

ICE-Installation Documentation

07.05.2008

Inhaltsverzeichnis

Introduction.....	2
Creating Databases and User.....	2
Installing Web Application.....	3
Checking Out ICE.....	3
Configuring ICE.....	4
Compiling ICE.....	6
Installing <iceproject> Web Application	6
Deploying Web Application.....	6
SingleSignOn Valve.....	7
Installing JDBC Drivers.....	7
Shared Classes.....	7
Configuring JK Connector.....	7
Creating Tables in ice Database.....	7
Creating Tables in sta Database.....	8
Further Steps in Filling Databases.....	9

Introduction

This documentation assumes that the following basic software has been already installed on the machine:

- Java
- MySQL
- php
- phpMyAdmin
- Web-Server (Apache)
- Servlet Container (Tomcat)
- Tomcat connector (mod_jk)

Creating Databases and User

For the ICE-Installation two databases are created: `ice` und `sta`.

The database names may differ, but in that case they should be customized correspondently in all configuration files. The `ice` database contains the entire pool of data which are available for all users and can be used to respond the ad hoc (online) requests. In the `sta` database the so called 'standard reports' are stored. Furthermore, information about the iCE-user and their rights are stored in the database as well.

```
CREATE DATABASE `ice`;  
CREATE DATABASE `sta`;
```

First the user `iceadmin` must be created. It must have a full access to the `ice` and `sta` databases, but the minimal range of rights for accessing the `mysql` database:

```
CREATE USER 'iceadmin'@'localhost' IDENTIFIED BY 'hannover_8';  
CREATE USER 'iceadmin'@'%' IDENTIFIED BY 'hannover_8';  
GRANT USAGE ON * . * TO 'iceadmin'@'localhost' ;  
GRANT USAGE ON * . * TO 'iceadmin'@'%' ;  
grant all privileges on ice.* to iceadmin@'%';  
grant all privileges on ice.* to iceadmin@localhost;  
grant all privileges on sta.* to iceadmin@'%';  
grant all privileges on sta.* to iceadmin@localhost;
```

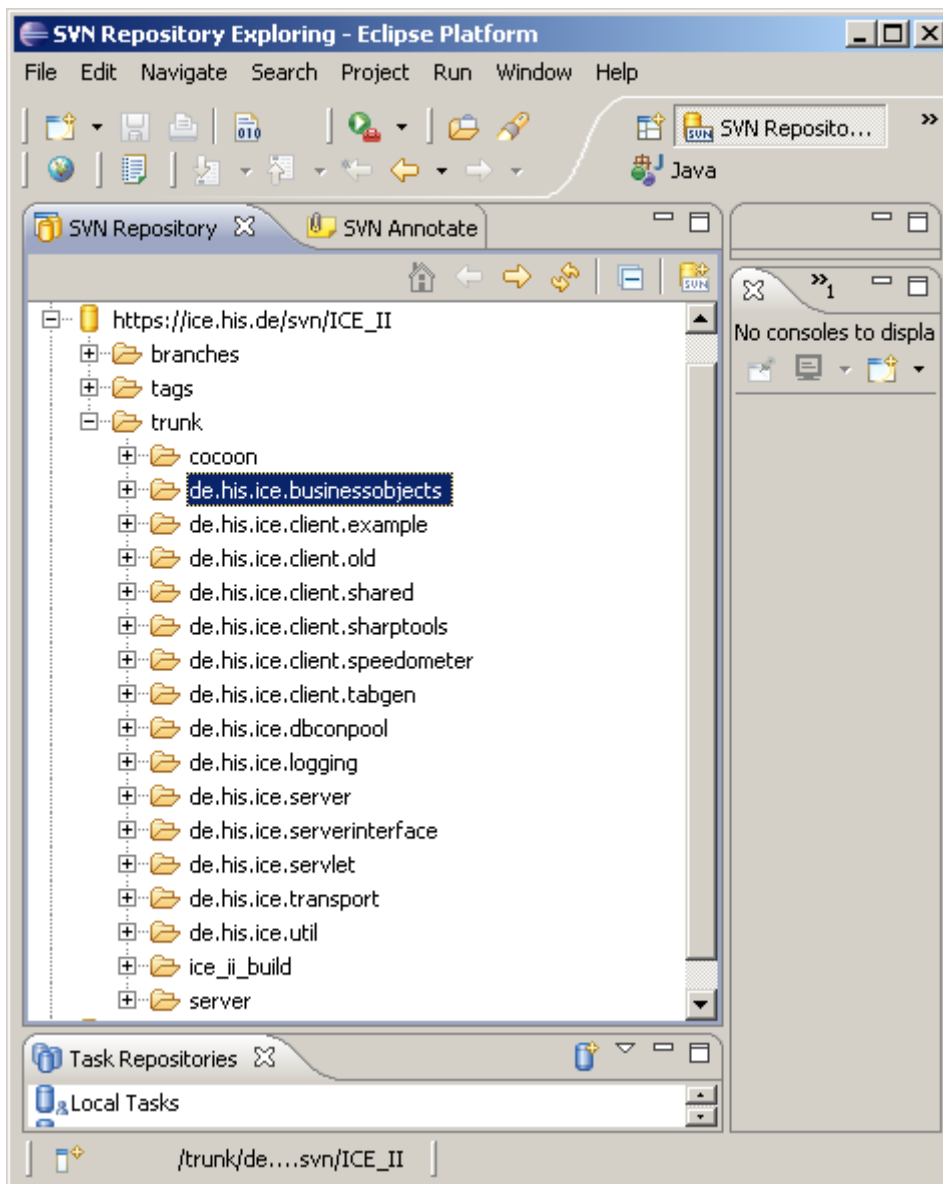
For the creation of the database tables you need a part of ICE java classes. For this reason the next you do is to checkout and to build ICE.

