jProject Business Suite

User Documentation

Version: 0.1.2

Table of Contents

Preface

1. Overview

The aim of this project is to provide an easy to use solution concerning the various basic task of working in projects from the human project resource point of view (not the project management!). It's not (just another) project management solution.

A project might be a lot of things. It can be a real project, a customer visit, a business contact, a vacation journey or other. The main focus is to provide not a lot of possibilities and various customization options (there are already a lot of business solutions which trying to do exactly this), but to create a small but capable core application with various possibilities to create input and output to this and other application.

We understand that we don't need to reinvent the wheel. But didn't you have the wish sometimes to access or collect data from systems which are outside "your" current application? Copy/Paste, Batch Runs, CSV, Directory Access, Queue Access - there are many ways to exchange data but its not quite common that `Application A` uses the same mechanism as `Application B` . So we started a 1st attempt with this small business application to find a solution. Step by step.

Client Maintenance
: Clients are divided in companies and persons. They can be maintained either via the internal subsystem or via a configurable LDAP accessor.

Users and Login
: Users can be maintained either via the internal subsystem or by specifying a JAAS compliant login module. The application itself provides 2 login modules. One for the internal user system and one for accessing user data held on a LDAP server.

Reporting
: Additional reports can be easily added via the plugin manager interface. Reports can be designed via various WYSIWYG editors provided for Jasper Reports.

Role Management
: The entire system uses different roles to associate relations between components. Even a user is just a role any component might play.

Import / Export [available from version 0.2 on]
: The openadapter™ Framework is used to set-up I/E tasks to various systems. This is one of the main features of this application. Therefore data can be easily "routed" throughout various external systems.

Business
: The main business functionality consists of maintaining projects, travels, time records and clients. You can create time based reports for your customers, travel reports for your own and you can even create and maintain invoices.

2. Where to start?

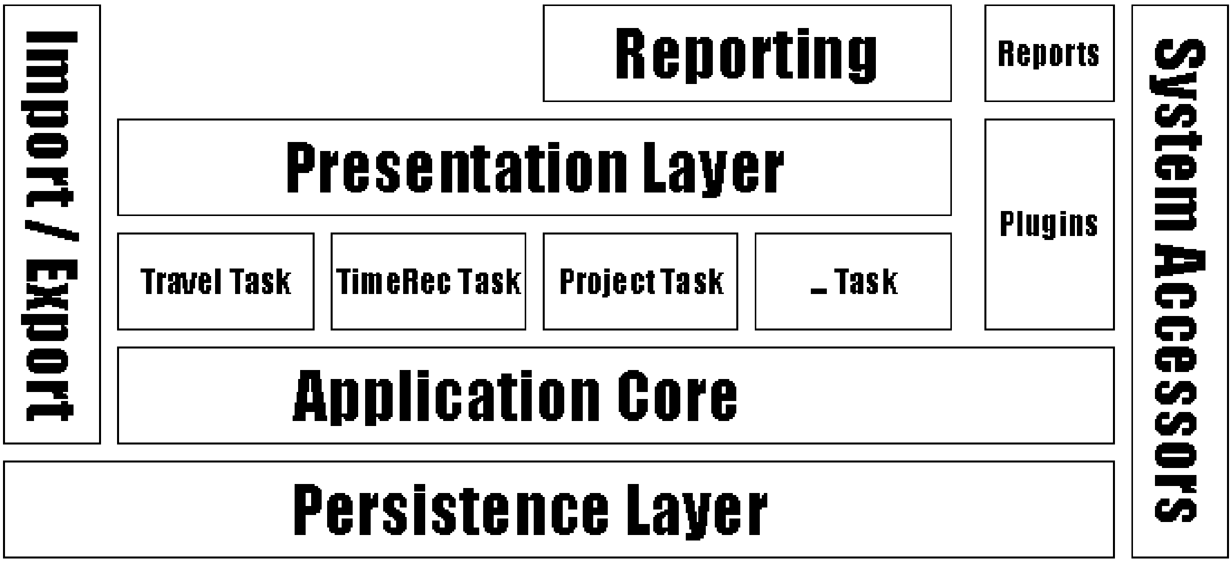
Follow this steps to learn about installing, configuring and using this application:

1. Read Chapter 1, *Architecture Overview* to understand the environments and components this application is build for.
2. Read Chapter 2, *Installation* for installing the web application, using Tomcat.
3. Read Chapter 3, *Configuration* for setting up this application to your needs.
4. Read Chapter 7, *Business Functions* for a step by step explanation of the main functionality.
5. Find answers to FAQ's on the project website.

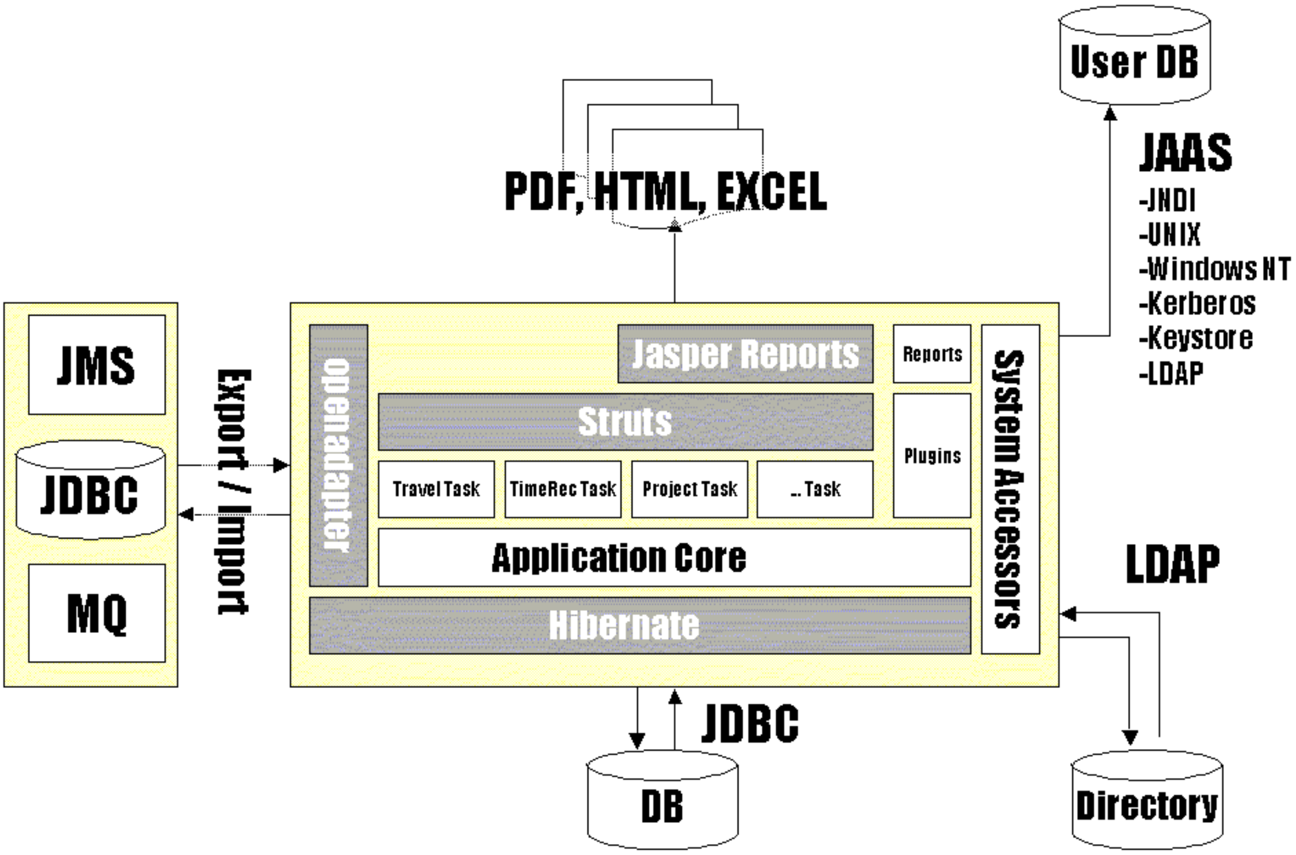
If you got further questions, use the forums provided on the `SourceForge` project page (http://sourceforge.net/projects/jsuite/).

Chapter 1. Architecture Overview

A high-level view of the application architecture:



Some of the components are covered by external frameworks. This diagram shows how the application interacts with other components:



Chapter 2. Installation

The application is shipped as a webarchive file (WAR). Every application server or servlet container has its own way how to install a WAR archive. You should have a look at the documentation of your servlet container / application server.

Note that no JDBC driver is shipped with the application, you have to provide them seperately on your own.

2.1. Preferences

1. A servlet container with servlet specification 2.3 and JSP 1.1. (or J2EE 1.3)
2. JDK 1.3 or upwards. If you want to use JAAS, JDK 1.4 is recommended. See Section 3.4, “Login and User Authentication” for more details.
3. A JDBC complient database driver supporting your database. Check your database documentation or webpage for additional information. E.g. the JDBC driver supporting free databases like MySQL can be downloaded directly from their webpage.

Note

You can use even a container with servlet spec 2.2 if you don't need special character encoding for UTF-8 e.g. japanese, russian or german mixed characters on a form which is done via a servlet filter. This filter used in the web.xml file is the only dependency to the servlet 2.3 specification.

2.2. Example [Tomcat]

Example 2.1. Installing on Tomcat

To install the application follow this steps:

1. Copy the war file in the `$TOMCAT_HOME$/webapps` directory or use the admin application.
2. If the automatic deployment process didn't start or Tomcat isn't configured for automatic deployment, restart Tomcat. Check the logs whether `/jproject` is shown as a deployed context, the `jproject` subdirectory under `$TOMCAT_HOME$/webapps` has been created and no error messages are thrown.

   ### Tip

   Sometimes you get following exception during the automatic deployment process: `WrapperException`

   This mainly happens if your application context gets reloaded but your servlet is still not available or found by the servlet container. A server restart should solve this problem.
3. However - if the automatic deployment didn't start at all and a restart of tomcat does not extract the application to the `$TOMCAT_HOME$/webapps` directory you might try to manually extract the contents.

   Just create the `jproject` directory in `$TOMCAT_HOME$/webapps` and unzip the contents of the WAR file to this directory.
4. Either place the JDBC driver jar/zip archive according to your application server settinngs into the class-

path or directly into the application library context. Simply copy the JDBC driver jar file therefore into the `$JPROJECT_HOME$/WEB-INF/lib`.

5. Check the correct installation by opening `http://$yourserver$:$yourport$/jproject/setup.jsp`. The setup page should be shown.

Before you can use the application you need to finish the steps described in the Chapter 3, *Configuration* section.

Chapter 3. Configuration

The application got one main setup file. However, if JAAS security and user authentication is used other setup files are mandatory as well. All setup and configuration files are stored in `$APP_HOME/WEB-INF/classes`.

jproject.properties
: This is the main setup file. Various properties can be setup. See Section 3.1, “Project property file [jproject.properties]” for more details.

log4j.properties
: Log4J is used as an application logger. Check http://logging.apache.org/log4j/docs/ for more details how to use the logger.

hibernate.properties
: Config File of the persistence layer. `Hibernate` (http://www.hibernate.org) is used the standard persistence mechanism. See Section 3.3, “Persistance Layer (Hibernate)” how to setup `Hibernate` for this application.

jproject.jaas
: Defines the LoginModule used by the system. Check Section 3.4, “Login and User Authentication” for more details.

policy.xml
: Standard, principal based policy file used by the policy provider shipped with this application. Define permissions to user based on principals used by JAAS. If you change the JAAS login module to other but those provided by the application (Standard and LDAP), you need to adjust this file to your principals and values.

jproject.policy
: Example policy file if the standard Sun / Java policy provider is used. Those entries needs to be placed in the standard VM policy file.

font.properties
: Fonts definition file if `PJA` must be used to support the reporting subsystem. (Linux, JDK 1.3).

3.1. Project property file [jproject.properties]

Table 3.1. Property

Property name	Purpose
`LOGGER`	Switches the logger on or off.
`STYLE`	Standard css file used by the system if no other are defined by the user.
`DEBUG_TIME`	If set to "true" JSP and action processing times will be shown on every page footer.
`ENCODE_PASSWORDS`	If set to "true" passwords in the user table will be encoded
`currency.name`	standard currency; valid values may depend on your VM [ADP - ATS - BEF - BGL - BOV - BYB - CLF -

Property name	Purpose
	DEM - ESP - EUR - FIM - FRF - GRD - IEP - ITL - LUF - MXV - NLG - PTE - RUB - USN - USS - XAF - XAG - XAU - XBA - XBB - XBC - XBD - XCD - XDR - XFO - XFU - XOF - XPD - XPF - XPT - XTS]
`currency.symbol`	standard currency symbol
`currency.useVMSettings`	If set to true (standard), VM settings will be used instead of above settings. Note that you might get problems with older VM versions; they will not show EURO as currency for european countries but the old country currency e.g. DM, etc.
`standard.policy.profile`	needs to be a valid profile defined in the policy file (see Section 3.6, “User/Principal Permissions”); this profile will be used for new created users
`languages`	languages supported by the system; e.g. de, en, de_DE, en_US, en_UK etc.
`global.security.usejaas`	If you are running on JDK 1.4 or higher, the use of JAAS for authorization is recommended. The Standard [jcon.jproject.jaas.BasePolicy] will handle the policy.xml file. Any other Policy Provider might me used. If you are using JDK 1.3, JAAS is not supported for authorization. However, if you switch off the use of JAAS via this property, no policy provider will be used, neither the standard one nor the one you might provide via the property [policy.provider]. The standard policy.xml file will be read in an normal way, and the authorization will take place as usual. No own implementation can be used if you switch this property off. If you want to use JAAS authorization even in an JDK 1.3 environment you might either use your own policy provider or use the standard policy provider of your JDK. An example policy file for use with Sun's policy provider implementation is shipped with this application.
`jproject.security.auth.policy`	Path to the standard policy file. Specify this, if you have problems with the automatic find process of the application.
`java.security.auth.login.config`	Path to the login module config file. Specify this, if you have problems with the automatic find process of the application.

