TrackStudio 2.7.5 Documentation

Copyright (c) 2002-2003. TrackStudio, Ltd. All rights reserved.

Table of Contents

1 Concepts 1

2 What's New 4

3 Installation 5

3.1 Requirements 5

3.2 TrackStudio/SA Configuration 6

3.2.1 TrackStudio Server Manager 7

3.2.2 Windows Service Management 9

3.2.3 How to Install an SSL Certificate. 10

3.3 TrackStudio/WAR Configuration 11

3.3.1 Configuration Files 12

3.4 Database Configuration 14

3.4.1 HypersonicSQL Configuration 14

3.4.2 PostgreSQL Configuration 15

3.4.3 Oracle Configuration 15

3.4.4 DB2 Configuration 16

3.4.5 MS SQL Configuration 16

3.4.6 Firebird Configuration 17

3.5 UNIX-Specific Notes 17

3.6 Update from 2.5 18

3.7 Debugging Errors 19

4 Getting Started 21

4.1 Creating a New User 22

4.2 Creating a New Project 23

4.3 Creating a New Bug 25

5 Task Management 28

5.1 Task Concepts 29

5.2 Subtasks 32

5.3 Filters 33

5.3.1 View Filter 34

5.3.2 Edit Filter 35

5.3.3 Filter Subscription 38
5.3.4 Notify by Email 39
5.4 Task 42
5.4.1 Edit Task 42
5.4.2 Customize Task 43
5.4.3 Export 44
5.5 Messages 45
5.5.1 Submit by Email 47
5.6 Uploads 48
5.7 Access Control 48
5.8 Reports 49
5.8.1 Report Types 49
5.8.2 List Reports 50
5.8.3 View Reports 51
5.9 Workflow Management 51
5.9.1 Categories 52
5.9.2 Edit Category 53
5.9.3 Workflows 54
5.9.4 Priorities 55
5.9.5 States 56
5.9.6 Message Types 57
5.9.7 Transitions 58
5.9.8 Customize Workflow 60
6 User Management 62
6.1 User Concepts 63
6.2 List of Users 64
6.3 User 64
6.3.1 Edit User 64
6.3.2 Change Password 65
6.3.3 Customize User 65
6.4 Status 67
6.4.1 List User Status 67
6.4.2 Edit User Status 68

7 Open API 70

7.1 Adapter Development 70

7.2 Adapter Structure 71

7.3 Building 74

8 Index 75

TrackStudio 2.7.5 Documentation

1 Concepts User

TrackStudio is one of the most powerful and scalable Java-based defect and issue management software applications.

Description

When developing TrackStudio, we tried to use as few objects and concepts in the system as possible and attempted to realize the necessary functionality through enhancing the objects already existing in the system. This synergy allowed us to create a system that's powerful, yet easy-to-use-and-understand.

The simplicity of the design and the clear concepts of the system are reflected in the user interface. We tried to create a logical user interface where you cannot get lost. Whatever the users do, they are always fully aware of the current state of the system and of the actions available to them.

We attempted to create a user interface that can comfortably manage a large number of tasks with least waste of time. We understand the problems that can arise when managing hundreds of users and projects, and we try to find the most effective and simple solutions to them.

We did our best to optimize the user interface for those people who spend a lot of time working with the system daily. You won't find here any multi-page wizards for creating a user or a task, because each of these operations can be performed with one mouse click wherever you are in the system. Creating complex objects (for example, a workflow) can be stopped and resumed at any step.

Let's look at the main objects, defined in the TrackStudio system and their purpose.

Object	Description
Task	A task is any object whose state is to be tracked. The flexibility of TrackStudio allows our customers not only to track the state of bugs and feature requests, but also to keep records of the equipment, to track the process of making advertising materials. The system supports a task hierarchy, which means that projects can contain subprojects and the latter can contain issues. This makes the user interface much easier: if you know how to create a project, you know how to create bugs. If you can create a workflow and set security rules for a certain project, it means you can create workflows and set security rules for project groups and even for issues. All the created objects (filters, workflows, custom fields) can be inherited, which means you do not have to declare the same custom fields and workflows for each new project. It is enough just to create a new project and it will automatically inherit all the properties common for this project group. You can also enhance the inherited objects to make them fit the specific project.
User	A user is the key object in the system. As for tasks, TrackStudio supports a hierarchy for users, which makes it possible to use the system effectively in medium sized and large organizations. The system allows transferring a part of an authority or even the entire authority to the subordinate managers without losing control over the system. Subordinate managers can have the same rights and privileges for their parts of a project as the project manager does over the whole project. These lower-level managers can use shared project rules, user roles, task categories, custom fields, and workflows, and can modify them according to their specific needs. The project manager has access to all the information and can create reports, and analysis relating to the project as a whole or to individual elements - all with only a few clicks of the mouse.
Workflow	The behavior of any task in the system is defined through the workflow. A workflow allows rules to be set for changing the states of a task in the system. A workflow consists of states and transitions while the transition matrix defines the description of the transitions.
Message	A message is used for changing the state of a task. The available types of messages are defined by the workflow. You can enter some additional information when saving a message. For example, with just a few mouse clicks, John, a developer, can mark the task as finished, set its resolution, specify the time spent on the task and type in an explanatory comment. An important feature of the system is that it allows creating messages to be created via e-mail using a very simple interface, similar to that used in web forms.
Category	Categories define the type of a task and are used to create links between tasks and workflows. When creating a category, you can specify its possible subcategories and user groups that can create and delete tasks of that type. For example, you can specify that only a manager can create tasks of the project type, or that a bug cannot have a project as its subtask.
User Status	A user status represents a user group. Every user group can have a certain set of privileges.

Filter	Filters are very important objects in the system. Not only do they allow you to search for tasks meeting the specified criteria, but they are also used to adjust e-mail notifications. As with all the other objects, filters and e-mail notification rules are inherited, which saves time configuring filters for each project. In TrackStudio every user can set their personal rules for e-mail notification. For example, they can specify that in the project group A they are interested only in the messages from the customer, in the project B they are interested only in those tasks where they are a handler, and they are interested in all activities in the issue C. Additionally, the user can receive e-mail notifications either each time a change that meeting the specified criteria occurs in the object, or at regular intervals. For example, a manager can receive a daily report containing all the information about the project entered into the system during the previous day.
Report	Applying a filter results in a list of tasks. Another way of getting a summary about a group of tasks is by using reports. For example, you can obtain information about the tasks created by every developer, or what percentage of user requests have been successfully fulfilled. The reports can be generated in the form of HTML, PDF, CSV, XML or XLS.
Custom Field	Custom fields allow you to enlarge the list of fields available for the system objects. Custom fields are not object-specific, so you can create them for tasks, users and even workflows.

2 What's New User

This topic describes some new features of TrackStudio.

Description

TrackStudio Enterprise is one of the most flexible Java-based issue tracking systems. It supports the widest range of commercial and open source application servers and DBMSs, and can be run on any computer platform. The following new features are introduced in version 2.7:

1) New **Open API** based on microkernel architecture is realized in TrackStudio 2.7. It means that there is a small kernel responsible for both the interaction of components and the system configuration, while practically all the system functionality is based on adapters with well-defined interfaces. Adapters can be arranged in a pipeline when the result of some method is passed over to the next adapter in the pipeline. The application functionality can be changed or enhanced without modifying the initial code of TrackStudio through using the adapters and arranging them in a pipeline.

2) TrackStudio 2.7 has a new **filter-based email notification system**. For instance, a users can indicate that they want to get e-mail notifications only for high-priority tasks and only if they are handlers for that task. One project can have a number of e-mail notification schemes. Users can use e-mail notification schemes created by other users and inherit them from other available projects. You can configure it in such a way that users will not get e-mail notifications on every change of the task, but the list of all changes in tasks/projects for a certain period of time will be sent to them at regular intervals.

3) TrackStudio 2.7 allows filtering the tasks not only by their parameters, but also by the message parameters (bug notes).

4) TrackStudio 2.7 has the **Bulk Edit tool**, which allows posting comments, changing handlers or task status for several tasks at a time, including those located in different projects.

5) TrackStudio 2.7 has a new JasperReports-based report generator allowing the creation list/detail/distribution and timesheet reports. Reports can be displayed as HTML, PDF, XLS, CSV and XML documents.

6) TrackStudio 2.7 is greatly enhanced regarding the **control over workflows, task categories and custom fields**.

7) TrackStudio 2.7 has new online and PDF **documentation**.

3 Installation Administrator

This section contains installation and configuration notes.

3.1 Requirements Administrator

This section describes hardware and software requirements.

Description

TrackStudio Enterprise requires the following system configurations:

Equipment	Requirement
Server hardware	**Minimum requirements:** Pentium III or equivalent, 600 MHz or higher, 256 MB memory; 1 GB operational disk space **Recommended requirements:** Pentium IV or equivalent, 1.6GHz or higher, 512 MB memory; 2 GB operational disk space
Server operating system	**One of the following:** Microsoft Windows NT/2000/XP Linux Sun Solaris Hewlett Packard HP-UX IBM AIX
Database server	**One of the following:** ORACLE 8i, 9i IBM DB2 8.1 MS SQL Server 2000 Borland Interbase 6.5 Firebird 1.0 PostgreSQL 7.3.1 HypersonicSQL 1.7.2

Application server	**One of the following:** BEA Weblogic 7.0 IBM WebSphere 5 Sun ONE Application Server 7 Tomcat 4.x JBoss 3.0.x Jetty 4.1 Caucho Resin 2.1.7
JDK	**One of the following:** Sun JDK 1.3.1 Sun JDK 1.4.x IBM JDK 1.3.1 BEA JRockit 3.1.5 BEA JRockit 7.0
Web Browser	**One of the following:** Microsoft Internet Explorer, version 5.0 to version 6.0 Mozilla 1.1
Email Client	**One of the following:** Microsoft Outlook Express, version 4.0 to version 6.0 Microsoft Outlook 98,2000,XP The Bat!

3.2 TrackStudio/SA Configuration Administrator

This section describes TrackStudio Enterprise configuration (Standalone distribution).

Description

You must have JRE 1.3.1 or higher installed on your system to run TrackStudio Enterprise Standalone (you don't need JDK for TrackStudio Enterprise Standalone). To install and configure the software you should perform the following steps:

1. Run HypersonicSQL (**hsql.exe**).

```
Opening database: test
HSQLDB server 1.7.2 is running
Use SHUTDOWN to close normally. Use [Ctrl]+[C] to abort abruptly
Mon Jun 09 09:11:18 MSD 2003 Listening for connections ...
```

2. Run TrackStudio Enterprise Server Manager (**sman.exe**).

3. Press the **Start** button and run the server. When the server is running, press the **Open Window** button.

4. Use the following to log in the system: **login=root, password=root**

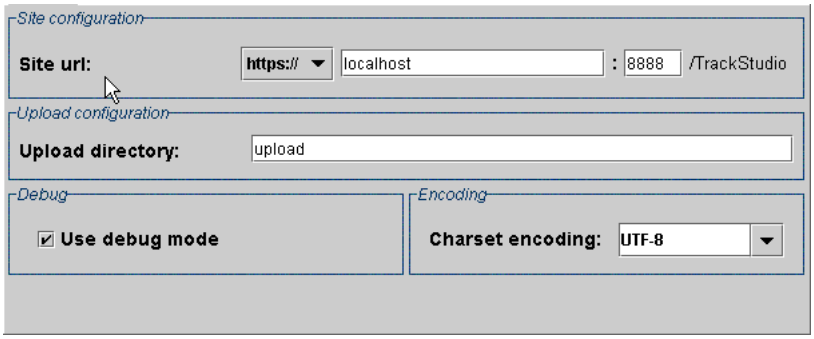
3.2.1 TrackStudio Server Manager Administrator

This section describes how to use TrackStudio Enterprise Server Manager to configure TrackStudio Enterprise.

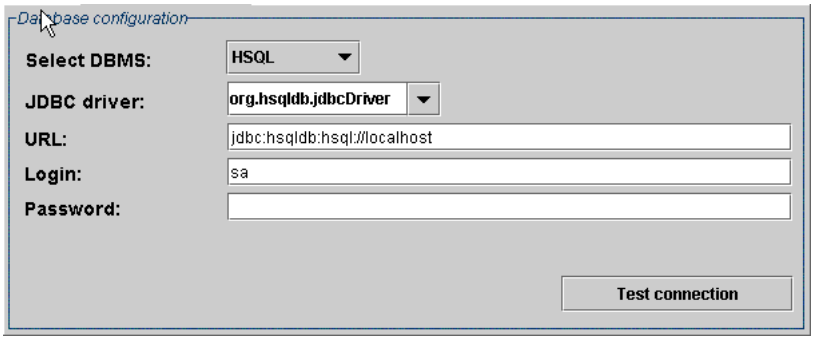
Description

Launch TrackStudio Enterprise Server Manager (**sman.exe**). Wait while it's loading, TrackStudio Enterprise Server Manager window will appear.

1. General. On this tab you can specify TrackStudio Enterprise URL (*http://localhost* by default) and port (*8888* by default), upload directory, and character encoding. You can also switch debug mode on/off.

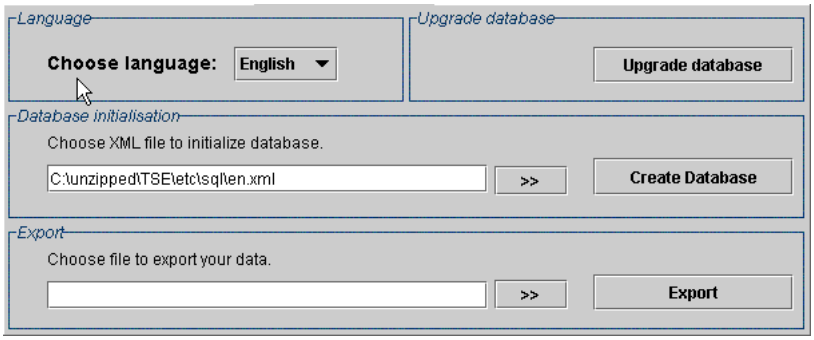


2. Database connectivity. Here you can configure database connection options: database URL, JDBC driver, Login and Password. In the JDBC driver field there is a default JDBC Driver for HypersonicSQL that you shouldn't usually need to modify. After you have specified all the necessary connection properties, you can test the database connection by clicking the **Test connection** button. Please note, that you should start DBMS before clicking the **Test connection** button.



3. Database management. On the **Database management** tab you should enter path to database creation localized XML data file. You can enter it manually or browse it. To create the TrackStudio Enterprise database using a selected file, click the **Create Database** button.

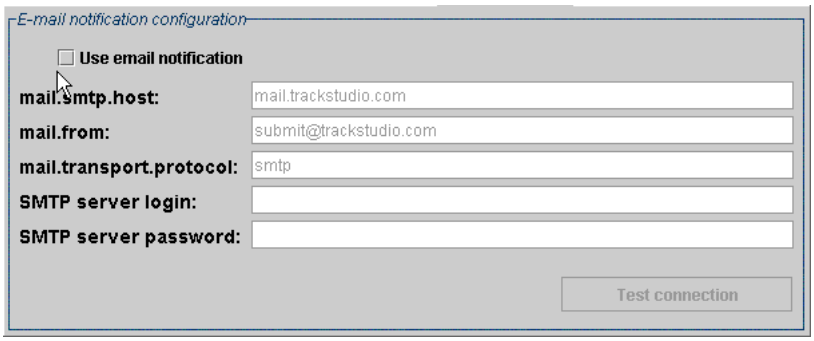
In the window that will be appear, you can click the **Details** button to see database creation details. If during the process of creating a new database an error occurred in one or more SQL operators, TrackStudio may work incorrectly or fail to work completely. Contact us, if such an error occurred. To upgrade the database from a previous version of TrackStudio Enterprise you should click the **Upgrade Database** button.



