TrackStudio 2.8.1 Documentation

Table of Contents

1 Concepts 1

2 What's New 4

3 Installation 7

3.1 Requirements 7

3.2 TrackStudio/SA Configuration 8

3.2.1 TrackStudio Server Manager 9

3.2.2 Windows Service Management 12

3.2.3 How to Install an SSL Certificate. 13

3.2.4 Importing and Exporting the Database 14

3.3 TrackStudio/WAR Configuration 16

3.4 Database Configuration 16

3.4.1 HSQLDB Configuration 16

3.4.2 PostgreSQL Configuration 17

3.4.3 Oracle Configuration 17

3.4.4 DB2 Configuration 18

3.4.5 MS SQL Configuration 18

3.4.6 Firebird Configuration 19

3.5 Configuration Files 19

3.6 Authorize Through the LDAP 23

3.7 External User Self-Registration 24

3.8 UNIX-Specific Notes 26

3.9 Update from 2.7 27

3.10 Debugging Errors 28

4 Getting Started 30

4.1 Creating a New User 31

4.2 Creating a New Project 32

4.3 Creating a New Bug 34

4.4 Creating a New Filter 36

4.5 Enabling E-Mail Notification 37

5 Task Management 39

5.1 Task Concepts 40

5.2 Subtasks 43

5.3 Filters 45

5.3.1 View Filter 46

5.3.2 Edit Filter 47

5.3.3 Filter Subscription 53

5.3.4 Notify by Email 54

5.4 Task 55

5.4.1 Edit Task 56

5.4.2 Customize Task 57

5.4.3 E-Mail Import 58

5.4.4 Export 59

5.5 Messages 60

5.6 Uploads 61

5.7 Access Control 62

5.8 Reports 63

5.8.1 Report Types 63

5.8.2 List Reports 65

5.8.3 View Reports 66

5.9 Workflow Management 67

5.9.1 Categories 67

5.9.2 Edit Category 68

5.9.3 Workflows 70

5.9.4 Priorities 71

5.9.5 States 71

5.9.6 Message Types 72

5.9.7 Transitions 73

5.9.8 Customize Workflow 76

6 User Management 78

6.1 User Concepts 79

6.2 List of Users 80

6.3 User 80
6.3.1 Edit User 80
6.3.2 Change Password 81
6.3.3 Customize User 82
6.4 Status 83
6.4.1 List User Status 83
6.4.2 Edit User Status 84
7 Advanced Topics 87
7.1 Full Text Search 87
7.2 E-Mail Notification 89
7.3 E-Mail Submission 93
7.3.1 Task Submission 94
7.3.2 Plain Message Submission 95
7.3.3 Form-Based Message Submission 96
7.4 Calculated Custom Fields 98
7.5 Data Analysis 103
7.6 Internationalization 105
7.6.1 Character Encoding 106
7.6.2 Localizing the Interface 106
8 Open API 109
8.1 Adapter Development 109
8.2 Adapter Structure 110
8.3 Building 113
9 Index 114

TrackStudio 2.8.1 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. Calculated custom fields are especially useful in filters, email notification rules, and distribution reports.

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 TrackStudio 2.8

User Interface

- Link-friendly URLs . You don’t have to log in to go to a TrackStudio page. Just open the necessary page and bookmark it in your browser. You can send your colleague a direct link to a report or to an uploaded file. From now on, you will never get a message saying your session has expired. TrackStudio will automatically log you in. You can upgrade your TrackStudio version on the server or even restart the server – most users will not even notice the break in its functioning.
- All forms are printer-friendly. Now you can print out any form, just use **File->Print** in your browser.
- User default project. Each user can be assigned a project that will be available for him right after he/she logs in.
- On-line help.
- Quick task search by keyword. You can search for a task by a keyword among all tasks, among tasks of a particular project, or among closed tasks. You can search for tasks faster still by combining the full text search with other filtering conditions.

Server Manager

- Configures LDAP connection
- Configures external user self-registration rules
- Supports transferring data between DBMSs via XML import/export

Task Management

- Task Numbers. Now when you create a task, it is given not only GUID, but also a task number. You can quickly edit a particular task if you know its number.
- Supports bulk edit for task budget, deadline, and priority.
- Long text description and message support. Now the maximal length of a task description or a bug-note is not limited by the peculiarities of a DBMS.
- Modify now means "modify after creating"
- The beginning of the task description is added to the task header.

Custom Fields

- BeanShell-based calculated custom fields. Calculated custom fields allow you to create very comprehensive filters, email notification rules and reports. Calculated custom fields, together with distribution reports, allows (↗ see page 103) you to find out what percentage of user requests are resolved within one day, or which developer fixes bugs most quickly.

Workflow

- Allows you to enter color hex in the **States** page
- Easier to fill the **Edit Category** and **Transition** pages
- **Submitter&Handler** settings in the **Categories** and **Transition** rules
- Supports the copy operation for workflows. You can copy a complex workflow with just one mouse click.

Filters

- Filter by submitter or handler status (group)
- **Last x messages** feature in the message filter
- Full text search
- Sort by multiple columns
- Safely delete the used filters
- The manager can edit subordinate filters.
- Supports the copy operation for filters. You can copy a filter with just one mouse click.

Email submission

- Flexible project-specific email submission rules. You can automatically process all incoming mail to the support email address. If an e-mail message meets the specified criteria, it is added to the system; if not, it can be deleted or forwarded to the specified address.
- Import email messages as new tasks or messages. The incoming e-mail messages can be added to the system as tasks or messages depending on the given configuration.

Email notification

- HTML and plain text email notifications. The plain text email notification is recognized in addition to the HTML email notification.
- The manager can edit subordinate e-mail notification rules.

Integration

- LDAP integration

Security

- LDAP support
- Configurable external user self-registration rules. You can specify who will be the manager for the newly created users, whether it is necessary either to create a separate project for each user or to give him/her access to those already existing, and a lot more.

The following new features are introduced in TrackStudio 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 HSQLDB 1.7.2

Application server	**One of the following:** BEA Weblogic 7.0-8.1 IBM WebSphere 5 Sun ONE Application Server 7 Tomcat 4.x JBoss 3.0.x Jetty 4.1 Caucho Resin 2.1.11
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 BEA JRockit 8.1
Web Browser	**One of the following:** Microsoft Internet Explorer, version 5.0 to version 6.0 Mozilla 1.1 Netscape 7.1
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 HSQLDB (**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**

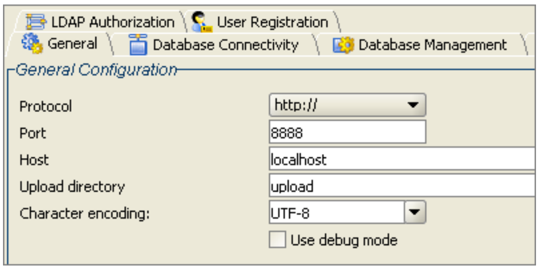
3.2.1 TrackStudio Server Manager Administrator

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

Description

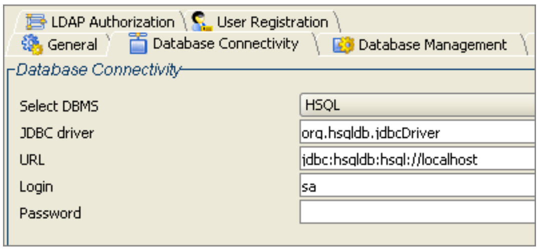
Launch TrackStudio Enterprise Server Manager (**sman.exe**). Wait while it's loading, TrackStudio Enterprise Server Manager window will appear.

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.



Protocol: http://
Port: 8888
Host: localhost
Upload directory: upload
Character encoding: UTF-8
□ Use debug mode

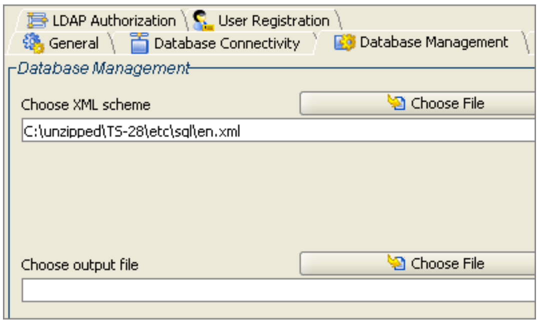
2. Database Connectivity. Here you can configure database connection options: database URL, JDBC driver, Login and Password. In the JDBC driver field there is a default JDBC Driver for HSQLDB that you shouldn't usually need to modify. After you have specified all the necessary connection properties, 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.



Select DBMS: HSQL
JDBC driver: org.hsqldb.jdbcDriver
URL: jdbc:hsqldb:hsql://localhost
Login: sa
Password:

3. Database Management. On the **Database Management** tab you should enter path to database creation 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. To export the database you should click the **Export Database** button.



Database Management

Choose XML scheme

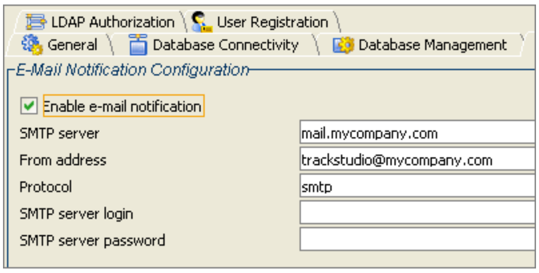
C:\unzipped\TS-28\etc\sql\en.xml

Choose output file

4. E-Mail 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:

- **SMTP Server** - SMTP host address;
- **From address** - when sending notification, this address will be in the **from** field;
- **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.



E-Mail Notification Configuration

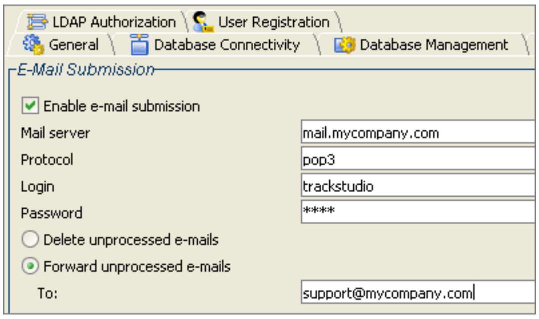
☑ Enable e-mail notification

SMTP server	mail.mycompany.com
From address	trackstudio@mycompany.com
Protocol	smtp
SMTP server login	
SMTP server password	

5. E-Mail Submission (optional). Here you can configure email submission properties. To turn this feature on, check **Enable e-mail submission**. You can edit the following properties:

- **Mail server** - POP3 host;
- **Protocol** - mail storing protocol, only *pop3* supported;
- **Login;**
- **Password;**
- **Delete unprocessed e-mails;**
- **Forward unprocessed e-mails;**

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



E-Mail Submission

☑ Enable e-mail submission

Mail server: mail.mycompany.com

Protocol: pop3

Login: trackstudio

Password: ****

○ Delete unprocessed e-mails

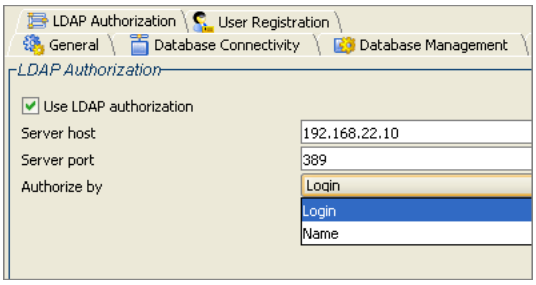
● Forward unprocessed e-mails

To: support@mycompany.com

6.LDAP Authorization (optional). On this tab you can set the options for the LDAP server used to authorize users. For the authorization of users via the LDAP server, click the **Use LDAP authorization** checkbox and specify the following parameters:

- **Server host** – the LDAP server host;
- **Server port** – the LDAP server port;
- **Authorize by** – specify user by name or by login.

You can test the connection to the LDAP server by pressing the **Test Connection** button. A more detailed description of authorizing a user via the LDAP server is given in the topic Authorize Through the LDAP (see page 23).



LDAP Authorization

☑ Use LDAP authorization

Server host: 192.168.22.10

Server port: 389

Authorize by: Login

Login

Name

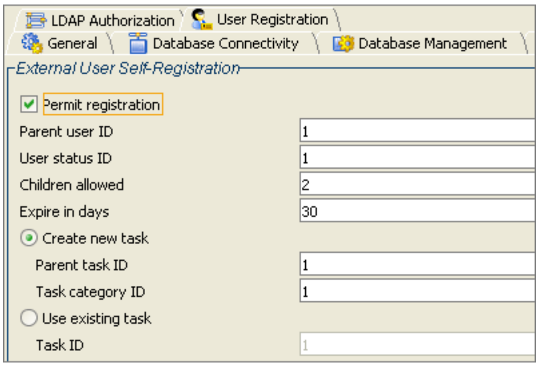
7.User Registration (optional). Here you can set the options for external user self-registration. In order for users to register themselves in the system, click the **Permit registration** checkbox and specify the following parameters:

- **Parent user ID** – the ID of the parent user;
- **User status ID** – the ID of the new user status (group);
- **Children allowed** – the maximum number of users the new user may create;
- **Expire in days** – the interval of time after which the user's login expires;

If you check the **Create new task** radio button you will be able to specify the following:

- **Parent task ID** – the ID of the task where a new task for the user is created;
- **Task category ID** – the ID of the task category;

If you check the **Use existing task** radio button you should specify the **Task ID**: the ID of the task the new user is granted access to.