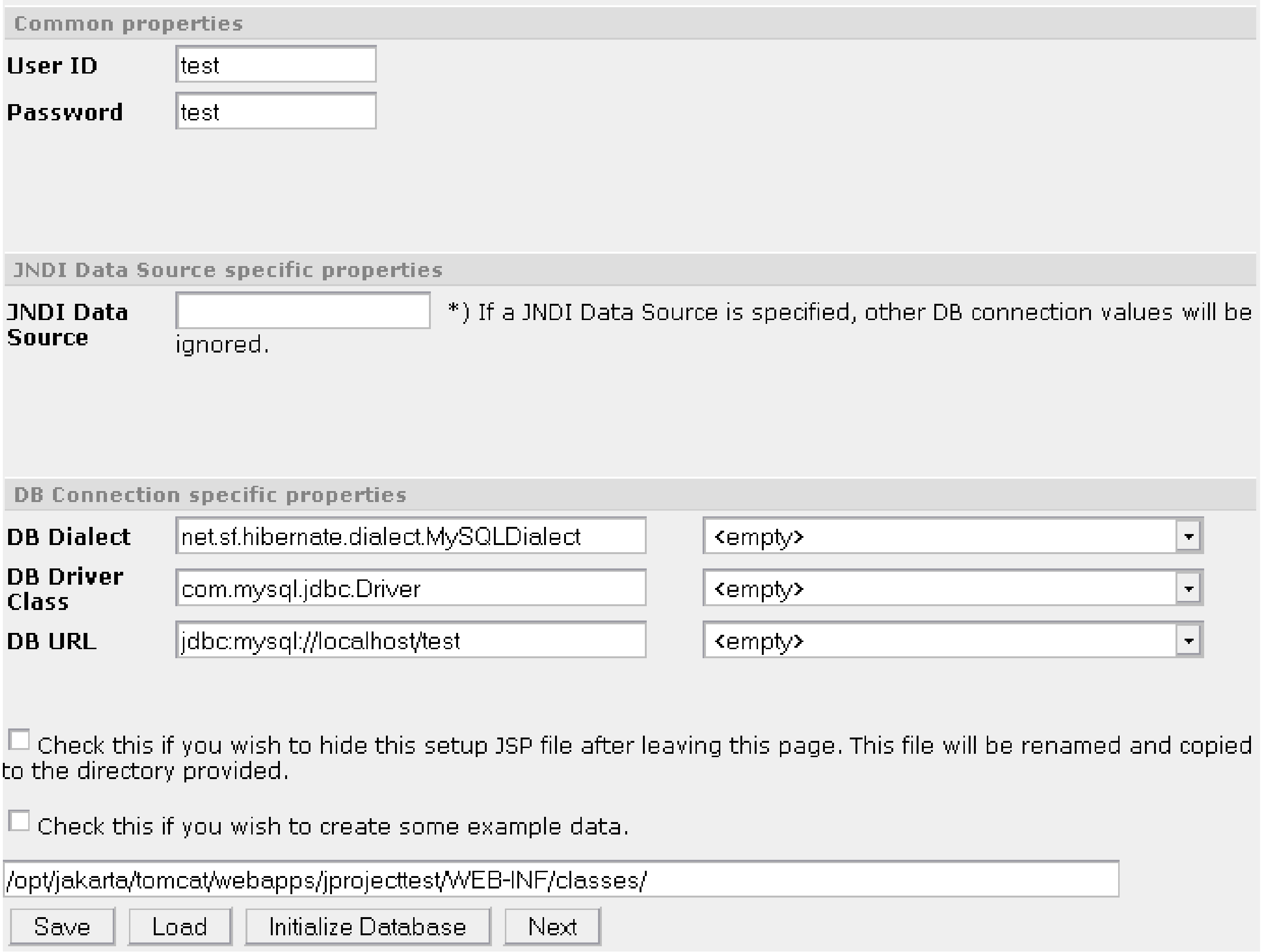
Property name	**Purpose**
`policy.provider`	Define the policy provider if needed. If you are using JDK 1.3 the standard application policy de.jcon.jproject.jaas.BasePolicy should be not declared as the policy provider. This policy only works from JDK 1.4 on.
`jproject.plugin.dir`	Path to the plugin directory relative to the web content directory. Specify this, if you have problems with the automatic find process of the application.
`report.compile.dir`	Path to the report directory where the compiled reports should be placed. Specify this, if you have problems with the automatic find process of the application.
`ldap.login.username`	LDAP user
`ldap.login.pwd`	LDAP password
`ldap.context.factory`	specifies the context factory used to access the server
`ldap.provider.url`	the URL to connect to
`ldap.search.context`	an optional context for defining an additional filter
`ldap.mapping.attr.uid`	the LDAP attribute used for uid verification
`ldap.mapping.attr.password`	the LDAP attribute used for password verification
`ldap.mapping.attr.roles`	the LDAP attribute used to get the roles needed for getting the right policy authorization
`person.accessor.factory`	Factory for person accessor. Use the LDAP factory if you want to access or LDAP directory as the pool for your client data. See Section 3.5, “LDAP Accessor config” for more details.
`company.accessor.factory`	NOT SUPPORTED IN VERSION 0.1.2 !! Factory for company accessor. Use the LDAP factory if you want to access or LDAP directory as the pool for your client data. See Section 3.5, “LDAP Accessor config” for more details.
`java.awt.headless`	Some "headless" operating systems (e.g. Linux) have problems by accessing classes from the graphical environment within Java. So this feature got introduced in JDK 1.4. See Section 3.7, “Reporting config” for more details.
`SSHA_SALT`	SSHA "salt" string to beeing applied for SSHA key generation and verification. Note that if you change this string, all SSHA keys stored in your database will not be able to beeing verified anymore.

3.2. Quick setup

1. Setup Section 3.3, “Persistance Layer (Hibernate)”. This is mandatory.
2. Setup Section 3.4, “Login and User Authentication”
3. Setup Section 3.5, “LDAP Accessor config”
4. Setup Section 3.6, “User/Principal Permissions”
5. Setup Section 3.7, “Reporting config”

3.3. Persistance Layer (Hibernate)

Just start the setup page with `http://$yourserver$:$yourport$/jproject/setup.jsp`.



Common properties

User ID	test
Password	test

JNDI Data Source specific properties

JNDI Data Source	*) If a JNDI Data Source is specified, other DB connection values will be ignored.

DB Connection specific properties

DB Dialect	net.sf.hibernate.dialect.MySQLDialect	<empty>
DB Driver Class	com.mysql.jdbc.Driver	<empty>
DB URL	jdbc:mysql://localhost/test	<empty>

□ Check this if you wish to hide this setup JSP file after leaving this page. This file will be renamed and copied to the directory provided.

□ Check this if you wish to create some example data.

/opt/jakarta/tomcat/webapps/jprojecttest/WEB-INF/classes/

1. You have to specify your `userid`, `password` and `DB Dialect`.
2. Specify either your `DataSource` OR
3. Specify your database connection by setting your `DB Driver` and `DB URL`.
4. Hit `Save` to save your property file to your chosen destination. (Standard is `$WEBAPP_HOME/jproject/WEB-INF/classes`). It needs to go in a path, which is available via the `CLASSPATH` settings.

Tip

You might not be able to save to your chosen directory due to the security constraints of your application server. If this happens you need to save the `hibernate.properties` file somewhere else and copy it manually to the origin destination.

You need to save the properties first before you can initialize the database. Make sure that you wait that amount of time your application server needs to reload changed resources. If the automatic reload feature is disabled for your server, you can change the settings and initialize the database only once. Further changes to the persistence setup within the same session, will not being applied until you restart your servlet container!

5. If you wish to initialize your database, hit `Initialize Database`.

   Note

   All tables will be dropped and recreated if you reinitialize your database. To prevent that this might happen by accident, you should remove the `setup.jsp` file after you finished the setup. You can try to do this automatically by selecting the appropriate checkbox. Again, your security settings might prevent you from doing this via an JSP file, so you have to do this manually.

   Check the output on the page if your tables have been successfully created. Should there be an error, check your settings.

An example `hibernate.properties` file might look like this:

```
#Property file generated by jProject Business Suite Setup.
#Thu Mar 11 11:55:00 CET 2004
hibernate.connection.driver_class=com.mysql.jdbc.Driver
hibernate.dialect=net.sf.hibernate.dialect.MySQLDialect
hibernate.jdbc.use_streams_for_binary=true
hibernate.jdbc.batch_size=0
hibernate.query.substitutions=true 1, false 0, yes 'Y', no 'N'
hibernate.query.imports=net.sf.hibernate.test, net.sf.hibernate.eg
hibernate.connection.username=test
hibernate.use_outer_join=true
hibernate.proxool.pool_alias=pool1
hibernate.jdbc.use_scrollable_resultset=true
hibernate.connection.url=jdbc\:mysql\://localhost/test
hibernate.connection.password=test
hibernate.show_sql=false
hibernate.connection.pool_size=1
hibernate.statement_cache.size=25
```

3.4. Login and User Authentication

You might either want to use the internal user verification system or the `Java Authentication and Authorization Service (JAAS)` provided by Java. This service has been included in the JDK 1.4.

Note

The Java Authentication and Authorization Service (JAAS) is a set of APIs that enable services to authenticate and enforce access controls upon users. It implements a Java technology version of the standard Pluggable Authentication. Check http://java.sun.com/products/jaas/overview.html for more details.

If the `global.security.usejaas` is set to `false` (standard), the internal user verification is used for authentication.

3.4.1. Java Authentication and Authorization Service (JAAS) Overview

You can use the JAAS Authentication Service by specifying a `LoginModule`. Several login modules are already shipped with JDK 1.4, e.g.

1. Java Naming and Directory Interface (JNDI)
2. UNIX Operating Environment
3. Windows NT
4. Kerberos
5. Keystore

By using JAAS, you can reuse your already existing user database or verification mechanism. You can even provide your own LoginModule. See http://java.sun.com/developer/technicalArticles/Security/jaasv2/index.html for more details.

An example `jproject.jaas` file might look like this:

```
jprojectJaas {
   de.jcon.jproject.jaas.login.StandardLoginModule required debug=true;
   /*de.jcon.jproject.jaas.login.LDAPLoginModule required debug=true;*/
   /*com.tagish.auth.DBLogin required dbDriver="sun.jdbc.odbc.JdbcOdbcDriver" dbURL="jdbc:odbc:DBLogi
   /*com.tagish.auth.win32.NTSystemLogin required returnNames=true returnSIDs=false defaultDomain="do
   /*com.tagish.auth.FileLogin required debug=true pwdFile="/path/to/passwd";*/
   /*com.sun.security.auth.module.Krb5LoginModule required debug=true useTicketCache=false;*/
  };
```

3.4.2. JAAS Setup

The property `global.security.usejaas` needs to be set to `true` if you want to use JAAS. You specify you login module in the `jproject.jaas` config file. The application comes with the support of two own login modules.

Note that the standard Sun modules:

1. com.sun.security.auth.module.NTLoginModule
2. com.sun.security.auth.module.UnixLoginModule

will not work with this application because the don't use callback handlers to validate application specifics enties of userid/password. They will only validate the user under which context the application is running on.