4. Email notification (optional). On this tab you can configure email notification properties. To turn this feature on you should check **Use email notification**. If checked, the following properties are enabled:

- **mail.smtp.host** - SMTP host address
- **mail.from** - when sending notification, this address will be in the **from** field
- **mail.transport.protocol** - only smtp supported
- **SMTP server login** - login to SMTP server (optional)
- **SMTP server password** - SMTP server password (optional)

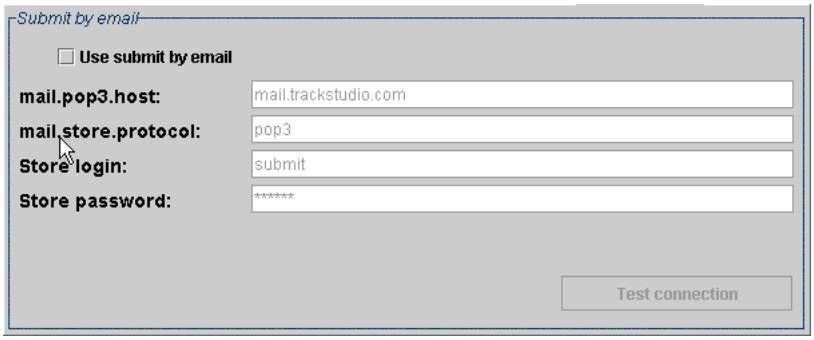
To test the connection to the email server, click the **Test connection** button.



5. Submit by email (optional). Here you can configure email submission properties. To turn this feature on, check **Use submit by email**. You can edit the following properties:

- **mail.pop3.host** - Email POP3 host
- **mail.store.protocol** - mail storing protocol, only pop3 supported.
- **Login**
- **Password**

You can also test the connection to the specified server by clicking the **Test connection** button.



6. Starting/Stopping TrackStudio Enterprise. To start TrackStudio Enterprise, press the **Start** button or select the **Start** item in the **Server** menu. To stop TrackStudio Enterprise, press the **Stop** button or the menu item. When started, you can press the **Open window** button and login into TrackStudio Enterprise. Also you can open your browser at the **URL** and **port** you have specified on the **General** tab.

Administrator login - **root**, administrator password - **root**.

7. Managing configuration. To save your current configuration options, press the **Save** button or the menu item. To load a configuration, press the **Reload** button. If you want to save the console output to a file, select the **File->Save output as...** menu item.

3.2.2 Windows Service Management Administrator

This topic describes how to install TrackStudio as Windows service.

Description

To launch an application as a service (a Windows NT/2000/XP service) in TrackStudio Standalone, you can use **Wrapper.exe**. In the directory **./etc** you can find the config files for starting Jetty (**jwrapper.conf**) and HypersonicSQL (**hwrapper.conf**) as services. This wrapper can also be used to run Jetty and HypersonicSQL from the command line.

Jetty

1) To run an application from the command line, execute the following command:

```
C:\unzipped\TSE>C:\unzipped\TSE\Wrapper.exe -c etc/jwrapper.conf
wrapper  | --> Wrapper Started as Console
wrapperp | port 1777 already in use, using port 1778 instead.
wrapper  | Launching a JVM...
jvm 1    | Wrapper (Version 3.0.3)
jvm 1    |
jvm 1    | 23:48:04.198 EVENT  Using default
           configuration: etc/jetty.xml
jvm 1    | 23:48:04.279 EVENT  Checking Resource aliases
jvm 1    | 23:48:06.452 EVENT  Starting Jetty/4.1
```

2) To install an application as a Windows service, execute the following command:

```
C:\unzipped\TSE>C:\unzipped\TSE\Wrapper.exe -i etc/jwrapper.conf
wrapper  | TrackStudio Enterprise installed.
```

3) To uninstall a Windows service, execute the following command.

```
C:\unzipped\TSE>C:\unzipped\TSE\Wrapper.exe -r etc/jwrapper.conf
wrapper  | TrackStudio Enterprise removed.
```

HypersonicSQL

1) To run an HypersonicSQL from the command line, execute the following command:

```
C:\unzipped\TSE>C:\unzipped\TSE\Wrapper.exe -c etc/hwrapper.conf
wrapper  | --> Wrapper Started as Console
wrapperp | port 1777 already in use, using port 1778 instead.
wrapper  | Launching a JVM...
jvm 1    | Wrapper (Version 3.0.3)
jvm 1    |
jvm 1    | Opening database: test
jvm 1    | HSQLDB server 1.7.2 is running
```

2) To install an application as a Windows service, execute the following command:

```
C:\unzipped\TSE>C:\unzipped\TSE\Wrapper.exe -i etc/hwrapper.conf
wrapper  | HypersonicSQL installed.
```

3) To uninstall a Windows service, execute the following command:

```
C:\unzipped\TSE>C:\unzipped\TSE\Wrapper.exe -r etc/hwrapper.conf
wrapper  | HypersonicSQL removed.
```

3.2.3 How to Install an SSL Certificate. Administrator

The following part deals with the installation of an SSL certificate for jetty.

Description

1) Create a keystore

```
keytool -genkey -alias my-cert -keyalg RSA
        -keystore .mykeystore
```

When creating a certificate, you must specify *keystorePassword* and *keyPassword*

2) Create a certificate request, CSR (into the file **cert.csr**)

```
keytool -certreq -alias my-cert -file cert.csr
        -keystore .mykeystore
```

3) Send CSR to Verisign (or some other company), in response you must get a **cert.crt**.

The following URL can be used for testing

https://www.thawte.com/cgi/server/test.exe

4) Convert cert.crt from PEM to DER (**cert.der**). You can use **openssl** to convert it:

```
openssl x509 -in cert.crt -out cert.der –outform DER
```

5) Import the certificate into the keystore:

```
keytool -import -alias my-cert -file cert.der -keystore .mykeystore
```

6) Edit **jetty.xml**:

```
<Call name="addListener">
    <Arg>
      <New class="org.mortbay.http.SunJsseListener">
```

```
        <Set name="Port">8443</Set>
        <Set name="MinThreads">5</Set>
        <Set name="MaxThreads">100</Set>
        <Set name="MaxIdleTimeMs">30000</Set>
        <Set name="LowResourcePersistTimeMs">2000</Set>
        <Set name="Keystore"><SystemProperty name="jetty.home"
             default="."/>.mykeystore</Set>
        <Set name="PoolName">Listener</Set>
    <Set name="Password">keystorePassword</Set>
    <Set name="KeyPassword">keyPassword</Set>
      </New>
    </Arg>
  </Call>
```

7) Change the protocol and port for **siteURL** in **trackstudio.properties**.

```
# URL of your site. Host name and port should be correct.
# We use this address in e-mail notification messages.

trackstudio.siteURL http://localhost:8443/TrackStudio
```

8) Launch jetty.

9) Open **https://localhost:8443/TrackStudio**

To create a self-signed certificate, do the following:

1) Create Certificate Authority. To do that, run

```
perl ./CA.pl -newca
```

or

```
./CA -newca
```

2) Create a certificate request:

```
keytool -certreq -alias my-cert -file cert.csr
        -keystore .mykeystore
```

3) Create a certificate

```
openssl ca -config /usr/share/ssl/openssl.cnf
           -out cert.crt -infiles cert.csr
```

4) Verify the certificate:

```
openssl verify -CAfile ./demoCA/cacert.pem cert.crt
```

5) Convert the certificate from PEM to DER:

```
openssl x509 -in cert.crt -out cert.der -outform DER
```

6) Import **cert.der** into the keystore.

3.3 TrackStudio/WAR Configuration Administrator

This section describes TrackStudio Enterprise configuration (WAR distribution).

Description

To install and configure TrackStudio Enterprise you should perform the following steps:

1. Run your DBMS and create a new database using the corresponding SQL script that you can find in the **sql** directory. If during the process of creating a new database an error occurred in one or more SQL operators, TrackStudio may work incorrectly or fail to work completely. Contact us, if such an error occurred. If you have used TrackStudio 2.5 before, you should perform the Update from 2.5 (see page 18)

2. Define the TrackStudio configuration in the files **trackstudio.properties**, **trackstudio.mail.properties** and **trackstudio.hibernate.properties**.

3. Use the **-Dtrackstudio.Home=** parameter to specify the directory name (without spaces) with the configuration files for the application server, or simply put the configuration files to the directory **WEB-INF** (in this case you don't need to use **-Dtrackstudio.Home**).

4. Perform the deployment of **TrackStudio.war**. You can also unpack WAR and perform the deployment of the directory structure.

5. Run the application server.

6. The application is available at **http://localhost:port/TrackStudio**

7. Use the following to log in the system: **login=root** and **password=root**

3.3.1 Configuration Files Administrator

This section describes TrackStudio configuration files.

Description

When you start a TrackStudio, five property files are loaded at startup:

- **trackstudio.properties**, a general-purpose configuration file
- **trackstudio.hibernate.properties**, database configuration file
- **trackstudio.mail.properties**, mail configuration file
- **trackstudio.license.properties**, license configuration file, you should not modify it
- **trackstudio.adapters.properties**, adapters configuration file, you should not modify it when you install TrackStudio Enterprise.

These files are read using the following rules:

- If the TrackStudio home directory is explicitly specified by setting the **trackstudio.Home** property using the -D option, the files are read relative to this directory
- Otherwise, the files are read relative to **WEB-INF** directory

1. Configure database connection parameters (**trackstudio.hibernate.properties**)

```
hibernate.dialect cirrus.hibernate.sql.HSQLDialect
hibernate.connection.url jdbc:hsqldb:hsql://localhost
hibernate.connection.driver_class org.hsqldb.jdbcDriver
hibernate.connection.username sa
hibernate.connection.password
```

2. Configure general TrackStudio parameters (**trackstudio.properties**)

```
#######################
# TrackStudio options #
#######################

# URL of your site. Host name and port should be correct.
# We use this address in e-mail notification messages.

trackstudio.siteURL http://localhost:8888/TrackStudio


# Upload directory. Should exist and be accessible
# We suggest you use absolute (not relative) path here

trackstudio.uploadDir upload


# Set yes to include debug information

trackstudio.debug yes


# Set character encoding, such as Windows-1252 or UTF-8

trackstudio.encoding UTF-8


# Send email notification, yes/no

trackstudio.sendMail no


# Include email submit form in email notification, yes/no.
# Please note, that this option requires trackstudio.sendMail yes

trackstudio.FormMailNotification no


# Handler for SSL protocol

java.protocol.handler.pkgs com.sun.net.ssl.internal.www.protocol
```

3. Configure email notification and email submission properties (**trackstudio.mail.properties**)

```
##############################
# Email notification options #
##############################

## Mail transport protocol. Should be smtp

mail.transport.protocol smtp


## SMTP host

mail.smtp.host mail.mycompany.com


## Send email notifications from this address

mail.from user@mycompany.com


## SMTP user. Required only if SMTP server requires authentication

mail.smtp.user


## SMTP password. Required only if SMTP server requires authentication
```

```
mail.smtp.password



############################
# Email submission options #
############################

## Protocol. Should be pop3

mail.store.protocol pop3


## POP3 host

mail.pop3.host mail.mycompany.com


## Check this mailbox for email submission messages

mail.pop3.user user


## POP3 server password

mail.pop3.password pass
```

3.4 Database Configuration Administrator

This section contains DBMS-specific configuration notes.

3.4.1 HypersonicSQL Configuration Administrator

This section describes how to configure HypersonicSQL.

Description

1. Start Hypersonic SQL:

```
java -cp hsqldb.jar org.hsqldb.Server -database TrackStudio
```

2. Execute **trackstudio-hsql.sql** (TrackStudio/WAR) or use the **Create Database (see page 7)** button (TrackStudio/SA):

```
java -cp hsqldb.jar org.hsqldb.util.DatabaseManager
```

Example

Sample JDBC configuration:

```
ConnectionURL: jdbc:hsqldb:hsql://localhost
DriverClass: org.hsqldb.jdbcDriver
UserName: sa
Password:
```

3.4.2 PostgreSQL Configuration Administrator

This section describes how to configure PostgreSQL.

Description

1. Start postmaster.

```
./postmaster -D ../data/ -i -h host.mycompany.com
```

2. Execute **trackstudio-pgsql.sql** (TrackStudio/WAR) or use the **Create Database (see page 7)** button (TrackStudio/SA).

```
./psql -f trackstudio-pgsql.sql
```

Example

Sample JDBC configuration

```
ConnectionURL: jdbc:postgresql://host.mycompany.com:5432/postgres
DriverClass: org.postgresql.Driver
UserName: postgres
Password: postgres
```

3.4.3 Oracle Configuration Administrator

This section describes how to configure ORACLE.

Description

1. Create Tablespace.
2. Create TrackStudio user
3. Grant DBA and Resource role to created user.
4. Start SQLPlus Worksheet and login as TrackStudio user
5. Execute **sql\trackstudio-oracle.sql** (TrackStudio/WAR) or use the **Create Database (see page 7)** button (TrackStudio/SA)

Notes

Oracle connection string includes database URL, JDBC driver, Login and Password. First part (before "@") of this URL is common, you have no need to modify it. After this character you need to enter your database location like "HostAddress:Port:ORACLE_SID". If you are using locally installed version of Oracle, HostAddress is "localhost". Default Oracle port is 1521, default ORACLE_SID is ORCL. In the JDBC driver field there is a default JDBC Driver for Oracle, that you don't usually need to modify.

Example

Sample JDBC configuration.

```
ConnectionURL: jdbc:oracle:thin:@localhost:1521:ORCL
DriverClass: oracle.jdbc.driver.OracleDriver
UserName:TS
Password:TS
```

3.4.4 DB2 Configuration Administrator

This section describes how to configure DB2.

Description

1. Create user tablespace and temp system tablespace.
2. Open a DB2 command window (DOS prompt) (Windows) or log into the server (UNIX)
3. Connect to the database

```
db2 connect to databasename user dbuser using password
```

4. Execute **sql\trackstudio-db2.sql** (TrackStudio/WAR) or use the **Create Database (see page 7)** button (TrackStudio/SA)

```
db2 -tvf trackstudio-db2.sql
```

Example

Sample JDBC configuration

```
ConnectionURL: jdbc:db2://192.168.22.10/TS
DriverClass: COM.ibm.db2.jdbc.net.DB2Driver
UserName: db2admin
Password:db2admin
```

3.4.5 MS SQL Configuration Administrator

This section describes how to configure Microsoft SQL Server.

Description

1. Start **Enterprise Manager** and create database.
2. Start **Query Analyzer** and execute **trackstudio-mssql.sql** (TrackStudio/WAR) or use the **Create Database (see page 7)** button (TrackStudio/SA).

Example

Sample JDBC configuration

```
ConnectionURL: jdbc:microsoft:sqlserver://192.168.22.114:1433;
DatabaseName=trackstudio;SelectMethod=cursor
DriverClass: com.microsoft.jdbc.sqlserver.SQLServerDriver
UserName: sa
Password:
```

3.4.6 Firebird Configuration Administrator

This section describes how to configure Firebird/Interbase.

Description

1. Start Firebird's **isql** program.
2. Create database

```
SQL> create database 'c:\trackstudio.gdb' user 'sysdba' password 'masterkey';
SQL>
```

3. Connect to the database

```
SQL> connect 'c:\trackstudio.gdb' user 'sysdba' password 'masterkey';
Commit current transaction (y/n)?y
Committing.
Database: 'c:\pm', User: sysdba
SQL>
```

4. Execute **sql\trackstudio-firebird.sql** (TrackStudio/WAR) or use the **Create Database (see page 7)** button (TrackStudio/SA)

Example

Sample JDBC configuration

```
ConnectionURL: jdbc:firebirdsql://localhost/c:/trackstudio.gdb
DriverClass: org.firebirdsql.jdbc.FBDriver
UserName: sysdba
Password:masterkey
```

3.5 UNIX-Specific Notes

This section contains UNIX-specific configuration notes.

Description

TrackStudio does not contain graphical libraries for generating colors and fonts and other AWT information. For that, Java relies on the system's libraries for providing that information. Thus an environment capable of providing AWT information and a graphics card (for exporting to static formats) are required.

In a Windows environment, nothing extra needs to be done to set up such an environment as it already exists. A GUI interface is already running and a graphics card already exists.

For non-Windows environments (such as Unix and Linux), such is usually not the case. You need to have X or some form of X running on such systems and point the display to the machine running X (such as running the command export *DISPLAY=192.168.0.16:0.0* in a korn shell). For best performance, TrackStudio recommends running X on the machine (or setting the DISPLAY to point to another machine running X). However, if that is not an

acceptable solution, there are alternative solutions available.