External User Self-Registration

☑ Permit registration
Parent user ID: 1
User status ID: 1
Children allowed: 2
Expire in days: 30
● Create new task
Parent task ID: 1
Task category ID: 1
○ Use existing task
Task ID: 1

A more detailed description of the parameters and their configuration is given in the topic External User Self-Registration (see page 24).

8. 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**.

9. 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 HSQLDB (**hwrapper.conf**) as services. This wrapper can also be used to run Jetty and HSQLDB 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.
```

HSQLDB

1) To run an HSQLDB 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  | HSQLDB installed.
```

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

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

3.2.4 Importing and Exporting the Database Administrator

To transfer the data stored in the TrackStudio database to another DBMS you can use the feature which allows the data to be exported into an XML file.

Description

This feature is available on the **Database Management** tab of the Server Manager application (available in TrackStudio/SA). To export the data into XML, perform the following steps:

1. Stop TrackStudio, if it is running.

2. Set the options for the connection with the database that the export will be performed from. This can be done on the **Database Connectivity** tab.

3. On the **General** tab specify the character encoding of the file the data will be exported from.

4. Specify the name of the file the data will be exported to.

5. Press the **Export Database** button and wait till the export process is finished.

As the result of the export process, you will have an XML file containing all the information from all the tables of the TrackStudio database. You can edit and view it using any text editor. It can also be imported to create a new database.

You can import the data from XML into a database only in the process of creating a new database in TrackStudio. To import the data, perform the following steps:

1. Stop TrackStudio, if it is running.

2. Set the options for the connection with an empty database containing no old tables or data from TrackStudio.

3. Specify the name of the imported XML file in the **Choose XML scheme** field.

4. Press the **Create Database** button and wait till the process of the database creation is finished.

The import is carried out in the following way:

1) Database tables are created.

2) SQL INSERTs based on the XML file are generated and they insert the data into the database.

3) Indexes and constraints are created.

Remarks

The import/export feature can be used to transfer data only between two similar versions of TrackStudio. In the event that you have a database from an old version of TrackStudio that you want to transfer to another DBMS you should first upgrade the database to the last version.

Notes

The SA and WAR versions use the same database scheme and differ only in the distributed components – no special actions are required to transfer the data between the WAR and SA versions.

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 2.7 (⧉ see page 27)

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**

3.4 Database Configuration Administrator

This section contains DBMS-specific configuration notes.

3.4.1 HSQLDB Configuration Administrator

This section describes how to configure HSQLDB.

Description

1. Start HSQLDB:

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

2. Execute **trackstudio-hsql.sql** (TrackStudio/WAR) or use the **Create Database (see page 9)** 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 9)** 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 9) 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 9)** 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 9)** 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 9)** 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 Configuration Files Administrator

This section describes TrackStudio configuration files.

Description

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

- **trackstudio.properties**, a general-purpose configuration file
- **trackstudio.hibernate.properties**, database configuration file

- **trackstudio.mail.properties**, e-mail configuration file
- **trackstudio.register.properties**, external user self-registration configuration file
- **trackstudio.ldap.properties**, LDAP 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


# Logout URL for your site.

#trackstudio.logoutURL http://localhost:8888/TrackStudio


# Upload directory. Should exists and be accessible
# We suggest you use absolute (not relative) path here
# For example, /mnt/upload or c:\\TrackStudio\\upload

trackstudio.uploadDir upload


# Set yes to log debug information

trackstudio.debug no


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

trackstudio.encoding UTF-8


# 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**)

```
###############################
# E-Mail notification options #
###############################
```

```
# Enable email notification, yes/no

trackstudio.sendMail no


## Mail transport protocol. Should be smtp

mail.transport.protocol smtp


## SMTP server

mail.smtp.host mail.mycompany.com


## Send email notifications from this address

mail.from trackstudio@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


## Connection timeout properties

mail.smtp.timeout 10000
mail.smtp.connectiontimeout 10000


#############################
# E-Mail submission options #
#############################

# Enable e-mail submission, yes/no.
# Please note, that this option requires email notification

trackstudio.FormMailNotification no


## 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 trackstudio


## POP3 server password

mail.pop3.password pass


## Delete or forward bad messages

mail.pop3.forward no
```

```
## Forward e-mail address

mail.pop3.fwdaddress
```

4. Configure external user self-registration properties (**trackstudio.register.properties**)

```
###########################################
# External user self-registration options #
###########################################

# Enable externa user self-registration, yes/no

trackstudio.registration no


# The ID number of the user parent to new users

registration.parentuser 1


# The status ID of the new user

registration.userstatus 1


# The maximum amount of sub-users that the new user can create

registration.children 2


# Number of days after which user login expired

registration.expireindays 30


# Specifies whether a new task is created for the new user
# or whether he/she is granted access to an existing task.
# Possible values: yes, no

registration.createtask yes


# The ID of the task that will be the parent to a new
# task of the user.
# Effect only with registration.createtask yes

registration.parenttask 1


# The ID of the category for the new task
# Effect only with registration.createtask yes

registration.taskcategory 1


# The ID of the existing task the new user is granted
# access to. In this case no new task is created.
# Effect only with registration.createtask no

registration.accesstask 1
```

5. Configure LDAP authorization properties (**trackstudio.ldap.properties**)

```
##############################
# LDAP configuration options #
##############################

# Use LDAP authorization, yes/no

trackstudio.useLDAP no
```

```
# LDAP server port

ldap.port 389


# LDAP server port host IP

ldap.host 192.168.22.10


# Authorize by name or login ?

ldap.authby login
```

3.6 Authorize Through the LDAP Administrator

A lot of organizations use the LDAP directory to store user accounts. TrackStudio can be configured to authorize users by their accounts on the LDAP server.

Description

The LDAP settings are stored in the file **trackstudio.ldap.properties**, and are also available via the Server Manager on the **LDAP Authorization** tab. You can use the following configuration parameters:

trackstudio.useLDAP – specifies whether the authorization on the LDAP server is used. Possible values: **yes, no**. If the parameter is set to **yes**, the user authorization on the LDAP server will be performed along with the usual authorization in the TrackStudio system.

ldap.host – specifies the LDAP server address (e.g. *192.168.22.10*).

ldap.port – specifies the server port (e.g. *389*).

ldap.authby – specifies which of the user parameters is used for the authorization on the server. It can take one of two values: **name** or **login**. It shows which of the TrackStudio user parameters is used as the user name or user login when authorizing on the LDAP server.

Even if you use LDAP authorization, you will have to register a new user in TrackStudio first. This new user login (or name) must be exactly the same as those on the LDAP server. By default, the user login is used for the authorization, so if you want to use the user name, make sure that the **ldap.authby** parameter is set correctly. A user can log into the system if his password matches the one stored in the DB or the one specified in LDAP. To avoid the authorization through the local database, you should remove **gran.app.adapter.auth.SimpleAuthAdapter** from the pipeline in the file **trackstudio.adapter.properties**.

Notes

When you change the password under the **Change Password** tab, the password is changed in the database, but not the LDAP.

Example

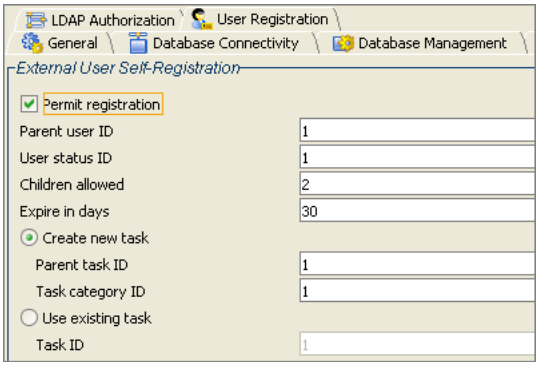
To use the MS Active Directory Service, the authorization must be performed by name. The user's name in TrackStudio and in the Active Directory Service must also coincide. When you enter your user login in TrackStudio, the system will search for the your user name and will use this name for the LDAP authorization.

3.7 External User Self-Registration Administrator

TrackStudio can be configured in such a way that the new user registration will be possible without the participation of the system administrator. In that case users can register with the system on their own and gain access to certain tasks.

Description

The registration settings are stored in the file **trackstudio.register.properties**, and are also available via the Server Manager on the **User Registration** tab.



External User Self-Registration

☑ Permit registration

Parent user ID	1
User status ID	1
Children allowed	2
Expire in days	30
● Create new task	
Parent task ID	1
Task category ID	1
○ Use existing task	
Task ID	1

You can use the following configuration parameters:

trackstudio.registration – specifies whether the external user registration is allowed. Possible values: **yes, no**. If the parameter is set to **yes**, a user sees the additional **Register** button on the login page. Using this button he/she can access the registration page. If the parameter is set to **no** the user will not be able to go to the registration page.

The following parameters for a new user are available:

registration.parentuser – specifies the ID number of the user parent to new users. For example, for new users to be able to register as child users of the *root* user, you should set this field to 1 (Admin ID).

registration.userstatus – specifies the status ID of the new user. For example, the ID number of the *manager* status is 1, while the ID number of the *tester* status is 3.

registration.children – specifies the maximum amount of sub-users that the new user can create.

registration.expireindays – specifies the number of days after which the created account expires. If the parameter is not specified or is set to *0* there will be no time limitations for the new users to use their accounts. In that case the account will expire when/if one of the parent users' accounts expires.

You should specify the project or task the new user has access to:

registration.createtask – specifies whether a new task is created for the new user or whether he/she is granted access to an existing task. Possible values: **yes, no**. The **yes** value is usually used when organizing a hosted service, when tasks for different self-registered users should be independent. In this case a new task is created and the new user is granted access rights to it. The **no** value is used to allow customers to register with the system and to enter the information about bugs for an existing project. The ID of the latter is specified in **registration.accesstask**.

registration.parenttask – specifies the ID of the task that will be the parent to a new task of the user.

registration.taskcategory – specifies the ID of the category for the new task. For example, the ID of the *project* category is 1, while the ID of the *bug* category is 2.

registration.accesstask – specifies the ID of the existing task the new user is granted access to. In this case no new task is created.

The status specified in **registration.userstatus** should be available for the user specified in registration.user. The category specified in **registration.taskcategory** should be available for the task specified in **registration.parenttask**. The values of all parameters are checked at the system start. If any parameters are incorrect TrackStudio will stop running. You can learn about what caused the error from the message text. To solve the problem you must correct the value of the invalid parameter and restart TrackStudio.

Example

Example 1. Suppose we need to organize a hosted service. To do that, new users with the *manager* status (ID 1) should register with the system. Their parent user should be *Admin* (ID 1), they should be allowed to create 2 sub-users and to use their account for 30 days, and they should have their own project for creating subtasks.

In this case the parameters should be set as follows:

```
trackstudio.registration yes
registration.parentuser 1
registration.userstatus 1
registration.children 2
registration.expireindays 30
registration.createtask yes
registration.parenttask 1
registration.taskcategory 1
```

Example 2. Suppose we need to grant customers access to the information about bugs in the system being developed now. To do that, new users should register with the system under the *tester* status (ID 3) and have *Admin* (ID 1) as their parent user. They should not be allowed to create sub-users, there should be no time limitations to the use of their account, and they should have access to the project with ID number *B01AE0FC4191AF12E0306BD990BC046A*.

In this case the parameters should be set as follows:

```
trackstudio.registration yes
registration.parentuser 1
registration.userstatus 3
registration.children 0
registration.expireindays 0
registration.createtask no
registration.accesstask B01AE0FC4191AF12E0306BD990BC046A
```

3.8 UNIX-Specific Notes Administrator

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.

3.9 Update from 2.7 Administrator

This section describes how to update from TrackStudio 2.7.

Description

To upgrade the system from TrackStudio 2.7 to TrackStudio 2.8, 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 2.7 until possible problems with TrackStudio 2.8 are solved.

2) Unpack the archive with TrackStudio 2.8 to a separate directory.

3) Stop the TrackStudio 2.7 instance.

4) If you are using the standalone version, run sman.exe from TrackStudio 2.8, go to the **Database Management** tab, 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 2.7 until the problem is solved.

Notes

- If you have been using an older version of TrackStudio, you should upgrade your database to version 2.7 first using the TrackStudio 2.7. 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.

- It is necessary to re-save those filters, that perform ordering by custom fields.
- TrackStudio 2.8 checks the logical correction of data and performs the re-enumeration of tasks at the startup. It can take several minutes, please, be patient.
- 2 minutes after the first launch TrackStudio performs re-indexing of the database. It can take 5-10 minutes. The performance can be low during the re-indexing, and the full text search will not work.
- XML Import/Export allows you to transfer the data of TrackStudio 2.8 between various DB, but it cannot be used to transfer data between various versions of TrackStudio.
- In version 2.8 **TASK_CREATE** role is enough to create tasks; you do not need to set **TASK_MODIFY**.
- To bulk edit the priority, deadline, or estimated time (budget), your users should have **MESSAGE_SCHEDULER** role.
- The functioning of unsupported browsers is blocked in TrackStudio 2.8.
- After you create a task, you can't change its category in TrackStudio 2.8.

3.10 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, JDK/JRE, 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 Log As…**

If you are using TrackStudio/WAR, turn on the option **trackstudio.debug yes** in the file **trackstudio.properties**. In this case debug logs are sent to stdout of the application servers and can be redirected to file. The information about the displayed error messages that is sent to stderr (and can be redirected too) is also required.

If you are using TrackStudio/SA, you should mark the checkbox **Use debug mode** on the **General** tab in the Server Manager. In this case, debug logs and error messages will be displayed in the server manager window. To save the logs, select **File->Save Log As**.