1. For standard validation against the application user tables use:

   ```
   de.jcon.jproject.jaas.login.StandardLoginModule required debug=true
   ```
2. For validating on LDAP or Active Directory use (see: user documentation for config parameters):

   ```
   de.jcon.jproject.jaas.login.LDAPLoginModule required debug=true
   ```
3. For validating on an NT domain (only available for NT/2000 OS, not XP!) use (see: http://free.tagish.net/jaas/doc.html):

   ```
   com.tagish.auth.win32.NTSystemLogin required returnNames=true returnSIDs=false defaultDomain="domain"
   ```
4. For validating on Linux / Win2000 domain with Kerberos protocol use:

   ```
   com.sun.security.auth.module.Krb5LoginModule required debug=true useTicketCache=false
   ```
5. For validating on database tables use (see: http://free.tagish.net/jaas/doc.html):

   ```
   com.tagish.auth.DBLogin required dbDriver="sun.jdbc.odbc.JdbcOdbcDriver"
   ```

```
dbURL="jdbc:odbc:DBLogin"
```

6. For validating on a file use (see: http://free.tagish.net/jaas/doc.html):

   ```
   de.jcon.jproject.jaas.login.StandardLoginModule required debug=true;
   ```

7. For standard validation against the application user tables use:

   ```
   com.tagish.auth.FileLogin required debug=true pwdFile="/path/to/passwd"
   ```

Note

Note that at the moment only the standard JAAS module has support for user profiles / roles defined within the `policy.xml` file! All other login modules will assign the standard user profile. This will be enhanced during the next releases.

de.jcon.jproject.jaas.login.StandardLoginModule (standard)

Wrappes the standard user verification mechanism within this JAAS module. If JAAS is enabled and this module is specified the user authentication takes place in the same way as JAAS wouldn't be enabled.

de.jcon.jproject.jaas.login.LDAPLoginModule

This login module can be used to access an LDAP Server for user authentication. You need to specify the following parameters in the `jproject.properties` file:

Table 3.2. Configuration properties for LDAP Login Module

Property name	Purpose
`ldap.login.username`	LDAP user
`ldap.login.pwd`	LDAP password
`ldap.context.factory`	specifies the context factory used to access the server
`ldap.provider.url`	the URL to connect to
`ldap.search.context`	an optional context for defining an additional filter
`ldap.mapping.attr.uid`	the LDAP attribute used for uid verification
`ldap.mapping.attr.password`	the LDAP attribute used for password verification
`ldap.mapping.attr.roles`	the LDAP attribute used to get the roles needed for getting the right policy authorization

An example config might look like this:

```
ldap.login.username=cn=admin,dc=jconhome,dc=com
ldap.login.pwd=admin
ldap.context.factory=com.sun.jndi.ldap.LdapCtxFactory
ldap.provider.url=ldap://jconserver/
ldap.search.context=ou=devel,dc=jconhome,dc=com
ldap.mapping.attr.uid=uid
ldap.mapping.attr.password=userPassword
ldap.mapping.attr.roles=employeeType
```

3.5. LDAP Accessor config

You need to set the `person.accessor.factory` property to `de.jcon.jproject.accessor.ldap.LDAPPersonBOAccessorFactory` (Standard is: `de.jcon.jproject.accessor.PersonBOAccessorFactory`) if you want to use the LDAP person accessor. In addition to this, you need to configure this accessor with following properties:

Table 3.3. Configuration properties for LDAP Accessor Module

Property name	**Purpose**
`ldap.person.ldapversion`	LDAP version
`ldap.person.login.username`	LDAP user
`ldap.person.login.pwd`	LDAP password
`ldap.person.context.factory`	specifies the context factory used to access the server
`ldap.person.provider.url`	the URL to connect to
`ldap.person.search.context`	an optional context for defining an additional filter
`ldap.person.create.template`	LDAP entry which is used a template for the creation of new records. This entry needs to be specified if you wish to create new persons/companies.
`ldap.person.mapping.attr.ID`	the LDAP attribute used for id verification; note that this attribute will be changed by the application if you connect to an existing LDAP system once this component will be enhanced by application specific values (e.g. ROLES, etc.)
`ldap.person.mapping.attr.$XYZ$`	Other LDAP attributes which needs to be mapped to an person object.
`ldap.person.mapping.pk.USERID`	Specify the primary key of a person object which is used in relation to other entries in the application database.

An example config might look like this:

```
ldap.person.ldapversion=3
ldap.person.login.username=cn=admin,dc=jconhome,dc=com
ldap.person.login.pwd=admin
ldap.person.context.factory=com.sun.jndi.ldap.LdapCtxFactory
ldap.person.provider.url=ldap://jconserver/
ldap.person.search.context=ou=devel,dc=jconhome,dc=com
ldap.person.create.template=cn=admin,ou=devel,dc=jconhome,dc=com
ldap.person.maximum.fetch=10000
ldap.person.mapping.attr.ID=internationaliSDNNumber
ldap.person.mapping.attr.LASTNAME=sn
ldap.person.mapping.attr.FIRSTNAME=givenName
ldap.person.mapping.attr.BIRTHDAY=birthDate
ldap.person.mapping.attr.CITY=l
ldap.person.mapping.attr.ZIPCODE=postalCode
ldap.person.mapping.attr.COUNTRY=preferredLanguage
ldap.person.mapping.attr.STREET=postalAddress
ldap.person.mapping.attr.POBOX=postOfficeBox
ldap.person.mapping.attr.PHONE1=homePhone
ldap.person.mapping.attr.PHONE2=telephoneNumber
ldap.person.mapping.attr.FAX=facsimileTelephoneNumber
ldap.person.mapping.attr.EMAIL=mail
ldap.person.mapping.attr.MOBILE=mobile
ldap.person.mapping.attr.WEB=pager
ldap.person.mapping.attr.DESCRIPTION=description
#
# not that this  mapping is just an example
# if there is a standard RCF schema for mapping this
# attributes, of course this should be used
#
ldap.person.mapping.attr.BANK_ACCOUNT=departmentNumber
ldap.person.mapping.attr.BANK=businessCategory
ldap.person.mapping.attr.BANK_ID=carLicense
#
# this USERID is part of the primary key and NOT the userid
# of a USERROLE
#
ldap.person.mapping.pk.USERID=destinationIndicator
```

The `company accessor` is setup in the same way.

Note

Note, that the LDAP person and company accessor is still an experimental feature. Although it might be possible to switch the accessors in an already running system (as long as the USERID feature uses the same key) its definitely not recommended. You should decide once you setup your system, which accessor you use. Of course you can export/import data at the same way independent which accessor type you use.

The system might even store some data redundant in database tables, even if you use the LDAP accessor. This has mainly technical reasons and is used for better SQL performance ;-(

3.6. User/Principal Permissions

Permission Config.

Use following values to define permissions:

$application$.create
  create permission

$application$.read
  read permission

$application$.update
  update permission

$application$.delete
  delete permission

$application$.report
  report permission

$application$.*

all permissions

If you want to specify permissions on your own, you have to add all permissions you need.

E.G. A test for a "read" permission will fail if you just specify a "create" permission. You have to add the read permission as well.

Supported permission sections:

1. home
2. logout
3. project
4. uow
5. travel
6. invoice
7. invoiceRec
8. timeRecord
9. car
10. person
11. company
12. role
13. group
14. relation
15. settings
16. user
17. password

Of course there are combinations of this sections which don't make sense at all. However, you can specify profile in a way you like.

For JDK 1.4 the recommended way to define your policies/permissions is via the internal `PolicyProvider` and the `policy.xml` file.

3.6.1. Standard Policy Provider Application - policy.xml

```
<permissions>
        <!-- Principal with role = standard -->
        <principal>
                <class>de.jcon.jproject.jaas.JaasRolePrincipal</class>
                <value>standard</value>
                <permission>
```

```
                        <class>de.jcon.jproject.jaas.StandardPermission</class>
                        <value>home.*</value>
                        <action></action>
                </permission>
                <permission>
                        <class>de.jcon.jproject.jaas.StandardPermission</class>
                        <value>logout.*</value>
                        <action></action>
                </permission>

                ....

                <permission>
                        <class>de.jcon.jproject.jaas.StandardPermission</class>
                        <value>project.read</value>
                        <action></action>
                </permission>
        </principal>

        <!-- Principal with role = admin -->
        <principal>
                <class>de.jcon.jproject.jaas.JaasRolePrincipal</class>
                <value>admin</value>
                <permission>
                        <class>de.jcon.jproject.jaas.StandardPermission</class>
                        <value>home.*</value>
                        <action></action>
                </permission>
                <permission>
                        <class>de.jcon.jproject.jaas.StandardPermission</class>
                        <value>logout.*</value>
                        <action></action>
                </permission>

                ...

                <permission>
                        <class>de.jcon.jproject.jaas.StandardPermission</class>
                        <value>project.*</value>
                        <action></action>
                </permission>
        </principal>
</permissions>
```

3.6.2. Standard Policy Provider Sun - jproject.policy

If you choose the use the standard policy provider shipped with your VM, you might define the principal permission in another way. Check your documentation how to setup your VM with Java security permissions.

E.G. Sun's VM uses properties files to define permissions. Check the programlisting.

```

// principal based - JDK 1.4
grant principal de.jcon.jproject.jaas.JaasRolePrincipal
        "standard" {
   permission de.jcon.jproject.jaas.StandardPermission "home.*";
   permission de.jcon.jproject.jaas.StandardPermission "logout.*";
   permission de.jcon.jproject.jaas.StandardPermission "project.read";
   permission de.jcon.jproject.jaas.StandardPermission "uow.*";
   permission de.jcon.jproject.jaas.StandardPermission "travel.*";
   permission de.jcon.jproject.jaas.StandardPermission "invoice.*";
   permission de.jcon.jproject.jaas.StandardPermission "timeRecord.*";
   permission de.jcon.jproject.jaas.StandardPermission "car.*";
   permission de.jcon.jproject.jaas.StandardPermission "person.*";
   permission de.jcon.jproject.jaas.StandardPermission "company.*";
   permission de.jcon.jproject.jaas.StandardPermission "role.*";
   permission de.jcon.jproject.jaas.StandardPermission "group.*";
   permission de.jcon.jproject.jaas.StandardPermission "relation.*";
   permission de.jcon.jproject.jaas.StandardPermission "settings.*";
};

// principal based - JDK 1.4
// this principal can do everything
grant principal de.jcon.jproject.jaas.JaasRolePrincipal
        "admin" {
   permission de.jcon.jproject.jaas.StandardPermission "home.*";
   permission de.jcon.jproject.jaas.StandardPermission "logout.*";
   permission de.jcon.jproject.jaas.StandardPermission "project.*";
   permission de.jcon.jproject.jaas.StandardPermission "uow.*";
   permission de.jcon.jproject.jaas.StandardPermission "travel.*";
   permission de.jcon.jproject.jaas.StandardPermission "invoice.*";
   permission de.jcon.jproject.jaas.StandardPermission "timeRecord.*";
   permission de.jcon.jproject.jaas.StandardPermission "car.*";
   permission de.jcon.jproject.jaas.StandardPermission "person.*";
   permission de.jcon.jproject.jaas.StandardPermission "company.*";
   permission de.jcon.jproject.jaas.StandardPermission "role.*";
   permission de.jcon.jproject.jaas.StandardPermission "group.*";
   permission de.jcon.jproject.jaas.StandardPermission "relation.*";
   permission de.jcon.jproject.jaas.StandardPermission "settings.*";
   permission de.jcon.jproject.jaas.StandardPermission "user.*";

};

// standard JDK 1.3 - no different roles/principals are supported
grant  {
   permission de.jcon.jproject.jaas.StandardPermission "home.*";
   permission de.jcon.jproject.jaas.StandardPermission "logout.*";
   permission de.jcon.jproject.jaas.StandardPermission "project.*";
   permission de.jcon.jproject.jaas.StandardPermission "uow.*";
   permission de.jcon.jproject.jaas.StandardPermission "travel.*";
   permission de.jcon.jproject.jaas.StandardPermission "invoice.*";
   permission de.jcon.jproject.jaas.StandardPermission "timeRecord.*";
   permission de.jcon.jproject.jaas.StandardPermission "car.*";
   permission de.jcon.jproject.jaas.StandardPermission "person.*";
   permission de.jcon.jproject.jaas.StandardPermission "company.*";
   permission de.jcon.jproject.jaas.StandardPermission "role.*";
   permission de.jcon.jproject.jaas.StandardPermission "group.*";
   permission de.jcon.jproject.jaas.StandardPermission "relation.*";
   permission de.jcon.jproject.jaas.StandardPermission "settings.*";
   permission de.jcon.jproject.jaas.StandardPermission "user.*";
};
```

3.7. Reporting config

The application uses the open source framework `Jasper Reports` as the core reporting system. There are a lot of tools who helps you writing your own reports or adjusting the standard reports shipped with the system. Additional reports can be easily added via the plugin interface (see Chapter 5, *Plugin Development and Configuration*). For additional information how to create new reports please refer to http://jasperreports.sourceforge.net. You can find even some links to good GUI based WYSIWYG report creation tools.

There are some prerequisites that the reporting system is able to run on an application server / servlet container. The Java VM or the operation system itself needs to support the so called `headless environment` feature. Thus the system need to support a graphical environment even if there is no graphical environment (e.g. X-Server) running.

At the moment there are no problems running on Windows, but on Linux. There are two solution to get the reporting system run on Linux:

1. Use JDK 1.4 and switch the headless.property via the jproject.properties file on.

2. If you have to use JDK 1.3, you need to install the PJA Toolkit from Eteks (see http://www.eteks.com/pja/en/).

3.7.1. Setup PJA Toolkit for headless environments (only for JDK 1.3)

1. Add `-Xbootclasspath/a:"$your_path_to$/pja.jar"` to the vm startup option of your servlet container. It's not enough to just adjust the CLASSPATH!!

2. Enable following properties in the Section 3.1, “Project property file [jproject.properties]” setup file:

```
awt.toolkit=com.eteks.awt.PJAToolkit
java.awt.graphicsenv=com.eteks.java2d.PJAGraphicsEnvironment
java2d.font.usePlatformFont=false
user.home=./WEB-Content/Web-inf
```

Chapter 4. National Language Support / Character Encoding

The application is shipped with the support for ENGLISH and GERMAN. Any other language might be easily added by providing a localized `ApplicationResources_$$.properties` resource bundle placed beneath `$WEBAPP/WEB-INF/classes/de/jcon/jproject/resources` where `$$` stands for the language code.