If your TrackStudio UNIX server does not have an X11 Server installed or the DISPLAY environment variable is not set, you may receive one of the following errors when executing your reports:

```
Can't connect to X11 window server using ':0.0'
  as the value of the DISPLAY variable., stack:
java.lang.InternalError: Can't connect to X11 window server using
  ':0.0' as the value of the DISPLAY variable.
at sun.awt.X11GraphicsEnvironment.initDisplay(Native Method)
```

or

```
Internal error: exception thrown from the servlet service
function (uri=/xxx/xxx2.jsp):
java.lang.NoClassDefFoundError: java/awt/SystemColor, stack:
java.lang.NoClassDefFoundError: java/awt/SystemColor
   at com.sas.visuals.BaseBorder.<init>(BaseBorder.java:209)
```

Possible circumventions follow:

1) Install the X11 Server and set the DISPLAY environment variable.

2) Upgrade to Java 1.4 and pass the parameter **-Djava.awt.headless=true** to java when you run it. This no longer requires Xvfb to be running, but it does require the X11 packages to be installed.

Where to specify the options will vary between servlet engines. For example, for Tomcat 3.x, you would specify these options in the **tomcat.bat** or **tomcat.sh** file for **TOMCAT_OPTS**. For Tomcat 4.x, you would specify these options in the **catalina.bat** or **catalina.sh** file for **CATALINA_OPTS**.

3) If your server does not have a display environment, and you cannot upgrade to jdk 1.4, you can install a virtual frame buffer. Xvfb is available at http://www.x.org . It simulates a display environment, and allows the jdk to drawn the images.

4) Try using the Eteks pure java AWT classes at http://www.eteks.com/pja/en. We know of one user who got it working, but we haven't tried it ourselves.

3.6 Update from 2.5 Administrator

This section describes how to update from TrackStudio 2.5.

Description

To upgrade the system from TrackStudio 2.5 to TrackStudio 2.7, you must do the following:

1) Backup your database. You can do this through the DBMS, for instance, by using the `imp/exp` utility in ORACLE. The backup will allow you to use version 2.5 until possible problems with TrackStudio 2.7 are solved.

2) Unpack the archive with TrackStudio 2.7 to a separate directory.

3) Stop the TrackStudio 2.5 instance.

4) If you are using the standalone version, run sman.exe from TrackStudio 2.7, go to the **Database management** tab, select your DB language, if necessary, and press the **Upgrade Database** button.

5) If you are using the WAR version, execute the update script for your DBMS.

6) If errors or problems occur whilst upgrading the DB, you should contact us and continue using TrackStudio 2.5 until the problem is solved.

Notes

- If you have been using an older version of TrackStudio, you should upgrade your database to version 2.5 first using the TrackStudio 2.5. In case there are any problems, please, contact us.
- The database initialization procedure (the **Create Database** button) must be performed only in the case of the initial installation of TrackStudio. If you have used TrackStudio before, you must use the **Upgrade Database** button.

Possible problems after upgrading the database.

- In version 2.7 task priorities are workflow-specific. Because of this, the upgrade will result in all the priorities becoming empty (set to `null`) except for those concerning the tasks belonging to the standard categories (bug, task, project. etc.).
- It is necessary to re-save those filters, that perform filtering by priority.
- The mechanism for e-mail notification has been changed, so users who want like to subscribe to e-mail notifications must modify their filters.
- In version 2.7 there may be relations between categories, so after upgrading the database you may have to set relations for the created categories.

3.7 Debugging Errors Administrator

This topic describes how to debug errors.

Description

If you are experiencing any problems with the application, please contact us either via e-mail or in the web forum.

Provide the following information in your bug report:

1) The version of TrackStudio.

2) The distributive type (standalone + JRE, standalone or WAR).

3) The OS, DBMS, Internet Browser.

4) Attach debug logs to your e-mail message. If you use Server Manager, it can be done through the menu **File->Save Output As…**

4 Getting Started User

This topic deals with the following questions: how to create a user or project, how to grant a user access to the new project, how to create a new task and how to track task changes.

Description

TrackStudio has 2 working modes - **User Management** and **Task Management**. The **User Management** mode is used for managing users and user groups. The **Task Management** mode is used for creating and deleting tasks, tracking task changes, building reports, etc.

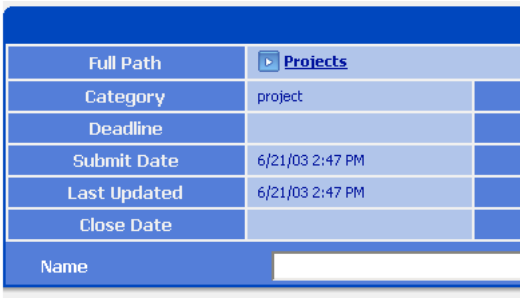
The menu in the upper part of the screen lets you switch between the working modes of the system - **User Management** and **Task Management**.



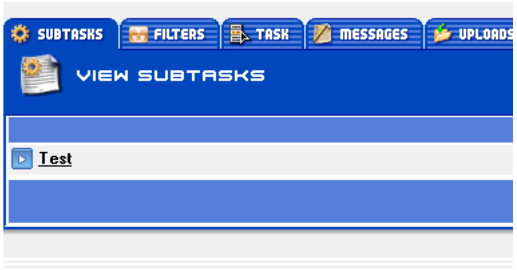
In the right-hand part there is a dropdown list allowing you to quickly select the current project.



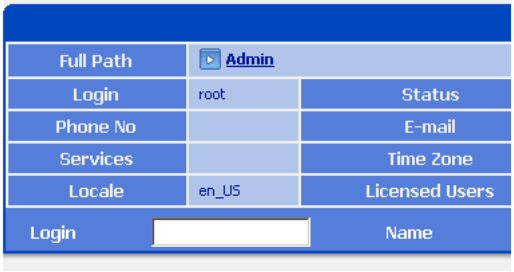
In the **Task Management** mode, the information about the current task is displayed in the central part of the screen.



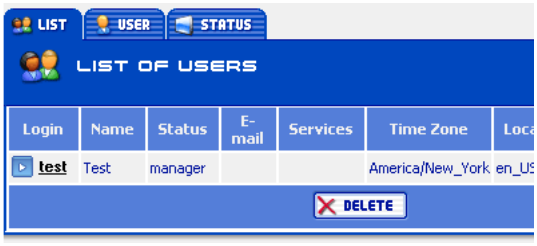
In the lower part of the screen you can see the **Task Control Area** containing information about the actions available for the current task. The available actions depend on the user's permissions.



In the **User Management** mode, the information about the current user is displayed in the central part of the screen.



In TrackStudio all the actions performed for users (for example, creating a user, viewing the user list, changing the password) are performed for the current user.



4.1 Creating a New User User

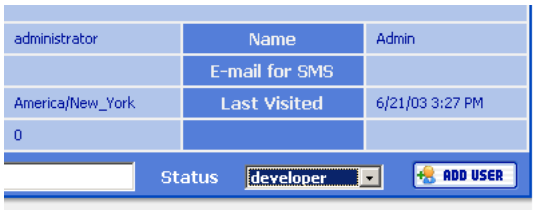
This topic describes how to create a user.

Description

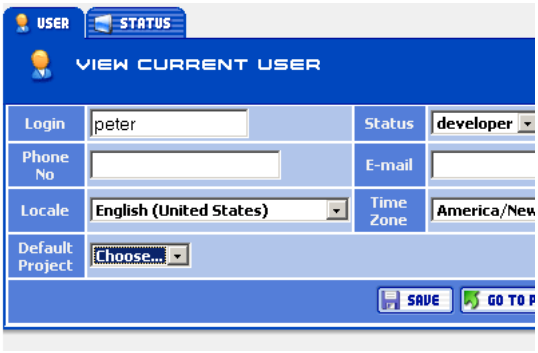
Log into the system as *root/root*. After logging into the system, you are in the **Task Management** mode, which is used for managing tasks. To create a user, select the **User Management** mode in the upper menu. The current user is *Admin* with the login *root*.



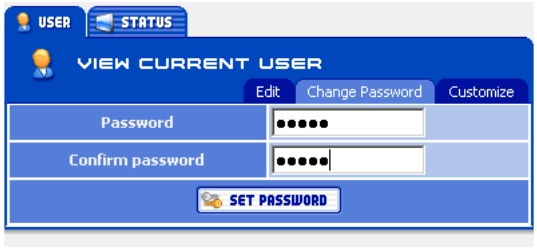
To create a subordinate user: enter the name (e.g. *Peter*); the login (e.g. *peter*); the user group (*developer*) and press the **Add User** button. Note that the newly created user has now become the current user and all further actions (editing, changing the password) will be applied to that user.



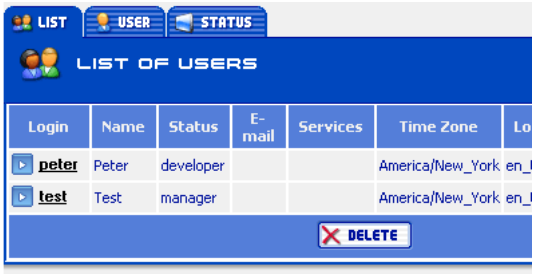
In the form that appears fill in the user's **e-mail**, select the **locale** and the **time zone**. After making the changes, press **Save**.



Select the **Change Password** tab and enter the password for the new user. After that press the **Set Password** button.



Select the *Admin* user in the **Full Path** line of the **Task Header** and TrackStudio will show the list of subordinate users in the lower part of the screen.



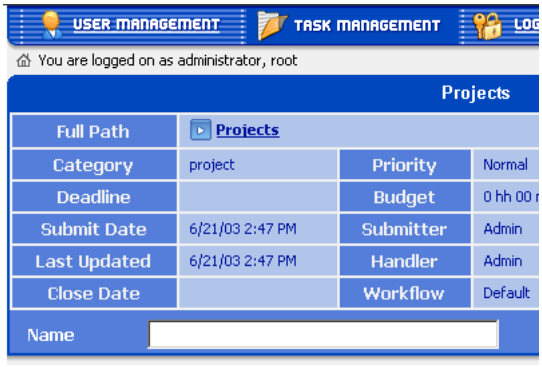
4.2 Creating a New Project User

This topic describes how to create a project.

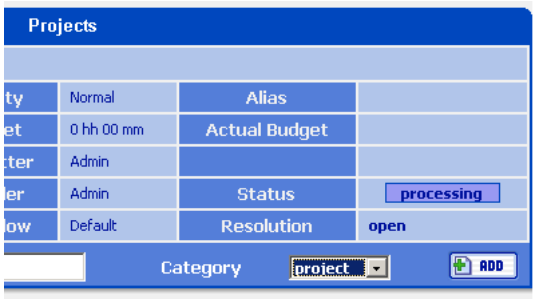
Description

Log into the system as *root/root* or switch to the **Task Management** mode (if you have not

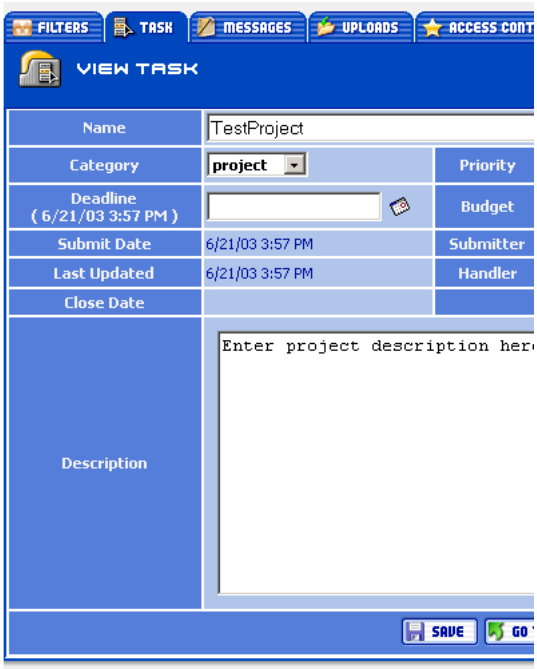
logged out from the system). The current task is a top level project called *Projects*.



To create a project – enter the task name (e.g. *TestProject*); select the task category (e.g. *project*) and press the **Add** button. Note that the newly created project has now become the current task and all the further actions will be applied to it.



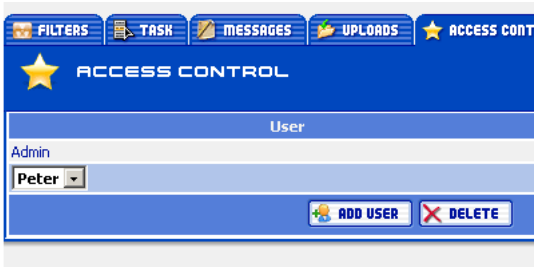
In the form that appears enter the project description in the **Description** field, set other options and press **Save**.



You will find yourself in the window for adding messages to this project. Here you can change the default handler for the project tasks or close the project.



Now you could add a user - e.g. *Peter* - to the list of those working on the project. To do that, go to the **Access Control** tab. You will see all the users who can work on this project. Select the user *Peter* from the dropdown list and press **Add User**.



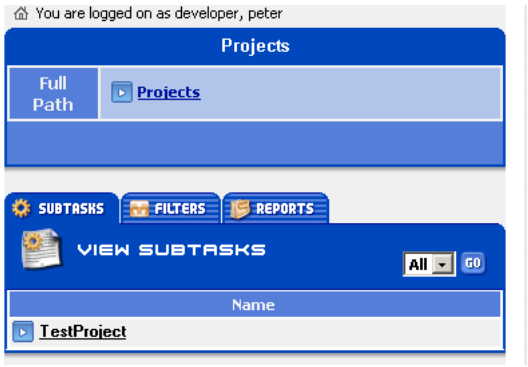
Log out after that.

4.3 Creating a New Bug User

This topic describes how to create a bug.

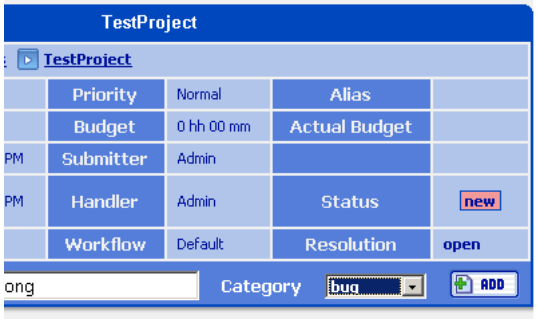
Description

Log into the system as *peter* (*Peter*), the current task is the top level task called *Projects*. Note that the user *Peter* has no access to the project *Projects* (but only to its subproject - *TestProject*), thus, Peter cannot create subprojects of *Projects* as well as perform many other actions.

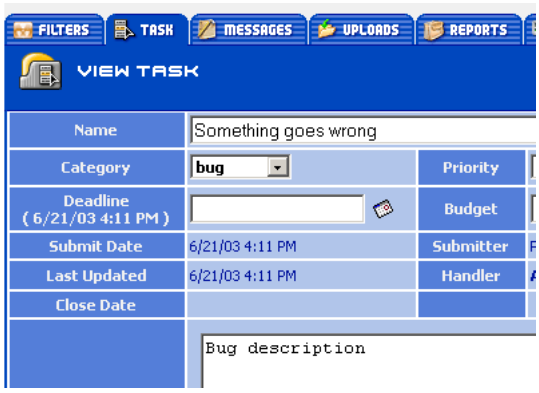


Select the project *TestProject* in the **Subtasks** window, after that it will become the current project.

To create a bug in the project *TestProject* – enter the task name *Something goes wrong*; choose the task type (*bug*), and press the **Add** button.



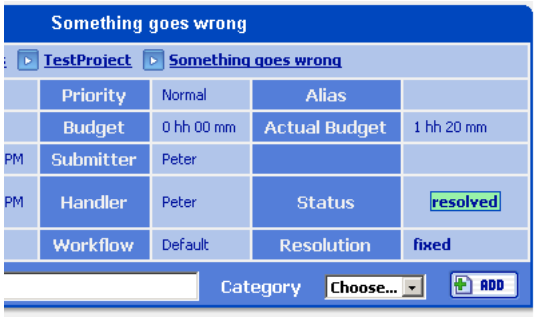
Enter a detailed description for the bug, specify its priority and press the **Save** button.



You will find yourself in the window for adding messages to the new bug. For example, to resolve the bug, choose the message type (e.g. *resolve*), specify the next handler (e.g. *Peter*), select the resolution (e.g. *fixed*), enter the time it took to fix the bug (e.g. *1 h 20 min*), type the comment and press **Save Message**.



Note that the **Task Header** - the actual budget, the status and the resolution - have now been changed for the task according to the added message.



5 Task Management User

This section describes task management features.