Notes

Displaying debug logs in the Server Manager window can cause serious performance degradation. Displaying debug logs does not seriously affect the performance of TrackStudio/WAR or TrackStudio/SA when it is running as a windows service.

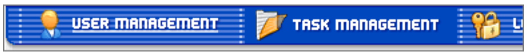
4 Getting Started User

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.

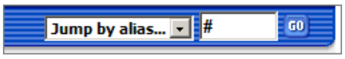
Description

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.

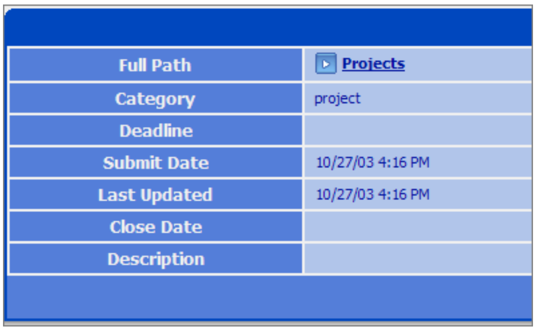
The menu in the upper part of the screen lets you switch between the working modes of the system - **User Management** and **Task Management**.



In the right-hand part there is a dropdown list allowing you to quickly select the current project. Input field allows you to jump to the necessary task if you know its number.

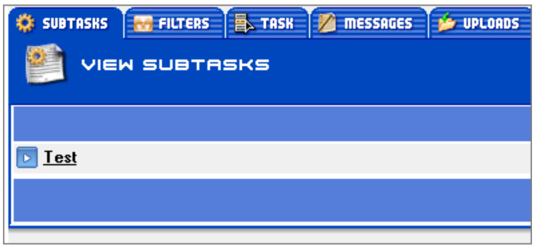


In the **Task Management** mode, the information about the current task is displayed in the central part of the screen.

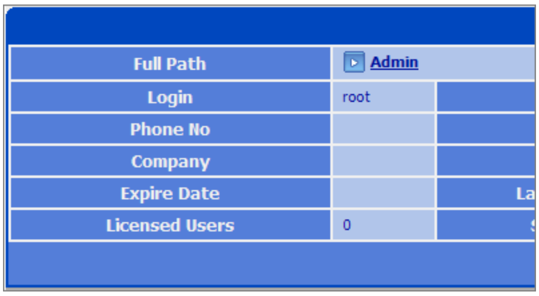


Full Path	Projects
Category	project
Deadline	
Submit Date	10/27/03 4:16 PM
Last Updated	10/27/03 4:16 PM
Close Date	
Description	

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.

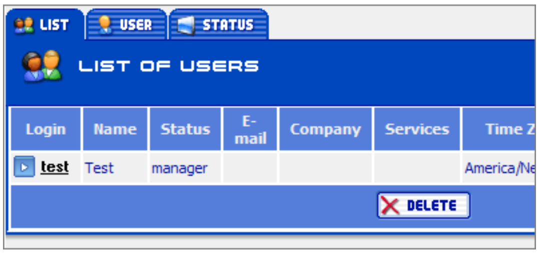


In the **User Management** mode, the information about the current user is displayed in the central part of the screen.



Full Path	Admin	
Login	root	
Phone No		
Company		
Expire Date		La
Licensed Users	0	

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.



Login	Name	Status	E-mail	Company	Services	Time Z
test	Test	manager				America/Ne

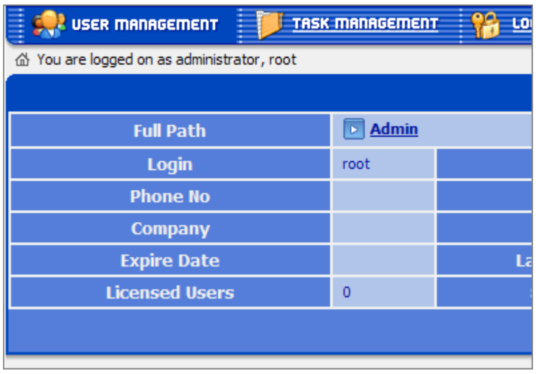
4.1 Creating a New User

User

This topic describes how to create a user.

Description

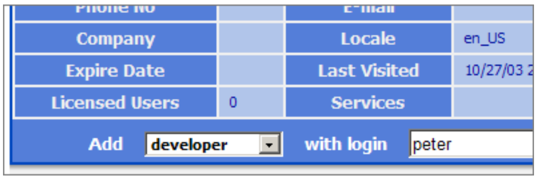
Log into the system as *root/root*. After logging into the system, you are in the **Task Management** mode, which is used for managing tasks. To create a user, select the **User Management** mode in the upper menu. The current user is *Admin* with the login *root*.



You are logged on as administrator, root

Full Path	Admin	
Login	root	
Phone No		
Company		
Expire Date		La
Licensed Users	0	

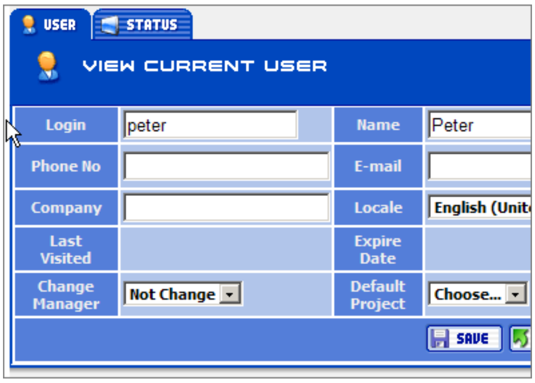
To create a subordinate user: enter the user group (*developer*); the login (e.g. *peter*); the name (e.g. *Peter*) and press the **Add User** button. Note that the newly created user has now become the current user and all further actions (editing, changing the password) will be applied to that user.



Phone No		E-mail	
Company		Locale	en_US
Expire Date		Last Visited	10/27/03 2
Licensed Users	0	Services	

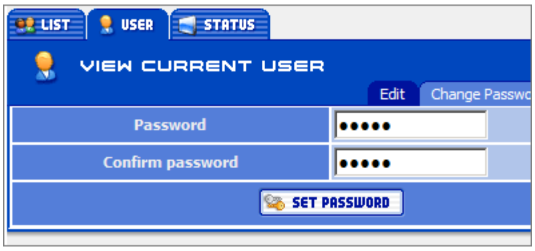
Add developer with login peter

In the form that appears fill in the user's **E-Mail**, select the **Locale** and the **Time Zone**. The default project field allows you to specify the project which will be available to the user right after he/she logs in. Access rights for the default project are automatically granted. After making the changes, press **Save**.



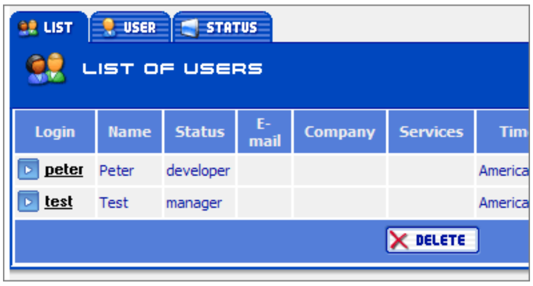
Login	peter	Name	Peter
Phone No		E-mail	
Company		Locale	English (Unit
Last Visited		Expire Date	
Change Manager	Not Change	Default Project	Choose...

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



Password	•••••
Confirm password	•••••

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.



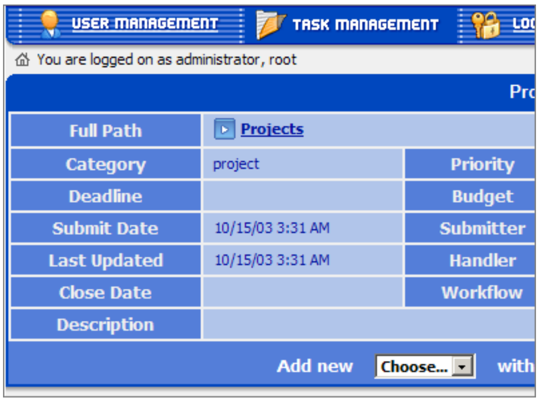
Login	Name	Status	E-mail	Company	Services	Tim
peter	Peter	developer				America
test	Test	manager				America

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*.

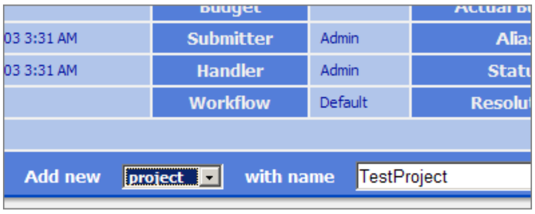


You are logged on as administrator, root

Full Path	Projects		
Category	project	Priority	
Deadline		Budget	
Submit Date	10/15/03 3:31 AM	Submitter	
Last Updated	10/15/03 3:31 AM	Handler	
Close Date		Workflow	
Description			

Add new Choose... with

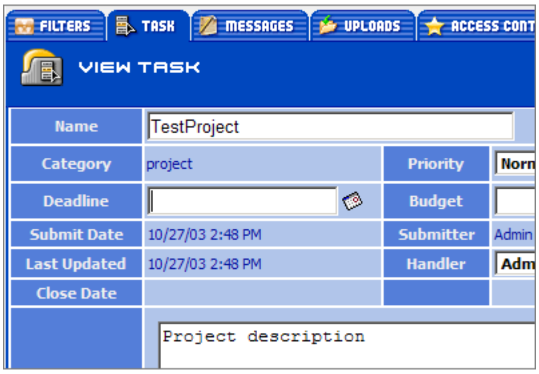
To create a project – enter the task name (e.g. *TestProject*); select the task category (e.g. *project*) and press the **Add** button. Note that the newly created project has now become the current task and all the further actions will be applied to it.



	Budget		Actual Bu
03 3:31 AM	Submitter	Admin	Alias
03 3:31 AM	Handler	Admin	Statu
	Workflow	Default	Resolut

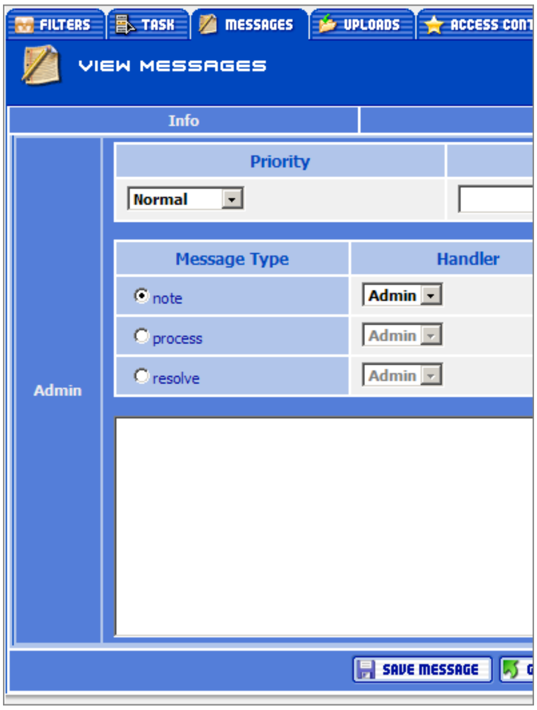
Add new project with name TestProject

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



Name	TestProject		
Category	project	Priority	Norm
Deadline		Budget	
Submit Date	10/27/03 2:48 PM	Submitter	Admin
Last Updated	10/27/03 2:48 PM	Handler	Adm
Close Date			
	Project description		

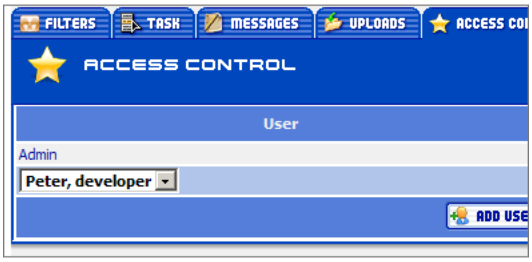
You will find yourself in the window for adding messages to this project. Here you can change the default handler for the project tasks or close the project.



Priority	
Normal	

Message Type	Handler
● note	Admin
○ process	Admin
○ resolve	Admin

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**.



User
Admin
Peter, developer

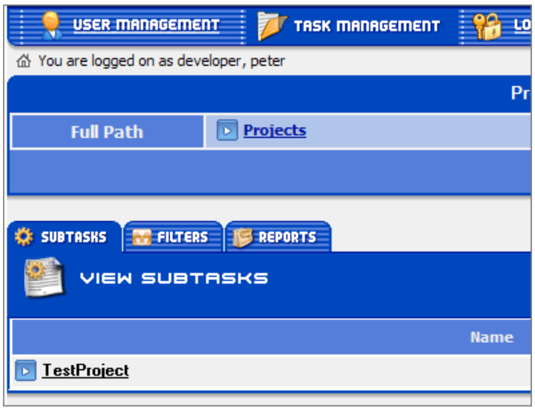
Log out after that.

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.

