PostgreSQL, see page 11.

### 6. Start the server

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

The server can be started with

```
$ server-start
```

### 7. Create the file `.sdmshrc`

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

Here is an example of the contents:

```
$ cat ~/.sdmshrc
User=SYSTEM
Password=GOHOME
```

## Installing a BICsuite Client

```
Host=localhost
Port=2506
Timeout=0
```

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

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

### 8. Install the convenience package

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

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

### 9. Install the examples (optionally)

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

#### a) Create the job servers

It just takes one script to create the job servers:

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

#### b) Load the workflow definitions

The following commands are entered to load the workflow definitions:

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

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

## Installing a BICsuite Client

The installation of a BICsuite scheduling client is easy and requires only a few simple actions which are explained in the following.

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

## Installing a BICsuite Client

### 1. Creating the user `bicsuite`

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

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

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

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

### 2. Unpack the software

Unpack the tar archive in the `bicsuite` home directory. For instance:

```
$ tar xvzf bicsuite-2.6.1.tgz
```

Create a symbolic link:

```
$ ln -s bicsuite-2.6.1 bicsuite
```

### 3. Create the configuration

#### a) User environment

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

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

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

## Jobserver Sample Installation

### b) Software environment

Several templates for the configuration files that should be used as a basis for the system configuration can be found under `$BICSUITEHOME/etc`. For a client installation we need the files `bicsuite.conf` and `java.conf`. These have to be copied without the `".template"` extension to the directory `$BICSUITECONFIG`.

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

The file `bicsuite.conf` configures some default settings and does not usually have to be modified.

The file `java.conf` describes the Java environment that is to be used and usually does not require any changes.

### 4. Create the file `.sdmshrc`

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

Here is an example of the contents:

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

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

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

## Jobserver Sample Installation

In the following a sample installation of a BICsuite jobserver will be shown.

### Scenario

On the machine `machine_42` a jobserver shall execute processes as user `arthur`. The HOME directory of the user is `/home/arthur`. The BICsuite server is installed on the machine `scheduling_server` and is listening on port 2506. The system password is `G0H0ME`.

### Requirements

On the machine `machine_42`, a BICsuite client was installed in the HOME directory `/home/bicsuite`.

The user `arthur` has the following access privileges on the client installation files:

- Read access on the files `java.conf` and `bicsuite.conf` in `$BICSUITECONFIG` and all files under `/home/bicsuite/lib`
- Read and execute privileges on all files in `/home/bicsuite/bin`

### Installation

To enable a jobserver to connect to the BICsuite scheduling server, the jobserver has to be configured within BICsuite scheduling server. In the following we execute the necessary steps using the `sdmsh` command line utility.

Alternatively this could also be done using the Web GUI.

1. Login as user `arthur` into the machine `machine_42`
2. Setting of the shell environment variables (`.bashrc`).

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

3. Testing the environment

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

A `SDMS>` prompt should appear (use `'exit'` to close `sdmsh`).

4. Create directories

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

## Jobserver Sample Installation

### 5. Create a Scope for machine `machine_42` using `sdmsh`

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

All directory paths must be full qualified. The use of shell environment variables is not supported.

### 6. Create a job server using `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'  
);
```

The `HTTPPORT` is necessary for the display of job logs and can be selected freely but must be unique for all jobserver on the same machine. All directory paths must be full qualified. The use of shell environment variables is not supported.

### 7. Create the Named Resource for the jobserver using `sdmsh`

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

### 8. Create an Environment for the job server using `sdmsh`

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

### 9. Create the Resource in the job server using `sdmsh`

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

## Installation with Postgres

10. Create the configuration file `$HOME/etc/jobserver.conf` for the jobserver containing the following:

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

11. Start the jobserver

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

If the jobserver should start automatically at boot time, the system administrator has to configure this accordingly.

## Installation with Postgres

### Introduction

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

### Installation

1. Download and install the latest version of Postgres

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

2. Configure the file `pg_hba.conf`

To enable the BICsuite Scheduling Server to identify itself to PostgreSQL with a user and password, the following line has to be added to the Postgres configuration file `pg_hba.conf`. This file is usually found under

```
/var/lib/pgsql/<version>/data:
```

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

PostgreSQL then has to be restarted.

3. Create the Postgres user `bicsuite`

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

## Installation with Postgres

```
$ 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
```

or, in the case of version 9:

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

The entered password will be required again later.

### 4. Create the repository database

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

```
$ createdb bicsuitedb
```

### 5. Create and initialise the database tables

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

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

### 6. Configure the database connection in the BICsuite Server configuration file

`$BICSUITECONFIG/server.conf`

Change the following properties as shown here:

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

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

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

### 7. Configure the BICsuite Java Class Path for the Postgres JDBC

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

For instance:

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



## Installation with MySQL

### Introduction

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

### Installation

1. Download and install the latest version of MySQL.

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

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

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

2. Create the MySQL user `bicsuite` and the database `bicsuitedb`

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

```
$ 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_password';
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. Create and initialise the database tables

## Installation with Ingres

Run the following commands:

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

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

Change the following properties as shown here:

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

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

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

For instance

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

## Installation with Ingres

### Introduction

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

### Installation

1. Install Ingres

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

2. Create the user `bicsuite`

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

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

## Installation with Ingres

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

continue
* create user 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. Create the repository database 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. Create and initialise the database tables

Run the following commands to create the required tables:

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

### 5. Configure the database connection in the BICsuite Server configuration file

\$BICSUITECONFIG/server.conf

Change the following properties as shown here:

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

## Installation with Oracle

### 6. Configure the BICsuite Java Class Path for the Ingres JDBC

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

For instance:

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

## Installation with Oracle

### Introduction

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

### Installation

1. Download and install the latest version of Oracle Express Edition.  
Remember your Oracle system password.

2. Create the Oracle user bicsuite

```
$ sqlplus SYSTEM/oracle_system_password
```

Create the user bicsuite with:

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

Assign the required access privileges:

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

3. Create and initialise the database tables

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

4. Configure the database connection in the BICsuite Server configuration file  
Edit `$BICSUITECONFIG/server.conf` and modify the following properties:

## Configuration of the TLS/SSL connections

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

Here it is also necessary to change the `Hostname` to the servers hostname or IP address.

### 5. Configure the BICsuite Java Class Path for the Oracle JDBC

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

For instance:

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

## Configuration of the TLS/SSL connections

### Introduction

Configuring encrypted communication within the BICsuite system is relatively easy. However, the required time and effort will vary depending upon your requirements.

The following options are available:

1. BICsuite without SSL/TLS communication
2. BICsuite with and without SSL/TLS communication
3. BICsuite exclusively with SSL/TLS communication

In turn, there are two methods of communicating with TLS/SSL:

1. Server-side authentication only This means that clients can verify whether the server with which they are communicating really is trustworthy. This setting just requires one key pair for the server. All communication is encrypted.
2. Server-side and client-side authentication With this configuration, both parties verify whether the identity of the communication partner is known to them. This setting is extremely secure but also more time-consuming since a key pair has to be generated for each client and obviously for the server as well. All communication is naturally encrypted.

## TLS/SSL configuration

Configuring the SSL/TLS communication essentially requires just a few steps.

### 1. Generate the key pairs for the server

The utility `keytool`, a component of Java (SE), is used to generate the key pairs. More information about this utility can be found at

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

The following procedure should work (for non-specialists):

a) Create a directory, e.g.