Installing Web Application

Checking Out ICE

The source code of the ICE-Project is stored on `ice.his.de` server and is managed by version control system **Subversion** (SVN). Checkout of ICE-Project from SVN can be made, for example, in Eclipse (open-source integrated development environment) using the Subversion Plugin:

- Change to SVN- Perspective (Window -> Open Perspective -> Other -> SVN Repository Exploring)
- In Tab SVN Repository open the entry for https://ice.his.de/svn/ICE_II. If the entry does not exist yet, creat it (right click: New -> Repository Location)
- Select the entry **trunk** there. The subfolders seen are the individual Eclipse-Projects. They shall be checked out separately.



Configuring ICE

The following files are of relevance for configuring ICE:

- Client
 - de/his/ice/logging/log4j_client.conf (in directory de.his.ice.logging/src)
 - de/his/ice/transport/transport.conf (in directory de.his.ice.transport/src)
 - de/his/ice/serverinterface/serverinterface_client.conf (in directory de.his.ice.serverinterface/src)
 - de/his/ice/client/tabgen/config/tabgen.conf (in directory de.his.ice.client.tabgen/src)
e.g. for configuration the language `language = en`
 - de/his/ice/client/old/config/client.conf (in directory de.his.ice.client.old/src)
e.g. for configuration the language `language = en`
 - static pages in ice_ii_build/htdocs directory, in particular
 - JNLP-Files (Web Start descriptor files) in directory ice_ii_build/htdocs/icewww/ice/ice_tab/text:
 - ◆ tabgen_application.jnlp (Web Start descriptor file for application)

```
<jnlp spec="1.5+" codebase="http://localhost/iceproject" href="icewww/ice/ice_tab/text/tabgen_application.jnlp">
  <information>
    <title>ICE II Tabellengenerierung</title>
    <vendor>ICE Projektgruppe, HIS GmbH</vendor>
    <homepage href="http://www.his.de"/>
    <description>ICE II Tabellengenerierung</description>
    <icon href="ice.gif"/>
    <offline-allowed/>
  </information>
```