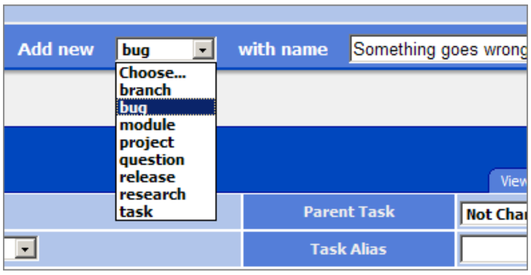


You are logged on as developer, peter

Full Path	Projects

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

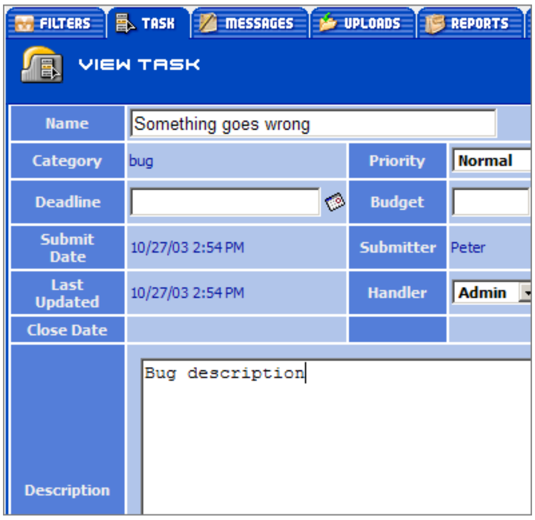
To create a bug in the project *TestProject* – choose the task type (*bug*), enter the task name (*Something goes wrong)*; and press the **Add** button.



Add new bug with name Something goes wrong

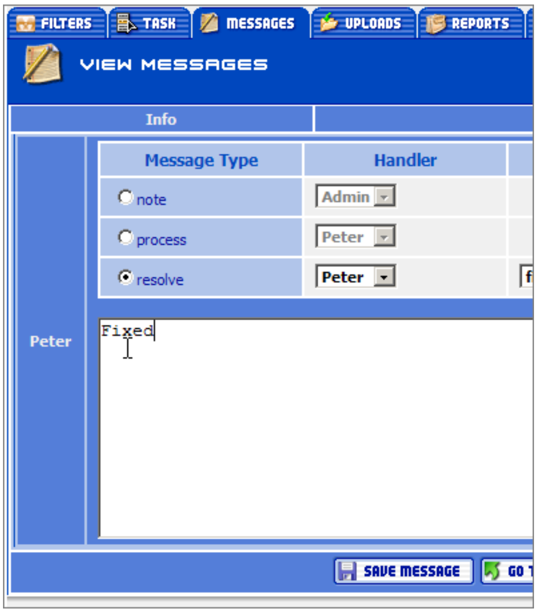
Parent Task	Not Cha
Task Alias	

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



Name	Something goes wrong		
Category	bug	Priority	Normal
Deadline		Budget	
Submit Date	10/27/03 2:54 PM	Submitter	Peter
Last Updated	10/27/03 2:54 PM	Handler	Admin
Close Date			
Description	Bug description		

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 the **Save Message** button.

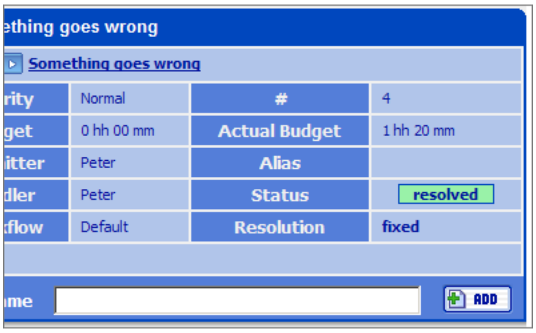


Message Type	Handler
○ note	Admin
○ process	Peter
● resolve	Peter

Peter

Fixed

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.


ething goes wrong

Something goes wrong

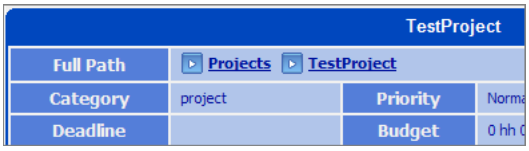
rity	Normal	#	4
get	0 hh 00 mm	Actual Budget	1 hh 20 mm
itter	Peter	Alias	
dler	Peter	Status	resolved
flow	Default	Resolution	fixed

4.4 Creating a New Filter User

The list of tasks under the **Subtasks** tab is displayed according to the selected filter. Use the dropdown list on the right to select the current filter. You can use the **Search** field to perform a full text search of the tasks in the filtering results.

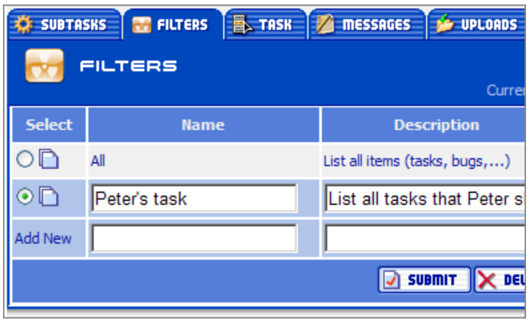
Description

Let’s create a filter that will display all tasks for which *Peter* is the handler. To do this, select *TestProject* - this filter will only be available for *TestProject* and its subtasks.


TestProject

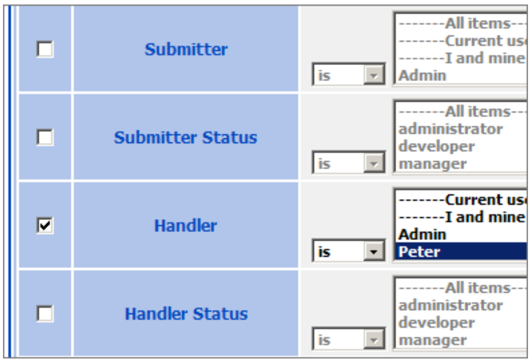
Full Path	Projects TestProject		
Category	project	Priority	Norma
Deadline		Budget	0 hh (

Move to the **Filters** tab. Enter the name for the filter (*Peter’s task*), its description (*List all tasks that Peter should resolve*) and press the **Save** button.



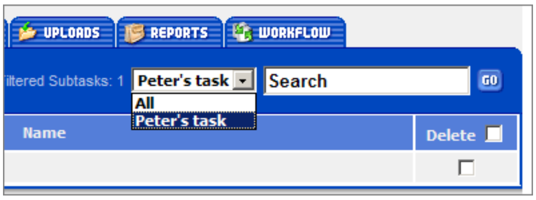
Select	Name	Description
○	All	List all items (tasks, bugs,...)
●	Peter's task	List all tasks that Peter s
Add New		

Move to the **Edit** tab. Select the filtering condition *Handler=Peter* and press the **Submit** button.



□	Submitter	is	-------All items-- -------Current us -------I and mine Admin
□	Submitter Status	is	-------All items-- administrator developer manager
☑	Handler	is	-------Current us -------I and mine Admin Peter
□	Handler Status	is	-------All items-- administrator developer manager

That's it – the filter has been created. Now you can use it to view the list of tasks under the **Subtasks** tab.



You can find more details on creating and editing filters in the topic Edit Filter (see page 47).

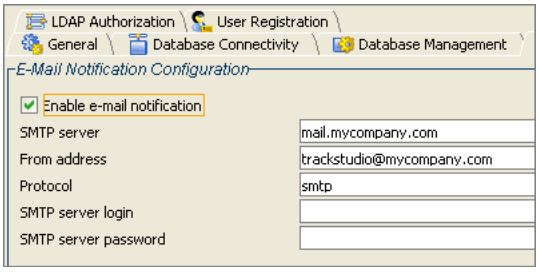
4.5 Enabling E-Mail Notification User

TrackStudio allows you to send out e-mail notifications when tasks are modified.

Description

To enable this notification system, do the following:

- Permit e-mail notification in **trackstudio.mail.properties**. You can also use the TrackStudio Enterprise Server Manager (see page 9).
- Specify the e-mail addresses of the users you want to receive the notifications.

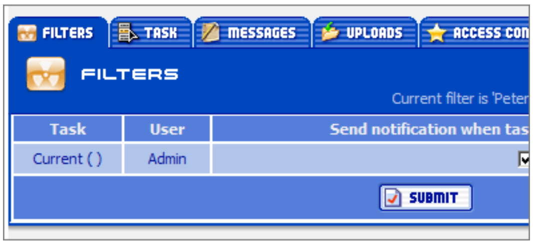


E-Mail Notification Configuration

☑ Enable e-mail notification

SMTP server	mail.mycompany.com
From address	trackstudio@mycompany.com
Protocol	smtp
SMTP server login	
SMTP server password	

- Enable the email notification (**Filter->Notify**) for the *Peter's task* filter and the *TestProject* task. The filter determines which tasks send e-mail notifications, while the task determines for which project the e-mail notification is enabled.



Current filter is 'Peter

Task	User	Send notification when tas
Current ( )	Admin	☑

- Now whenever the task with *handler=Peter* in the project *TestProject* is created or modified you will receive an e-mail notification.

Notes

When you mark the **Send notification** checkbox, you enable the email notification for only the task (and their subtasks) which belong with this checkbox. Suppose you have task *A* and its subtasks - *SubA*, *SubB*. You create some filter for the task *A* - it is available both for *SubA* and *SubB*. Then you select *SubA*, goto **Filter->Notify**, and check the **Send notification** checkbox. Now you enable the email notification for the task *SubA* and its subtasks only. If you create the subtask *SubSubA*, you will receive a notification. If you create subtasks *SubSubB* or *SubC*, you will not receive notification. Please make sure that you enable the email notification for the right task. You can find more details about how the e-mail notification feature works in the topic E-Mail Notification (see page 89).

5 Task Management User

This section describes task management features.

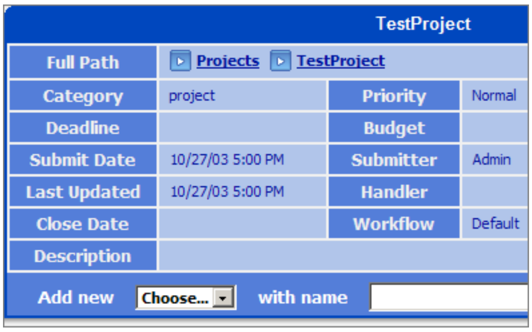
Description

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. On the right side of the menu, there is a dropdown list which allows the user to quickly select a project or task by its number. Only active projects are displayed in the project list. Active projects are those that are not finalized.

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.

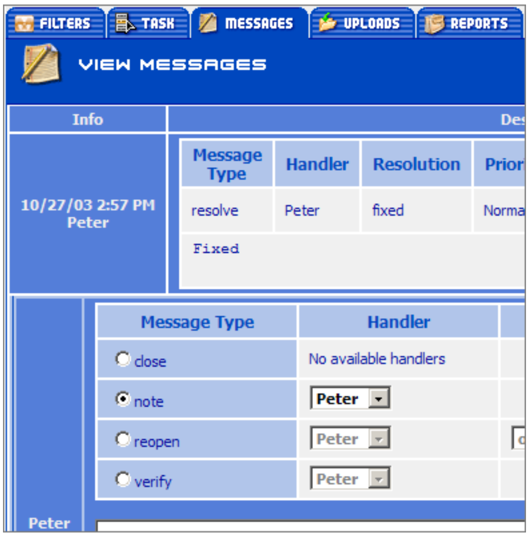


TestProject

Full Path	Projects TestProject		
Category	project	Priority	Normal
Deadline		Budget	
Submit Date	10/27/03 5:00 PM	Submitter	Admin
Last Updated	10/27/03 5:00 PM	Handler	
Close Date		Workflow	Default
Description			

Add new Choose... with name

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.



Info	Des
10/27/03 2:57 PM Peter	

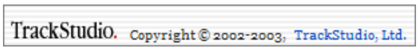
Message Type	Handler	Resolution	Prior
resolve	Peter	fixed	Norma

Fixed

Message Type	Handler
○ close	No available handlers
● note	Peter
○ reopen	Peter
○ verify	Peter

Peter

In the **Status Area** you can find the information about the system.


TrackStudio. Copyright © 2002-2003, TrackStudio, Ltd.

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
#	Unique numeric task number.	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 current task and 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) You have rights to view this task and add subtasks to it 2) 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.	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).

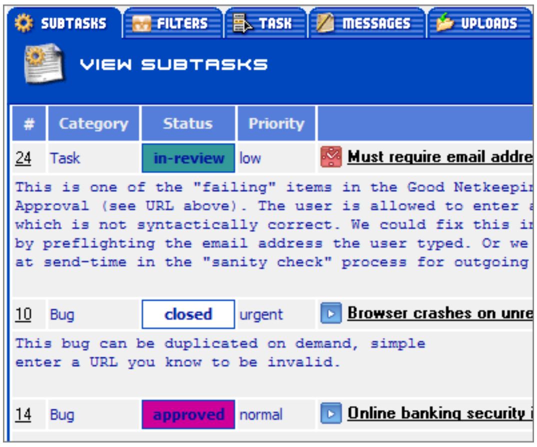
5.2 Subtasks User

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

Description

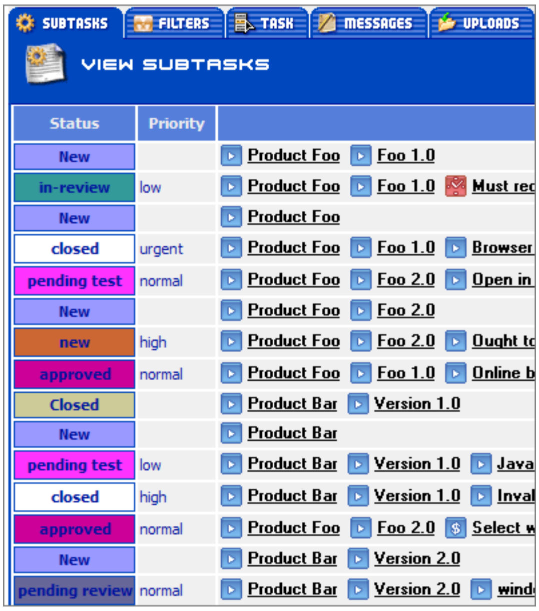
The **Subtasks** window allows you to view the information about subtasks of the current task. If the current task has no subordinate tasks, this window is not available. The list of displayed columns depends completely on the selected filter, except that the **Task Name** is always displayed. All the 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. The **Search** field allows you to quickly find a task by keyword among those tasks which have passed the filter. You can find more details about the query language syntax in the topic Full Text Search (see page 87).