Description

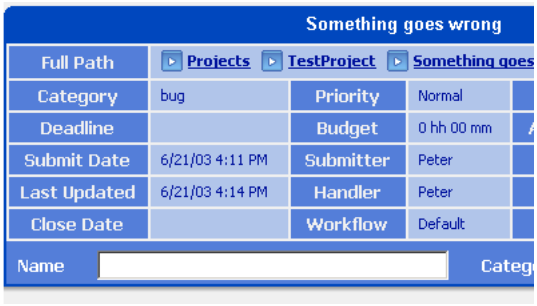
The Task Management window consists of the following parts:

- The **Main Menu** with the **Project Dropdown** list.
- Information about the logged users and their groups.
- **Task Header**
- **Task Control Area**
- **Status Area**

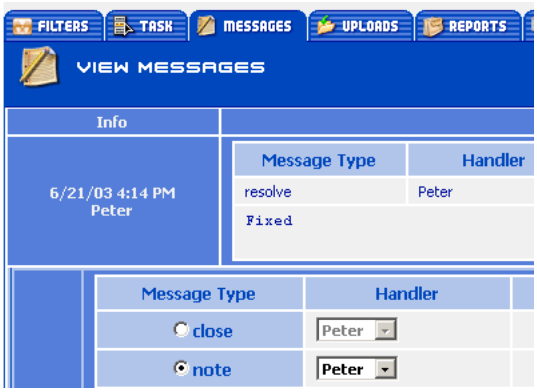
Using the **Main Menu**, the user can switch between the working modes of the application (**Task Management** and **User Management**). In the right-hand part of the menu there is a dropdown list allowing the user to quickly select a project.



The **Task Header** contains the information about the current task. All the available actions (creating subtasks, viewing messages, exporting, generating reports) are applied only to the current task.



The **Task Control Area** contains the information about the actions available for the current task. The availability of actions depends on the user's privileges.



In the **Status Area** you can find the information about the version of the system, about the

time when the page was generated and also about how effectively the request cache is used. **Cache Hit Ratio** is estimated separately for the task cache and for the user cache. If the system is functioning normally, both figures exceed 95%.



5.1 Task Concepts User

This topic contains general task concepts.

Description

TrackStudio supports the hierarchy of tasks. The **Task Header** contains the information about the current task and its position in the hierarchy. Every created object (tasks, filters, reports, categories, workflows) is assigned to the current task, i.e. it will be available for the current task as well as for all its subtasks. To create a global object available for every task in the system, you must create an object for the root task.

For each task TrackStudio assigns an icon:

Icon	Description
	No budget or deadline defined
	Budget defined for task and actual budget less than budget
	Deadline defined for task and current date less than deadline
	Budget defined for task and actual budget greater than budget
	Deadline defined for task and current date greater than deadline

Following table provides list of task properties

Task Property	Description	Task Header	Task Tab
Full Path	Shows the full task path in the tasks hierarchy.	Yes	No
Name	Store **Task Name**, the size is limited to 160 bytes	Yes	Yes

Category	Tasks should have one of the following categories for easy management: *project* - when you create a new project, select this category for the new task. A project can include other bugs or projects. *bug* - bug/error/defect. When a project member finds a bug in the project, he/she should create a new task and assign it the *bug* category. *task* - a project can include some subtasks. If you want to create a subtask, create a new task in the current project and select this category for it. *research* - use this task category to add a research to your project. *question* - a user can add task with this category, if he/she has any questions. *branch, release, module* - these categories can be used for easy management of multiple related bugs. You can extend category list for your project, if you wish.	Yes	Yes
Deadline	This is the date when the current task must be finished.	Yes	Yes
Submit Date	Date when the task was submitted.	Yes	Yes
Last Updated	Date of the last task update. The last update date for a project is the maximum of the last update of any of its subtasks.	Yes	Yes
Close Date	Date when the task was closed (if the task is closed)	Yes	Yes
Priority	Task priority. It can vary from *low* (lowest priority) to *support* (highest priority). The priority can be customized on a per-category basis.	Yes	Yes
Budget	Estimated time (in hours) allocated for the task. This is working time, not calendar time.	Yes	Yes
Submitter	The user who has submitted this task.	Yes	Yes
Handler	The current task handler. When the handler makes some changes to the task (for example, a developer fixes a bug), he/she can assign this task to another user (for example, to tester for checking) or leave it with the same handler. In either case, any task at any time can have only one handler responsible for it. This system helps avoid the situation when there are a lot of issues and nobody knows how to resolve them.	Yes	Yes
Task Alias	Alternative short task name, used in some places if available. Generally, you should set **Task Alias** for your projects. Alias names are also useful for jumping to tasks from the site title page.	Yes	Yes

Actual budget	Elapsed task processing time (working time, not calendar time). The actual budget for the project is the sum of actual budgets of all its subtasks.	Yes	Yes
Status	Task state. Tasks with different status are highlighted with different colors. A newly created task has the *new* status. If a manager processes the task, its status changes to *processing*. After solving the task it becomes *resolved*. The task may be then verified (by a tester), and its status changes to *verified*. After the task is done it becomes *closed*. A task can be *reopened*, when its status is *resolved*, *verified* or *closed*. A reopened task gets the *processing* status.	Yes	Yes
Resolution	Resolution reflects the current task condition. It can be *open, duplicate, fixed, not a bug, not fixable, suspended*.	Yes	Yes
Parent task	A user can move the current task (with all subtasks) to another parent task. The new parent of the task must meet the following requirements: 1) Only tasks with categories marked as **Is Project** shown here. 2) You have rights to view this task and add subtasks to it 3) This task is not a child of the moved task.	No	Yes
Description	Here you can place the task description, if its name is not informative enough. The description size is limited to 1800 bytes	No	Yes
Custom Fields	User-defined custom fields.	Only marked as **visible**	Yes

Example

Suppose some bugs appear in versions of your software both for Windows and Linux. The versions for different platforms are being developed by different teams and you need to track the bug fixes individually for every version. There are two common ways to ensure this:

- You can add to the system one issue, add notes about the systems in which it appears and track the status by adding more notes. This way you cannot track the exact time of fixing the bug, the time spent for each OS, etc.
- You can make individual copies of the issue for each OS. In this case you will have problems tracking the total time spent on this issue and discussing the problems common for both platforms. This method may also lead to numerous bugs and problems in managing them when the bug becomes apparent in later beta versions of the product.

In TrackStudio you can create several versions of each issue (in fact, each of them is a lower level task of a special kind) and set up individual properties (handler, access rights etc.) for each of them. You will be able to view both the general information on the issue (e.g. the total

time spent on fixing the bug in all versions) and the version or configuration-specific information (e.g. the list of all bugs not yet fixed in the Windows version or the list of overtime issues).

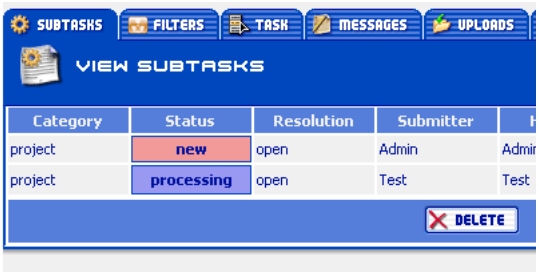
5.2 Subtasks User

This section describes how to list subtasks of the current task (**Task Management->Subtasks** tab).

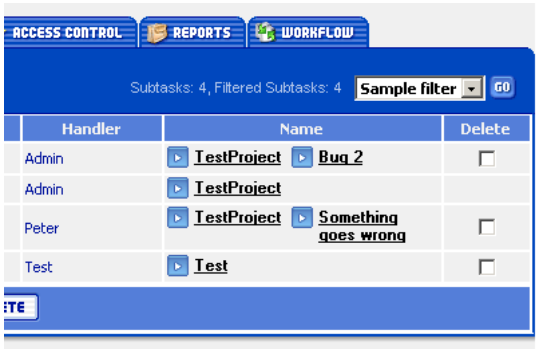
Description

The **Subtasks** window allows you to view the information about subtasks of the current task. If the current task has no subordinate tasks, this window is not available. The list of displayed columns depends completely on the selected filter, except that the **Task Name** is always displayed. All the filters are permanent and the user has to customize them only once. For example, you can adjust the system of filters so that when you access one of your projects, you will see all overdue issues, and when you access another project, you will see the list of questions submitted to you. To select a filter for the current task, use the drop down list in the right-hand part of the window.

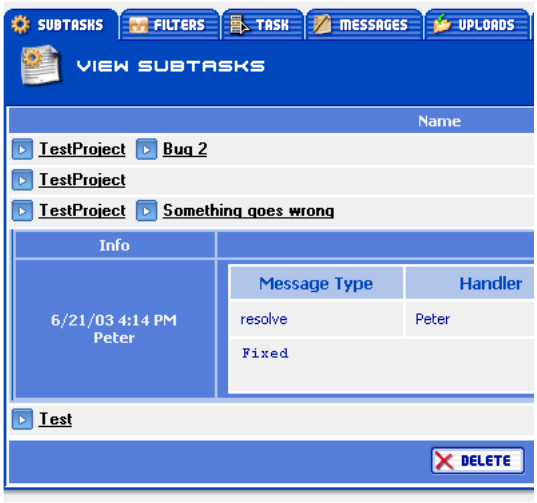
For example, just the task list can be displayed.



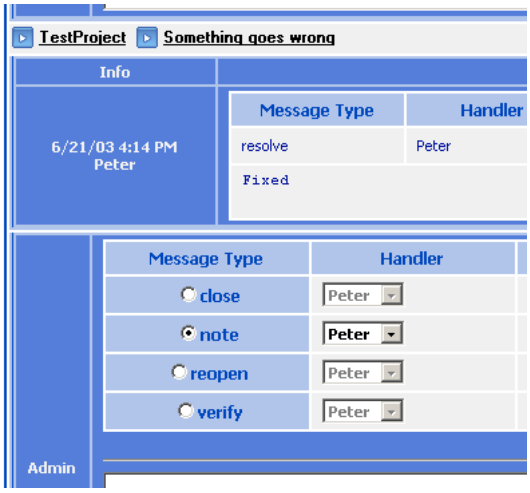
The task list with all the subtasks can be displayed.



The task list with messages can be displayed.



The task list with messages and the bulk edit tool can be displayed. To create a message, you must type the message text, specify the message type, its handler and resolution and press the **Save Message** button.



If there are more subtasks than the current filter allows to be displayed on one page, a slider appears so you can choose which page you want.



To delete a task or a message you must check the task or the message and press the **Delete** button. You cannot delete a task that has subtasks.

5.3 Filters User

This topic describes how to view the available filter list, create new filters, and enable subscription and filter-based email notification (**Task Management->Filters** tab).

Description

Filters are one of the most powerful tools within TrackStudio, they are used for filtering tasks and messages, setting the email notification rules and designing reports. The process of filtering results in a set of tasks and messages meeting certain conditions. Filtering tasks by messages is a very powerful feature of TrackStudio, which allows, for example, getting the list of tasks that were modified the day before and the list of all the comments for those tasks. If necessary, the result of filtering can be sent via email at regular intervals.

TrackStudio allows inheriting filter permissions, as is the case with many other objects. It means that a filter specified for a certain group of projects will be available in all its subprojects. A filter can be modified only if the current task is the one for which the filter was created.

Filters can be either private or shared. Private filters are available only for the user who created them. If the user has no access rights to some task, he/she can create only private filters for it.

When modifying a filter, you can set the filter parameters both for tasks and for messages.

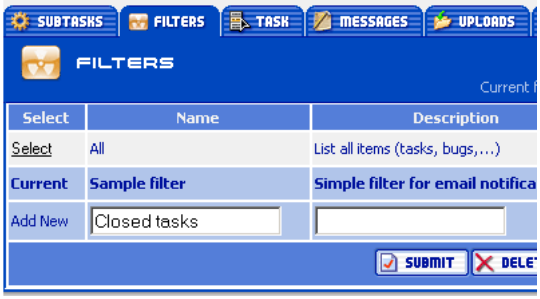
5.3.1 View Filter User

This topic describes how to view the available filter list, create or delete filters (**Task Management->Filters->View** tab).

Description

General filter properties includes:

Filter Property	Description
Select	Use the **Select** hyperlink to select current filter. You can view or edit the selected filter, enable email notification or subscription.
Name	Filter name
Description	Filter description.
Private	Private filters visible only for owner, non-private filters visible to all users that have access rights to this task or their subtasks. If you have no access rights for some task, you can create only private filters for it.
Task	Parent task for filter. Filter will be available for this task and its subtasks
Owner	The user who created the filter. You can't modify or delete foreign filters, but you can use them for task filtering and email notification.
Delete	Use this checkbox to select filter for delete. Please note, that you can't delete used, foreign or selected filters.



To create new filter, fill **Name** and **Description** fields, check, is this filter private or not, and press the **Submit** button. To delete filter check some filter and press the **Delete** button. Please note, that you can't delete used filter.

5.3.2 Edit Filter User

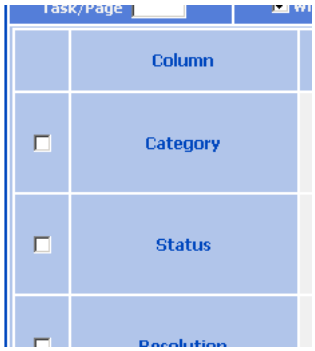
This section describes how to edit filters (**Task Management->Filters->Edit** tab).

Description

Look at the filter parameters for tasks. On the top you can select attributes (by marking the corresponding checkboxes), which you want your tasks to contain. You can display tasks with subtasks, tasks that have budget and/or deadline, tasks that are overbudget and overtime. You can use the **Task/Page** field to specify the number of tasks per page.



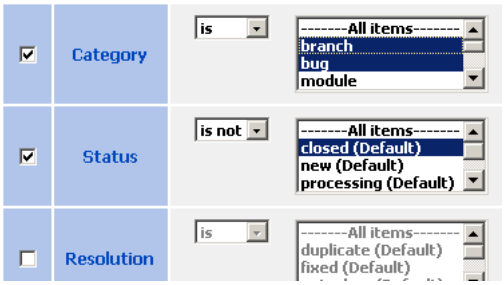
Below you can select and deselect columns that you want to display/hide. To display a column, mark the corresponding checkbox.



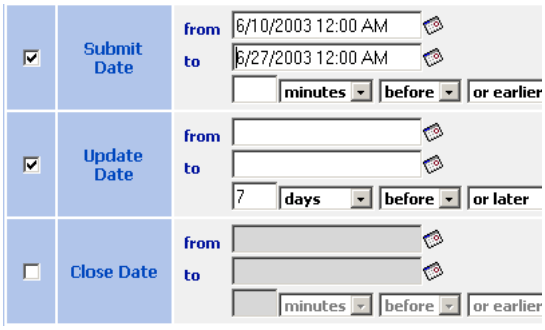
The task list can be sorted by any column. To sort the list you have to check one of two radio buttons: - **Asc** (to sort the columns in ascending order); **Desc** (to sort in descending order).



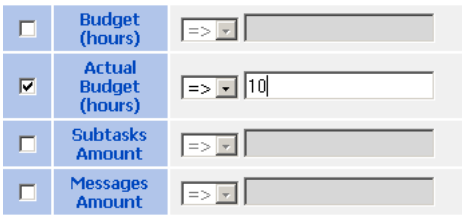
You can display only the tasks with the specified **Status**, **Resolution, Priority, Submitter** or **Handler**. The possible values of these properties can be selected from the lists (hold Ctrl to select multiple items).



You can also specify the boundary dates for such properties, as **Deadline**, **Submit Date**, **Update Date**, **Close Date**. In this case, only the tasks falling within the specified time scope will be displayed. You can use the calendar to specify the date or enter it manually. You can also set relative date, for example, filter condition *"Update Date is 7 days before or later"* shows tasks that were modified last week.

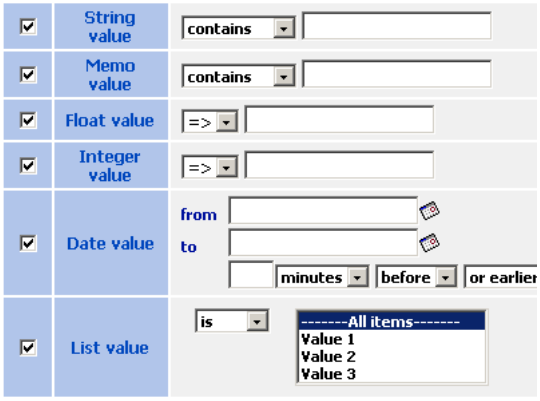


For numerical properties, such as **Budget**, **Actual Budget**, **Subtasks Amount**, **Messages Amount**, you can display tasks with values of these properties, greater or less that given (by selecting **=>** or **<=** from the drop-down list).