Change the property.languages property of the setup file to enable your language for the application.

You can even specify country specific languages, e.g. en_US, en_UK.

Chapter 5. Plugin Development and Configuration

The application provides an plugin interface. Plugins are shipped as an zipped file usually named like the java package name + the version (e.g. `de.jcon.jproject.reports.timerecord_0.0.1.zip`).

5.1. How to write a plugin?

You have to provide a main plugin class which implements the `de.jcon.jproject.plugin.JProjectPlugin` interface. Check the Java Doc of this class for additional information.

Your plugin class together with all depended classes have to be packed in an standard `JAR` file. This file and the plugin deployment descriptor needs to be packed together in the above mentioned ZIP archive.

Your plugin archive structure should therefore look like this example:

```
[de.jcon.jproject.reports.timerecord_0.0.1.zip]             // zip archive
   |
   -- [de.jcon.jproject.reports.timerecord_0.0.1]           // directory
         |
         -- [de.jcon.jproject.reports.timerecord_0.0.1.jar] // jar file
               |
               -- ...                                       // plugin class structure
               -- [jProjectPlugin.xml]                      // deployment descriptor
               -- [pluginsrc.zip]                           // optional sources
```

The deployment descriptor should look like:

```
<plugin>
   <!-- Name of the plugin -->
   <name>Project Time Report</name>
   <!-- Plugin class -->
   <class>de.jcon.jproject.reports.timerecord.ProjectTimeReport</class>
</plugin>
```

The `Plugin Manager` will install the plugins once use start the application automatically. The `initialize()` method is called during startup. Any initialization needed should be placed there.

5.1.1. Report Plugins

The report plugin provides a way to add additional reports automatically to the system. In addition to the standard plugin needs, a report plugin consists of following items (the same example as above is used):

ProjectTimeReport class

The report class. You need to extend the `de.jcon.jproject.reports.Report`.

ProjectTimeReport XML Jasper Report File

This is the Jasper Report file. This file needs to get compiled that it can be used with in the system. This is needed only once and can be achieved by the compile link within the application.

Note

If you changed a report xml file or a report has been new installed, you need to compile this report. You need to have the compile permission to be able to compile reports (e.g. `travel.compile.report`, `timeRec.compile.report`, `...`).

ProjectTimeReport_XX properties

Language property files to support internationalization. This properties should be used (although the are not mandatory).

ProjectTimeReportDataSource class

If you need an own DataSource (this is not an standard Java DataSource but an Jasper Report class!) to access your data, you can ship this with your report.

report.xml

Report deployment descriptor.

```
<report>
        <name>ProjectTimeReport</name>
        <description>Standard Time Record Report</description>
        <!-- e.g. invoice, travel, timeRecord, person, company, ... -->
        <category>timeRecord</category>
        <datasource>de.jcon.jproject.reports.timerecord.ProjectTimeReportDataSource</datasource>
        <file>ProjectTimeReport.xml</file>
        <!-- e.g.  category.'report'.name -->
        <permission>timeRecord.report.projectTimeReport</permission>
</report>
```

5.2. How to install a plugin?

You just need to copy your plugin ZIP archive to the plugin folder defined in your Section 3.1, “Project property file [jproject.properties]” file (Property: property.plugin.dir). The plugin gets automatically installed during the next server start.

Note

At the moment you still need to "compile" the report provided by the report plugin manually via the application feature itself.

Chapter 6. Export / Import Subsystem

> **Note**
>
> The export / import functionality described below will be available from version 0.2 on.

The application uses the openadaptor™ Framework (http://www.openadaptor.org) as the provider for their Import / Export functionality. There are internal `Sinks` (see Section 6.1, “ openadapter Framework”) defined for the file system and for an JMS server.

At the moment following functionality is provided by this subsystem:

1. Import of travel items from an PDA (Palm)
2. ... others are pending

6.1. openadapter™ Framework

> **Note**
>
> All the description and example pictures in this section are used from the Open Adapter [http://www.openadaptor.org] documentation. They are just used to give a rough overview of this framework. All rights are reserved byopenadapter™.

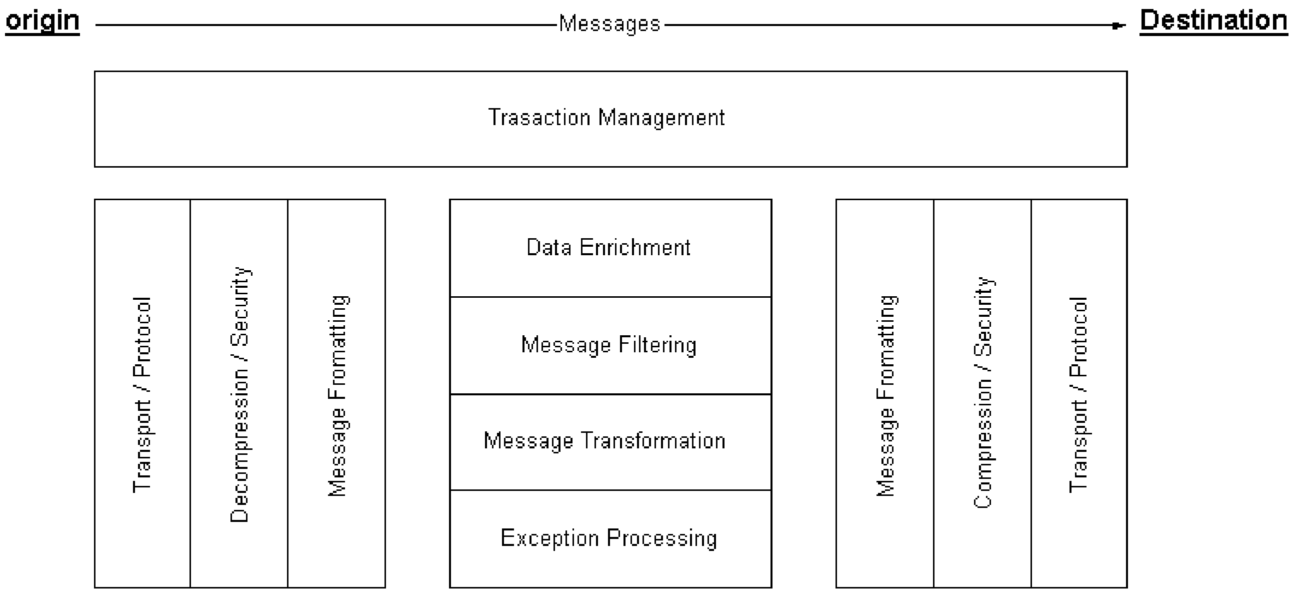
Theopenadapter™ framework is a message-based system integration toolkit. System integration work can be as simple as processing a file or as complicated as building real-time feeds between systems.

The framework abstracts the process of sending messages between systems (or transports). It provides a framework of "ready built" components that can be quickly assembled to integrate systems (by using simple configuration files rather than by writing actual program code).

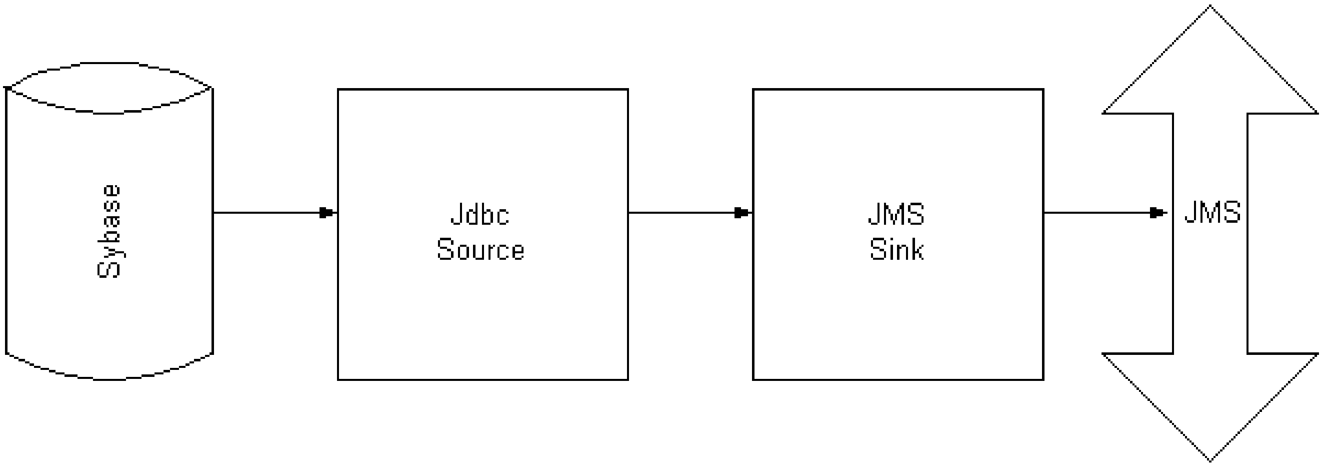
The framework currently allows for the construction of adaptors (this framework will also be extended to allow for the construction of message-based services and gateways).

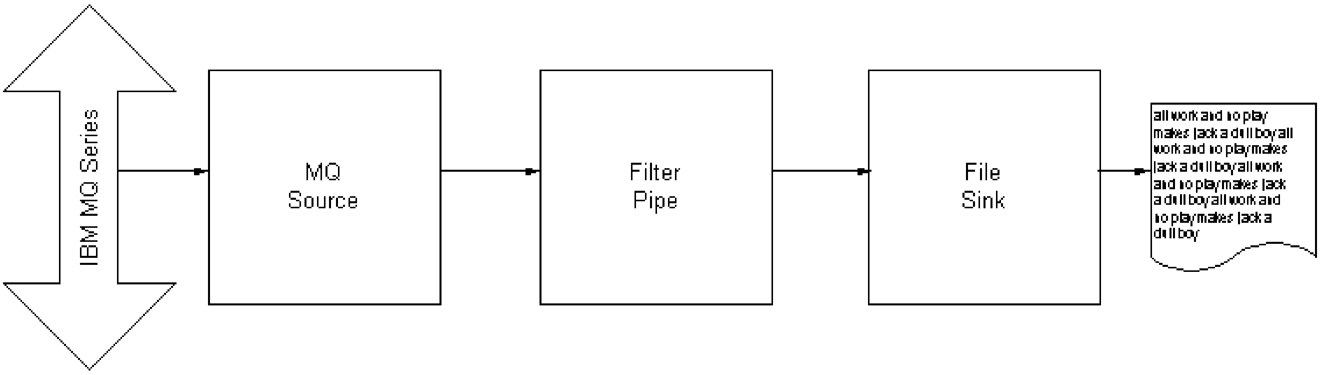
An adaptor is a one way link between one or more origins and one or more destinations. The origins and destinations could be many different things, e.g., flat files, TCP/IP sockets, databases, specific system APIs, middleware, RMI services, JMS queues, etc.

The different stages of the work required to send a message is illustrated below:



An adaptor is a single process - each Source component is run as a separate thread of execution. Below are examples of very simple adaptors:




IBM MQ Series
MQ Source
Filter Pipe
File Sink
all work and no play makes Jack a dull boy all work and no play makes Jack a dull boy all work and no play makes Jack a dull boy all work and no play makes Jack a dull boy

6.2. Configuration

FixMe: [to be done]

Chapter 7. Business Functions

The application provides some standard business functionality. You can manage your business travels, customer projects, time records according to your project definition, development groups, customer contacts and other. You can even easily create an invoice on the fly just from released project tasks. And all printouts are fully customizable through Jasper Report development tools.

Just have a look at two examples:

Example 7.1. Working as a single contractor or on a freelance base.

You are working on customer projects at the client side. You are contracted by another company and work for their customers. You have following business needs:

1. Keep track on your project work both on the customer side (often exists an project management or ERP system) and on the contractors side.
2. Sometimes you can get a report from the customer which you need to being confirmed by the customer so that you can get to invoice those recorded times to your contractor. Sometimes the contractor got its ERP system and you need to do it there again or the don't have anything and you are free to provide this on your own.

   At least you have to keep all synchronized and at the end you need to summarized all your work for your invoice.
3. Sometimes you need to invoice expenses as well or you need for other reasons to summarize your business travels by car.

You can all do this with this application. All you need is to get this application online so you are able to access it on all sides. You still have to possibly maintain your customer/contractor specific project management tools, but you can parallel record your projects time with your application.

If you are working on many projects throughout a calculation period you can do this for all projects within one application. Your use cases might look like:

1. Create or access a company record for your customer and your agency [once]
2. Create or import a project and specify project tasks [once]
3. Record your project times on a regularly base [daily/weekly]

   Its planed that this can be done with a PDA as well
4. Record your business travels either online or directly in your car on your PDA [daily]
5. Create a report at the end of your calculation period from your time records to let it sign by the customer if needed.
6. Release your time records.
7. Create an invoice from your selected and released time records.

Example 7.2. A midrange company works as a contractor/agency for other customers

Following business needs might be possible:

1. You have to manage your customer projects and the development of this projects.
2. You need to collect your project times from persons working on this project.
3. You need to invoice those times and other items.