For example, just the task list can be displayed.



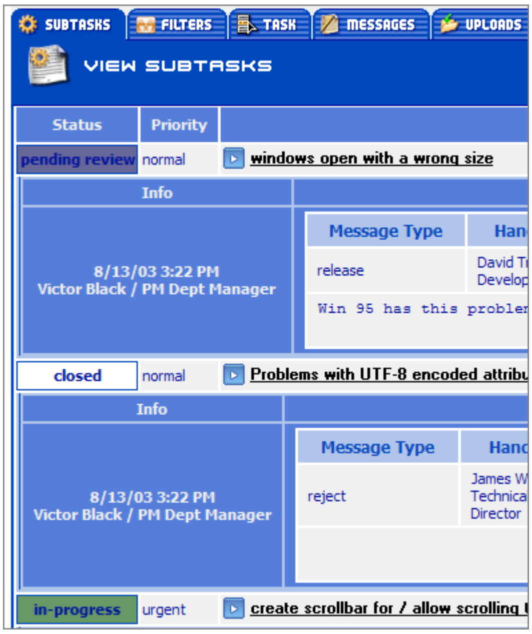
#	Category	Status	Priority	
24	Task	in-review	low	Must require email addre
This is one of the "failing" items in the Good Netkeepi Approval (see URL above). The user is allowed to enter which is not syntactically correct. We could fix this i by preflighting the email address the user typed. Or we at send-time in the "sanity check" process for outgoing				
10	Bug	closed	urgent	Browser crashes on unre
This bug can be duplicated on demand, simple enter a URL you know to be invalid.				
14	Bug	approved	normal	Online banking security

The task list with all the subtasks can be displayed.



Status	Priority	
New		Product Foo Foo 1.0
in-review	low	Product Foo Foo 1.0 Must rec
New		Product Foo
closed	urgent	Product Foo Foo 1.0 Browser
pending test	normal	Product Foo Foo 2.0 Open in
New		Product Foo Foo 2.0
new	high	Product Foo Foo 2.0 Ought tc
approved	normal	Product Foo Foo 1.0 Online b
Closed		Product Bar Version 1.0
New		Product Bar
pending test	low	Product Bar Version 1.0 Java
closed	high	Product Bar Version 1.0 Inval
approved	normal	Product Foo Foo 2.0 Select w
New		Product Bar Version 2.0
pending review	normal	Product Bar Version 2.0 wind

The task list with messages can be displayed.



Status	Priority	
pending review	normal	windows open with a wrong size

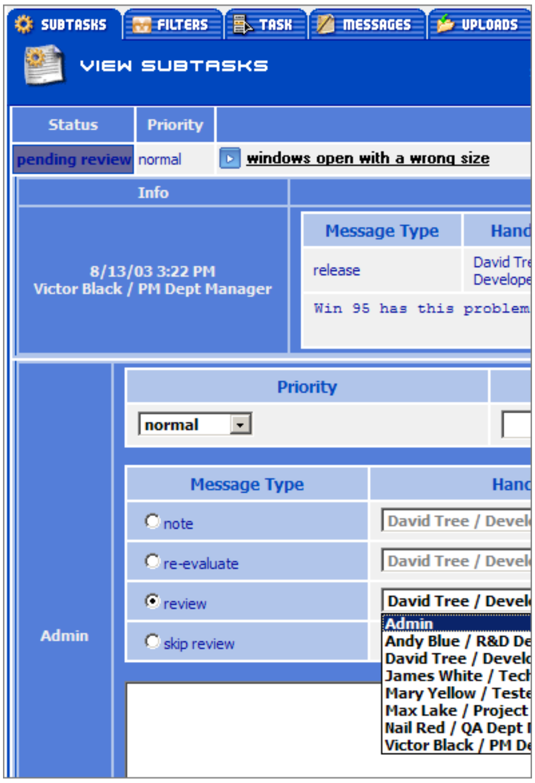
Info	
8/13/03 3:22 PM Victor Black / PM Dept Manager	Message Type: release Han: David Tr Develop Win 95 has this probler

Status	Priority	
closed	normal	Problems with UTF-8 encoded attribu

Info	
8/13/03 3:22 PM Victor Black / PM Dept Manager	Message Type: reject Hanc: James W Technica Director

Status	Priority	
in-progress	urgent	create scrollbar for / allow scrolling t

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. When using the **bulk processing tool,** messages are created only for those tasks that have a bug-note entered for them.



Status	Priority	
pending review	normal	windows open with a wrong size

Info

8/13/03 3:22 PM
Victor Black / PM Dept Manager

Message Type	Hand
release	David Tre Develope

Win 95 has this problem

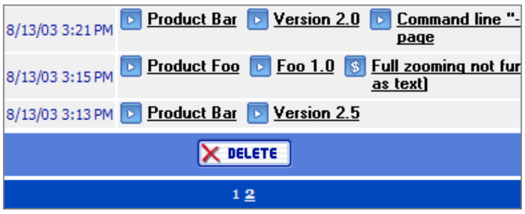
Admin

Priority: normal

Message Type	Hand
○ note	David Tree / Devel
○ re-evaluate	David Tree / Devel
● review	David Tree / Devel
○ skip review	

Admin
Andy Blue / R&D De
David Tree / Develo
James White / Tech
Mary Yellow / Teste
Max Lake / Project
Nail Red / QA Dept I
Victor Black / PM De

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.



8/13/03 3:21 PM	Product Bar	Version 2.0	Command line "- page
8/13/03 3:15 PM	Product Foo	Foo 1.0	Full zooming not fur as text)
8/13/03 3:13 PM	Product Bar	Version 2.5	

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 filters, 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.

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.

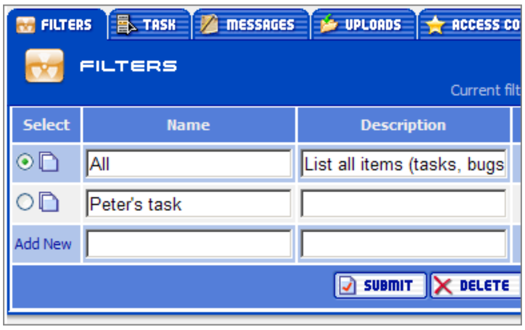
5.3.1 View Filter User

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** button to select current filter. You can view or edit the selected filter, enable email notification or subscription.
Copy	Use the **Copy** button to copy a filter. When a filter is copied, the settings of the email notification and the filter subscription are not copied. The owner of the new filter is the user who copied it.
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 selected filter. Please note, that you cannot delete the selected filter and the *All* filter. If you delete the filter that is being used to filter tasks or build reports, the *All* filter will be set instead of the deleted one. The settings of the e-mail notification and the filter subscription will be deleted when you delete the filter.



Select	Name	Description
●	All	List all items (tasks, bugs
○	Peter's task	
Add New		

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.

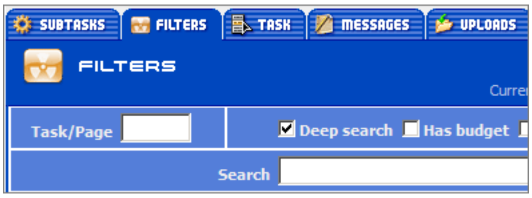
5.3.2 Edit Filter User

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

Description

Look at the filter parameters for tasks. At the top you can select the attributes (by marking the corresponding checkboxes) that your tasks should have. You can display tasks that have a budget (**Has budget**) and/or a deadline (**Has deadline**), or tasks that are beyond their budget (**Is overbudget**) and time (**Is overtime**). You can use the **Task/Page** field to specify the number of tasks per page. You can also use the **Deep search** feature to search and filter tasks through the entire hierarchy beginning with the current task. If this option is turned off, only subtasks of the current task will be included in the list. If this option is turned on, the recursive filtering through subtasks of the current task is performed.

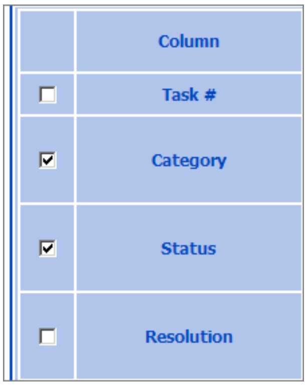
The **Search** field allows you to search for specific words and phrases within a particular task. You can perform a full text search without saving keywords in the filter – to do that, use the **Search** field on the **Subtasks** page. Note that the full text search is performed only through the tasks that pass the filter. This makes it possible to find tasks in the current project containing the word *CVS*, for example. You can find more details about the full text search query language in the topic Full Text Search (see page 87).



Task/Page ☑ Deep search ☐ Has budget ☐

Search

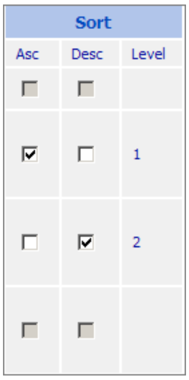
Below you can select and deselect columns that you want to display or hide. To display a column, mark the corresponding checkbox. If some field is checked, but no value is specified, the filtering will not be performed, but the corresponding column will be displayed in the list of tasks.



	Column
□	Task #
☑	Category
☑	Status
□	Resolution

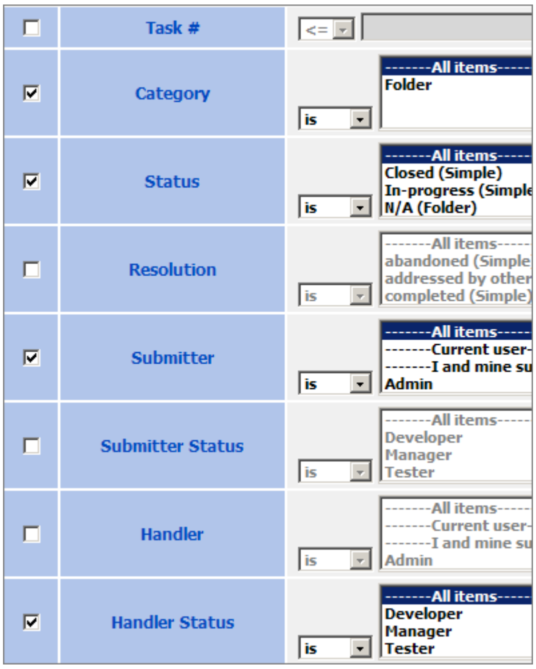
The list of tasks can be sorted either by one field or by several fields. For example, you can sort the list of tasks by categories and then sort those within the same category by the date they were created. The order of sorting is specified in the **level** field.

You can sort in either descending or ascending order. To sort the list in the ascending order, mark the checkbox in the necessary row in the column **ASC**; to sort in descending order, mark the box in the column **DESC**.



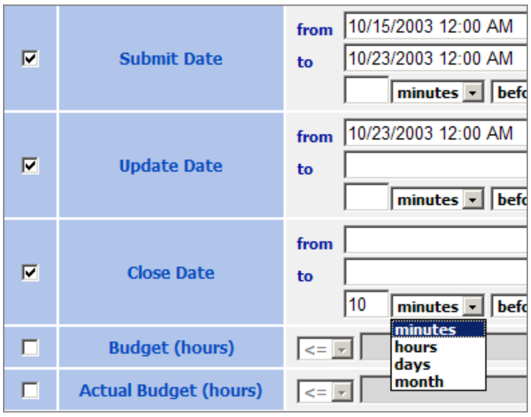
Sort		
Asc	Desc	Level
□	□	
☑	□	1
□	☑	2
□	□	

The **Task #** field allows you to search and filter tasks by their number. You can display only the tasks with the specified **Category, Status**, **Resolution, Priority, Submitter, Submitter Status, Handler** or **Handler Status**. The possible values of these properties can be selected from the lists (hold **Ctrl** to select multiple items).



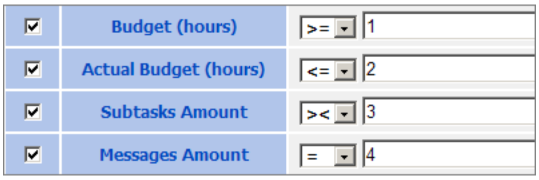
□	Task #	<=	
☑	Category	is	-------All items----- Folder
☑	Status	is	-------All items----- Closed (Simple) In-progress (Simple N/A (Folder)
□	Resolution	is	-------All items----- abandoned (Simple addressed by other completed (Simple)
☑	Submitter	is	-------All items----- -------Current user- -------I and mine su Admin
□	Submitter Status	is	-------All items----- Developer Manager Tester
□	Handler	is	-------All items----- -------Current user- -------I and mine su Admin
☑	Handler Status	is	-------All items----- Developer Manager Tester

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.