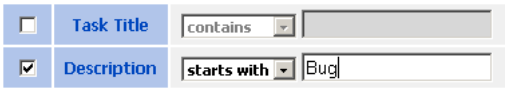


Filtering and sorting data by **custom fields** is supported. The list of custom fields available for filtering is defined as the list of all custom fields specified for the current task and its parent tasks, plus the list of custom fields specified for all the workflows visible to the current

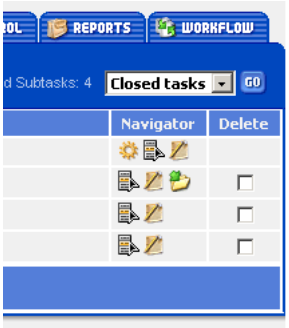
task.



You can view tasks that contain (or start with) certain text in the **Name** or in the **Description**.



Navigator is a convenient tool for accessing the subtask list, the task description, the message list or the attachment list straight from the **Subtasks** window.



Let's see the filter parameters for messages. Filtering tasks by messages allows you to display the tasks having at least one message meeting the specified conditions.

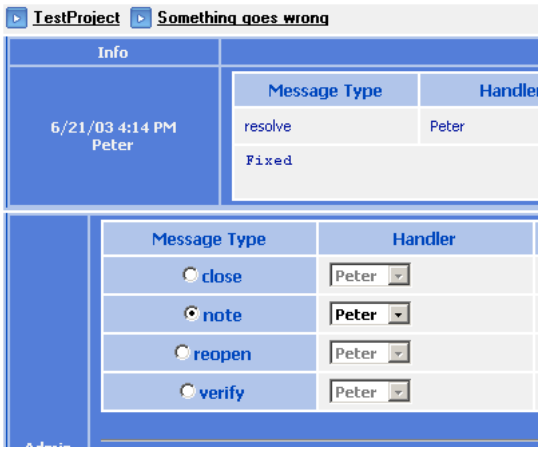
If the **View Messages** option is on, the message list is displayed along with all the tasks. You must set this checkbox if you want to see the list of messages for the task.



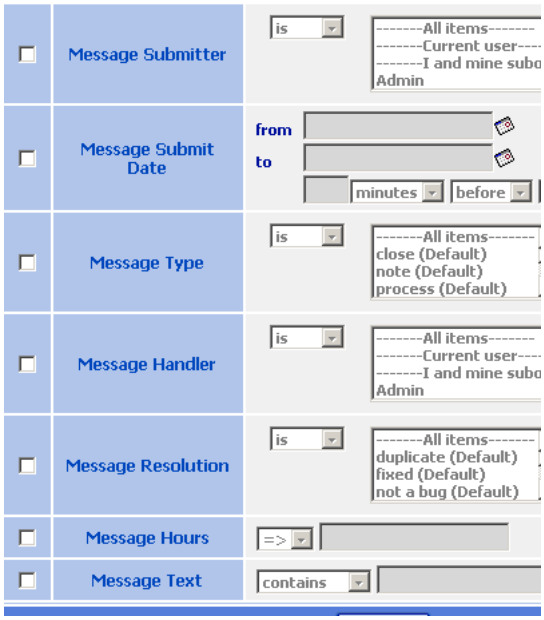
If the **Message Filter** is on, messages are also filtered along with the tasks. This means that for each task found not all the messages will be displayed - only those meeting the filtering conditions.

The **Bulk Processing Tool** allows you to view messages from a number of tasks at once. For each task you can post a comment, specify the number of hours devoted to it and appoint a new handler. For example, this feature gives you the opportunity to quickly change

the status for a large number of tasks, or add important information for several tasks at one time.



Message Submitter, Message Submit Date, Message Type, Message Handler, Message Resolution, Message Hours and Message Text allow you to specify the parameters for filtering the messages. These parameters are combined using the AND operator, i.e. a task is displayed only if it has at least one message meeting all the specified filtering conditions



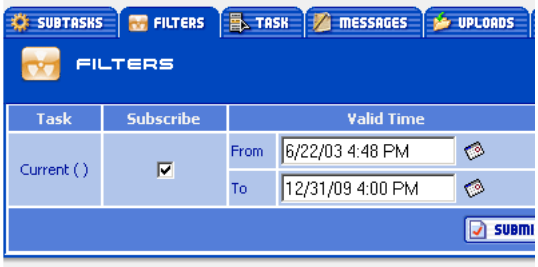
After you have specified the display and sort conditions, press the **Submit** button to save them.

5.3.3 Filter Subscription User

This section describes how to subscribe to a filter (**Task Management->Filters->Subscribe** tab).

Description

Subscription to filters can be useful if you want to periodically receive updated information about the condition of your project. You can receive lists of non-closed bugs, bugs updated on the previous day and much more. To subscribe to a filter, select the filter you need and click on the **Subscribe** tab. You should enter the filter subscription page.



You have to fill in the following fields:

Property	Description
Task	Indicates the current task. You will receive email notification on the subtasks of the selected task. Please note that you can receive email notification based on some filter for multiple tasks.
Subscribe	To subscribe, check this checkbox on. To unsubscribe, check this checkbox off.
Valid time	You can specify **start and stop date** of subscription. You will receive emails only when the current date is between the two.
Next Run	Then next time when the filter should be executed. Generally, you should not modify this field.
Interval	You should select **mailing interval** - from 30 minutes to 1 month.

When you have specified all subscription options, press the **Submit** button.

5.3.4 Notify by Email User

This topic describes the email notification feature (**Task Management->Filters->Notify** tab).

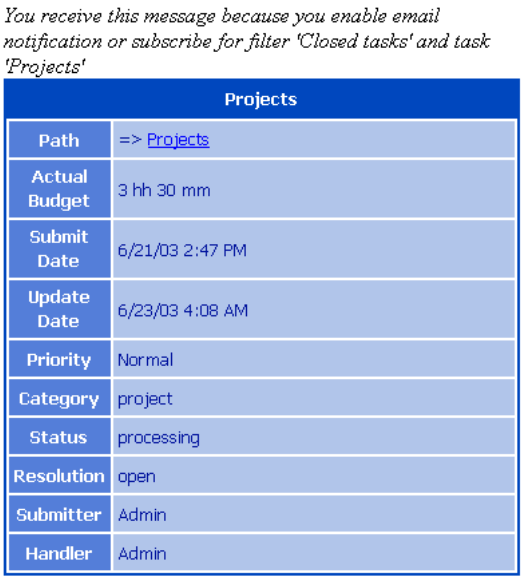
Description

TrackStudio has a powerful filter-based system of email notifications. To get an email notification about a certain task or message type, the user must create a filter, go to the **Notify** tab and mark that he/she would like to get email notifications about the tasks meeting the filtering conditions for the current task and its subtasks. The **Notify** checkbox must be set personally by every user, the manager cannot force all the users to subscribe, though he/she can limit sending notifications to some categories of users on certain events.

As to filter subscription, its message format is completely similar to that of email notification, but the latter is sent when the task status is changed, while filter subscription is sent at

regular intervals.

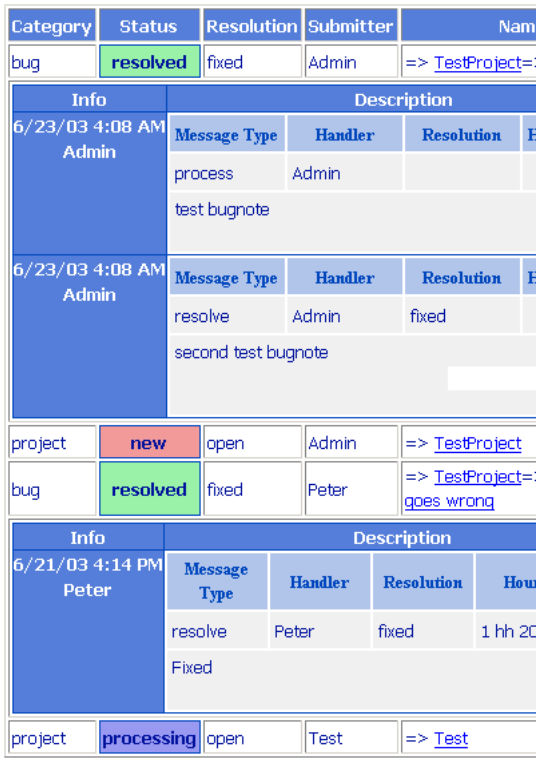
Email notifications are sent out in the HTML format. At the beginning of each email notification message there is information about why you have received this email notification. After that you will find the information about the task whose status has been changed.



You receive this message because you enable email notification or subscribe for filter 'Closed tasks' and task 'Projects'

Projects	
Path	=> Projects
Actual Budget	3 hh 30 mm
Submit Date	6/21/03 2:47 PM
Update Date	6/23/03 4:08 AM
Priority	Normal
Category	project
Status	processing
Resolution	open
Submitter	Admin
Handler	Admin

If the task has subtasks, the list of subtasks is displayed depending on the filtering conditions.



Category	Status	Resolution	Submitter	Nam
bug	resolved	fixed	Admin	=> TestProject=:

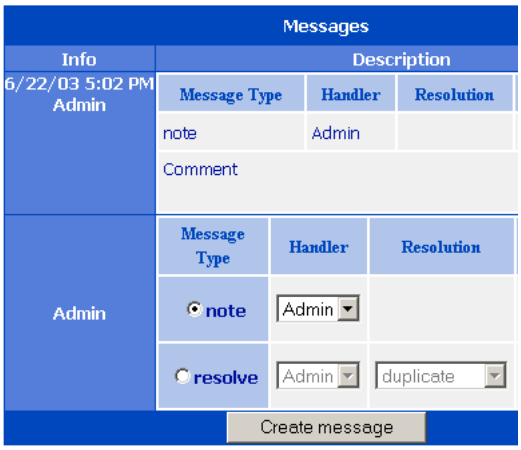
Info	Description			
6/23/03 4:08 AM Admin	Message Type	Handler	Resolution	
	process	Admin		
	test bugnote			
6/23/03 4:08 AM Admin	Message Type	Handler	Resolution	
	resolve	Admin	fixed	
	second test bugnote			

project	new	open	Admin	=> TestProject
bug	resolved	fixed	Peter	=> TestProject=: goes wrong

Info	Description			
6/21/03 4:14 PM Peter	Message Type	Handler	Resolution	Hou
	resolve	Peter	fixed	1 hh 2(
	Fixed			

project	processing	open	Test	=> Test

Then follows the list of messages for the changed task. If submitting via email is enabled, the form for submitting via e-mail is displayed.



For example, if the user creates a filter allowing only the tasks with *priority=high* and among them only those messages where message *handler=customer*, he/she will get email notifications only when high-priority tasks are created and when the customer adds some comments.

When a new task is created, TrackStudio gets the list of parent tasks for the current task. After that every task and every available filter is checked whether the checkbox on this tab is set. If the checkbox is set and the task meets the filtering conditions, an email notification is sent. Please note that when creating a new task, the filtering conditions for messages are not checked even if they are specified.

When a new message is added, similar operations are performed and additionally it is checked if the added message meets the filtering conditions for messages AND if sending email notification messages is allowed by the workflow.

If the task meets a number of filters the user is subscribed to, only one email notification message will be sent - that for the task, which is at the lowest level in the hierarchy.

Note that a filter can be specified for one project and the subscription to it can be allowed for various subprojects and tasks.

The following filter parameters are not checked during the filtering condition check:

- **With Subtasks**
- **View Messages**
- **Filter Messages**
- **Bulk Processing Tool**

Example

Here are some examples of using filters for email notification.

1) *“message submitter is not current user”*. If subscribed to this filter, a user will get email notifications both when a new task is added and when some other user (not him/herself) adds a message (bugnote).

2) *“(task) handler is current user”*. If subscribed to this filter, a user will get email notifications

only if he/she is the handler of the task.

3) *"(task) priority = high and message submitter is Customer"*. If subscribed to this filter, a user will get email notifications both when a high-priority task is added and when the *Customer* adds some messages to the task.

Subscribing to the *All* filter for a certain bug or task is similar to using the **"watch"** mode in some systems, i.e. a user will get email notifications whenever a change is made to the task status or any messages are added.

5.4 Task User

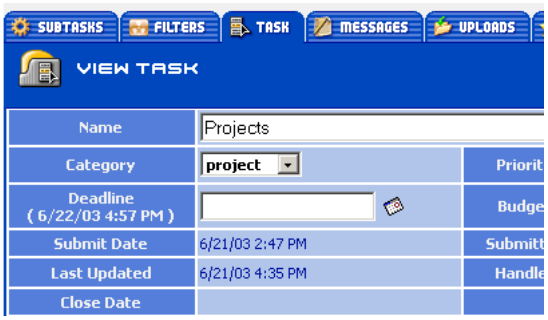
This section describes how to view or modify some important properties of the current task (**Task Management->Task** tab).

5.4.1 Edit Task User

This section describes how to create a new task or modify the existing one (**Task Management->Task** tab).

Description

To create a subtask of the current task, you must enter the **Task Name**, choose its category and press the **Add** button. The task **handler**, **budget** and **deadline** will be inherited from the upper level task.



The purpose of the fields is described in Task Concepts (see page 29) .

Not every task field can be modified at the **Task** tab. Some fields, e.g. submit date, are set when the task is created and they cannot be modified. Some other fields, e.g. **Handler**, can be modified using the message feature. There are also estimated fields - e.g. **Actual Budget** - and fields for which the values are set automatically (e.g. **Close Date**).

After changing the task properties, press either the **Save** button or the **Go to parent** button. When the **Save** button is pressed, the system saves the changes, sends out e-mail

notifications to users and opens the **Messages** tab. When the **Go to parent** button is pressed, the system saves the changes, sends out e-mail notifications to users and opens the **Subtasks** tab of the upper level task.

It is better to use the **Save** button if you want to add a message right after creating the task. The **Go to parent** button is useful when you enter into the system information about a number of tasks.

Notes

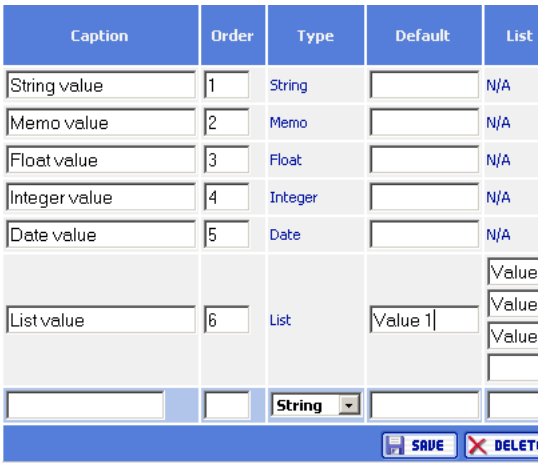
Be especially careful when changing the **Parent Task**. Such system objects as custom fields, filters, categories and workflows can be inherited. When the parent task is changed, the list of the parent tasks and inherited objects can also change, which can result in unpredictable consequences.

5.4.2 Customize Task Project Manager

This section describes how to customize task fields (**Task Management->Task->Customize** tab).

Description

You may want to create some custom fields such as version, release, platform, and other. All custom fields created for a task will be also available in all subtasks of the current task. If you need category-specific custom fields you should use workflow customization.



You have to fill in the following properties:

Property	Description
Caption	Custom field name. It can be *software platform, release, version, customer name* and so on.
Order	An integer value which indicates the position of the field on the user page, the system sorts custom fields by this field.

Default	Default field value (100, for example). For list, the default value should be equal to one of the list items.
List	Contains possible values for dropdown lists. To fill this property enter first list item, then press the **Save** button, then enter the second and save that, etc.
Required	Indicates whether a field value is required or not. The users can't save a task if they do not fill in all required properties.
Type	Custom field type. Please check out next table for more information about available custom field types.
Visible	Indicates whether field is visible in the **Task Header** or not.
Send by email	This field defines whether this property should be included in the email notification and subscription messages sent to a user by email.