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

b) This directory will afterwards contain the private key for the server and should therefore be protected accordingly, e.g.

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

c) Generate a key pair using `keytool`, e.g.

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

**IMPORTANT:** Both passwords (`keypass` and `storepass`) *must* be identical!

d) The servers public key must later be given to the clients. To do this, it first has to be extracted from the key file.

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

The `storepass` (`secret`) is required here.

### 2. Modify the server configuration

The server configuration now has to be modified to notify the server about its keys. You also have to decide whether you want to use just server-side authentication or authentication by both the server and the client. A decision also has to be made with regard to non-encrypted communication.

```
#  
# SSLPort: port for encrypted communication  
# (inactive = 0)  
#  
SSLPort=2507  
#  
# KeyStore defines which key file is to be used  
# $BICSUITEHOME/etc/certs/avrkeystore  
#  
KeyStore=/home/bicsuite/etc/certs/avrkeystore
```

## Configuration of the TLS/SSL connections

```
#
# The KeyStorePassword is required to read out the key store
#
KeyStorePassword=secret
#
# TrustStore defines which file contains the (public) keys of the
# valid communication partners
#
TrustStore=/home/bicsuite/etc/certs/svrkeystore
#
# The TrustStorePassword is required to read out the trust store
#
TrustStorePassword=secret
#
# ClientAuthentication defines whether clients have to authenticate
# themselves (true) or not (false)
#
ClientAuthentication=true
#
# Port defines the port for non-encrypted communication
# (inactive = 0)
# If all ports are configured as being inactive,
# port 2506 is opened for non-encrypted communication
#
Port=2506
#
# ServicePort defines the port for accessing the system in
# emergencies (inactive = 0)
#
ServicePort=2505
```

If `ClientAuthentication=true`, the server requires both the key store and the trust store. If client authentication is not necessary, only the key store is required.

### 3. Client configuration

If no client authentication is necessary, clients only need access to a trust store to be able to verify the servers identity. If the identity of the clients has to be verified as well, a key store will also need to be generated for each client. This is done analogue to the creation of the key store for the server.

### 4. Modify `.sdmshrc`

The use of an "ini" file is mandatory if `sdmsh` or the standard utilities are to communicate with the server via TLS/SSL. There are three possibilities here:

- a) `$BICSUITECONFIG/sdmshrc`
- b) `$HOME/.sdmshrc`
- c) A file specified when the utility is called

## Configuration of the TLS/SSL connections

The information required for the secure connection is the decisive factor here. For instance:

```
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
```

This permits symmetrical authentication. If only server-side authentication is necessary, the lines concerning the key store can be deleted.

### 5. Job server configuration

The job servers now have four new configuration parameters corresponding to the server parameters:

```
KEYSTORE
TRUSTSTORE
KEYSTOREPASSWORD
TRUSTSTOREPASSWORD
```

These are supplemented by another parameter which states whether encrypted communication is desired:

```
USE_SSL
```

If client authentication is to be used, a key pair has to be created for each job server. This is done in precisely the same way as has already been described for the BICsuite Server.

You must obviously make sure that the configuration parameter `RepoPort` in particular is set correctly.

### 6. Getting it all together

Last, but not least, the public keys now have to be swapped between the individual communication partners using `keytool`. To define the servers public key as being trustworthy, for instance, it is added to the client's trust store. For example:



## Installing the Zope server

```
$ 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
```

A storepass is naturally required to do this.

If client authentication is required as well, the client certificates will also need to be added to the servers trust store.

## Installing the Zope server

### Introduction

A Zope Application Server has to be installed before you can use the BICsuite!Web userinterface.

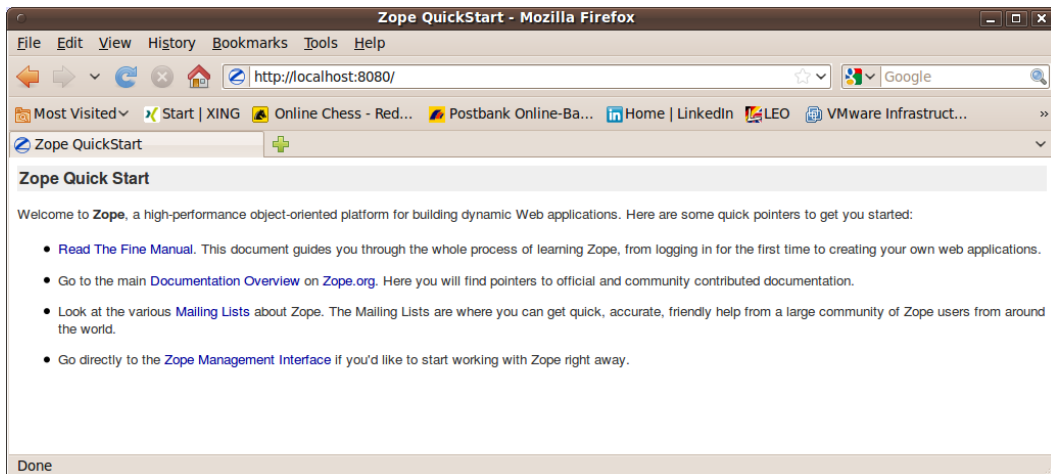


Figure 2.1: Zope Quick Start page

## Installing the Zope server

### Installation

#### 1. Install virtualenv

```
$ easy_install virtualenv
```

#### 2. Create a virtual Python environment for the Zope installation

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

#### 3. Install the Zope2 software

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

#### 4. Create a Zope instance for BICsuite!Web user interface

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

You can use any password of your choice. This will be required again later again. The username must be `sdmsadm` though.

The Zope server is started briefly for testing:

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

In the web browser, the URL

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

should now be displayed on the Zope QuickStart page as in Figure 2.1.

The Zope instance is now stopped again:

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

#### 5. Install the BICsuite!Web components

The Zope installation has to be extended with several modules before the BICsuite!Web components can be installed.