This might be achieved by completing following tasks:

1. Create or access a company record for your customer [once]
2. Create or access person records for your employees, give them a user role so they can record their project times with the system[once]
3. Create or import a project and specify project tasks [once]
4. Assign persons to the project group [once]
5. Let the project group members record their project times on a regularly base [daily/weekly]

   Its planed that this can be done with a PDA as well
6. Let the project group members record their business travels online or via PDA [daily]

   Note

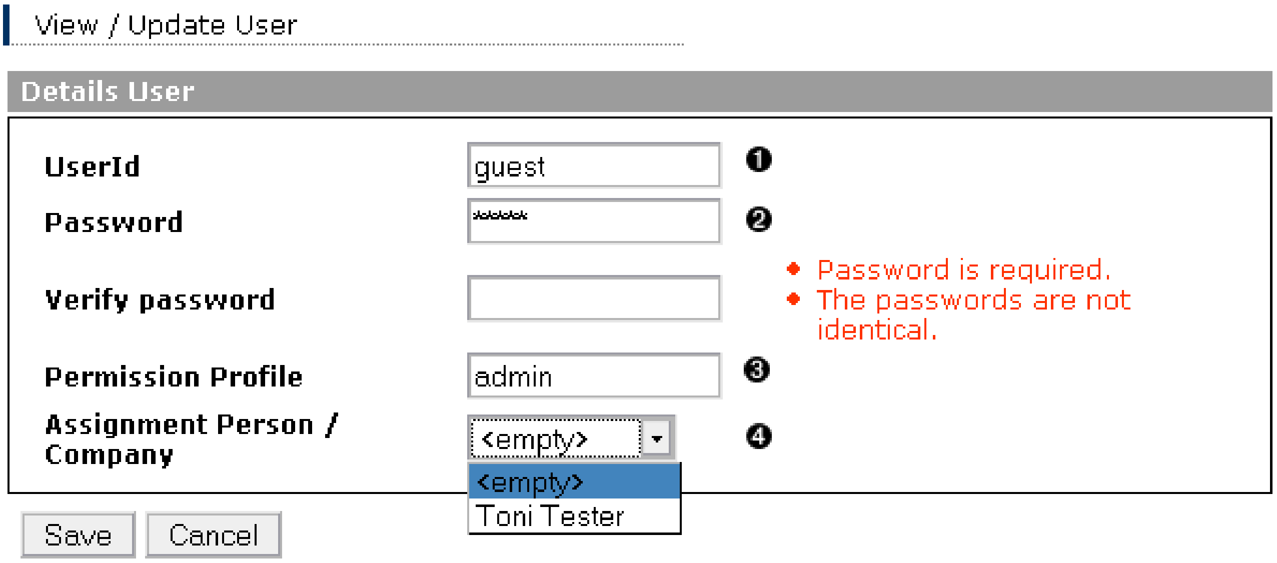
   In one of the next version you can setup your bluetooth export/import task. So, if you got this part of the application installed in your location, the time records and/or business travels can be automatically collected from the PDAs and imported into the system once your employee entries the location. All you need are PDAs, mobiles or smartphones which are bluetooth capable and a server which acts as a bluetooth host.
7. Let the project group members release their time records.
8. Create an invoice from the released time records.

7.1. User management

Use this subsystem to manage your users.

Note

You can only manage your users with this application if you do not use JAAS or if you use the standard JAAS authentication module (see Section 3.4, “Login and User Authentication”), shipped with the application. By using any other login module (LDAP, Kerberos, ...) any changes made with this part of the application regarding userid and password will not being reflect during the login process.



View / Update User

Details User

UserId	guest	❶
Password	*****	❷
Verify password		• Password is required. • The passwords are not identical.
Permission Profile	admin	❸
Assignment Person / Company	<empty>	❹

❶ The userid. You can change to userid everytime.

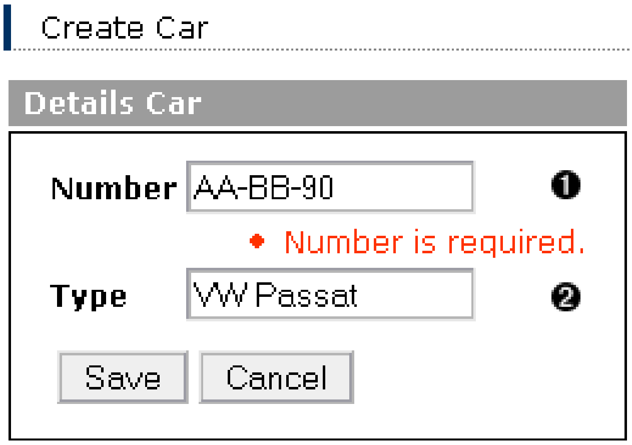
❷ Password

❸ This value needs to be a permission config defined in Section 3.6, “User/Principal Permissions”

❹ You might assign a person or company component to this user. You will only see entries from the user itself or entries which have a visibility of `protected` or `public`.

7.2. Car management

Use this subsystem to manage your cars.



Create Car

Details Car

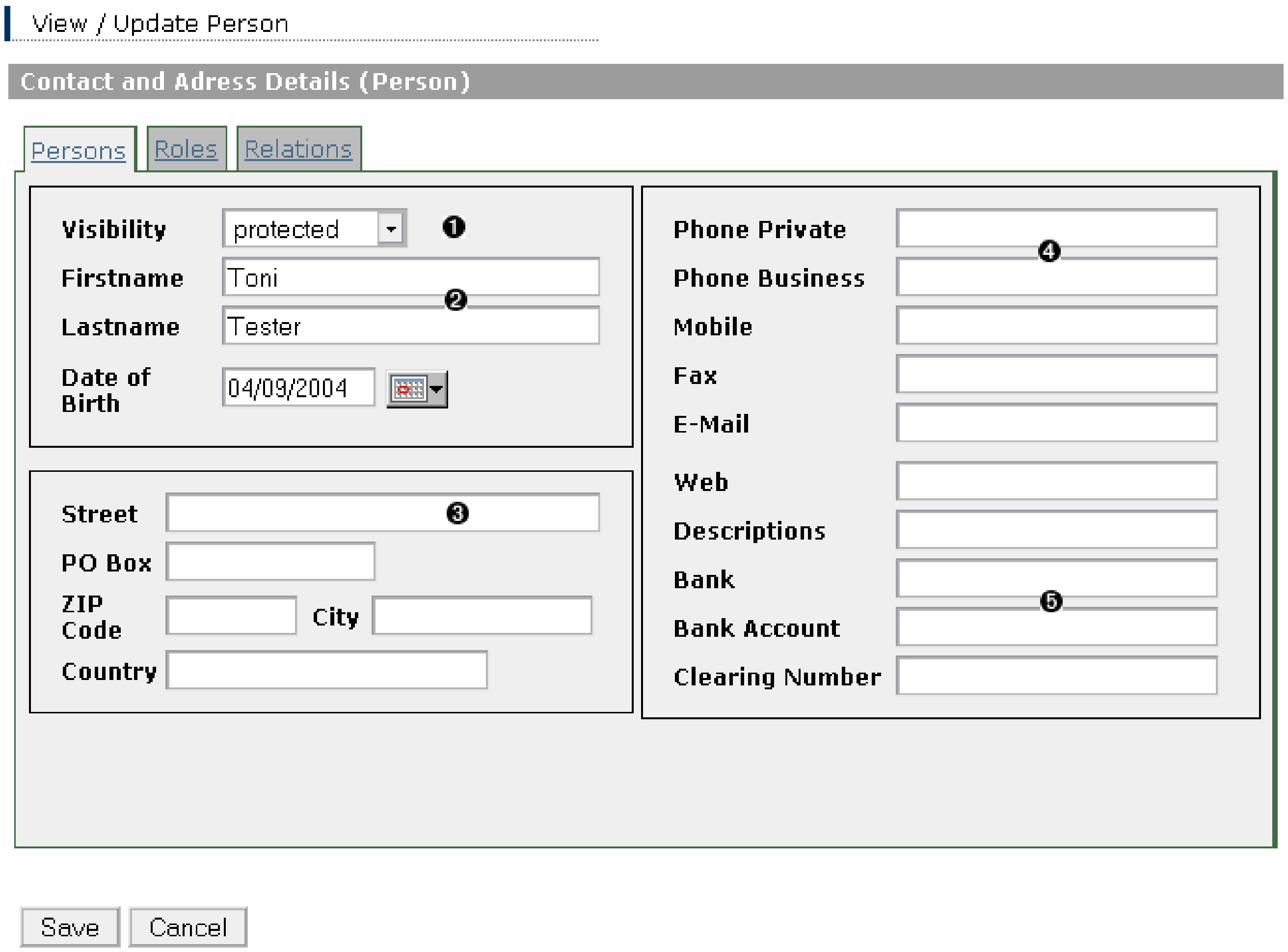
Number	AA-BB-90	❶
	• Number is required.	
Type	VW Passat	❷

❶ The number of the car.

❷ The type or brand of the car

7.3. Person / Company management

Use those subsystems to manage your persons/companies.



View / Update Person

Contact and Adress Details (Person)

Visibility	protected
Firstname	Toni
Lastname	Tester
Date of Birth	04/09/2004

❶ The visibility of this component.

Table 7.1. Component Visibility

Visibility	Impact
`private`	Only the owner (=who created this component) can see and change this component.
`user`	Only the owner and the user associated with this component (=component user role) can see and change this component.
`protected`	All users can see this component, but only the owner can change it.
`public`	All can see and change this component.

Please note that there is no locking done by the system. Refer to Section 8.1, “Locking Concepts” for more information.

❷ General data

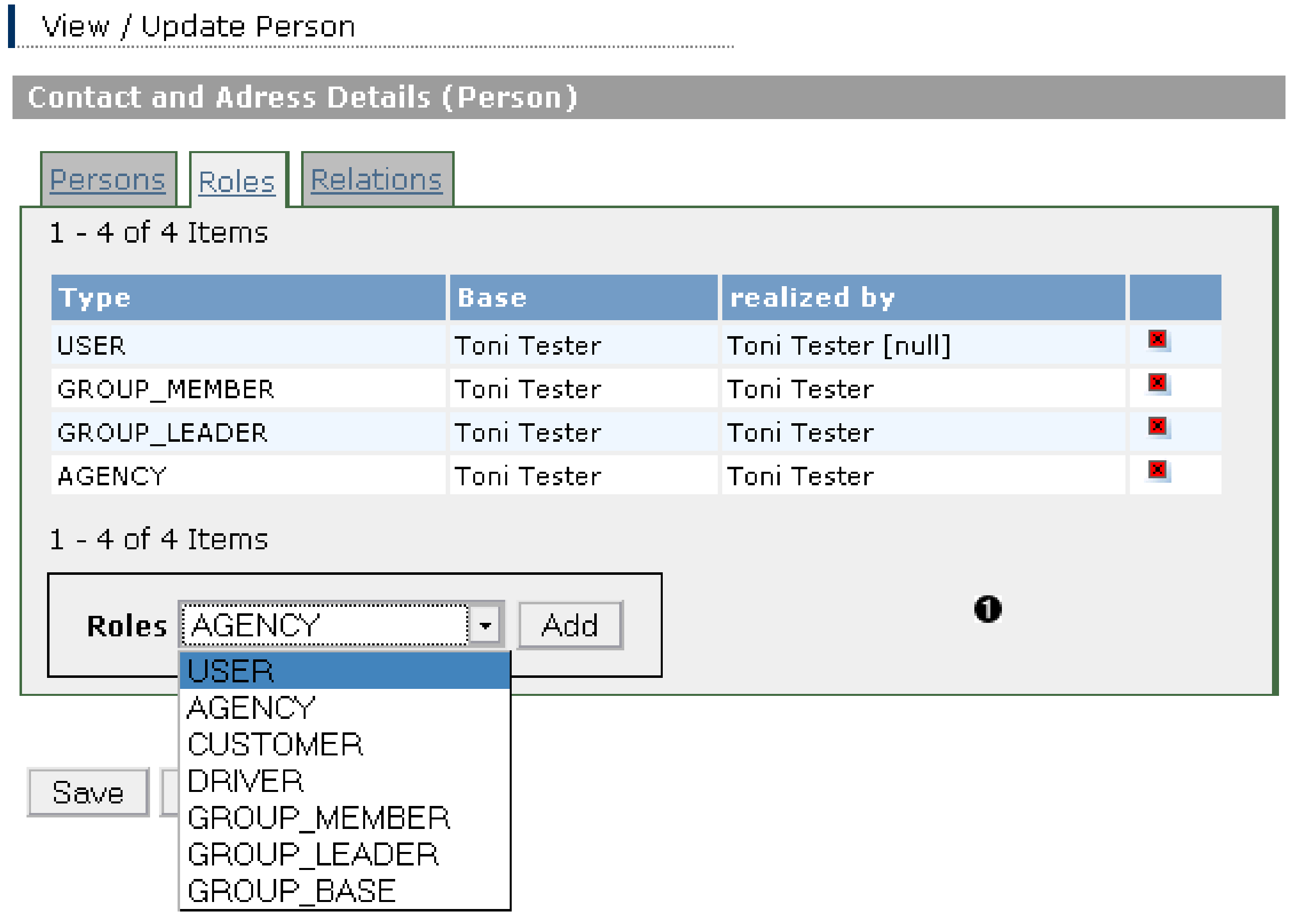
❸ Adress information. You need to provide this if you want to use the invoice reports.