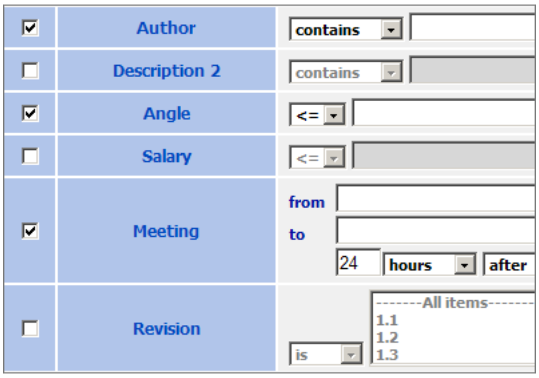
☑	Submit Date	from 10/15/2003 12:00 AM to 10/23/2003 12:00 AM minutes befo
☑	Update Date	from 10/23/2003 12:00 AM to minutes befo
☑	Close Date	from to 10 minutes befo
□	Budget (hours)	<=
□	Actual Budget (hours)	<=

For numerical properties, such as **Task #, Budget, Actual Budget, Subtasks Amount, Messages Amount,** you can display tasks with properties that are equal, not equal, greater than or equal, and less than or equal compared to the specified value (by selecting **=, ><, =>** or **<=** from the drop-down list).



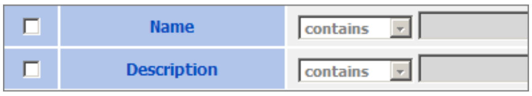
☑	Budget (hours)	>=	1
☑	Actual Budget (hours)	<=	2
☑	Subtasks Amount	><	3
☑	Messages Amount	=	4

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



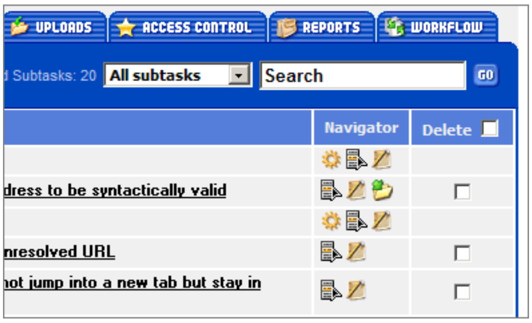
☑	Author	contains	
□	Description 2	contains	
☑	Angle	<=	
□	Salary	<=	
☑	Meeting	from to 24 hours after	
□	Revision	is	-------All items------- 1.1 1.2 1.3

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



□	Name	contains	
□	Description	contains	

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.

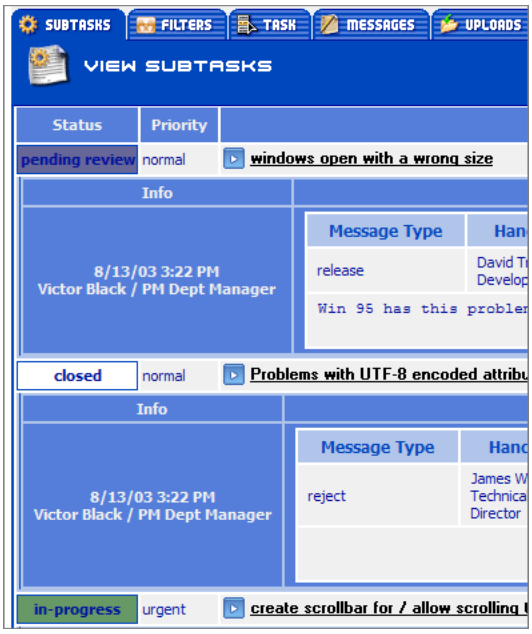


Subtasks: 20 All subtasks Search

	Navigator	Delete □
dress to be syntactically valid		□
nresolved URL		□
not jump into a new tab but stay in		□

Let’s see the filter parameters for messages. Filtering tasks by messages allows you to display the tasks which have 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. You can specify the number of messages to be displayed in the list. You can choose the first few messages or the last few messages. Messages in all lists are sorted by the date of their creation in ascending order.



Status	Priority	
pending review	normal	windows open with a wrong size

Info		
8/13/03 3:22 PM Victor Black / PM Dept Manager	Message Type	Han
	release	David T Develop
	Win 95 has this probler	

Status	Priority	
closed	normal	Problems with UTF-8 encoded attribu

Info		
8/13/03 3:22 PM Victor Black / PM Dept Manager	Message Type	Hanc
	reject	James W Technica Director

Status	Priority	
in-progress	urgent	create scrollbar for / allow scrolling t

The **Filter Messages** must be on if you wish to filter messages in tasks. If it is off, only tasks will be filtered. For example, if the following condition is specified – *Message Handler=John and Message Filter=on*, all the tasks will be found, but only the messages that meet *Message Handler=John* will be displayed.

If the following condition is specified – *Message Handler=John and Message Filter=off*, all the tasks that have at least one message where *handler=John* will be displayed (if **view messages=on**, all messages in these tasks will be displayed no matter who their handler is).

You can specify from the beginning what number of messages should be filtered. Suppose you specified that the last 20 messages should be filtered. TrackStudio would filter the last 20 messages from the list according to the filtering conditions. The rest of the messages will not be processed or displayed.

View Messages and **Filter Messages** are checked in the following order:

1) The condition **Filter Messages** is applied first; it leaves the specified number of the first and the last messages in each task.

2) Then filtering by **submitter**, **handler** and other fields is performed.

3) And then **view messages** displays the specified number of messages at the beginning or at the end of the list.

For example, if you specified:

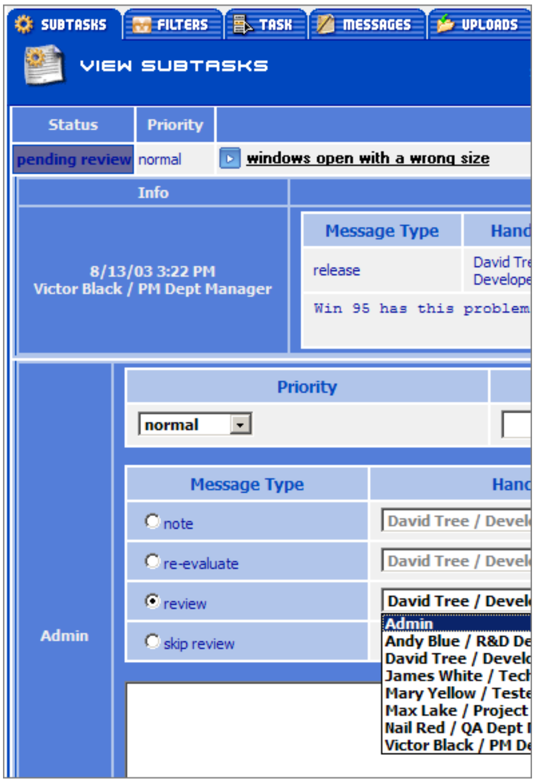
```
Filter Messages: 5 last
Submitter: John
View Messages: 3 first
```

TrackStudio will find the last 5 messages for each task first, then it will keep the tasks where *submitter=John*, and if there are more than 3 such messages, it will display only the first 3 of them.

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**. You can also change its **priority**, **deadline** and **budget**.For example, this feature gives you the opportunity to quickly change the **status**, **deadline** and **budget** for a large number of tasks, or add important information to several tasks at once.

Please note that behavior of the message editor in the **Messages** window differs from that in the **Subtasks** window. In the **Subtasks** window the **Bulk Processing Tool** is applied to the task only if there is a non-empty **Message Description** specified (i.e. the **Bulk Processing Tool** does not allow you to change the **priority**, **handler** or **resolution** if no bug-note is specified).

If you turned on the **Bulk Processing Tool** with the **View Messages** option off, you would see only the form for entering messages in the list of tasks. The messages themselves would not be visible.



Status	Priority	
pending review	normal	windows open with a wrong size

Info		
8/13/03 3:22 PM Victor Black / PM Dept Manager	Message Type	Hand
	release	David Tre Develope
	Win 95 has this problem	

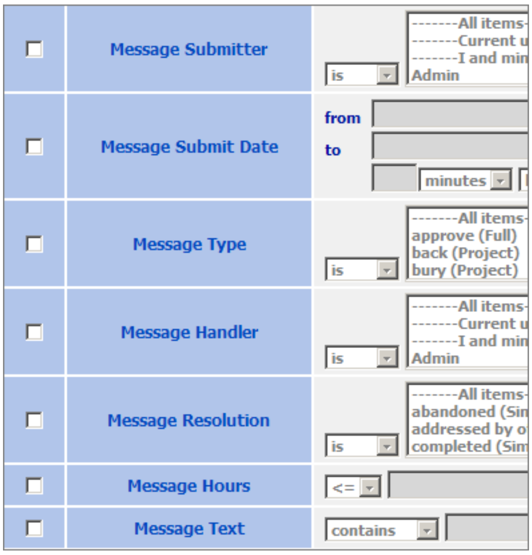
Admin

Priority: normal

Message Type	Hand
○ note	David Tree / Devel
○ re-evaluate	David Tree / Devel
● review	David Tree / Devel
○ skip review	

Admin
Andy Blue / R&D De
David Tree / Develo
James White / Tech
Mary Yellow / Teste
Max Lake / Project
Nail Red / QA Dept I
Victor Black / PM De

Message Submitter, Message Submit Date, Message Type, Message Handler, Message Resolution, Message Hours and **Message Text** allow you to specify the parameters for filtering 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).



□	Message Submitter	is	-------All items- -------Current u -------I and min Admin
□	Message Submit Date	from to	minutes
□	Message Type	is	-------All items- approve (Full) back (Project) bury (Project)
□	Message Handler	is	-------All items- -------Current u -------I and min Admin
□	Message Resolution	is	-------All items- abandoned (Sin addressed by o completed (Sim
□	Message Hours	<=	
□	Message Text	contains	

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

Example

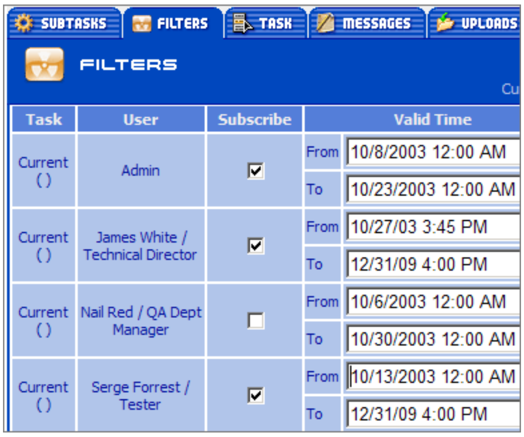
Name	Description	Settings
Mine Tasks	Current use is handler	Category=All; Status=All; Handler=current user; Priority=All
My Open	Open, current user is handler	Category=All; Status=new/processing; Handler=current user; Priority=All
My Today	Due today for current user!	Category=All; Status=new/processing; Handler=current user; Deadline=1 days after or early; Priority=All
Overdue	Past due date	Category=All; Status=new/processing; Handler=current user; Deadline=1 minutes ago or early; Priority=All
Still Warm	Issues that were closed within the last week	Category=All; Status=closed; Close date=7 days before or later

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.



Task	User	Subscribe	Valid Time	
Current ()	Admin	☑	From	10/8/2003 12:00 AM
			To	10/23/2003 12:00 AM
Current ()	James White / Technical Director	☑	From	10/27/03 3:45 PM
			To	12/31/09 4:00 PM
Current ()	Nail Red / QA Dept Manager	□	From	10/6/2003 12:00 AM
			To	10/30/2003 12:00 AM
Current ()	Serge Forrest / Tester	☑	From	10/13/2003 12:00 AM
			To	12/31/09 4:00 PM

You have to fill in the following fields:

Property	Description
Task	Indicates the current task. You will receive email notification on the current task and the subtasks of the current task. Please note that you can receive e-mail 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 (**From**) and stop (**To**) date of subscription. You will receive e-mails 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.

The manager can subscribe those subordinate users who have access to the current task to receive filtering results periodically. Note that users can subscribe themselves to a filter even if they do not have access to the task. As with the e-mail notification, the filter subscription is enabled for the current task, not for the parent task of the filter. This means you can enable the subscription to the filter separately for each project. When you have specified all subscription options, press the **Submit** button.

5.3.4 Notify by Email User

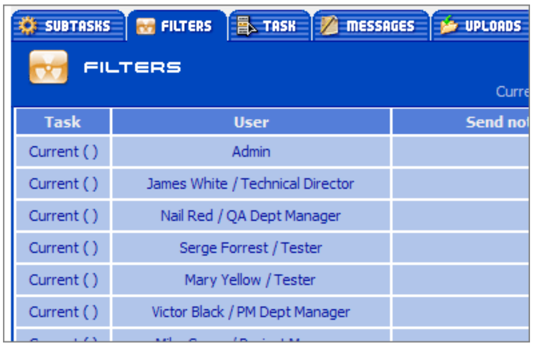
This topic describes the email notification feature (**Task Management->Filters->Notify** tab).

Description

TrackStudio has a powerful filter-based system of email notifications.

To activate email notifications, you should

1. Choose a **filter** describing what changes in the task should invoke an email notification. To activate the e-mail notification for any tasks, you should choose the *All* filter.
2. Make the **task** for which the e-mail notification should be activated the current one. The notification system will also be activated for all subtasks of the current task. To send out notifications when any tasks are modified, you should activate the email notification for the root task.
3. Mark the **Send notification** checkbox. If the subscription to the given filter is already activated for the parent task of the current task, it will be available for the latter automatically and the checkbox will be gray.