- ◆ tabgen_applet.jnlp (Web Start descriptor file for applet)

```
<jnlp spec="1.5+" codebase="http://localhost/iceproject" href="icewww/ice/ice_tab/text/tabgen_applet.jnlp">
  <information>
    <title>ICE II Tabellengenerierung</title>
    <vendor>ICE Projektgruppe, HIS GmbH</vendor>
    <homepage href="http://www.his.de"/>
    <description>ICE II Tabellengenerierung</description>
    <icon href="ice.gif"/>
    <offline-allowed/>
  </information>
  <security>
    <all-permissions />
  </security>
  <resources>
    <property name="sun.swing.enableImprovedDragGesture" value="true"/>
    <j2se version="1.5.0+" href="http://java.sun.com/products/autodl/j2se"/>
    <jar href="icewww/classes/ice/lib2/iceII_client.jar"/>
    <jar href="icewww/classes/ice/lib2/TableLayout.jar"/>
    <jar href="icewww/classes/ice/lib2/log4j-1.2.9.jar"/>
    <jar href="icewww/classes/ice/lib2/looks-2.0.4.jar"/>
  </resources>
  <applet-desc documentBase="http://localhost/iceproject/icewww/ice/ice_tab/text" name="AppletStarter" main-class="de
  <param name="icedbnr" value="1"/>
  <param name="icedbthema" value="Hochschulstatistik"/>
  </applet-desc>
</jnlp>
```

- HTML-pages with embedded applet tag in directory ice_ii_build/htdocs/icewww/ice/ice_tab/text:
 - ◆ tabGen-BestExt6_1.htm
 - ◆ tabGen-BestInt6_1.htm
 - ◆ tabGen-Aktuell.htm
 - ◆ finden_wsdb.htm

- ◆ StandTabVerw.htm
- ◆ UserGroupsVerw.htm
- Server
 - de/his/ice/logging/log4j_server.conf (in directory de.his.ice.logging/src)
 - de/his/ice/serverinterface/serverinterface_server.conf (in directory de.his.ice.serverinterface/src)
 - config/ICETab.cfg (in directory server/src)
 - config/STATab.cfg (in directory server/src)
- Databases
 - ice_ii_build/META-INF/context.xml

```
<Context path="/iceproject">
  <Realm className="org.apache.catalina.realm.DataSourceRealm" debug="99" dataSourceName="jdbc/sta"
  localDataSource="true" userTable="users" userNameCol="anmeldung" userCredCol="passwd"
  userRoleTable="userroles" roleNameCol="role"/>

  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs/ice"
  prefix="localhost_access_log." suffix=".txt" pattern="combined" resolveHosts="false"/>

  <Resource name="jdbc/sta" auth="Container" type="javax.sql.DataSource"
  driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost/sta?autoReconnect=true"
  username="iceadmin" password="hannover_8" maxActive="20" maxIdle="10" maxWait="-1"/>
  <Resource name="jdbc/ice1" auth="Container" type="javax.sql.DataSource"
  driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost/ice?autoReconnect=true"
  username="iceadmin" password="hannover_8" maxActive="20" maxIdle="10" maxWait="-1"/>
</Context>
```

- cocoon/META-INF/context.xml

```
<Context path="/cocoon">
  <Realm className="org.apache.catalina.realm.DataSourceRealm" debug="99"
  dataSourceName="jdbc/sta" localDataSource="true" userTable="users" userNameCol="anmeldung"
  userCredCol="passwd" userRoleTable="userroles" roleNameCol="role"/>
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs/ice"
  prefix="localhost_access_log." suffix=".txt" pattern="combined" resolveHosts="false"/>
  <Resource name="jdbc/sta" auth="Container" type="javax.sql.DataSource"
  driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost/sta?autoReconnect=true"
  username="iceadmin" password="hannover_8" maxActive="20" maxIdle="10" maxWait="-1"/>
</Context>
```

- cocoon/WEB-INF/cocoon.xconf

```
<datasources>
  <jdbc name="standard">
    <dburl>jdbc:mysql://localhost/sta</dburl>
    <user>iceadmin</user>
    <password>hannover_8</password>
  </jdbc>
  <jdbc name="tdb1">
    <dburl>jdbc:mysql://localhost/ice</dburl>
    <user>iceadmin</user>
    <password>hannover_8</password>
  </jdbc>
</datasources>
```

While building the `war` target the jar archives with the client classes used within a new table generation tool are signed. It is required in order that the application on the client machine could get the necessary permissions to perform security-relevant actions, such as loading and saving files locally, for instance. Therefore, all client configuration files should be customized accordingly before compiling.