## Installing the Zope server

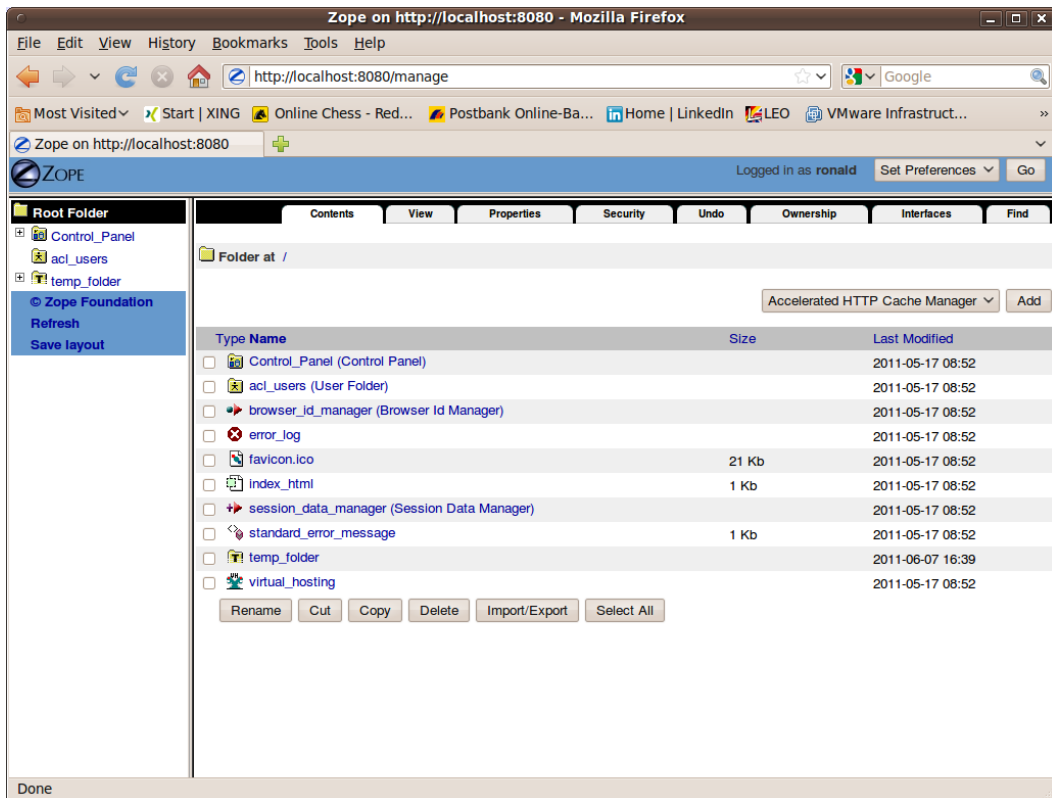


Figure 2.2: Zope Management screen

```
$ 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 .
```

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

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

The Zope Management user interface is now opened at the address

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

## Installing the Zope server

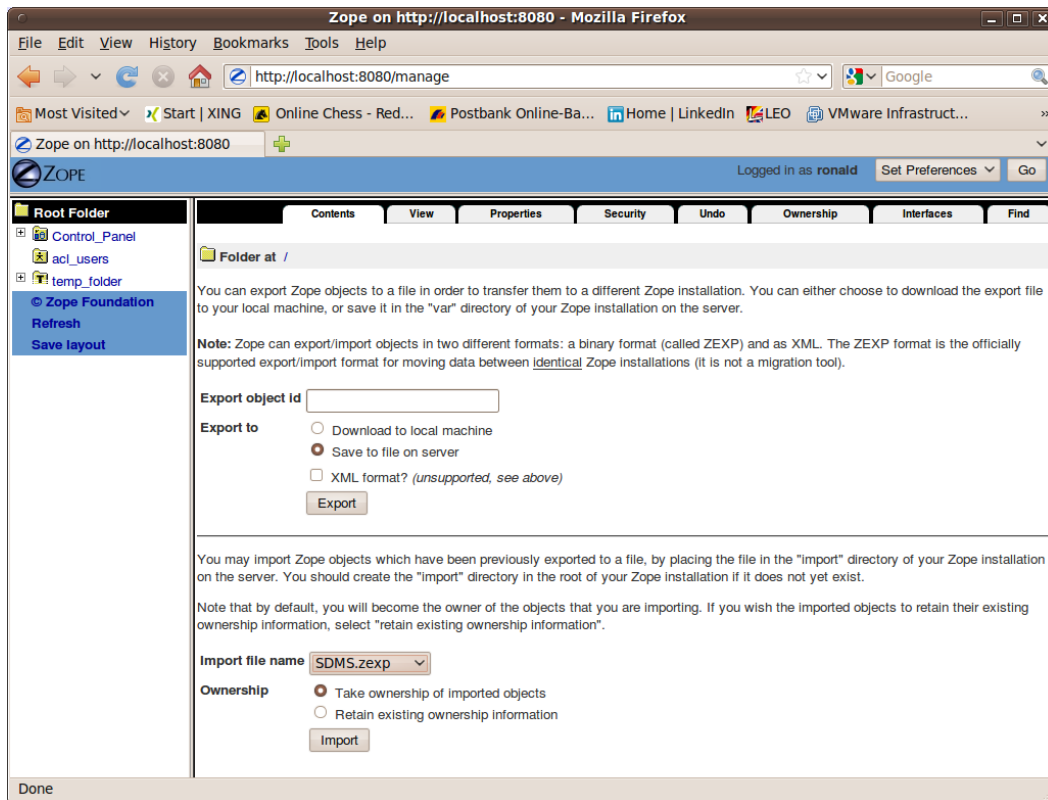


Figure 2.3: Zope Import dialogue

in a browser (see also Figure 2.2). This is done with the user `sdmsadm` together with the password you have assigned.

The front end software is now loaded into Zope (Import button, see Figure 2.3):

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

If everything has been carried out without any errors, the interface will look like the screenshot in Figure 2.4.

### 6. Configure the server connections

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

The `SDMSServers` Python script is edited in the folder `Custom`. This script delivers a dictionary which has to contain an entry for every BICsuite Server

## Installing the Zope server

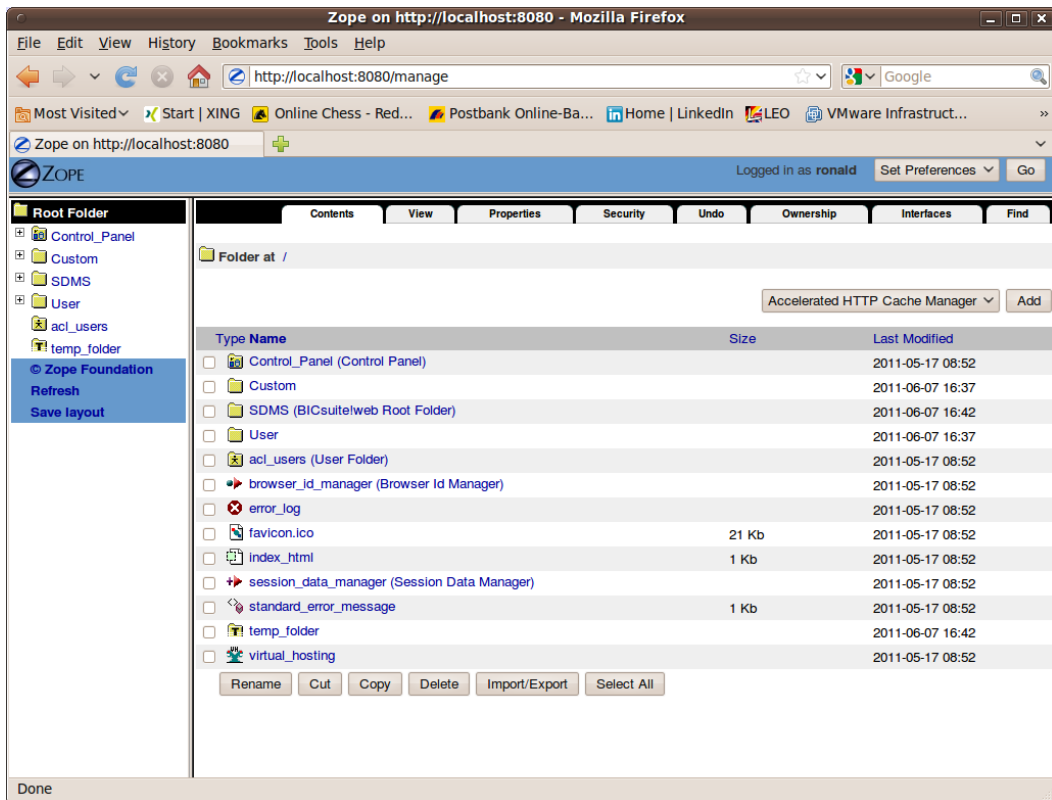


Figure 2.4: Zope result window

that is to be addressed by this BICsuite!Web installation. The entry looks like this:

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

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

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

    # Optional property stating whether BICsuite!Web should
    # cache the server connections
    'CACHE'     : 'Y'

    # Optional property stating for how long cached BICsuite!Web
```

## Installing the Zope server

```
# server connections should continue to be valid
# Default setting is 60 seconds, only significant if 'CACHE' : 'Y'
'TIMEOUT' : '60'
}
```

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

If a server is to be addressed via a secure SSL connection, the following additional properties have to be defined as well:

```
# Connection is set up via Secure Socket Layer
'SSL' : 'true',

# If stated, the identity of the BICsuite! Server is verified
# The stated file must contain the BICsuite!Server server certificate
'TRUSTSTORE' : 'truststore.pem',

# If the BICsuite!Server requires client authentication,
# this property must be defined and the stated file
# must contain the client's certificate and private key.
# The certificate must be known to the server in its trust store.
'KEYSTORE' : 'keystore.pem'
```

Note:

When using SSL, for performance reasons it is advisable to use cached server connections because setting up a secure connection is a resource-intensive operation.

### 7. Open the BICsuite!Web interface

The user interface is now available at the address

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

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

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

## Installing the Https extensions

Installing the Zope https extension will take some time. It is not particularly complicated, but it is important to carefully and precisely follow the instructions and ideally to understand them as well.

## Installing the Zope server

### 1. Download the M2Crypto module

The M2Crypto module can currently be found at

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

Unpack it in `$HOME`

### 2. Install M2Crypto in your virtual environment

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

To test whether the installation was completed successfully, you can simply try to load the module in Python:

```
$ $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
>>>
```

If this triggers an error message, an `ldconfig` may be required or you may have to close and reopen the terminal.

### 3. Patch the Zope installation

The next step ensures that, depending upon the configuration, Zope will actually use the M2Crypto module that has now been installed.

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

### 4. Create the files required by Zope https

The files required by Zope can be saved to the "etc" directory of the Zope instance, although this is not mandatory. In this guide, it is assumed that this is indeed the case. If you want to adapt this procedure, it is assumed that you have a sound understanding of how both TLS/SSL and https function.

First of all, navigate to the "etc" directory:

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

An SSL Certificate Authority is now created:

## Installing the Zope server

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

The following parameters can be entered as an example:

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

You can now create an SSL server certificate:

- **Generate a key for the 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
```

- **Remove the password from serverkey.pem:**

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

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

- **Generate a Certificate Signing Request:**

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

The following parameters can be entered as an example:

```
Country Name (2 letter code) [AU]:DE  
State or Province Name (full name) [Some-State]:Bavaria  
Locality Name (eg, city) []:Schrobenhausen  
Organization Name (eg, company) [Internet Widgits Pty Ltd]: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 []:
```

*Important !!!*

`my_hostname` must be replaced with the name that is used by the browser to address the https host !!!



## Installing the Zope server

- Sign the certificate:
  - a) Save openssl.cnf (usually requires root privileges)

```
# cp /etc/ssl/openssl.cnf /etc/ssl/openssl.cnf_save
```
  - b) Edit openssl.cnf:

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

The following entries are modified:

```
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) Sign:

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

Example:

```
Enter pass phrase for ./cakey.pem: super_secret
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
```
  - e) Restore the old openssl.cnf

```
# mv /etc/ssl/openssl.cnf_save /etc/ssl/openssl.cnf
```
- Write the SSL server certificate:

```
$ cat servercert.pem serverkey.pem > ssl_server_cert.pem
```
- Create ssl entropy pool :

```
$ dd if=/dev/random of=ssl_entropy_pool.dat bs=1024 count=1
```
- Create ssl\_dh\_init:

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

## 5. Configure Zope https

If https is to listen at port 8085 and

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

the following should be added to the file `$HOME/bicsuiteweb/etc/zope.conf` after the closing tag `</http-server>`:

## Installing the Zope server

```
<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/ssl_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>
```

If communication should only take place via https, the section `<http-server>` to `</http-server>` is to be commented out.

### 6. Restart Zope

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

We use `runzope` so that we see any errors straight away:

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

`$HOME/bicsuiteweb/log/event.log` is also helpful for analysing problems.

### 7. Once it has functioned properly, cancel `runzope` (Ctrl+C) and restart Zope

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

To start BICsuite!web:

In the browser

`http://hostname:8080/SDMS`

or

`https://hostname:8085/SDMS`

The first time BICsuite!Web is started, the server certificate has to be confirmed as being trustworthy in the browser. This is all you need to do in Firefox. In Internet Explorer, however,

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

has to be imported under Internet Options → Content → Publishers... → Trusted Root Certification Authorities to prevent warnings from constantly being displayed.

# 3 Installation in a Windows environment

## Installing the BICsuite Server

### Introduction

This guide assumes that you have unpacked the BICsuite distribution to a local directory. We recommend using `C:\Program Files`. The unpacked distribution should now be in the directory `C:\Program Files\bicsuite`.

The command shell is required for some of the installation steps. Many of the instructions or examples in this guide assume that you are working in the command line window. In this case, in the examples `C:>` is shown as the prompt. This naturally does not have to be entered. Many operations can often also be (more easily) performed using the Windows interface. You can obviously choose the method that is easiest for you.

You should also note that for typesetting reasons some lines in the examples are shown with line breaks. This is evident where the line has an indent and there is no prompt at the beginning of the line. When copying the commands, you should *not* use a return at the end of the line.

### Installation

1. Create the installation directory

Create a folder for installing the BICsuite software. In the following we refer to this directory as `INSTALLDIR`.

Example:

```
C:> cd \Program Files
C:> mkdir bicsuite
```

2. Unpack the BICsuite distribution in `INSTALLDIR`

`INSTALLDIR` should now have a subdirectory called `bicsuite-release`.

Example:

```
C:\Program Files\bicsuite\bicsuite-2.6.1
```

3. Set the `BICSUITEHOME` environment variable

## Installing the BICsuite Server

Close all the Windows command line prompts. Under Control Panel → System → Advanced → Environment Variables, set the variable `BICSUITEHOME` as the system variable for the BICsuite distribution directory.

Example:

```
BICSUITEHOME=C:\Program Files\bicsuite\bicsuite-2.6.1
```

### 4. Set the optional environment variables `BICSUITECONFIG`

We advise you not to save the BICsuite configuration files under `%BICSUITEHOME%`, but to use a separate directory.

Close all the Windows command line prompts. Under Control Panel → System → Advanced → Environment Variables, set the variable `BICSUITECONFIG` as the system variable for the BICsuite configuration directory.

Example:

```
BICSUITECONFIG=C:\Program Files\bicsuite\etc
```

This will make it easier to later upgrade to a newer release of BICsuite.

### 5. Create the configuration folder

If the configuration files are not to be saved under `%BICSUITEHOME%`, you will have to create the configuration folder. For instance:

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

Configuration files are required that can be created from the provided templates regardless of whether a separate folder is to be used for the configuration or not. These will then be modified at a later stage of the installation.

Open a Windows command line window. If you have not set the environment variable `%BICSUITECONFIG%`, use `%BICSUITEHOME%\etc` instead.

Simply copying the Configuration template is adequate for the time being:

```
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 the file `BICSUITE_CONF.BAT`, the value of the variable `BICSUITELOGDIR` can be modified as required so that the system logging takes place outside of the installation directory. In this case, you will obviously have to make sure that this directory actually exists.