❹ Contact information

❺ Bank information.

7.3.1. Roles

You need to specify roles if this component has to get into relations to other components. If the component can not play a specific role, it will not be possible to assign this component to others.



View / Update Person

Contact and Adress Details (Person)

1 - 4 of 4 Items

Type	Base	realized by	
USER	Toni Tester	Toni Tester [null]	☒
GROUP_MEMBER	Toni Tester	Toni Tester	☒
GROUP_LEADER	Toni Tester	Toni Tester	☒
AGENCY	Toni Tester	Toni Tester	☒

1 - 4 of 4 Items

Roles: AGENCY

❶ The roles this component can "play".

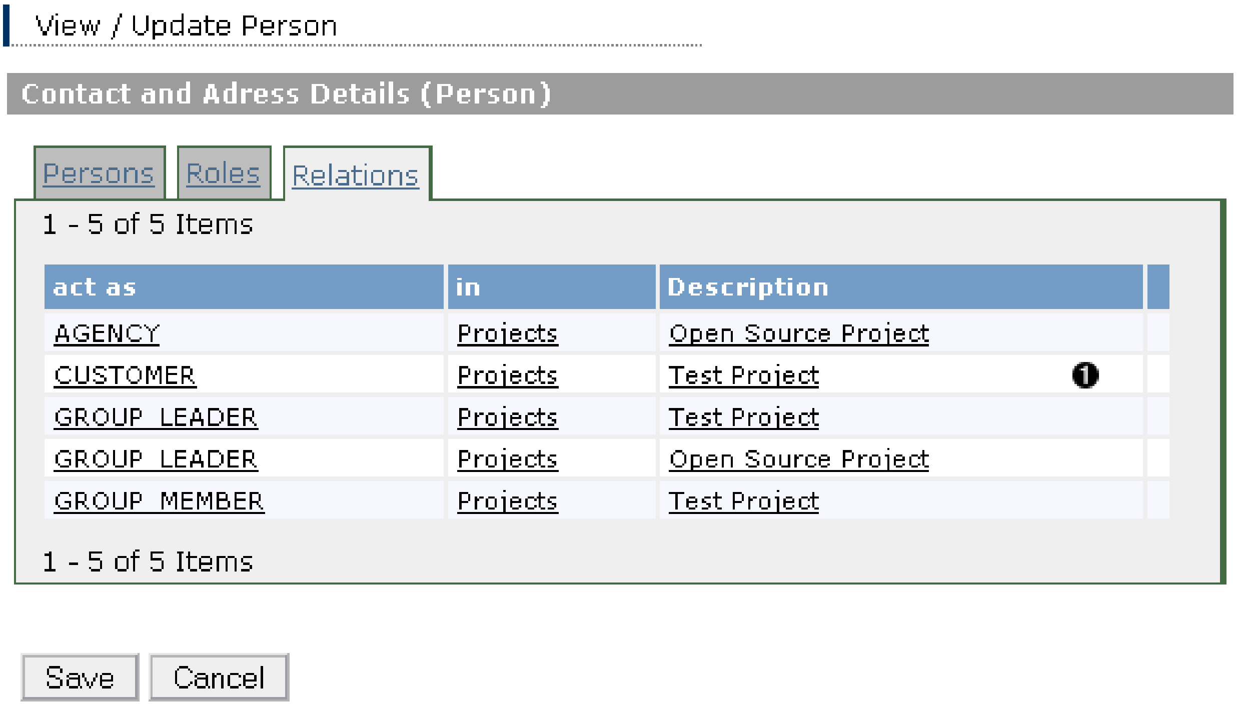
Table 7.2. Role Types

Type	Impact
`GROUP_MEMBER`	This component can become a group member (at them moment supported by: person, company, project)
`GROUP_LEADER`	This component can become a group leader or sub leader (at them moment supported by: person, company, project)
`GROUP_BASE`	This component can manage groups (at them moment supported by: projects)
`USER`	This component can act as an user (at them moment supported by: person, company, project). The standard values of userid and password if you create a new user role are equally to the name of the component.
`CUSTOMER`	This component can act as a customer (at them mo-

Type	Impact
	ment supported by: person, company, project). It will be shown in the project customer list.
`AGENCY`	This component can act as an agency (at them moment supported by: person, company, project). It will be shown in the project agency list.

7.3.2. Relations

This page shows all relations of the component.



1 - 5 of 5 Items

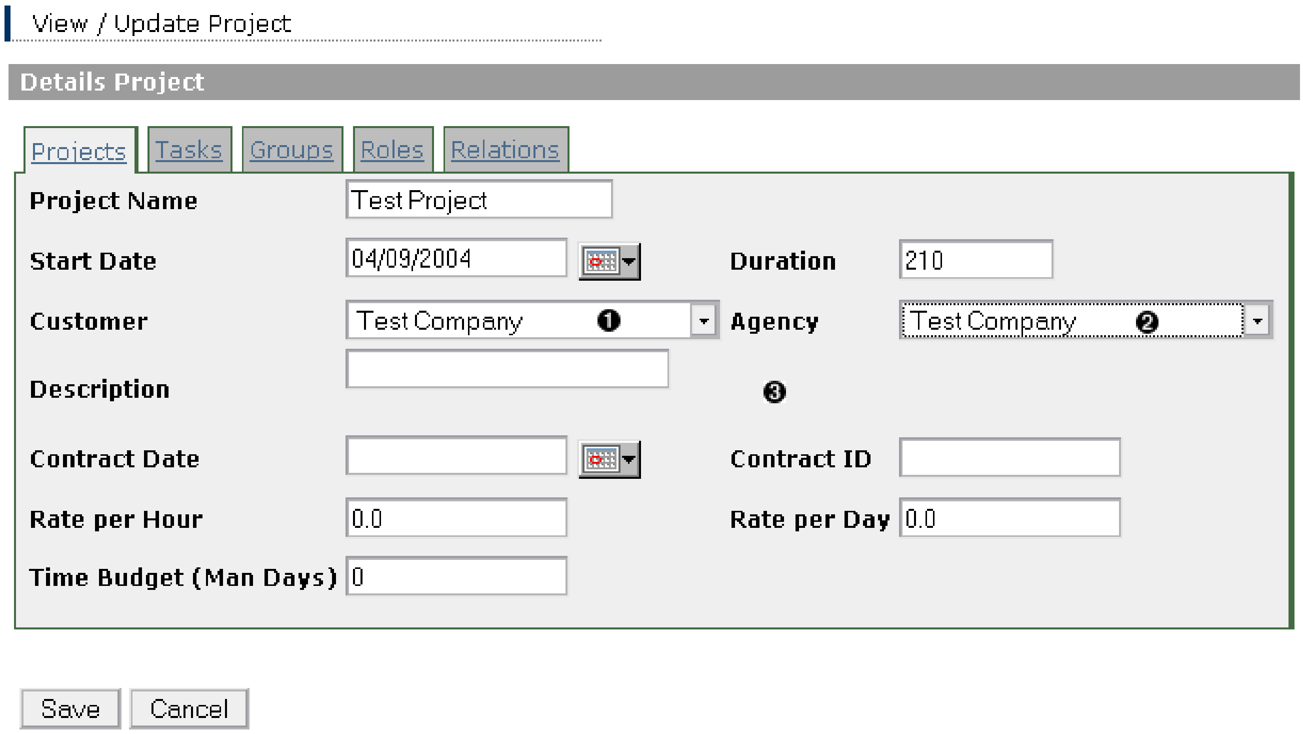
act as	in	Description
AGENCY	Projects	Open Source Project
CUSTOMER	Projects	Test Project
GROUP_LEADER	Projects	Test Project
GROUP_LEADER	Projects	Open Source Project
GROUP_MEMBER	Projects	Test Project

1 - 5 of 5 Items

❶ All relations the component has. As long as a component has relations to other component, it can't be deleted.

7.4. Project management

Use this subsystem to manage your projects.



Project Name	Test Project		
Start Date	04/09/2004	Duration	210
Customer	Test Company ❶	Agency	Test Company ❷
Description		❸	
Contract Date		Contract ID	
Rate per Hour	0.0	Rate per Day	0.0
Time Budget (Man Days)	0		

❶ The customer of this project. You can choose components which can play the customer role.

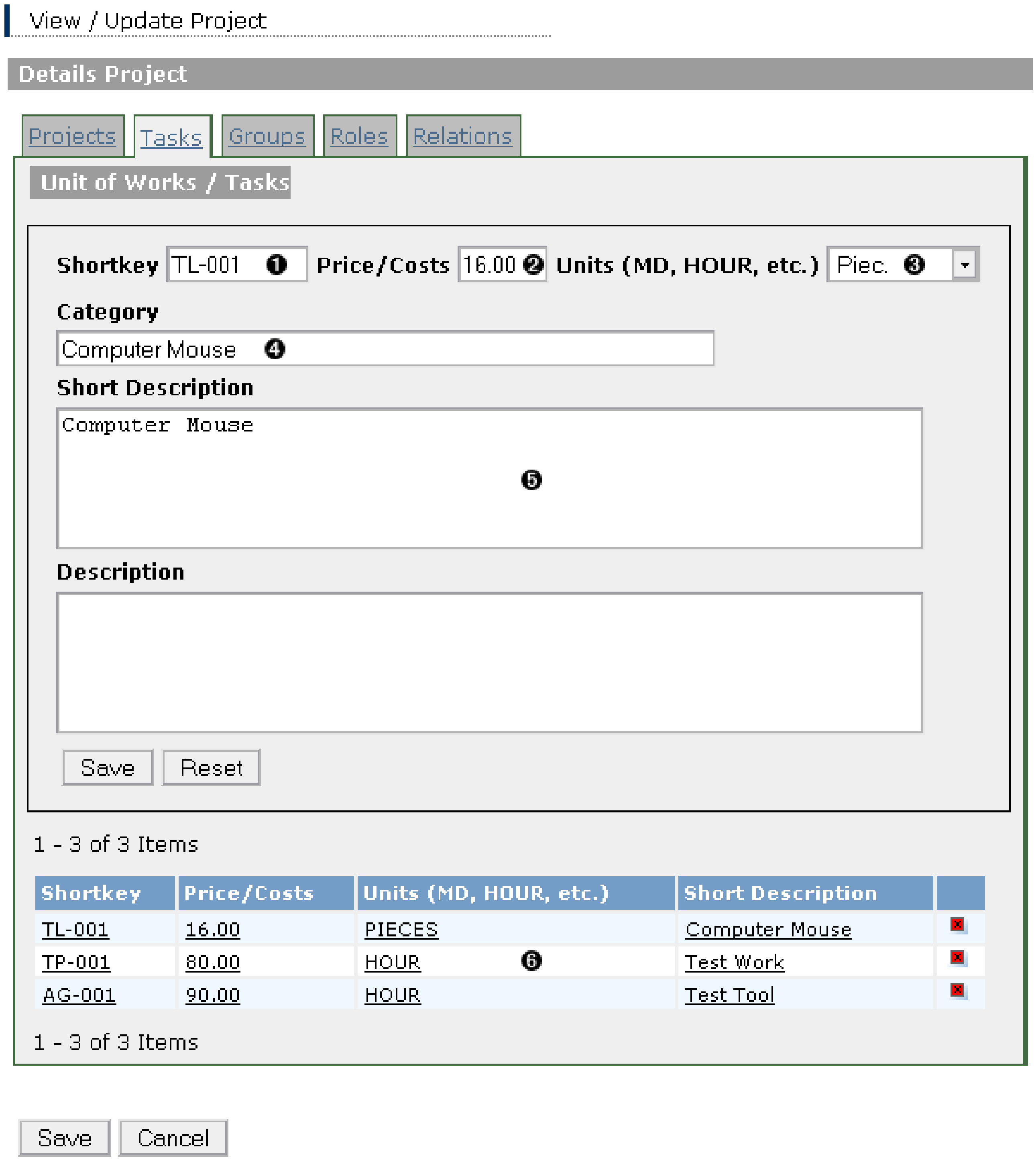
❷ The agency of this project. You can choose components which can play the customer role.

If an agency is specified its address will be used for the invoice report, otherwise the customer address will be used.

❸ General project information.

7.4.1. Tasks

A project consists of tasks.



View / Update Project

Details Project

Unit of Works / Tasks

Shortkey TL-001 ❶ Price/Costs 16.00 ❷ Units (MD, HOUR, etc.) Piec. ❸

Category

Computer Mouse ❹

Short Description

Computer Mouse ❺

Description

1 - 3 of 3 Items

Shortkey	Price/Costs	Units (MD, HOUR, etc.)	Short Description	
TL-001	16.00	PIECES	Computer Mouse	☒
TP-001	80.00	HOUR ❻	Test Work	☒
AG-001	90.00	HOUR	Test Tool	☒

1 - 3 of 3 Items

❶ A free format key you can specify for the tasks. This is not a primary key, so two task might have the same shortkey value although this is not really recommendable.

❷ The "price" of this task.

The number will be used for calculation of the total sum if somebody performs this task.

❸ The unit of this task. This unit is shown on the invoice.