Task	User	Send not
Current ( )	Admin	
Current ( )	James White / Technical Director	
Current ( )	Nail Red / QA Dept Manager	
Current ( )	Serge Forrest / Tester	
Current ( )	Mary Yellow / Tester	
Current ( )	Victor Black / PM Dept Manager	

Once a user is created, it is recommended that you subscribe him/her to e-mail notifications for any changes in any tasks (activate the *All* filter for the root task). If the user receives too many notifications, he can unsubscribe for the root task notifications and activate only notifications for certain projects and task types.

The email notification system is configured separately for each user and each user can define his own rules for sending out e-mail notifications. He/she can create a filter and activate e-mail notifications for a project he has no access to. In this case he will be receiving e-mail notifications only when those subtasks of the project are modified that he has access to. The manager can subscribe (or unsubscribe) a user to any filter only if the user has access to the task or the project. For example, if a developer has no access to the root task, the manager cannot subscribe him/her to receiving e-mail notifications when subtasks of the root task are modified.

You can find more details about the configuration of the e-mail notification system in the topic E-Mail Notification (⧉ see page 89).

5.4 Task User

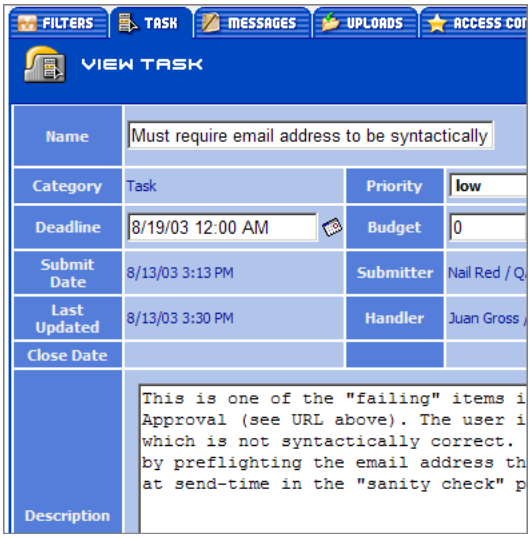
This section describes how to view or modify some important properties of the current task (**Task Management->Task** tab).

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 choose the task **Category**, enter its **Name** and press the **Add** button. The task **handler**, **budget** and **deadline** will be inherited from the upper level task.



Name	Must require email address to be syntactically		
Category	Task	Priority	low
Deadline	8/19/03 12:00 AM	Budget	0
Submit Date	8/13/03 3:13 PM	Submitter	Nail Red / Q
Last Updated	8/13/03 3:30 PM	Handler	Juan Gross /
Close Date			
Description	This is one of the "failing" items i Approval (see URL above). The user i which is not syntactically correct. by preflighting the email address th at send-time in the "sanity check" p		

The purpose of the fields is described in Task Concepts (⧉ see page 40).

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. 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

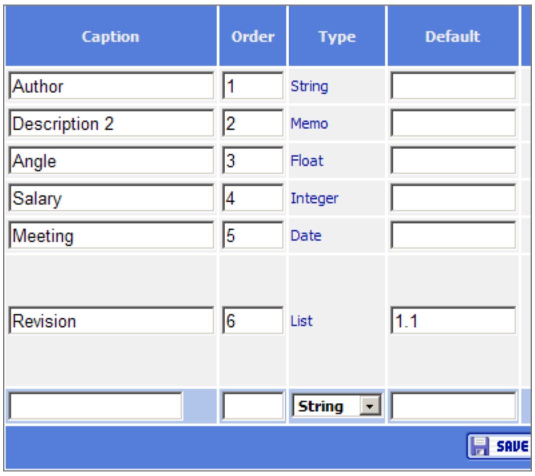
Be especially careful when changing the **Parent Task**. Such system objects as custom fields, filters, categories and workflows can be inherited. When the parent task is changed, the list of the parent tasks and inherited objects can also change, which can result in unpredictable consequences.

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.



Caption	Order	Type	Default
Author	1	String	
Description 2	2	Memo	
Angle	3	Float	
Salary	4	Integer	
Meeting	5	Date	
Revision	6	List	1.1
		String	

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.
Type	Custom field type. Please check out next table for more information about available custom field types.
Default	Default field value (100, for example). For list the default value should be equal to one of the list items.
Formula	Pressing this button opens the expression editor for calculated custom fields. A field is considered calculated if there is an expression specified for it. You can get more details about calculated custom fields in the topic Calculated Custom Fields (⧉ see page 98).
List of Values	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.

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

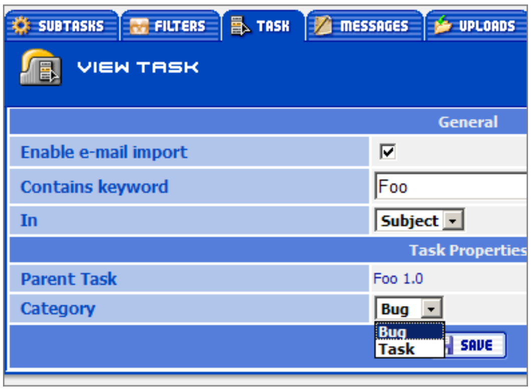
After all the required fields are filled in, press the **Save** button to save the created or modified custom field. To delete a custom field or dropdown list item select the field or list item and click the **Delete** button. You can delete or modify fields created only on the current level in task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

5.4.3 E-Mail Import Project Manager

This section describes how to define task submission rules (**Task Management->Task->E-Mail Import** tab).

Description

To enable task submission, you should open the **Task->E-Mail Import** form and set the following options:



General

Enable e-mail import ☑

Contains keyword Foo

In Subject

Task Properties

Parent Task Foo 1.0

Category Bug

- **Enable e-mail import** – enables task submission for the current task.
- **Contains keyword** – allows you to specify the keyword either in the subject or in the body of the message. Keywords make it possible to use one mailbox for importing messages into several projects, therefore you should use different keywords for different projects. If no keyword is specified, the e-mail is imported regardless of its body or subject. The e-mail import rules with the empty keyword list are checked after the rules with a keyword specified. The keyword check is case-insensitive.
- **In** – allows you to specify the field in which to search for the keyword. The **subject** and **body** fields are supported.
- **Category** – the category of the created tasks. The category must be associated with a workflow which has the start state.

To enable Task Submission, you should check **Enable e-mail import** and press **Save**. To disable Task Submission, you should uncheck **Enable e-mail import** and press **Save**