If the Java executable is not in the path given here, the variable `BICSUITEJAVA` has to be set to the qualified path in the file `JAVA JAVA_CONF.BAT`. For example:

## Installing the BICsuite Server

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

6. Download and install a database management system supported by BICsuite.

BICsuite for Windows currently supports the following systems:

- Postgres (page 35)
- MySQL (page 37)
- Microsoft™ SQL Server (page 40)
- Ingres (page 42)
- Oracle (page 44)

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

7. Modify the PATH environment variable

Close all the Windows command line prompts. Navigate to Control Panel → System → Advanced → Environment Variables and add the following string to the Path system variable:

```
%BICSUITEHOME%\bin
```

Example:

```
PATH= ... ;C:\Program Files\bicsuite\bicsuite-2.6.1\bin
```

8. An up-to-date Java Runtime Environment has to be installed if this has not already been done

A JRE can be downloaded from <http://www.oracle.com/technetwork/java/index.html> and installed.

9. Start BICsuite Server

Open a Windows command line window if you have not already done so.

Run the BICsuite command line utility RUN\_SERVER.

```
C:> RUN_SERVER
```

Leave this window open for the next steps.

10. Create the optional file `.sdmshrc` in the Windows user's home directory

When this file is created, it will be read by most of the BICsuite command line utilities to supplement any missing parameters.

This means that when you call `sdmsh`, for example, you won't have to enter the host port user and password each time.

Sample file:

## Installing the BICsuite Server

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

The important thing to note here is that this file may contain the password for accessing the Scheduling System. It should therefore be protected against access (including read access) by other users.

### 11. Test

Open a Windows command line window.

Run the BICsuite command line utility `sdmsh`:

```
C:> sdmsh
```

Or, if you haven't created an `.sdmsrc` file:

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

A corresponding prompt is displayed once you have successfully logged on to the server. You can now send commands to the Scheduling Server.

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

System information about the running BICsuite Server should now be displayed.

The utility can be exited again as follows:

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

### 12. Install the convenience package

The convenience package installs a commonly used configuration for an exit state model. In the following command line it is assumed that the `.sdmshrc` file exists. Should this not be the case, the command line has to be modified analogue to the previous step.

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

### 13. Install the examples

Installing the provided sample workflow definitions requires just a few commands:

## Installation with Postgres

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

### 14. Automatically start the BICsuite Server

To start the BICsuite Server automatically when you log on, you just have to save one shortcut to the start script in the startup folder.

The start script is called:

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

### 15. Automatically start the job server

To start the job server automatically when you log on, you just have to save one shortcut to the start scripts in the startup folder.

The scripts are:

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

## Installation with Postgres

### Introduction

This guide does not claim to precisely describe the installation of the database system. Details about this can be found in the Postgres documentation. Normally, though, this guide should be adequate for performing a "standard" installation. For typesetting reasons some lines in the examples are shown with line breaks. This is evident where the line has an indent and there is no prompt at the beginning of the line. When copying the commands, you should *not* use a return at the end of the line.

### Installation

1. Download and install the latest version of Postgres

At the end of the installation, use stackbuilder to install the Postgres JDBC Driver (pg-JDBC) as well.

The Postgres password you chose during the installation will be required again later.

Note:

## Installation with Postgres

If the Postgres service does not start, it may be blocked by your firewall or antivirus software. Running the Postgres service as an administrator may remedy this problem.

### 2. Create the Postgres user bicsuite

Open a Windows command line window and navigate to your Postgres installation directory.

Example:

```
C:> cd /d C:\Program Files\PostgreSQL\9.0\bin
```

Run the postgres createuser command to create the user bicsuite who is also permitted to create databases:

Example:

```
C:> cd /d C:\Program Files\PostgreSQL\9.0\bin
C:> .\createuser -U postgres -W -P bicsuite
```

Use the Postgres password (see 1.).

Remember the bicsuite password you have entered here.

### 3. Create the repository database bicsuitedb

If you have not already done so, open a Windows command line window and navigate to your Postgres installation directory. Run the postgres createdb command to create a database under the name bicsuitedb.

Example:

```
C:> cd /d C:\Program Files\PostgreSQL\9.0\bin
C:> .\createdb -U bicsuite -W -O bicsuite bicsuitedb
```

Use the assigned bicsuite password (see 5.1.2.).

### 4. Create and initialise the database tables

Open a Windows command line window if you have not already done so. Navigate to the BICsuite sql directory.

Example:

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

Run the postgres psql command to initialise the BICsuite repository.

Example:

```
C:> C:\Program Files\PostgreSQL\9.0\bin\psql -U bicsuite
-f pg\install.sql bicsuitedb
```



## Installation with MySQL

Use the assigned bicsuite password (see 5.1.2.).

### 5. Configure the database connection in the BICsuite Server configuration file

Open a Windows command line window if you have not already done so. If you have not set the environment variable `%BICSUITECONFIG%`, use the location `%BICSUITEHOME%\etc` instead.

Edit `%BICSUITECONFIG%\server.conf` and modify the following properties:

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

The hostname must be the same as the computers hostname:

Entering `echo %USERDOMAIN%` in a Windows command line window will return this name.

```
Hostname=<enter the hostname here>
```

### 6. Configure the BICsuite Java Class Path for the Postgres JDBC

Open a Windows command line window if you have not already done so. If you have not set the environment variable `BICSUITECONFIG`, use the location `%BICSUITEHOME%\etc` instead of `%BICSUITECONFIG%`.

Edit `%BICSUITECONFIG%\JAVA_CONF.BAT` so that the environment variable `BICSUITEJDBC` is set to postgres JDBC jar.

Example:

```
SET DBMSHOME=C:\Program Files\PostgreSQL
SET BICSUITEJDBC=%DBMSHOME%\pgJDBC\postgresql-8.4-702.jdbc4.jar
```

## Installation with MySQL

### Introduction

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

For typesetting reasons some lines in the examples are shown with line breaks. This is evident where the line has an indent and there is no prompt at the beginning of the line. When copying the commands, you should *not* use a return at the end of the line.

## Installation

1. Download and install the latest version of MySQL Community Version

The Instance Configuration Wizard is launched after the software has been downloaded.

Select:

- a) Detailed Configuration  
→ Next
  - b) Developer Machine  
→ Next
  - c) Transactional Database Only  
→ Next
  - d) Retain InnoDB tablespace Settings  
→ Next
  - e) Manual Setting (10 connections are more than adequate)  
→ Next
  - f) Add firewall exception for port  
Port 3306, retain Enable TCP and Enable Strict Mode  
→ Next
  - g) Retain Standard Character set  
→ Next
  - h) Retain Install AS Windows Service  
Set Include bin directory in Windows path  
→ Next
  - i) enter root\_password (remember this!)  
→ Next  
→ Execute
2. Create the MySQL user bicsuite and the database bicsuitedb  
Open a Windows command line window and run the MySQL command line utility.

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

Create the user bicsuite and make a note of the password.

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

Create the database

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

## Installation with MySQL

Grant the user bicsuite all privileges to this database

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

### 3. Create and initialise the database tables

Open a Windows command line window if you have not already done so.