There are six possible custom field types:

Type	Description
String	String of symbols
Memo	Text area
Float	Floating point value
Integer	Integer value
Date	Date/Time value
List	Drop-down list with values specified in the **List** field

After all the required fields are filled in, press the **Save** button to save the created or modified custom field. To delete a custom field or dropdown list item select the field or list item and click the **Delete** button. You can delete or modify fields created only on the current level in task hierarchy.

5.4.3 Export Project Manager

This section describes how to export tasks (**Task Management->Task->Export** tab).

Description

To export the current task with all subtasks and messages just press the **Submit** button. The created XML will be zipped and attached to the current task. Look for it within the **Uploads** tab.

Remarks

You can export data to other formats with the custom Export Adapter.

Example

```
<gran-pm-export>
```

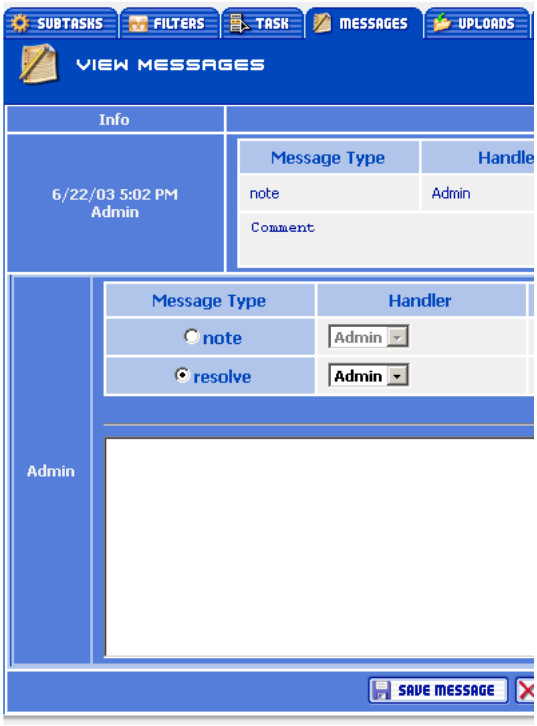
```
  <defect>
    <path> => ROOT => User Activity => Enter title here. </path>
    <id>A42754AA0BAD18E0E0303BD58EAC420F</id>
    <parent_id>A3EEB42C51F5F5EEE0303BD58EAC1707</parent_id>
    <name><![CDATA[Enter title here.]]></name>
    <abudget>27 hh 00 mm</abudget>
    <submitdate>24/06/2002 13:42</submitdate>
    <updatedate>24/06/2002 16:52</updatedate>
    <priority>Normal</priority>
    <category>bug</category>
    <status>new</status>
    <resolution>open</resolution>
    <submitter>User</submitter>
    <handler>User</handler>
    <info><![CDATA[]]></info>
    <udfs>
    </udfs>
    <files>
    </files>
    <messages>
      <message>
        <status>note</status>
        <timestamp>24/06/2002 15:53:13</timestamp>
        <user>User</user>
        <time>10 hh 00 mm</time>
        <resolution>open</resolution>
        <description><![CDATA[10 hours of work]]></description>
      </message>
      <message>
        <status>note</status>
        <timestamp>24/06/2002 16:52:52</timestamp>
        <user>User</user>
        <time>10 hh 00 mm</time>
        <resolution>open</resolution>
      </message>
    </messages>
  </defect>
</gran-pm-export>
```

5.5 Messages User

This section describes messages (**Task Management->Messages** tab).

Description

Any task can contain messages. Messages provide the communications between the group members. They may be progress reports, questions, comments etc.



You have to fill in following fields:

Property	Description
Date	Message submission date
Submitter	User who submitted this message
Message Type	Select the message type from the list of available message types. It may be simply a note (comment), or you can resolve, close or reopen a task by selecting the message type. TrackStudio shows you a restricted list of message types based on the task workflow, current task status and user group.
Handler	Select the next handler of the task. For example, if the task is a bug, you, as a manager, can specify a developer who will work on this bug. After resolving it the developer should change the handler to a tester, who will verify the developer's work. TrackStudio shows you a restricted list of users based on the task workflow, current task status, message type and user group.
Resolution	In the resolution field you can enter the task **resolution** (*fixed*, *not a bug*). Each message type can have its own resolution set.
Hours	Specify time spent on processing a task in the **time** field. It can be useful for creating time reports on the job.
Delete	Use this checkbox to select message for deletion.
Text	Enter comment in the text field. The size is limited to 1800 bytes

To save a message, you have to press the **Save Message** button at the bottom of the page. To save a message and go to the list of subtasks of a parent task, you have to press the **Go**

to parent button at the bottom of the page. You can also delete a message by selecting one or more messages and clicking the **Delete** button.

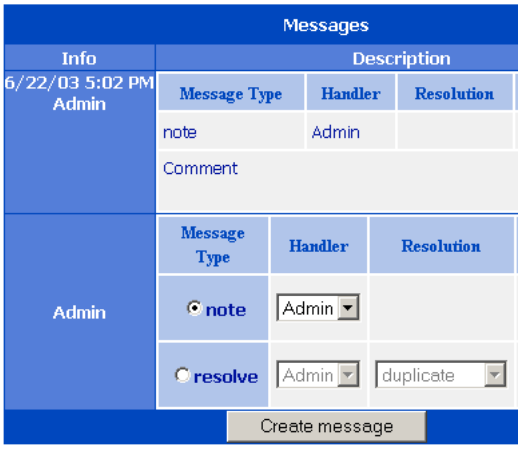
5.5.1 Submit by Email User

This section describes how to submit bugnotes by email.

Description

The Submit by Email feature provides easier and faster task processing. Using **Submit by Email** you can process task only with your email client, you don't need to go to the TrackStudio site. All you need to do is just send an email message.

When a user adds a bug-note, users will receive an email notification message from TrackStudio with the task description, subtasks and bug-notes list. After the last note in this message, you will find a form.



Using this form you can specify **message type** (note, resolve, etc., depending on your workflow), next **handler**, **resolution,** and **time** spent on this note. All these fields are the same as when you add a task note. When you have entered all these fields, press the **Create Message** button to generate the email message.

After the message has been generated, include the **bug-note** in it after the generated string. Add an **attachment** if you wish, but do not edit existing message text, subject or recipient address.

Finally, you should send this message to TrackStudio. This note will be added to the task and you will receive an email notification about it.

Notes

To use the Submit by Email feature you should enable JavaScript in your email program. For example, in the Outlook Express you should change the security level (Tools/Options/Security/Virus Protection/Select the Internet Explorer security zone to use) from 'Restricted sites zone (More secure)' to 'Internet zone (Less secure, but more

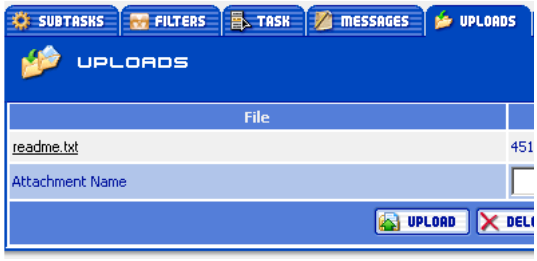
functional)'.

5.6 Uploads User

This section describes how to upload a file (**Task Management->Upload** tab).

Description

The **Uploads** tab provides a way to attach notes, files, links to other items, and e-mails to the current task. This dialog is also used to modify existing attachments and notes. Which of these actions you can perform depends on your privileges.



To upload a file attachment, specify the path to the file you want to attach using the **Attachment Name** field, or click the **Browse** button and find the file using the **Open File** dialog. Then press the **Upload** button. To download the uploaded file just select the link from the uploads list. If you want to delete an attachment, click the **Delete** button.

Remarks

For TrackStudio Host you should pay a small fee for uploaded files. The fee is charged monthly for the disk space.

5.7 Access Control Project Manager

This section describes how to use the access control feature (**Task Management->Access Control** tab).

Description

For managing large developer teams TrackStudio uses the system of delegation of authority (rights). Let's assume that we have two managers: John and Smith. The top manager can grant access to the *Foo* group of projects to *John*, and access to the *Bar* group of projects to *Smith*. After that both managers can grant access to specific projects, tasks and releases to individual developers or delegate this right to a lower level manager.

In all cases the company management can access the summary or detailed information. The summary information includes the total amount of time spent by the *Foo* and *Bar* managers.

The detailed information includes the list of tasks and employees involved in them, the detailed work description and the time spent on each task.

At the same time access to information can be restricted so that the *Bar* manager will not even know that the *Foo* project exists, let alone the number of employees in another team and the number of projects/tasks assigned to it. So, the authorized persons (top managers) can access and analyze any information (this would be impossible if the data were stored in different databases), but the lower level managers and the employees can see only the information required for their work. If necessary, however, one manager can be granted access to some of other manager's tasks).

If a user or manager has access to an upper level project, he/she will be automatically granted access to all subordinate projects. But the persons, who have access to lower level projects, cannot access their ascendants (they can see only their names - this is required for navigating the tree of projects).



To permit a user access to a task, you have to select it from the **User** drop-down list and press the **Add User** button. You can forbid access to any user by deleting him/her from the access list - just select user and click the **Delete** button. To delete an inherited item you should select the **Access Control** tab of the parent task.

5.8 Reports User

This section describes reports (**Task Management->Reports** tab).

Description

TrackStudio has a report generator that allows you to design various reports based on the results of filtering. You can use both the filters set for the current task and those inherited from higher level tasks. The created report will be available both for the current task and for its subtasks.

5.8.1 Report Types User

This topic describes available report types.

Description

List Report

Shows the list of subtasks of the current task in the form of a list. Tasks, bug notes, task fields and messages are displayed according to the filter settings.

Detail Report

Shows all the information about subtasks of the current task. The lists of subtasks as well as the messages are displayed according to the filter settings.

Distribution Report

Allows statistical analysis of the data distribution. To design this kind of report, you must specify what data should be displayed on the X and Y axes, which parameter should represent the data and which function should be calculated for the selected parameter.

The parameters of the X and Y axes can be any parameters having a fixed quantity of values, including custom fields of the list type. Data can be any numeric field, including custom fields of the Integer and Double types. Note that not every possible parameter combination can make sense.

Money Report

Lets you gather information on the intensity of working with subtasks of the current task. Using this filter, you can specify the filtering conditions for tasks or messages. For example, you can specify a period of time that you are interested in getting information about. A report constitutes a table where the developers are placed along the X axis while subtasks of the current task are displayed along the Y axis (if the filter has the “including tasks” option on, subtasks of the current task are also displayed).

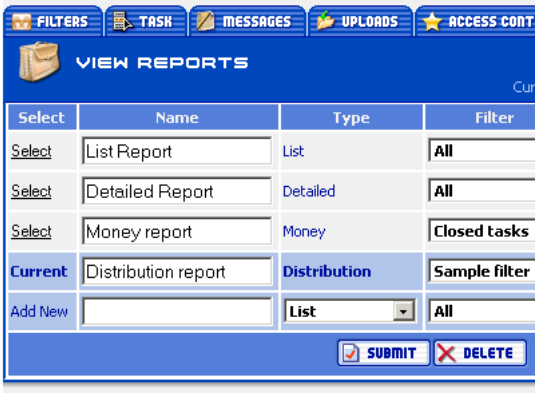
In the cells of the table you can see the information about how many hours a certain developer devoted to the current task during the period of time specified in the filter.

5.8.2 List Reports User

This topic describes how to a create report (**Task Management->Reports->List** tab)..

Description

To generate a report, you must select it by pressing the **Select** link, choose the format of the report layout (the report formats currently supported are HTML, PDF, MS Excel (XSL), CSV and XML), specify the necessary parameters and press the **Submit** button.

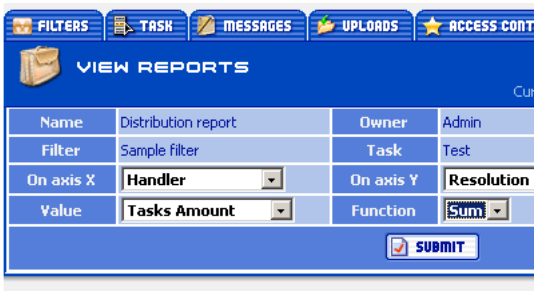


To create a report, you must specify its **Name**, **Description** and **Type**, choose a **Filter** for the report to be based on, indicate whether the report is private and press the **Submit** button.

5.8.3 View Reports User

This topic describes how to generate a report (**Task Management->Reports->View** tab)..

Description



To design a report, you must select it by pressing the **Select** link, choose the **format** of the report layout, specify the necessary parameters and press the **Submit** button.

5.9 Workflow Management Project Manager

This section describes how to use workflows (**Task Management->Workflow** tab).

Description

Workflow management allows the user to define new types of primary items (such as *build, task, bug*), indicate different states and transition rules, and e-mail notification regulations for them. You need to do the following steps to utilize workflow:

- Create empty (blank) workflow.
- Add set of priorities.
- Add set of states; define start and final states.

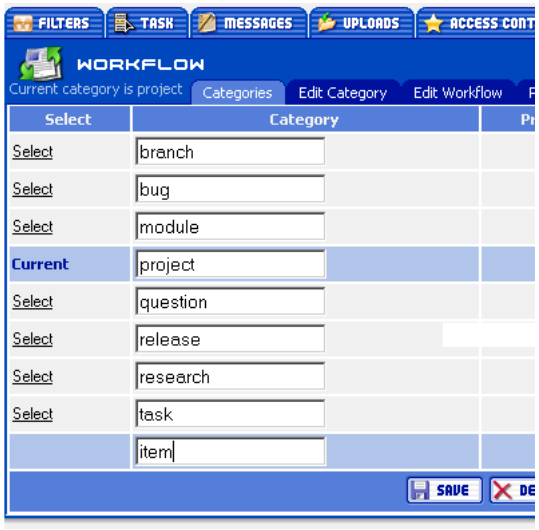
- Add set of message types to your workflow. Each message type should have appropriate set of resolutions. For example, message type **resolve** can have the following resolution set - *fixed, not fixable, duplicate problem, not a bug***.**
- Define transitions (transition rules between states), set up email notification and security rules for messages. A single message type can have several corresponding transitions; for example, message type *note* can have transitions from *new* to *new*, from *processed* to *processed*, from *resolved* to *resolved*.
- Create a task **category** and associate your workflow with this category.
- Create a new task and indicate the category you created as the task category.

5.9.1 Categories Project Manager

This section describes how to create a task category (**Task Management->Workflow->Categories** tab).

Description

Category is a certain task type, which is indicated during its creation and is directly connected with workflow. Several categories may have the same workflow.



You have to fill in the following properties:

Column	Description
Category	The name of the category.
Workflow	Workflow associated with this category.
Task	A certain task in the task hierarchy. You need to have access rights for this task level to edit or delete this category.
Is Project	Means that this category is for projects. Such tasks are displayed in the task list in the upper right-hand corner and in some other lists.

Delete	Use this checkbox to select a category for deletion. Please note, that you can't delete used or selected categories.

Please click the **Save** button to save a category. Please click the **Delete** button to delete a choosen category.

Remarks

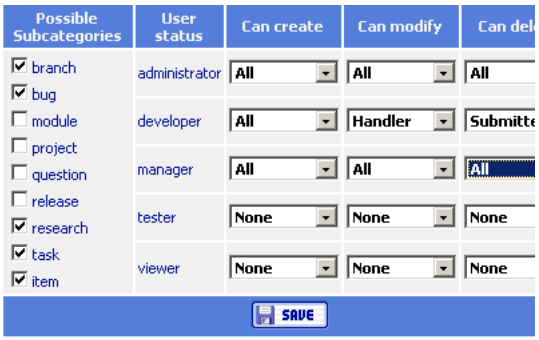
After you create a task, you can change its category only to those associated with the same workflow.

5.9.2 Edit Category Project Manager

This topic describes additional category properties (**Task Management->Workflow->Edit Category** tab).

Description

Edit Category allows you to adjust various parameters of the categories. The table on the left (possible subcategories) allows you to specify which categories can be set as subcategories of the current one. For example, you can specify that inside *projects* it is possible to create other *projects*, *tasks* and *bugs*, but a *bug* cannot have a *project* as its subtask.



The table on the right allows you to specify which groups of users can create their own tasks in this category, as well as edit and delete them.

You can select possible actions for users of all available status for a task category

Option 1	Option 2	Description
None	Can create	Nobody can create subtask of selected category
None	Can modify	Nobody can modify tasks of selected category
None	Can delete	Nobody can modify tasks of selected category
All	Can create	Everybody can create subtasks of selected category.
All	Can modify	Everybody can modify tasks of selected category.