Furthermore there are several property files for the internationalization of the labels used in GUI. The property files for a new table generation tool are placed in directory `de.his.ice.util.resource`, for the other tools – in client directories (`de.his.ice.client.old.tabsearch`, `de.his.ice.client.old.tabadmin`, `de.his.ice.client.old.tabupdate`, `de.his.ice.client.old.useradmin`, `de.his.ice.client.old.tabgen`) respectively.

Compiling ICE

For compiling ICE source code a java build tool called Ant (Version 1.7 and later) is used.

The project `ice_ii_build` is of great importance for compiling ICE sources and creating ICE web application. Both the ICE-Documentation and a build file `build.xml` are located here. The build file is used for compiling the entire ICE-Project and preparing a war archive, required for the installation of ICE in a Tomcat servlet container. As a result of executing `war` target a directory `ice_ii_build/webapps` will be created, where the entire application incl. Client/Server classes and the web pages is built. The following actions are thereby performed:

- Context file `context.xml` is copied to the directory `META-INF`
- File `web.xml` is copied to `WEB-INF`
- ICE server classes are copied to `WEB-INF/classes`
- Log4j library is copied to `WEB-INF/lib`
- JSP pages go to `jsp` directory
- Directory `icewww` contains static web pages (subfolder `ice`) as well as client classes (subfolder `classes/ice`)
- Directory `shared/lib` is created, and jar archive `icecocoon.jar` is copied here. The archive contains classes, which are used both by `cocoon` and `iceproject` web applications. That's why it must be placed in Tomcat directory `shared/lib` later
- directory `admintools` is created. It contains the administration tools `HintsAdmin.jar`, `KeysAdmin.jar`, `KeywordsAdmin.jar`, `Thcodnam.jar`
- Finally the archive `iceproject.war` with the content of `ice_ii_build/webapps` directory is created and saved in the `webapps` directory

Installing <iceproject> Web Application

Deploying Web Application

Before the installation of `iceproject` web application Tomcat must be shut down.

To install the ICE web application the archive `iceproject.war` should be copied to the Tomcat `webapps` directory. After Tomcat having been restarted, a new application is „deployed“ to servlet container automatically. Tomcat unpacks the archive, creates a file structure for the web application and saves the context file `context.xml` from the archive's folder `META-INF` into the directory `%TOMCAT_HOME%/conf/Catalina/localhost` under the name `iceproject.xml`, in case the file does not exist yet. The context file contains various

important configuration directives for the web application, such as datasources specifications, logging, settings for tomcat user authentication realm etc.

SingleSignOn Valve

Normally, users must authenticate themselves to each web application individually. Uncomment the following entry in `%TOMCAT_HOME%/conf/server.xml` if you would like a user to be authenticated the first time they encounter a resource protected by a security constraint, and then have that user identity maintained across all web applications contained in this virtual host.

```
<Valve
className="org.apache.catalina.authenticator.SingleSignOn"/>
```

Installing JDBC Drivers

Both web applications - `iceproject` and `cocoon` - require JDBC drivers for creating connection to the database. Placed into the `%TOMCAT_HOME%/common/lib` the drivers are available both to Tomcat itself and to all web applications.

Shared Classes

While building the application, the jar archive `icecocoon.jar` is created as well. It can be found in `webapps/shared/lib` and must be copied to the tomcat directory `TOMCAT_HOME%/shared/lib`. Thus the classes in this jar are shared across all web applications.

```
cp /path/to/ice_ii_build/webapps/shared/lib/icecocoon.jar %TOMCAT_HOME
%/shared/lib
```

Configuring JK Connector

Apart from that, the web applications `iceproject` and `cocoon` must be mapped to the Tomcat JK connector's worker (see `uriworkermap.properties` and `workers.properties` for details).

After all files having been copied to the appropriated locations and customized, Tomcat must be restarted for the changes to take effect.

Creating Tables in ice Database

Database tables can be created using the tool `AdminTableHandler`. To install this Java tool copy the ICE Server classes to any directory within the file system. Using the classes located in the directory `%TOMCAT_HOME%/webapps/iceproject/WEB-INF/classes` is to prefer however.