Run the following commands:

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

### 4. Download and unpack the MySQL (Connector/J) JDBC Driver

e.g. under C:\Program Files\MySQL

The MySQL JDBC jar archive can then be found e.g. under:

```
C:\Program Files\MySQL\mysql-connector-java-5.1.16
```

### 5. Configure the database connection in the BICsuite Server configuration file

Open a Windows command line window if you have not already done so. If you have not set the environment variable BICSUITECONFIG, use the location %BICSUITEHOME%\etc instead of %BICSUITECONFIG%.

Edit %BICSUITECONFIG%\server.conf and modify the following properties:

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

The hostname must be the same as the computers hostname:

Entering `echo %USERDOMAIN%` in a Windows command line window will return this name.

```
Hostname=<enter the hostname here>
```

### 6. Configure the BICsuite Java Class Path for the MySQL JDBC

Open a Windows command line window if you have not already done so. If you have not set the environment variable BICSUITECONFIG, use the location %BICSUITEHOME%\etc instead of %BICSUITECONFIG%.

Edit %BICSUITECONFIG%\JAVA\_CONF.BAT so that the environment variable BICSUITEJDBC is set to MySQL JDBC jar.

Example:

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

## Installation with Microsoft™ SQL Server

### Introduction

This guide does not claim to precisely describe the installation of the database system. Refer to the SQL Server documentation for more information. Normally, though, this guide should be adequate for performing a "standard" installation. For typesetting reasons some lines in the examples are shown with line breaks. This is evident where the line has an indent and there is no prompt at the beginning of the line. When copying the commands, you should *not* use a return at the end of the line.

### Installation

1. Download and install the latest version of SQL Server Express

During the installation, configure the combined (SQL and Windows) authentication.

2. Create the MySQL user bicsuite and the database bicsuitedb

Open a Windows command line window and run the SQL Server command line utility.

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

Create the user bicsuite with:

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

Close the SQL Server command line utility:

```
1> exit
```

Start the SQL Server command line utility under the user bicsuite and create the database:

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

Close the SQL Server command line utility:

## Installation with Microsoft™ SQL Server

```
1> exit
```

### 3. Create and initialise the database tables

Open a Windows command line window if you have not already done so.  
Run the following commands:

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

### 4. Download and install the JDBC driver for SQL Server

Install it under C:\Program Files as recommended by Microsoft.

### 5. Configure the database connection in the BICsuite Server configuration file

Open a Windows command line window if you have not already done so. If you have not set the environment variable BICSUITECONFIG, use the location %BICSUITEHOME%\etc instead of %BICSUITECONFIG%.

Edit %BICSUITECONFIG%\server.conf and modify the following properties:

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

The hostname must be the same as the computers hostname:

Entering `echo %USERDOMAIN%` in a Windows command line window will return this name.

```
Hostname=<enter the hostname here>
```

### 6. Configure the BICsuite Java Class Path for the SQL Server JDBC

Open a Windows command line window if you have not already done so. If you have not set the environment variable BICSUITECONFIG, use the location %BICSUITEHOME%\etc instead of %BICSUITECONFIG%.

Edit %BICSUITECONFIG%\JAVA\_CONF.BAT so that the environment variable BICSUITEJDBC is set to SQL Server JDBC jar.

Example:

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

## Installation with Ingres

### 7. Configure the TCP/IP access to the SQL Server

Open the SQL Server Configuration Manager from the start menu:

Start → All Programs → Microsoft SQL Server 2008 R2 → Configuration Tools  
→ SQL Server Configuration Manager

Now perform the following steps:

- a) Under SQL Server Network Configuration, select Protocols for SQLEXPRESS
- b) Double-click TCP/IP to edit the TCP/IP connection properties
- c) Set Activated to "Yes"
- d) Switch to the IP Address tab
- e) Under IPALL, clear the Dynamic TCP Ports box
- f) Under IPALL, set the TCP Port box to 1433

## Installation with Ingres

### Introduction

This guide does not claim to precisely describe the installation of the database system. Details about this can be found in the Ingres documentation. Normally, though, this guide should be adequate for performing a "standard" installation. For typesetting reasons some lines in the examples are shown with line breaks. This is evident where the line has an indent and there is no prompt at the beginning of the line *not* use a return at the end of the line.

### Installation

#### 1. Download and install the latest version of Ingres Community Edition

In the Ingres setup wizard, select:

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

#### 2. Create the user bicsuite

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

## Installation with Ingres

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

```
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. Create the repository database `bicsuitedb`

If you have not already done so, open a Windows command line window and create the repository database as the user `bicsuite`.

```
C:> createdb bicsuitedb
```

### 4. Create and initialise the database tables

Open a Windows command line window if you have not already done so. Navigate to the BICsuite sql directory.

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

Run the `ingres sql` command to initialise the BICsuite repository.

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

### 5. Configure the database connection in the BICsuite Server configuration file

Open a Windows command line window if you have not already done so. If you have not set the environment variable `BICSUITECONFIG`, use the location `%BICSUITEHOME%\etc` instead of `%BICSUITECONFIG%`.

A password must be assigned for your Windows user!

Edit `%BICSUITECONFIG%\server.conf` and modify the following properties:

## Installation with Oracle Express Edition

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

Replace `<windows password>` with the Windows password for bicsuite.

The hostname must be the same as the computers hostname:

Entering `echo %USERDOMAIN%` in a Windows command line window will return this name.

```
Hostname=<enter the hostname here>
```

### 6. Configure the BICsuite Java Class Path for the Ingres JDBC

Open a Windows command line window if you have not already done so. If you have not set the environment variable `BICSUITECONFIG`, use the location `%BICSUITEHOME%\etc` instead of `%BICSUITECONFIG%`.

Edit `%BICSUITECONFIG%\JAVA_CONF.BAT` so that the environment variable `BICSUITEJDBC` is set to ingres JDBC jar.

Example:

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

## Installation with Oracle Express Edition

### Introduction

This guide does not claim to precisely describe the installation of the database system. Details about this can be found in the Ingres documentation. Normally, though, this guide should be adequate for performing a "standard" installation. For typesetting reasons some lines in the examples are shown with line breaks. This is evident where the line has an indent and there is no prompt at the beginning of the line *not* use a return at the end of the line.

### Installation

1. Download and install the latest version of Oracle Express Edition

Make a note of your Oracle system password.

2. Create the Oracle user bicsuite

Open a Windows command line window and run the Oracle SQLPlus Server command line utility.



## Installation with Oracle Express Edition

```
C:> sqlplus SYSTEM/oracle_system_password
```

Create the user bicsuite with:

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

Assign the required access privileges:

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

### 3. Create and initialise the database tables

Open a Windows command line window if you have not already done so. Navigate to the BICsuite sql directory and run the following commands:

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

### 4. Configure the database connection in the BICsuite Server configuration file

Open a Windows command line window if you have not already done so. If you have not set the environment variable BICSUITECONFIG, use the location %BICSUITEHOME%\etc instead of %BICSUITECONFIG%.

Edit %BICSUITECONFIG%\server.conf and modify the following properties:

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

The hostname must be the same as the computers hostname in both the example above and the configuration file: Entering `echo %USERDOMAIN%` in a Windows command line window will return this name.