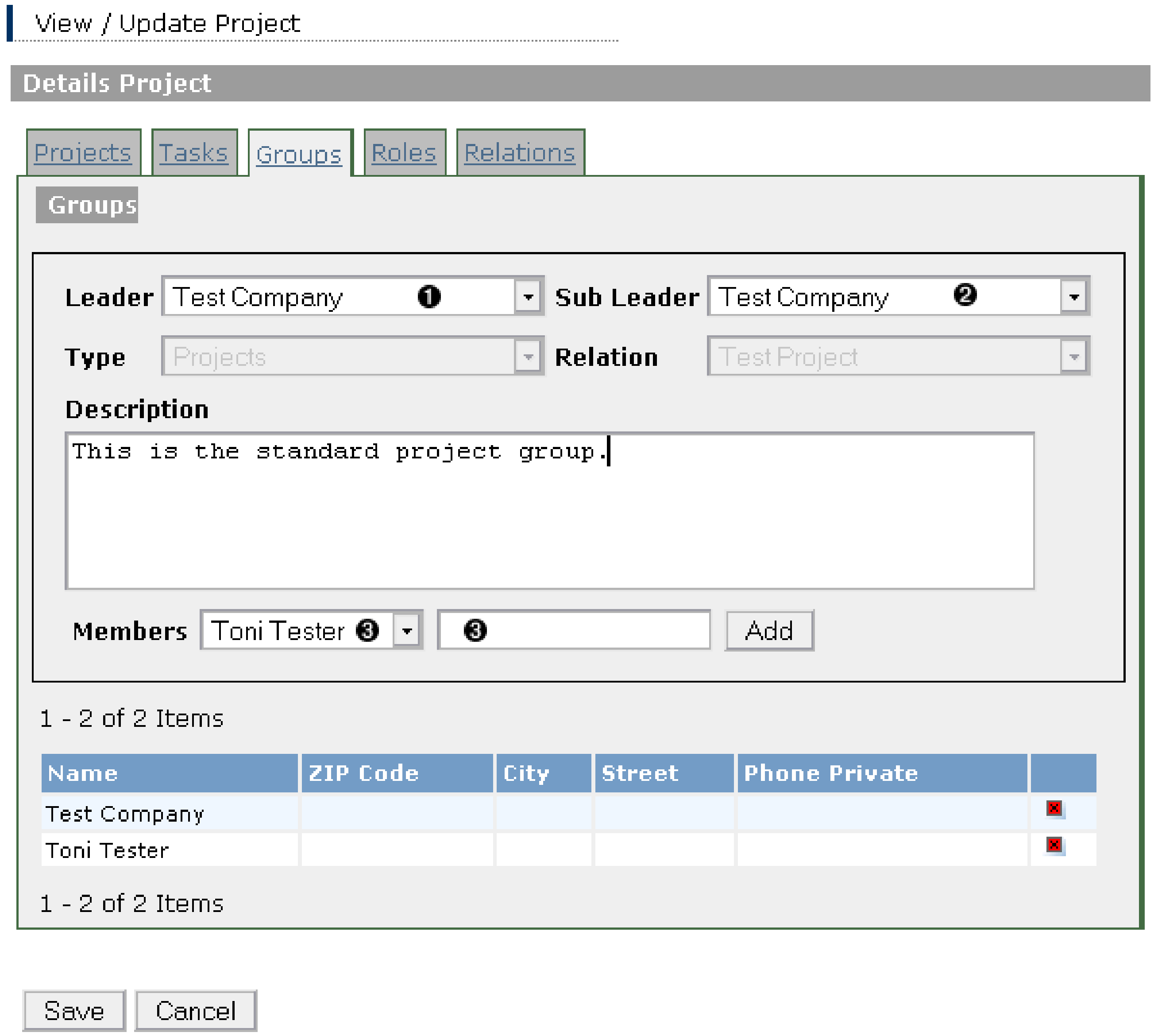
❹ An optional category. This category can be shown on the invoice as well.

❺ Descriptions of the task.

❻ Tasks for this project.

7.4.2. Groups

You can enable user to report time records to you (if you are the manager of an project) by assigning them to the project group. By doing this they will be able to see this project on their project selection list for time records.



View / Update Project

Details Project

Groups

Leader	Test Company ❶	**Sub Leader**	Test Company ❷
Type	Projects	**Relation**	Test Project

Description

```
This is the standard project group.
```

Members Toni Tester ❸ ❸ Add

1 - 2 of 2 Items

Name	ZIP Code	City	Street	Phone Private	
Test Company					☒
Toni Tester					☒

1 - 2 of 2 Items

❶ The leader of this project. You can choose components which can play the group leader role.

❷ The sub-leader/sub-agent of this project. You can choose components which can play the group leader role.

❸ Choose components which can play the group member role and add them to this group.

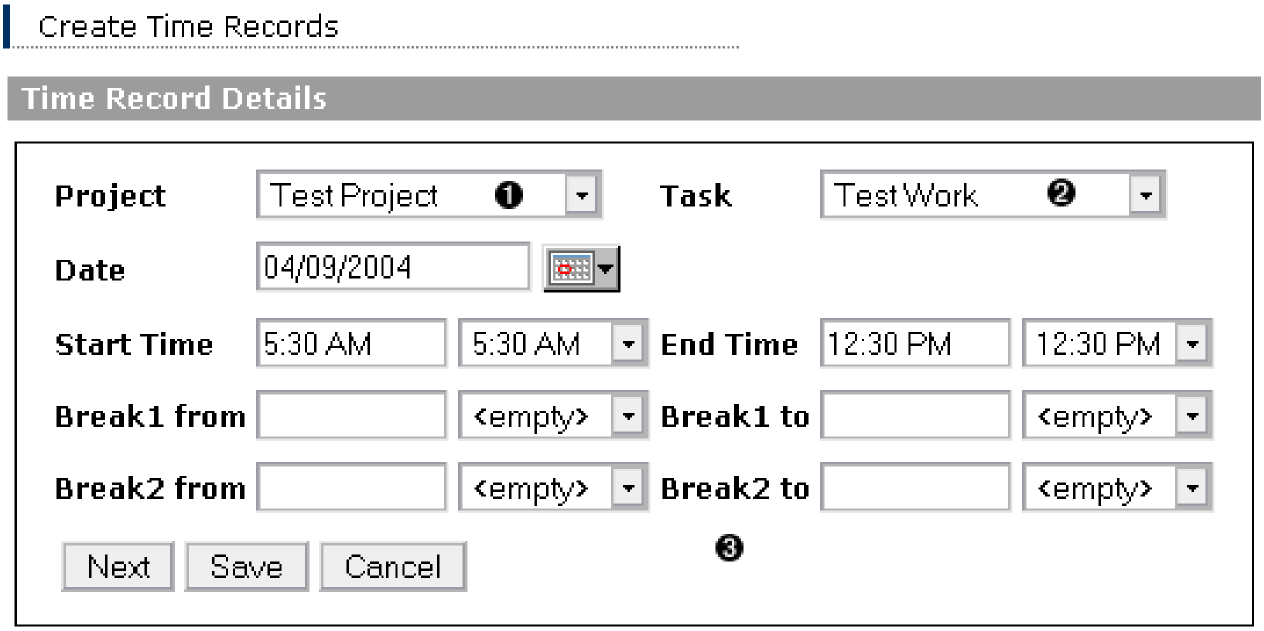
Within the dropdown list you can only choose components which are visible for you. However you might allow other components who can act as a group member to being added to this project group. The only need is that the have to have a user role as well.

If an entry is selected in the drop down list, this entry will be added to the list no matter what stands in the textfield.

If nothing is selected and you have specified a valid system user who can play the group member role, this user is added to the group member list by pressing the `Add` button.

7.5. Time Record management

You can manage the times, project members have needed to work with/for certain tasks.



Create Time Records

Time Record Details

Project: Test Project

Task: Test Work

Date: 04/09/2004

Start Time: 5:30 AM 5:30 AM

End Time: 12:30 PM 12:30 PM

Break1 from: <empty>

Break1 to: <empty>

Break2 from: <empty>

Break2 to: <empty>

❶ Select the project for which you want to assign this time record.

❷ Select the task to which you want to assign this time record.

❸ Time record properties to define. A net and a gross sum will be calculated out of your values which is used for time record reports and invoices.

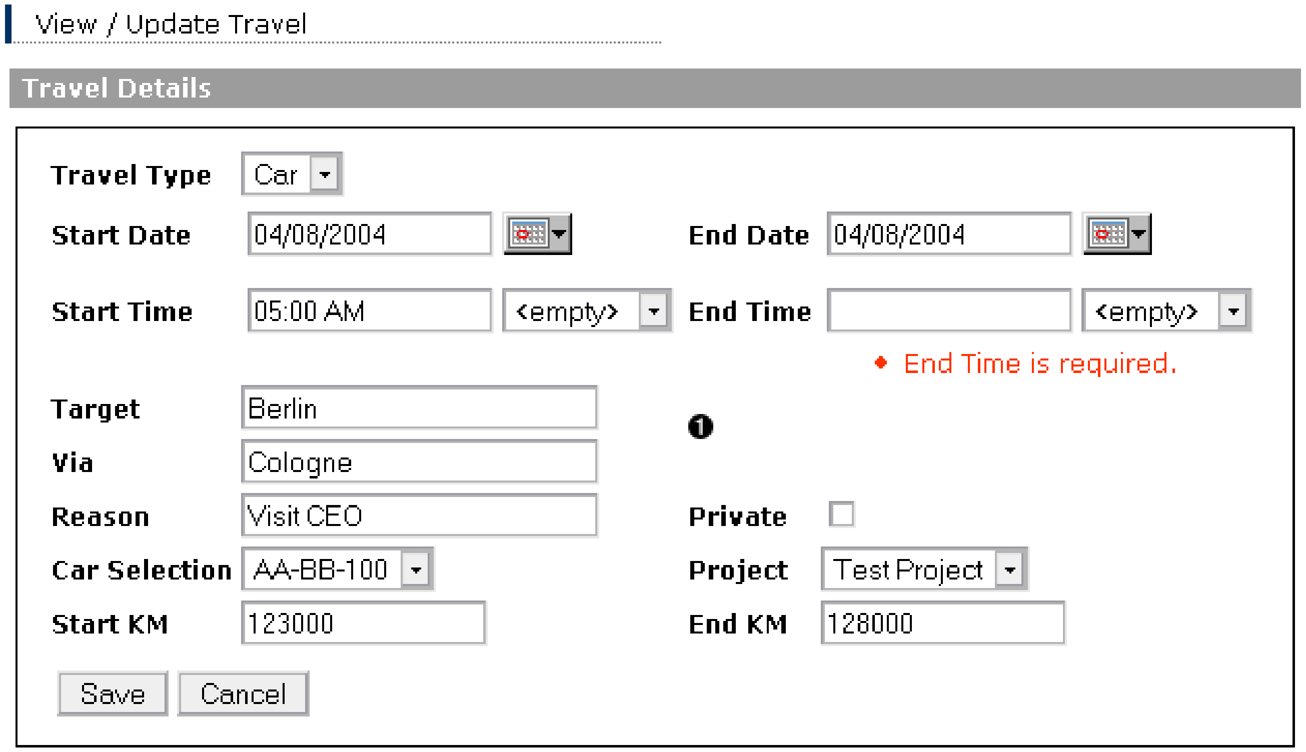
A time record runs through certain states in the order `[created] -> [released] -> [cleared] -> [deleted]`

Table 7.3. Time Record States

State	Impact
`created`	This is the initial state of a time record. As long as a record has this state it can be changed.
`released`	If you release time records the can be used to invoice. If a user A releases time record to foreign projects, they can't be changed anymore by this user. The project manager B of the project can choose to accept them or to revoke them. Once he revoke the time records they will be set back to the `created` state and user A will be able to change them again. Time records released to "own" project tasks can be changed even in this state.
`cleared`	The state a time record has once it is used in an invoice. It can't be changed anymore.
`deleted`	Fix me: Not used.

7.6. Travel management

Use this subsystem to manage your business travels



View / Update Travel

Travel Details

Travel Type: Car

Start Date: 04/08/2004 | End Date: 04/08/2004

Start Time: 05:00 AM <empty> | End Time: <empty>

◆ End Time is required.