All	Can delete	Everybody can delete tasks of selected category.
Submitter	Can create	Only submitter (of parent task) can create subtasks with selected category.
Submitter	Can modify	Only task submitter can modify tasks of selected category
Submitter	Can delete	Only task submitter can delete tasks of selected category
Handler	Can create	Only handler (of parent task) can create subtasks with selected category.
Handler	Can modify	Only task handler can modify tasks of selected category
Handler	Can delete	Only task handler can delete tasks of selected category

If you don't have the editing privileges for the given category, you won't be able to change the category settings of higher-level categories and user status, but you will be able to create your own (lower-level) categories and user status and modify them. This way you can add new categories and user status to the tracking process.

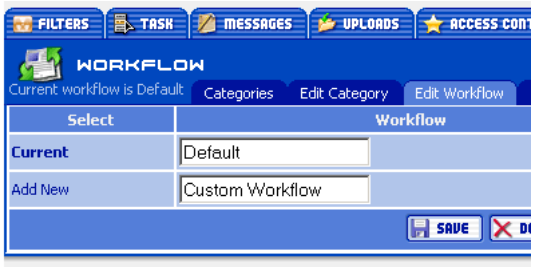
5.9.3 Workflows Project Manager

This section describes how to create or modify a workflow (**Task Management->Workflow->Edit Workflow** tab).

Description

Workflow is a collection of states and transitions that defines the tracking process of primary items. States, messages and transitions map the path items follow. Once established, workflows are assigned to tasks. This system enables you to first define workflow processes, then use workflow to track items at various levels. Several projects may use the same workflow, or workflows may be modified (extended) as needed for individual projects.

To connect a workflow with a task, a category mechanism is used. When a task is created, you indicate the category which defines the workflow. **On every hierarchy level tasks and categories are accessible if they are defined on the given level and the higher level(s).**



You have to fill in the following properties:

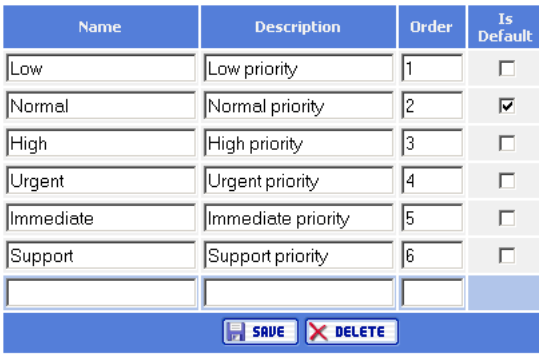
Column	Description
Select	Current workflow selector.
Workflow	Name of the workflow.
Task	Indicates related task in task hierarchy. You need to have access rights for this task level to edit this category.
Delete	Use this checkbox to select a workflow for deletion. Please note, that you can't delete a workflow that's selected or in use.

To choose an active workflow please use the **Select** hyperlink. Click the **Save** button to save the workflow. Click the **Delete** button to delete the workflow.

5.9.4 Priorities Project Manager

This topic describes how to create or modify priorities (**Task Management->Workflow->Priorities** tab).

Description



For each workflow you can specify its own priorities.

Property	Description
Name	The name of the priority
Description	the description of the priority
Order	Specifies the order in which priorities will be displayed.
Is Default	Specifies the default priority. When creating a task, the default priority will be automatically set for it.

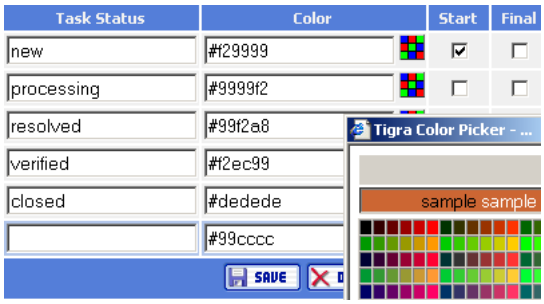
Please click the **Save** button to save a priority. Please click the **Delete** button to delete a chosen priority.

5.9.5 States Project Manager

This section describes how to define task states (**Task Management->Workflow->States** tab).

Description

A status (or state) is a position in the workflow where a primary item resides. When an item has a given status, and a handler who is responsible for performing a specific task, that task must be completed before the item can be transferred to the next state.



Items are moved from state to state using messages. *New* and *processed* are basic examples of states. When an item with default workflow (*task, bug*) is first submitted to TrackStudio, it resides in the *new* status. It cannot leave the *new* status until the user who has ownership of it performs the task of assigning it. When that user assigns it, the *process* message occurs and the item moves to the *processed* status. It will remain in this status until the appropriate action is taken by the user it was assigned to, and then it will move to the *resolved* status. As this example shows, states provide accountability and allow items to be transferred through the workflow process.

You have to fill in the following fields:

Property	Description
Task Status	Task status name
Color	Color of field in the **Task Header** and **Subtask List**.
Start	Indicates that this state is start state. Every workflow must have one **start state**. Start status is the status of a newly created task.
Final	Indicates that this state is a final state. Every workflow can have one or more **final states**. When a final status is reached which the handler is cleared and the close date is stamped.
Delete	Use this checkbox to select a state for deletion.

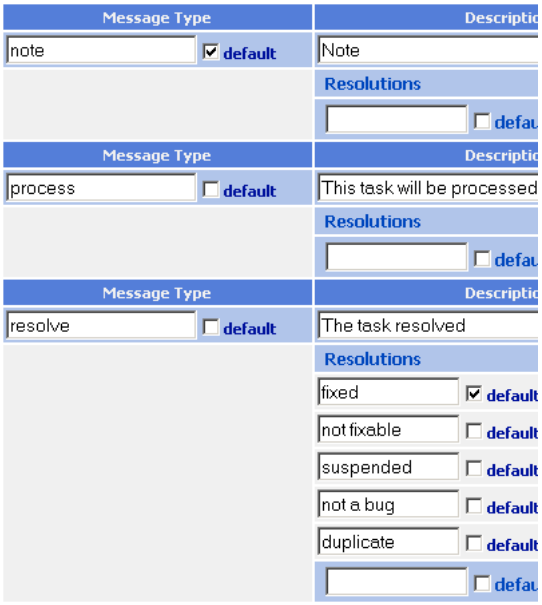
Click the **Save** button to save a task state. Click the **Delete** button to delete a task state.

5.9.6 Message Types Project Manager

This section describes how to define message types (**Task Management->Workflow->Message Types** tab).

Description

A message moves primary items from status to status in the tracking system. For example, a recently submitted item (*bug*, for example) is in the *new* status. When this item needs to be assigned to a user, such as an developer, the user who has current ownership (handler) of the item uses the *process* message to move the item to the *processed* state. When a resolution is reached, the developer (in this example) uses the *resolve* message to move the item through the workflow from the *processed* state to the *resolved* state.



Every message type has a set of available resolutions. For example, resolve message can have the following resolutions available - *fixed, not fixable, duplicate*. If your message does not have a set of resolutions established this message won't change the resolution.

You have to fill in the following properties:

Property	Description
Name	Message type name. You can specify a message type as default, TrackStudio selects the default message type automatically when you create a new message.
Description	Message type description
Resolution	Resolution name. You can specify a resolution as default, TrackStudio selects the default resolution automatically when you create a new message. To fill this property enter the first resolution, then press the **Save** button, then enter the second and save that, etc.

Delete	Use this checkbox to select a message type or resolution for deletion.

Click the **Save** button to save new and modified message types and resolutions. Click the **Delete** button to delete message types and resolutions.

5.9.7 Transitions Project Manager

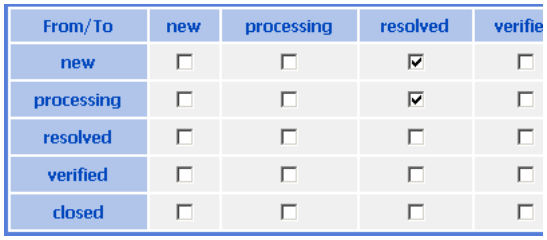
This section describes how to define a Transition Matrix (**Task Management->Workflow->Transitions** tab).

Description

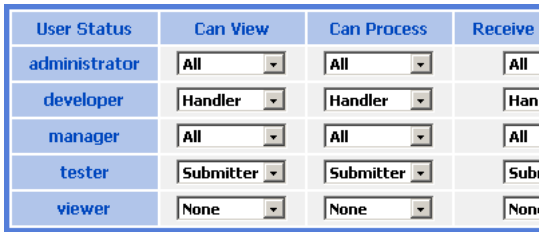
The **Transitions** tab is used for creating a Transition Matrix for every message type, and setting email processing and notification rules for the users of each level of task hierarchy.

The matrix is a table that has all states available. The left column represent initial (**From**) states, the first column - destination (**To**) states. If you check an item, it means that transitions for given message types are permitted from initial state to destination state.

For example, the following matrix for message type *resolve* means "*Resolve* is transition from *new* or *processing* state to *resolved* state".



You can select possible actions for users of available status for each message type.



Transition Option	User Option	Description
Can View	None	Nobody can view this message in the transition history
Can Process	None	Nobody can do this transition (save this message)
Receive Notification	None	Nobody can receive email notification

Can View	All	All users with this status can view this message in the transitions history (previously sent messages).
Can Process	All	All users with this status can do this transition (save this message).
Receive Notification	All	All users with this status and enabled email notification for filter receive a notification message when someone does this transition (saves this message).
Can View	Submitter	Task submitters with this status can view this message in the transitions history (previously sent messages).
Can Process	Submitter	Task submitters with this status can do this transition (save this message).
Receive Notification	Submitter	Task submitters with this status and enabled email notification for filter receive a notification message when someone does this transition (saves this message).
Can View	Handler	Task handlers with this status can view this message in the transitions history (previously sent messages).
Can Process	Handler	Task handlers with this status can do this transition (save this message).
Receive Notification	Handler	Task handlers with this status and enabled email notification for filter receive a notification message when someone does this transition (saves this message).

Click the **Save** button to save the modified transition matrix.

Remarks

If you don't have editing privileges for a given workflow, you won't be able to change the state transition diagram and processing rules for higher level user groups, but you will be able to create your own (lower level) user status and expand the workflow for that user status. This is done so you can add a new user status to the tracking process.

Example

You can define a mixed behaviors scheme:

You can have one workflow status where the *All* items are selected because you prefer that every one can attend a task and solve it (this is a free attention mode). Maybe your testing team work jointly, where one is the handler but every one can work on a task and move it to another state (for example send it to the development team because a bug has been detected).

You can have another workflow status where the *Handler* items are selected because you prefer that only one (the handler) can attend a task and solve it (this is a restricted attention mode). Maybe in your development team the members work individually, and the handler of a task is the only one that can work on it, and the only one that can move it to another state (for

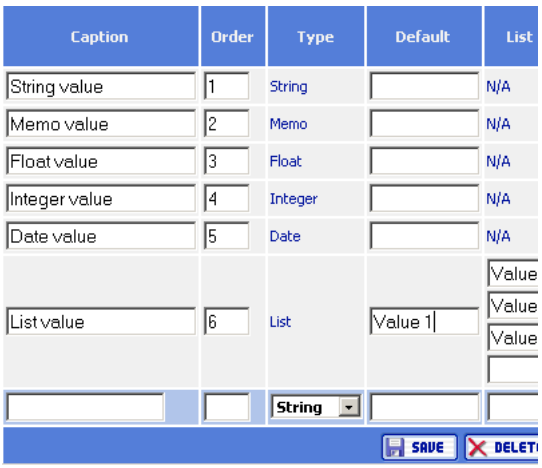
example send it to the testing team when a bug was implemented and now the project has to be tested).

5.9.8 Customize Workflow Project Manager

This topic describes how to define workflow-specific custom fields (**Task Management->Workflow->Customize** tab).

Description

You may want to create some properties such as version, release, platform and others. All these custom fields will be available only for the tasks with the defined workflow.



You have to fill in the following properties:

Property	Description
Caption	Custom field name. It can be *software platform, release, version, customer name* and so on.
Order	An integer value which indicates the position of the field on the user page, the system sorts custom fields by this field.
Default	Default field value (100, for example). For list, the default value should be equal to one of the list items.
List	Contains possible values for dropdown lists. To fill this property enter first list item, then press the **Save** button, then enter the second and save that, etc.
Required	Indicates whether a field value is required or not. The users can't save a task if they do not fill in all required properties.
Type	Custom field type. Please check out next table for more information about available custom field types.
Visible	Indicates whether field is visible in the **Task Header** or not.
Send by email	This field defines whether this property should be included in the email notification and subscription messages sent to a user by email.

There are six possible custom field types:

Type	Description
String	String of symbols
Memo	Text area
Float	Floating point value
Integer	Integer value
Date	Date/Time value
List	Drop-down list with values specified in the **List** field

After all the required fields are filled in, press the **Save** button to save the created or modified custom field. To delete a custom field or dropdown list item select the field or list item and click the **Delete** button.

6 User Management Project Manager

This section describes how to manage users and user groups.

Description

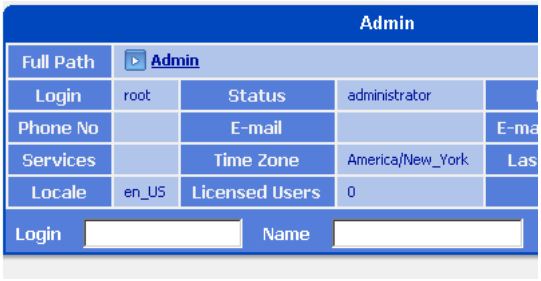
The **User Management** window consists of the following parts.

- The **Main Menu** with the **Project Dropdown** list.
- Information about logged users and their groups.
- **User Header**
- **User Control Area**
- **Status Area**

Using the **Main Menu**, a user can switch between the working modes of the application (**Task Management** and **User Management**).



The **User Header** contains information about the current user.

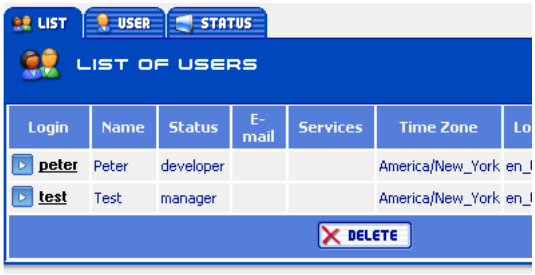


Here you can find the following user properties:

Property	Description
Full Path	In this field information about the position of the current user in the user hierarchy is displayed.
Login	The user's login
Status	The user status (a user group). Each user can belong to only one group.
Name	Contains the user's name.
Phone No	Contains the user's phone number.
E-mail	Contains the user's e-mail.
E-mail for SMS	Here you can specify an additional e-mail. This field is not used at the moment.
Services	Contains the services available only for the current user. This property is used only in TrackStudio Host.

Time Zone	Contains the time zone of the current user. All the data is displayed according to the specified **Time Zone**.
Locale	Contains the locale of the current user. All dates, as well as figures with floating points must fit the specified format.
Last Visited	Contains the date when the current user last logged into the system.
Licensed Users	Contains the number of child users available for the current user. If this value is set to 0, the number of users is not limited. When the system checks on this limitation, it checks not only the closest parent, but also all the upper parents.

The **User Control Area** contains information about the actions available for the current user. The availability of actions depends on the user's privileges.



In the **Status Area** you can find information about the version of the system, when the page was generated and also about how effectively the request cache is used. **Cache Hit Ratio** is estimated separately for the task cache and for the user cache. If the system is functioning normally, both figures exceed 95%.



6.1 User Concepts Project Manager

This topic contains general user concepts.

Description

TrackStudio supports the hierarchies of users. It means that each user can have one or more child users. Every user can also be associated with one or more user groups that can be assigned to the parent user and their child users.

The list of user properties can be found in User Management (see page 62).

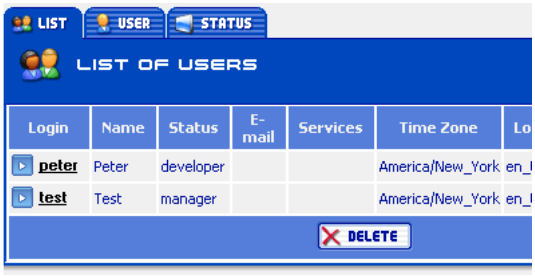
6.2 List of Users Project Manager

This section describes the list of child users of the current user (**User Management->List of Users** tab).