```
Hostname=<enter the hostname here>
```

### 5. Configure the BICsuite Java Class Path for the Oracle JDBC

Open a Windows command line window if you have not already done so. If you have not set the environment variable BICSUITECONFIG, use the location %BICSUITEHOME%\etc instead of %BICSUITECONFIG%.

Edit %BICSUITECONFIG%\JAVA\_CONF.BAT so that the environment variable BICSUITEJDBC is set to Oracle JDBC jar.

Example:

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

## Installing the Zope server

### Introduction

A Zope Application Server has to be installed before you can use the BICsuite!Web user interface.

For typesetting reasons some lines in the examples are shown with line breaks. This is evident where the line has an indent and there is no prompt at the beginning of the line *not* use a return at the end of the line.

Some files from the distribution are required to install the BICsuite!Web user interface. The first three steps of the BICsuite Server installation have to be performed before Zope can be installed.

### Installation

1. Download and install Python 2.7 from [www.python.org](http://www.python.org)

The default directory for the Python installation is `C:\Python27`. In our example, we are using the installation directory `C:\Program Files\Python27`. In the following we refer to this directory as `PYTHONDIR`.

2. Download and install Python `setuptools`

Important: Make sure you use the correct version for Python 2.7

3. Optionally, download and install `pywin32`

Important: Make sure you use the correct version for Python 2.7

This step is only necessary if the BICsuite!Web Zope server is to be started as a service.

4. Install the Zope2 software

Open a Windows command line window. Navigate to the virtual Python environment for the Zope installation and install Zope2.

Example:

```
C:> cd /d C:\Program Files\Python27
C:> Scripts\easy_install -i http://download.zope.org/Zope2/index/2.13.6 Zope2
```

Note:

In a test environment under Windows XP, this command returned an error (`rmdir directory not empty`) the first time it was run and had to be repeated.

5. Create a Zope instance for BICsuite!Web

Open a Windows command line window. Navigate to the virtual Python environment for the Zope installation and create the Zope instance `bicsuiteweb`.

## Installing the Zope server

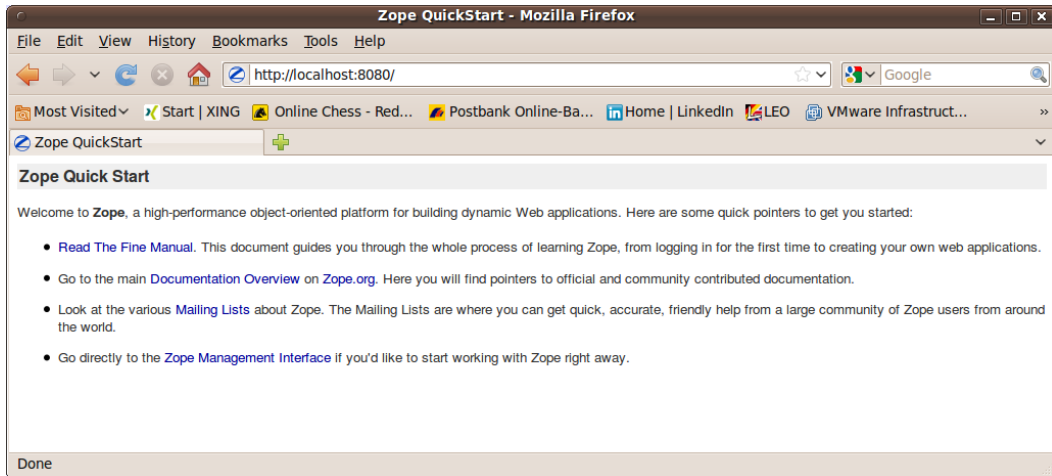


Figure 3.1: Zope Quicks Start Seite

The used password here can be replaced by any password of your own choice. The username must be `sdmsadm` though.

Example:

```
C:> cd /d C:\Program Files\Python27
C:> Scripts\mkzopeinstance -d ..\bicsuiteweb -u sdmsadm:sdmsadm_password
```

Important:

If Oracle has been installed on this computer, the Zope standard port 8080 may already be in use. In this case, the port will have to be changed in the file

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

Here it is assumed that there are no conflicts concerning the use of the port and that Zope can be addressed at port 8080.

Test:

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

In the browser, the URL `http://localhost:8080` should now display the Zope quick start page as shown in Figure 3.1.

Zope can now be exited again with `CRTL+C` or by closing the Windows command line window.

## Installing the Zope server

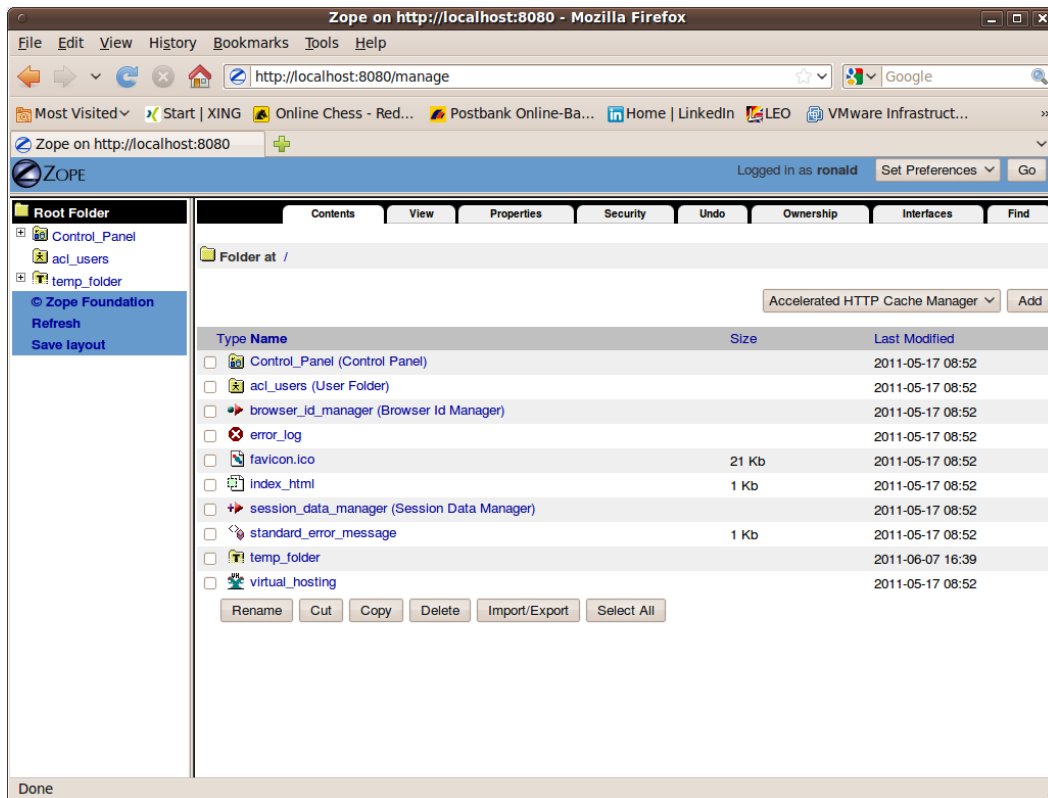


Figure 3.2: Zope Management Oberfläche

### 6. Install the BICsuite!Web components

The Zope installation has to be extended with several components before the BICsuite!Web components can be installed.