Target: Berlin

Via: Cologne

❶

Reason: Visit CEO | Private: □

Car Selection: AA-BB-100 | Project: Test Project

Start KM: 123000 | End KM: 128000

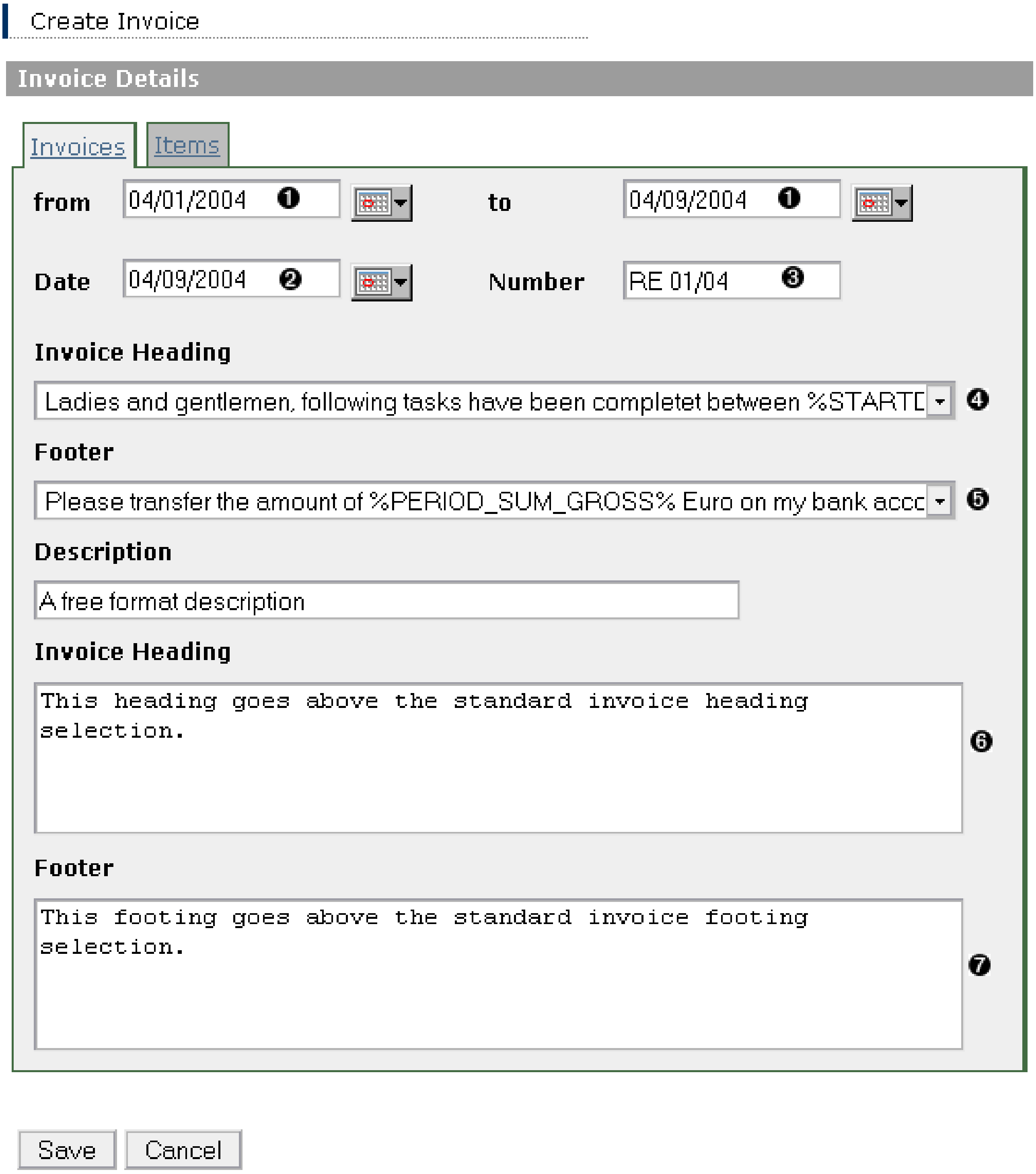
❶ Details for a private or business travel. You might choose a car or a project for assigning this travel.

7.7. Invoice management

Use this subsystem to manage your invoices. You can create invoices either directly from the time record list page or from the invoice list page.

If you create an invoice form the time record page, all released time records shown in the list will be automatically added to this invoice. This is unfortunately the only way you can add time records to an invoice. At the moment you can't even add time records to allergy existing invoices. This will be possible during the next releases.

If you create an invoice from the invoice list page, you can add certain amounts to specified project tasks. No time records can be added.



Create Invoice

Invoice Details

from 04/01/2004 ❶ to 04/09/2004 ❶

Date 04/09/2004 ❷ Number RE 01/04 ❸

Invoice Heading

Ladies and gentlemen, following tasks have been completet between %STARTD ❹

Footer

Please transfer the amount of %PERIOD_SUM_GROSS% Euro on my bank acco ❺

Description

A free format description

Invoice Heading

This heading goes above the standard invoice heading selection. ❻

Footer

This footing goes above the standard invoice footing selection. ❼

❶ If you want to cover a certain period in your invoice you can specify this here. You can use this values as a variable reference in your heading and footing variables.

❷ The invoice date.

❸ The invoice number.

❹ This value is used within the invoice report

Select a pre-defined heading (variable name is `HEADING`). You can manage this headings in your settings. (See Section 7.8, “Settings”)

❺ This value is used within the invoice report

Select a predefined footing (variable name is `FOOTING`). You can manage this footings in your settings. (See Section 7.8, “Settings”)

❻ This value is used within the invoice report

A free format header.

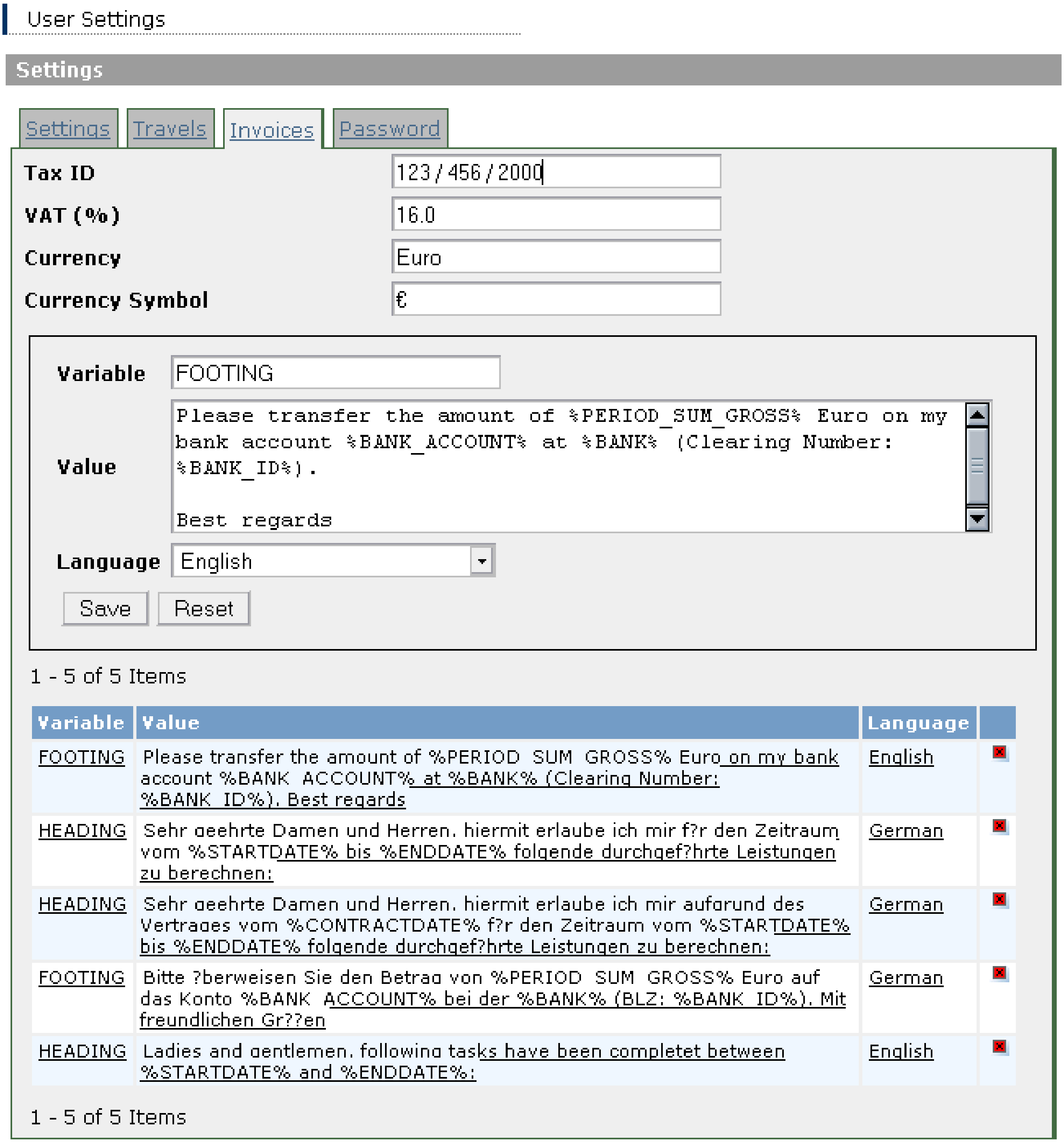
❼ This value is used within the invoice report

A free format footer.

The standard value of your currency is determined by your language settings. If you specified e.g. de_DE the currency will be EURO and all invoices will be calculated and shown in EURO. If you specify en_US, the currency will be DOLLAR, etc. The standard invoice report uses this values.

However the `Standard Invoice report with additional currency` uses the currency values specified in your user settings.

7.8. Settings



User Settings

Settings

Tax ID	123 / 456 / 2000
VAT (%)	16.0
Currency	Euro
Currency Symbol	€

Variable	FOOTING
Value	Please transfer the amount of %PERIOD_SUM_GROSS% Euro on my bank account %BANK_ACCOUNT% at %BANK% (Clearing Number: %BANK_ID%). Best regards
Language	English

1 - 5 of 5 Items

Variable	Value	Language	
FOOTING	Please transfer the amount of %PERIOD_SUM_GROSS% Euro on my bank account %BANK_ACCOUNT% at %BANK% (Clearing Number: %BANK_ID%). Best regards	English	☒
HEADING	Sehr geehrte Damen und Herren, hiermit erlaube ich mir f?r den Zeitraum vom %STARTDATE% bis %ENDDATE% folgende durchgef?hrte Leistungen zu berechnen:	German	☒
HEADING	Sehr geehrte Damen und Herren, hiermit erlaube ich mir aufgrund des Vertrages vom %CONTRACTDATE% f?r den Zeitraum vom %STARTDATE% bis %ENDDATE% folgende durchgef?hrte Leistungen zu berechnen:	German	☒
FOOTING	Bitte ?berweisen Sie den Betrag von %PERIOD_SUM_GROSS% Euro auf das Konto %BANK_ACCOUNT% bei der %BANK% (BLZ: %BANK_ID%). Mit freundlichen Gr??en	German	☒
HEADING	Ladies and gentlemen, following tasks have been completet between %STARTDATE% and %ENDDATE%:	English	☒

1 - 5 of 5 Items

❶ Variable management.

Settings

You can changed the layout for this user. A css file with this name needs to be found in the `theme` directory.

Chosse the language for this user.

Travels

You can specify the standard directory for your PDA import.

Note

This feature is not supported until version 0.2

Invoices

Manage variables you can use for the invoice reports.

The variables `HEADING` and `FOOTING` will be shown in the appropriate drop down lists on the invoice page. All other variables can be referenced by `%$VARIABLE_NAME%` in other variables or reports.

Password

You can change your password on this page.

Chapter 8. Technical issues

8.1. Locking Concepts

From version 0.2 on there will be a business object based locking. Version 0.1.x has no locking concept implemented.

Note

Note that there is for reason no optimistic, pessimistic or logical locking of data. So if the same objects (projects, persons, companies, ...) for one user are requested for change (e.g. if more user share the same user id like in the test application or public objects are shared within the application), the first will "win", the other will be notified, that the object has been changed by another user.

8.2. Database problems

There are databases which don't support transactions and foreign key constraints.

Its strongly recommended that you use transactions and support for foreign key constraints. The application relies on the referential integrity of the database to stay in a consistent stage. If there is no support by the database the application might get corrupt or inconsistent (because table entries will be able to delete which still are foreign keys in other tables)

8.2.1. MySQL and transactions

MySQLs standard table type MYISAM has no support for transactions and foreign key constraints (at least up to version 4.x). You have to use the INNODB table type by either defining `default-table-type=INNODB` as the standard table type or by explicitly setting the table type in the `create table ..` statement.

If you can't change the standard table type property you have to manually create the tables as follows:

1. Use the setup page as described in Section 3.3, “Persistance Layer (Hibernate)”.
2. If the `Initialize Database` action was successful you will see the SQL statements on the page. Just copy them and add at the end of each table statement `TYPE=InnoDB`.
3. Go the MySQL console or any other program where you can execute SQL statements and execute all of them (drop, create, alter table).

8.2.2. MySQL and automatic reconnect

In some circumstances you might get `SocketExceptions` using MySQL. This is a common issue of the standard driver properties from the MySQL JDBC driver.

You can solve this problem by usinge the `autoReconnect` property within your Hiberate configuration.

```
DB Url: jdbc:mysql://localhost/test?autoReconnect=true
```

8.2.3. MySQL and UTF-8 encoding

If you encounter problems with values read from the database and "special characters" e.g. german "Umlaute" you need to add addtitional parameters to your JDBC url for MySQL.

The standard encoding for this application is UTF-8. MySQL/ JDBC JConnector has problems with storing UTF-8 values. Try to specify the encoding directly in the JDBC URL, via the `useUnicode` and `characterEncoding` MySQL JDBC driver properties:

```
DB Url: jdbc:mysql://localhost/test?autoReconnect=true&useUncicode=true&characterEncoding=ISO-8859-1
```

Chapter 9. Further Development

Following developments are planned for the 0.2 release

1. Initial implementation of the openadaptor™ framework including the proper import from PDA travels.
2. Enable the already existing PDA program for recording travels to work with the new application export / import structure.
3. Let time records being able to being add to already existing invoices (?? maybe this will be not released due to tax revision reasons ??).
4. Support for user roles / profiles for all JAAS login modules (at the moment only the standard login module supports this).
5. Performance enhancements.