The configuration file `config/ICETab.cfg` must be accordingly customized. It contains the directives required by various ICE applications. The property names start with „\$“, value names – with #. The most important directives for the creation of ICE database tables are those specifying the parameters required for creating the database connection: `DatabaseHost`, `DatabaseName`, `DatabaseURL`, `DatabaseDriver`, `UserName`, `UserPassword`, `openDatabase`, `dbsystem`.

To run the tool change in a console window to the directory where the ICE Server classes are located:

```
cd %TOMCAT_HOME%/webapps/iceproject/WEB-INF/classes
```

Set the classpath before starting:

```
export CLASSPATH=./:/path/to/jdbcdriver:/path/to/log4jjar
```

To run the tool the Java application `AdminTableHandler` in the package `dbimport` must be launched:

```
/path/to/java dbimport.AdminTableHandler [control word] [database]
[database segment]
```

The first parameter specifies, what tables should be either created or deleted: (CreateTable:tabname, CreateSingle, CreateMultiple, CreateDBDependent, CreateSingleAndMultiple, DropSingle, DropMultiple, DropDBDependent, DropAll).

Thereby the following configuration files in package `config` are analysed:

```
AdminTables_ice_mysql.conf, AdminTables_sta_mysql.conf,
AdminTables_ice_oracle.conf, AdminTables_sta_oracle.conf,
AdminTables_ice_informix.conf, AdminTables_sta_informix.conf.
```

Which of these files is actually taken depends on the database instance (`ice` or `sta`) and the database system (properties from the config files `ICETab.cfg` and `STATab.cfg` respectively).

It would be sufficient to create all single (`CreateSingle`) and multiply tables (`CreateMultiple`) first:

```
java dbimport.AdminTableHandler CreateSingle ice iceadmin
java dbimport.AdminTableHandler CreateMultiple ice iceadmin
```

Using the script `scripts/fillltables_ice_en.sql`, located on the server, the initial data are inserted into the ICE database tables:

```
mysql -u iceadmin -p ice <fillltables_ice_en.sql
```

The script `scripts/import_testdata_ice.sql` inserts a couple of test data into the ICE database. It can be used for testing the functionality of ICE system:

```
mysql -u iceadmin -p ice <import_testdata_ice.sql
```

Creating Tables in sta Database

The tool `AdminTableHandler` can be used for creating tables in `sta` database as well. For the tool configuration the config file `config/STATab.cfg` is of relevance and must be customized. The `sta` database contains only the single tables. To create them launch:

```
/pfad/zu/java dbimport.AdminTableHandler CreateSingle sta iceadmin
```

The script `scripts/fillltables_sta_en.sql`, located on the server, inserts the initial data into the `sta` tables:

```
mysql -u iceadmin -p sta <fillltables_sta_en.sql
```

Besides, the update types supported by the system are imported as well using this script.

The script `scripts/import_testdata_sta.sql` inserts a couple of test data into the `STA` database. It can be used for testing the functionality of ICE system:

```
mysql -u iceadmin -p sta <import_testdata_sta.sql
```

Finally, two users are created:

- User `ice` with password `ice` with role `icesu` in group `ICE`
- User `guest` with password `guest` with role `iceadmin` in group `GUEST`

Further Steps in Filling Databases

Having been installed the ICE-System can now be filled with information. Prior to importing the data stocks, the key tables `tabellenmerkmale` and `tabellenschluessel` must be filled first. For managing ICE keys the administration tool `KeyAdmin.jar` is used.

For the successful searching for the datastocks later the administrating of the keywords is of great importance. The keyword relevant information is stored in the ice database tables `schlagworte`, `merkschlagworte`, `ausprschlagworte`. The keywords are imported into the database using the administration tool `tools.KeywordGenerator`.

The keywords are also used for searching for the standard reports stored in the sta database. Therefore, they must be imported into the sta database too. The tool `tools.StandardSWupdate` assists in reading keywords from a file and importing them into the database tables `schlagwort`, `schluessel`, `schlagschluessel`. Should any standard reports already exist in the database, they are finally reassigned on the basis of the updated key word list (thereby `hashtabelle` is filled anew).

Access data to ICE-Demo system:

URL: `http://localhost/ice`

Login: `ice`

Password: `ice`