```
C:> cd /d C:\Program Files\bicsuite\bicsuiteweb
C:> mkdir Extensions
C:> cd Extensions
C:> copy "%BICSUITEHOME%\zope\*.py" .
C:> cd ..\Products
C:> mkdir BICsuiteSubmitMemory
C:> cd BICsuiteSubmitMemory
C:> copy "%BICSUITEHOME%\zope\BICsuiteSubmitMemory\*.py" .
C:> cd ..\..\import
C:> copy "%BICSUITEHOME%\zope\SDMS.zexp" .
```

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

Open a Windows command line window.

## Installing the Zope server

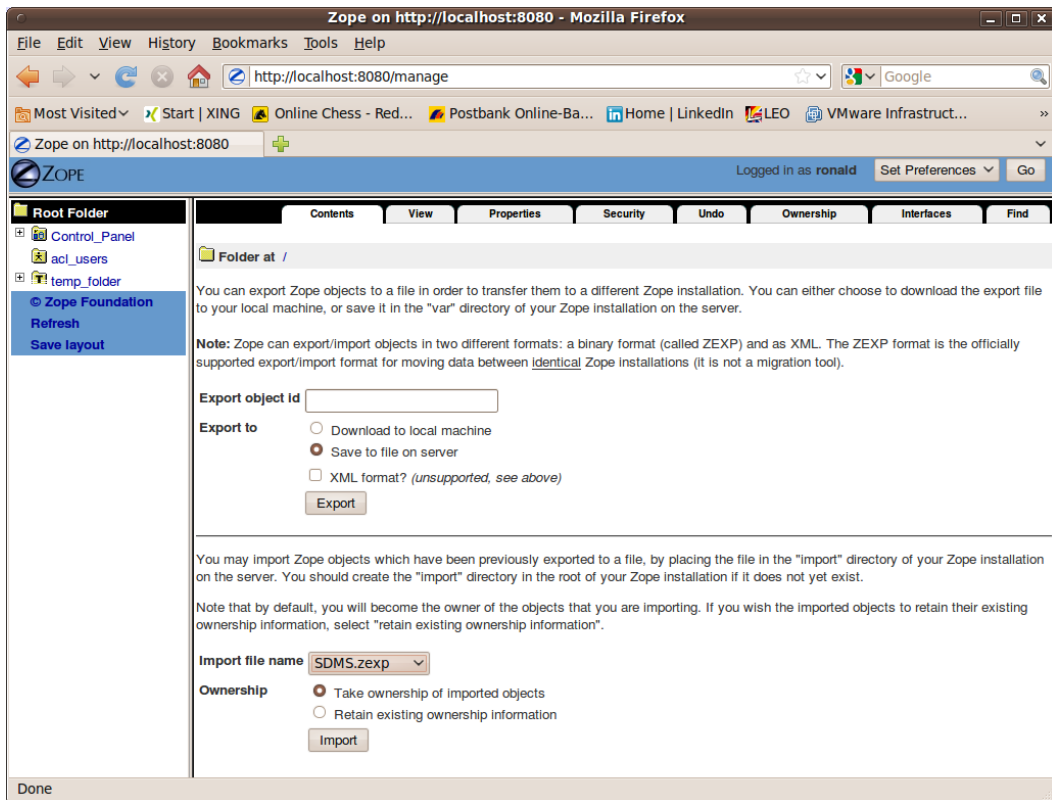


Figure 3.3: Zope Import Dialog

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

The Zope Management user interface, like in Figure 3.2 is now opened at the address

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

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

The front end software is now loaded into Zope (Import button, see also Figure 3.3):

- a) Import it into the folder `/SDMS.zexp`.
- b) In the folder `/SDMS/Install`, mark and copy the folders `User` and `Custom`.

## Installing the Zope server

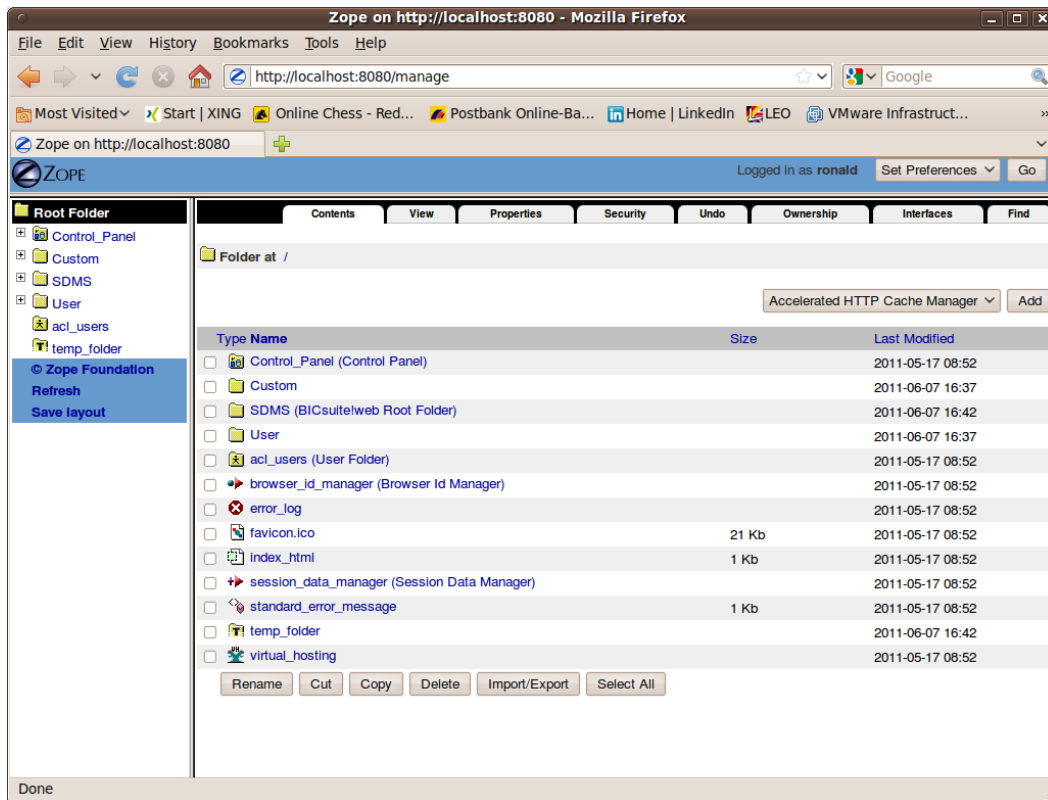


Figure 3.4: Zope result window

c) Create the folders User and Custom in the folder / by pasting them.

If everything has been carried out without any errors, the interface will look like the screenshot as in Figure 3.4.

### 7. Configure the server connections

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

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

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

## Installing the Zope server

```
# Port at which the BICsuite!Server is addressed
'PORT'      : '2506',

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

# Optional property stating whether BICsuite!Web should
# cache the server connections
'CACHE'     : 'Y'

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

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

If a server is to be addressed via a secure SSL connection, the following additional properties have to be defined as well:

```
# Connection is set up via Secure Socket Layer
'SSL'       : 'true',

# If stated, the identity of the BICsuite! Server is verified
# The stated file must contain the BICsuite!Server server certificate
'TRUSTSTORE' : 'truststore.pem',

# If the BICsuite!Server requires client authentication,
# this property must be defined and the stated file
# must contain the client's certificate and private key.
# The certificate must be known to the server in its trust store.
'KEYSTORE'  : 'keystore.pem'
```

### Note:

When using SSL, for performance reasons it is advisable to use cached server connections because setting up a secure connection is a resource-intensive operation.

### 8. Open the BICsuite!Web interface

The user interface is now available at the address

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

## Installing the Zope server

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

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

### 9. Automatically start the BICsuite!Web interface

To have BICsuite!Web interface start automatically when you log on, create a shortcut to

```
C:\Program Files\bicsuite\bicsuiteweb\bin\runzope.bat
```

in your startup folder.