Description

The **List of Users** window allows you to view the current user's list of child users. If the current user has no child users, this window is not available.

For each user the following properties are displayed: **Login, Name, Status, E-mail, Services, Time Zone, Locale**, and **Last Visited Date**. The purpose of the fields is described in User Management (see page 62) .



To delete a user, you must check one or several users and press the **Delete** button. You cannot delete users who are or were the submitters or handlers of tasks. To delete such users you must first delete all the objects they are associated with.

6.3 User User

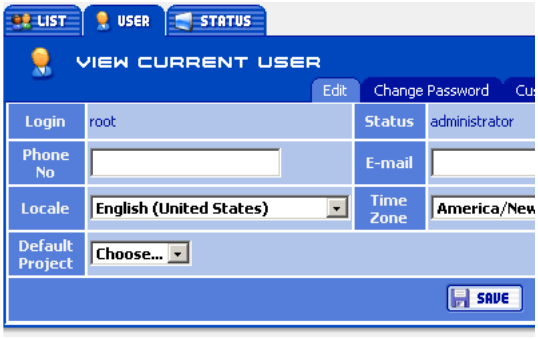
This section describes how to view or modify some important properties of the current user (**User Management->User** tab).

6.3.1 Edit User User

This section describes how to create a new user or modify the existing one (**User Management->User->Edit** tab).

Description

To create a user, you must enter their Login and Name, specify their **Status** and press **Add User**. The detailed description of the user properties can be found in User Management (see page 62).



After modifying the user properties, press either the **Save** button or the **Go to parent** button. When the **Save** button is pressed, the system will save the changes. When the **Go to parent** button is pressed, the system will save the changes and open the **List of Users** tab of the upper level task.

It is better to use the **Save** button if you want to add one user, while the **Go to parent** button is useful when you want to add several users at the same time.

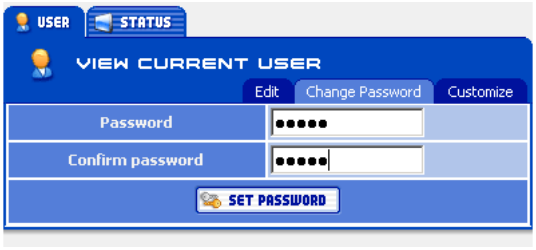
Notes

When you change locale or time zone, you should logout and login again.

6.3.2 Change Password User

This section describes how to change a user's password (**User Management->User->Change Password**).

Description

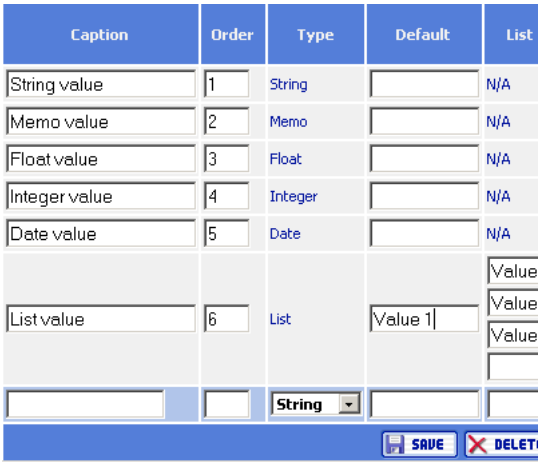


Enter your new password in the same field and enter it again in the **Confirm password** field to confirm it, then press the **Set password** button.

6.3.3 Customize User Project Manager

This section describes how to customize a user's properties (**User Management->User->Customize**).

Description



You have to fill in the following properties:

Property	Description
Caption	Custom field name. It can be *salary, company* and so on.
Order	An integer value which indicates the position of the field on the user page, the system sorts custom fields by this field.
Default	Default field value (100, for example). For list, the default value should be equal to one of the list items.
List	Contains possible values for dropdown lists. To fill this property enter first list item, then press the **Save** button, then enter the second and save that, etc.
Required	Indicates whether a field value is required or not. The users can't save a task if they do not fill in all required properties.
Type	Custom field type. Please check out next table for more information about available custom field types.
Visible	Indicates whether field is visible in the **User Header** or not.

There are six possible custom field types:

Type	Description
String	String of symbols
Memo	Text area
Float	Floating point value
Integer	Integer value
Date	Date/Time value
List	Drop-down list with values specified in the **List** field

After all the required fields are filled in, press the **Save** button to save the created or modified custom field. To delete a custom field or dropdown list item select the field or list item and click the **Delete** button.

6.4 Status Project Manager

This section describes how to configure a user group (**User Management->Status** tab)..

Description

The system level privileges can be assigned using groups (user status). Later on, when you create user accounts, the new users can be assigned to a group and the group privileges will be automatically granted to them.

Remarks

To allow users having new user status to create tasks or messages, you must select the task categories or message types, which they have permitted to create in their workflows with the new status.

6.4.1 List User Status Project Manager

This section describes how to create user status (user group) or view existing ones (**User Management->Status->List**).

Description

General filter properties include:

Property	Description
Select	Use the **Select** hyperlink to select current filter. You can view or edit selected user status.
Status	Status name
Private	Is status private or not. Private status can be assigned only to the owner.
Owner	The user who creates the user status. Only the owner can modify the user status.
Delete	Use this checkbox to select a filter for deletion. Please note, that you can't delete used, foreign or selected user status.

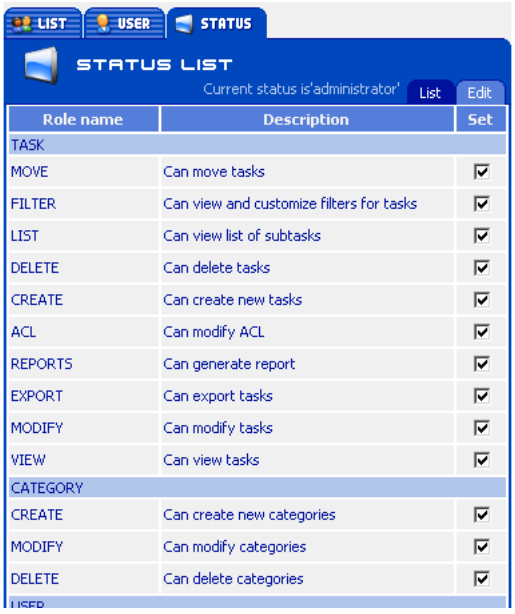


To create new user status enter user status name, set is status private or not and press the **Save** button. To delete a filter check the filter and press the **Delete** button.

6.4.2 Edit User Status Project Manager

This section describes how to modify user status settings (**User Management->Status->Edit**).

Description



Following is a brief synopsis of privilege types and how they control access.

Privileges group	Description
Workflow privileges	Grant users the ability to view, create, modify and delete workflows
Category privileges	Grant users the ability to create, modify and delete task categories.

User UDF privileges	Grant users the ability to view, create, modify and delete custom fields for users.
User privileges	Grant users the ability to view, create, modify and delete users, move users from one team to another and change user password.
Message privileges	Grant users the ability to view, create, modify and delete messages (notes), change task status, resolution and current handler.
Task privileges	Grant users the ability to view, create, modify and delete tasks, move tasks between projects, customize task filters, set user rights for projects and tasks, export tasks and generate reports.
Task UDF privileges	Grant users the ability to view, create, modify and delete custom fields for tasks and projects.
Upload privileges	Grant users the ability to view, create and delete attached files.

To save status description press the **Submit** button.

Notes

You cannot grant any privileges if you do not have them yourself.

7 Open API Developer

This topic describes TrackStudio Open API.

Description

Open API is used for modifying the present functionality and for enhancing TrackStudio as well as for integrating it with other applications.

Unlike other similar software, TrackStudio has microkernel architecture. It means that there is a small kernel responsible for both the interaction of components and the system configuration, while practically all the system functionality is based on adapters with well-defined interfaces. Adapters can be arranged in a pipeline when the result of some method in one adapter is passed over to the next adapter in the pipeline. The application functionality can be modified or enhanced without any changes in the initial code of TrackStudio. Pipeline architecture gives a simple solution to such tasks as audit, additional security checks, method parameter logging and return value modifying.

Adapters are subdivided into external and kernel ones. External adapters are used for integrating TrackStudio with other applications, while kernel ones provide the basic functionality, such as working with tasks, messages, workflows, etc.

Using external adapters, you can realize the following functionality:

- Additional methods for user authentication (external database, LDAP)
- Arranging the recipient list and the format of email notifications (for example, the email notifications can be sent in the form of plain text messages).
- Data export to various formats.
- Data import from email, integration with CVS, customer support tasks.
- Various recurrent operations, for example, the recurring export of data from TrackStudio into other systems.

Using kernel adapters, you can modify or enhance practically the entire system functionality. For example, you can add a trigger that will respond to a certain transition in a workflow.

7.1 Adapter Development Developer

This topic contains adapter development overview.

Description

An adapter is a class realizing the `gran.app.adapter.Adapter` interface. Each subsystem in TrackStudio has its own interface, inherited from

`gran.app.adapter.Adapter`. For instance, to implement an export adapter, you must implement the class, realizing the `gran.app.export.ExportAdapter` interface. Adapters are stateless components, i.e. they do not have an internal state and do not remember the history of the previous calls.

The list of the loadable adapters can be found in trackstudio.adapter.properties, for example:

```
# External adapters

adapter.auth gran.app.adapter.auth.SimpleAuthAdapter
adapter.export gran.app.adapter.export.XMLExportAdapter
adapter.email gran.app.adapter.email.HTMLFilterNotifyAdapter
adapter.pop3 gran.app.adapter.pop3.BasePOP3Adapter
```

If you need to execute a pipeline of some adapters (implementing the same interface) to perform some operation, you must put them in one line using ';' as a separator, for example:

```
adapter.service gran.app.adapter.service.BasePOP3ServiceAdapter;
                gran.app.adapter.service.BaseFilterServiceAdapter
```

If there are two identical adapters in the list, only the first of them will be executed. If there is an adapter not realizing the required interface in the list, it is not loaded.

7.2 Adapter Structure Developer

This section describes common adapter structure.

Description

TrackStudio takes each type of adapter as consisting of three components: the adapter interface, the proper adapter and AdapterManager realizing the pipeline.

Let's take the realization and interaction of the system components in the work of adapters:

1) Interface. The adapter interface must extend `Adapter` or `KernelAdapter`. The following requirements are demanded of the method signature:

- The name of the interface must be `SomethingAdapter`
- Every method must either throw `GranException` or not throw an exception.
- The names of methods must end with `Impl`
- If a method returns a value, this method must have a parameter of a returning type named result, which must come last in the list of parameters. For example,

```
  public boolean authorizeImpl(User user,
                               String password,
                               boolean result)
                throws GranException;
```

- Persistent objects (with a rare exception) are passed either by their string identifier or in the collections `java.util.Collection`, `java.util.LinkedList`, etc. To continue working in the case of the ID passing of the object, you must open a Hibernate session and load the object. When using the object list you do not have to open a session (remember that in this case each object in the collection must be Hibernate-initialized). If executing an adapter results in changes in the persistent object, you must always open and close the session. It is also recommended to use transactions in this case.

Interface example:

```
// $Id: OpenAPI.dtx,v 1.5 2003/07/08 14:33:58 maximkr Exp $
package gran.app.adapter.template;

import gran.app.adapter.Adapter;
import gran.exception.GranException;

public interface TemplateAdapter extends Adapter // or KernelAdapter
{
    public String methodThatReturnSomethingImpl(String param,
                                                String result)
                  throws GranException;

    public void methodThatReturnNothingImpl(String param)
                throws GranException;
}
```

2) An adapter has the following structure:

```
// $Id: OpenAPI.dtx,v 1.5 2003/07/08 14:33:58 maximkr Exp $
package gran.app.adapter.template;

import gran.exception.GranException;
import gran.tools.Logger;

public class BaseTemplateAdapter implements TemplateAdapter
{
    private static Logger log = new
            Logger("gran.app.adapter.template.TemplateAdapter");

    public boolean init()
    {
        return true;
    }

    public String getDescription()
    {
        return "Base Template Adapter";
    }

    public String methodThatReturnSomethingImpl(String param,
                                                String result)
                  throws GranException
    {
        return param + " OK";
    }

    public void methodThatReturnNothingImpl(String param)
                throws GranException
    {
        return;
    }
}
```

Within an adapter, methods can be called only through `AdapterManager.` Direct calling *Impl-methods is not recommended as it may cause problems when enhancing the system.

3) SomeAdapterManager controls the lists of adapters supporting the defined interface and is responsible for the correct passing of the parameters. The realization of SomeAdapterManager may have the following structure:

```
// $Id: OpenAPI.dtx,v 1.5 2003/07/08 14:33:58 maximkr Exp $
package gran.app.adapter.template;

import java.util.Collection;
import java.util.Iterator;
import gran.exception.GranException;

public class TemplateAdapterManager
{
    private Collection am = null;

    public TemplateAdapterManager(Collection adapters)
    {
        am = adapters;
    }


    public String methodThatReturnSomething(String param)
                  throws GranException
    {
        String result = null;
        for (Iterator iter = am.iterator(); iter.hasNext();) {
            TemplateAdapter adp = (TemplateAdapter) iter.next();
            result = adp.methodThatReturnSomethingImpl(param, result);
        }
        return result;
    }

    public void methodThatReturnNothing(String param)
                throws GranException
    {
        for (Iterator iter = am.iterator(); iter.hasNext();) {
            TemplateAdapter adp = (TemplateAdapter) iter.next();
            adp.methodThatReturnNothingImpl(param);
        }
    }
}
```

The system enhancement is carried out through classes realizing the existing interfaces (for example, gran.app.adapter.export.ExportAdapter). At the same time you do not have to modify the initial system code, the adapter interface or AdapterManager.

To call the adapters, a singleton class AdapterManager is used. This class stores the list of all adapters available on the system and allows registering new adapters in the system. To call a method (e.g. for exporting), you must execute the following:

```
AdapterManager.getInstance().getExportAdapterManager()
                            .export(taskid, userid);
```

Sample adapters:

1) An adapter realizing the user authentication.

2) A debugging adapter logging all the information about the called adapters and their parameters.

3) An adapter for exporting data.

7.3 Building Developer

This topic describes building TrackStudio from the source code.

Description

If you have purchased TrackStudio Enterprise with the source code, you can build the application from it. To do that, you will need:

- ant 1.5 or higher
- JDK 1.4 or higher. Please, note that though TrackStudio Enterprise can work under JDK 1.3.1, you need JDK 1.4 to build it.

To build the Standalone distribution with JRE, execute the following command:

```
> ant  sadist-jre
```

To build the Standalone distribution without JRE, execute the following command:

```
> ant  sadist
```

To build the WAR distribution, execute the following command:

```
> ant wardist
```

To speed up the process of building, you can use jikes, for example:

```
ant -Dbuild.compiler=jikes sadist-jre
```

If you are experience any problems in process in building the application, please, contact us.

Index

A

Access Control 48
Adapter Development 70
Adapter Structure 71

B

Building 74

C

Categories 52
Change Password 65
Concepts 1
Configuration Files 12
Creating a New Bug 25
Creating a New Project 23
Creating a New User 22
Customize Task 43
Customize User 65
Customize Workflow 60

D

Database Configuration 14
DB2 Configuration 16
Debugging Errors 19

E

Edit Category 53
Edit Filter 35
Edit Task 42
Edit User 64
Edit User Status 68
Export 44

F

Filter Subscription 38
Filters 33
Firebird Configuration 17

G

Getting Started 21

H

How to Install an SSL Certificate. 10
HypersonicSQL Configuration 14

I

Installation 5

L

List of Users 64
List Reports 50
List User Status 67

M

Message Types 57
Messages 45
MS SQL Configuration 16

N

Notify by Email 39

O

Open API 70
Oracle Configuration 15

P

PostgreSQL Configuration 15
Priorities 55

Windows Service Management 9
Workflow Management 51
Workflows 54

R

Report Types 49
Reports 49
Requirements 5

S

States 56
Status 67
Submit by Email 47
Subtasks 32

T

Task 42
Task Concepts 29
Task Management 28
TrackStudio Server Manager 7
TrackStudio/SA Configuration 6
TrackStudio/WAR Configuration 11
Transitions 58

U

UNIX-Specific Notes 17
Update from 2.5 18
Uploads 48
User 64
User Concepts 63
User Management 62

V

View Filter 34
View Reports 51

W

What's New 4