5.4.4 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

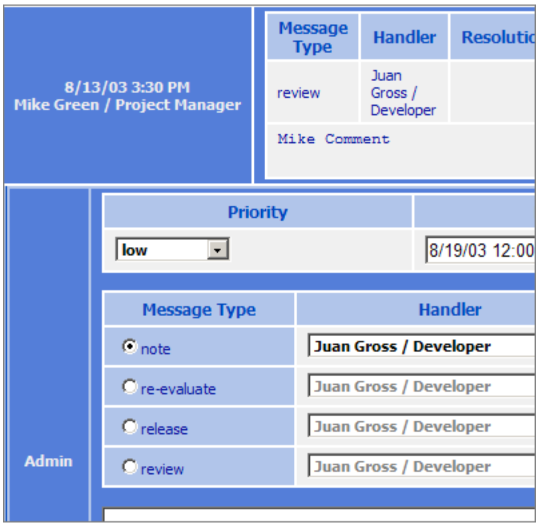
```
<gran-pm-export>
  <defect>
    <path> => ROOT => User Activity => Enter title here. </path>
    <id>A42754AA0BAD18E0E0303BD58EAC420F</id>
    <parent_id>A3EEB42C51F5F5EEE0303BD58EAC1707</parent_id>
    <name><![CDATA[Enter title here.]]></name>
    <abudget>27 hh 00 mm</abudget>
    <submitdate>24/06/2002 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.



8/13/03 3:30 PM
Mike Green / Project Manager

Message Type	Handler	Resolutio
review	Juan Gross / Developer	

Mike Comment

Priority	
low	8/19/03 12:00

Admin

Message Type	Handler
● note	Juan Gross / Developer
○ re-evaluate	Juan Gross / Developer
○ release	Juan Gross / Developer
○ review	Juan Gross / Developer

You have to fill in following fields:

Property	Description
Info	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.
Priority	Task priority. It can vary from *low* (lowest priority) to *support* (highest priority). The priority can be customized on a per-category basis.

Deadline	This is the date when the current task must be finished.
Budget	Estimated time (in hours) allocated for the task. This is working time, not calendar time.
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.

You can change the priority, deadline and budget fields only if you have enabled the scheduler.

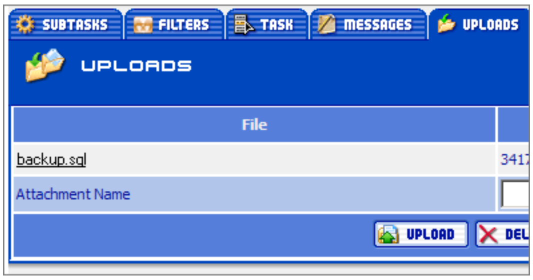
To save a message, you have to press the **Save Message** button at the bottom of the page. To save a message and go to the list of subtasks of a parent task, you have to press the **Go to parent** button at the bottom of the page. You can also delete a message by selecting one or more messages and clicking the **Delete** button.

5.6 Uploads User

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

Description

The **Uploads** tab provides a way to attach files to the current task. This dialog is also used to modify existing attachments. 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).

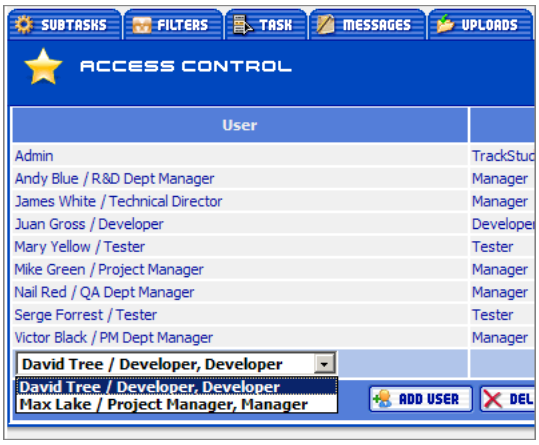
Description

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).



User	
Admin	TrackStuc
Andy Blue / R&D Dept Manager	Manager
James White / Technical Director	Manager
Juan Gross / Developer	Develope
Mary Yellow / Tester	Tester
Mike Green / Project Manager	Manager
Nail Red / QA Dept Manager	Manager
Serge Forrest / Tester	Tester
Victor Black / PM Dept Manager	Manager

David Tree / Developer, Developer

David Tree / Developer, Developer
Max Lake / Project Manager, Manager

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.

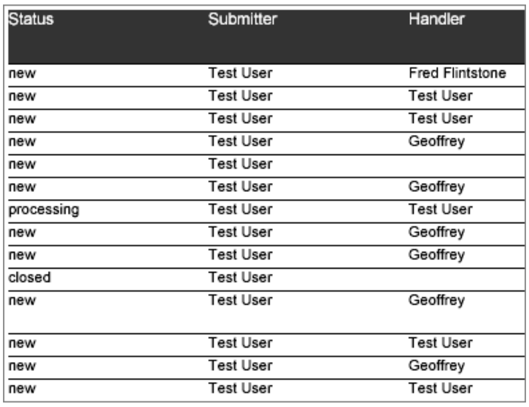
5.8.1 Report Types User

This topic describes available report types.

Description

List Report

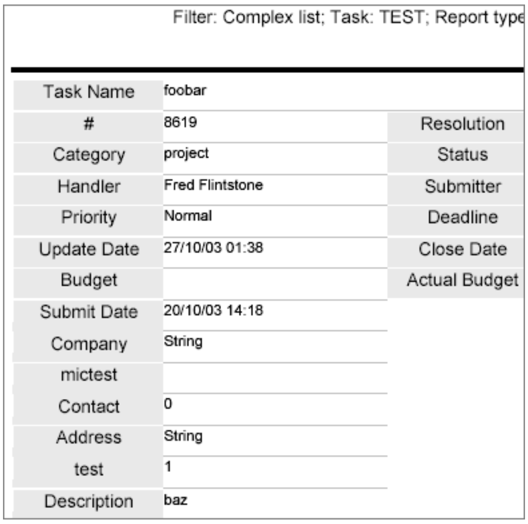
Shows the list of subtasks of the current task in the form of a list. Tasks, bug notes, task fields and messages are displayed according to the filter settings.



Status	Submitter	Handler
new	Test User	Fred Flintstone
new	Test User	Test User
new	Test User	Test User
new	Test User	Geoffrey
new	Test User	
new	Test User	Geoffrey
processing	Test User	Test User
new	Test User	Geoffrey
new	Test User	Geoffrey
closed	Test User	
new	Test User	Geoffrey
new	Test User	Test User
new	Test User	Geoffrey
new	Test User	Test User

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.



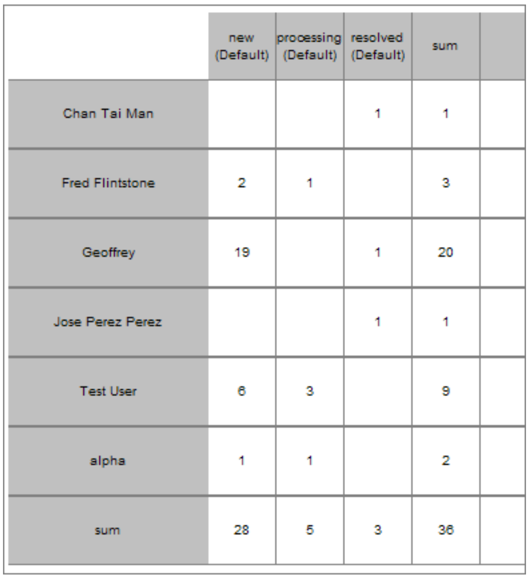
Filter: Complex list; Task: TEST; Report type

Task Name	foobar	
#	8619	Resolution
Category	project	Status
Handler	Fred Flintstone	Submitter
Priority	Normal	Deadline
Update Date	27/10/03 01:38	Close Date
Budget		Actual Budget
Submit Date	20/10/03 14:18	
Company	String	
mictest		
Contact	0	
Address	String	
test	1	
Description	baz	

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.

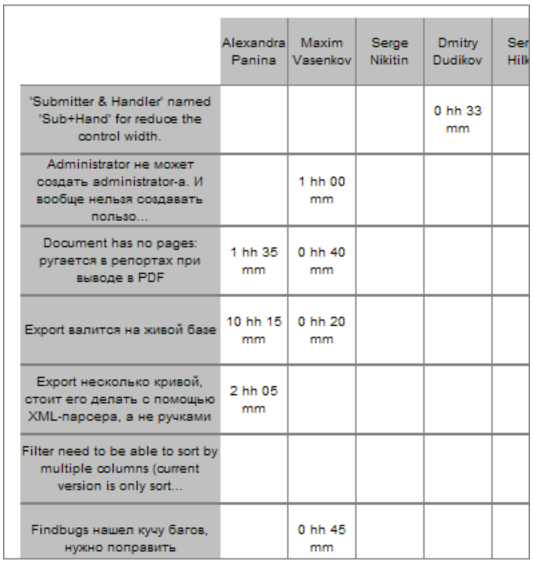


	new (Default)	processing (Default)	resolved (Default)	sum	
Chan Tai Man			1	1	
Fred Flintstone	2	1		3	
Geoffrey	19		1	20	
Jose Perez Perez			1	1	
Test User	6	3		9	
alpha	1	1		2	
sum	28	5	3	36	

Money Report

Lets you gather information on the intensity of working with subtasks of the current task. Using 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.



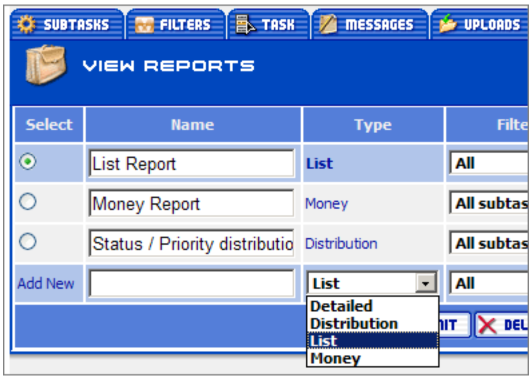
	Alexandra Panina	Maxim Vasenkov	Serge Nikitin	Dmitry Dudikov	Ser Hilk
'Submitter & Handler' named 'Sub+Hand' for reduce the control width.				0 hh 33 mm	
Administrator не может создать administrator-а. И вообще нельзя создавать пользо...		1 hh 00 mm			
Document has no pages: ругается в репортах при выводе в PDF	1 hh 35 mm	0 hh 40 mm			
Export валится на живой базе	10 hh 15 mm	0 hh 20 mm			
Export несколько кривой, стоит его делать с помощью XML-парсера, а не ручками	2 hh 05 mm				
Filter need to be able to sort by multiple columns (current version is only sort...					
Findbugs нашел кучу багов, нужно поправить		0 hh 45 mm			

5.8.2 List Reports User

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

Description

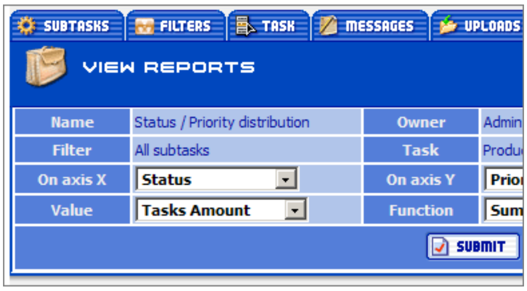
To create a report, you must specify its **Name**, **Description** and **Type**, choose a **Filter** for the report to be based on, indicate whether the report is private and press the **Submit** button.



Select	Name	Type	Filte
●	List Report	List	All
○	Money Report	Money	All subtas
○	Status / Priority distributio	Distribution	All subtas
Add New		List	All

Detailed
Distribution
List
Money

To generate a report, you must select it by pressing the **Select** button, 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.



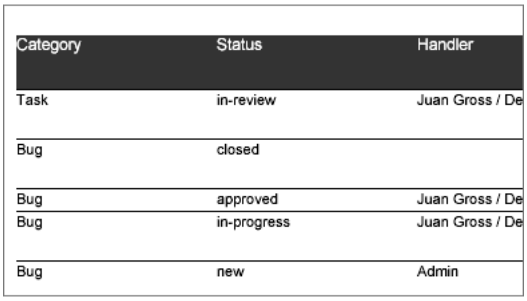
Name	Status / Priority distribution	Owner	Admin
Filter	All subtasks	Task	Produ
On axis X	Status	On axis Y	Prio
Value	Tasks Amount	Function	Sum

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.



Category	Status	Handler
Task	in-review	Juan Gross / De
Bug	closed	
Bug	approved	Juan Gross / De
Bug	in-progress	Juan Gross / De
Bug	new	Admin

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:

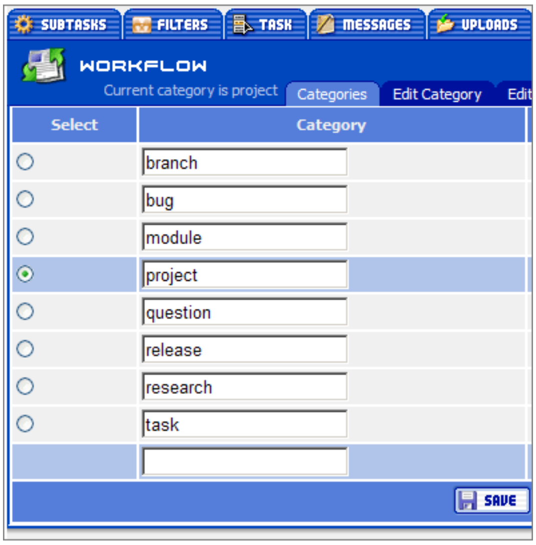
- Create empty (blank) workflow.
- Add set of priorities.
- Add set of states; define start and final states.
- Add set of message types to your workflow. Each message type should have appropriate set of resolutions. For example, message type **resolve** can have the following resolution set - *fixed, 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.



Select	Category
○	branch
○	bug
○	module
●	project
○	question
○	release
○	research
○	task

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.
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.

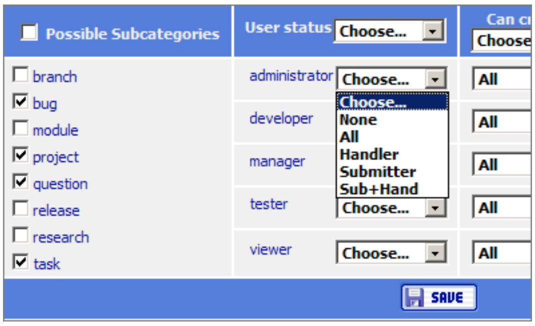
After creating a category, you should go to the **Edit Category** tab and specify who can create, delete or edit tasks within this category. You should also declare the created category as a possible subcategory.

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.



□ Possible Subcategories

□ branch
☑ bug
□ module
☑ project
☑ question
□ release
□ research
☑ task

User status	Choose...	Can cr Choose
administrator	Choose...	All
developer	Choose... None All Handler Submitter Sub+Hand	All
manager		All
tester	Choose...	All
viewer	Choose...	All

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

User	Action	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
Sub+Hand	Can create	Only submitter (of parent task) or handler (of parent task) can create subtasks with selected category.
Sub+Hand	Can modify	Only task submitter or handler can modify tasks of selected category
Sub+Hand	Can delete	Only task submitter or 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.

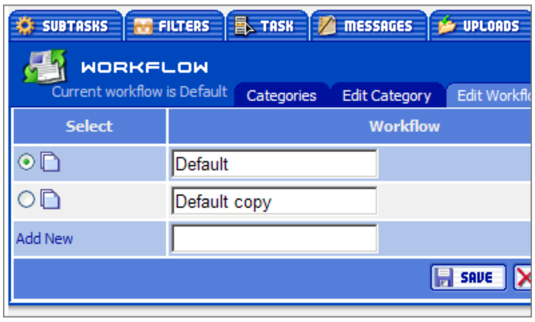
5.9.3 Workflows Project Manager

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).**



Select	Workflow
●	Default
○	Default copy
Add New	

You have to fill in the following properties:

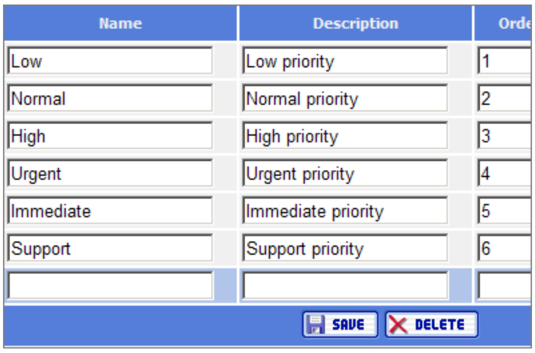
Column	Description
Select	Current workflow selector.
Copy	Copy the selected workflow.
Workflow	Name of the workflow.
Task	Indicates related task in task hierarchy. You need to have access rights for this task level to edit this category.
Delete	Use this checkbox to select a workflow for deletion. Please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

To choose an active workflow please use the **Select** button. 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



Name	Description	Orde
Low	Low priority	1
Normal	Normal priority	2
High	High priority	3
Urgent	Urgent priority	4
Immediate	Immediate priority	5
Support	Support priority	6

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.

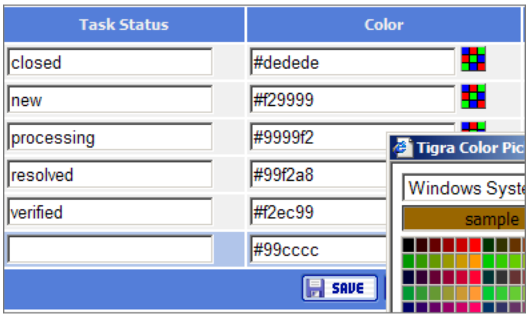
Please click the **Save** button to save a priority. Please click the **Delete** button to delete a chosen priority.

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.



Task Status	Color
closed	#dedede
new	#f29999
processing	#9999f2
resolved	#99f2a8
verified	#f2ec99
	#99cccc

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.

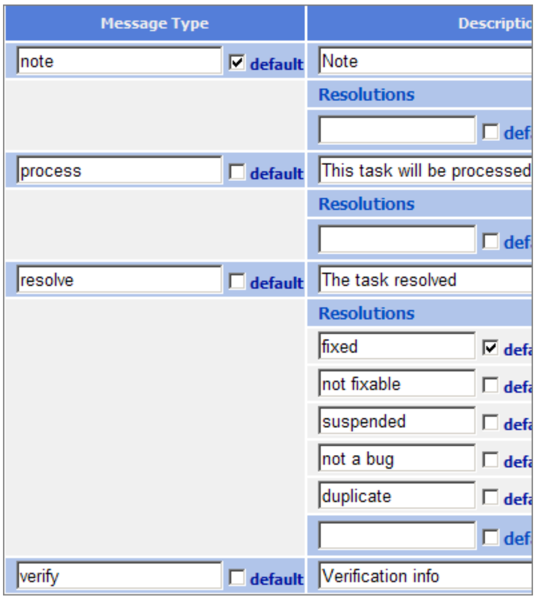
Click the **Save** button to save a task state. Click the **Delete** button to delete a task state.

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 state to state 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.



Message Type	Descriptic
note ☑ default	Note
	Resolutions
	☐ def
process ☐ default	This task will be processed
	Resolutions
	☐ def
resolve ☐ default	The task resolved
	Resolutions
	fixed ☑ defa
	not fixable ☐ defa
	suspended ☐ defa
	not a bug ☐ defa
	duplicate ☐ defa
	☐ def
verify ☐ default	Verification info

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
Message Type	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

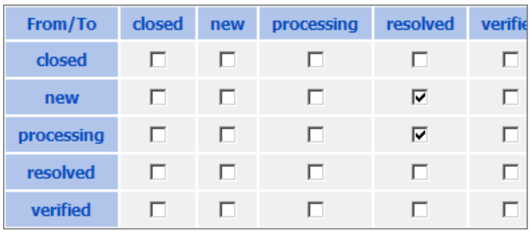
This section describes how to define a Transition Matrix (**Task Management->Workflow->Transitions** tab).

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.

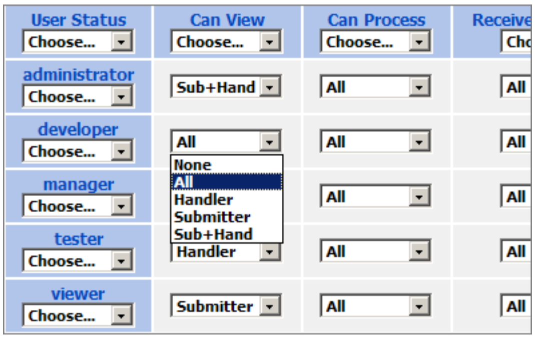
The matrix is a table that has all states available. The left column represent initial (**From**) states, the first column - destination (**To**) states. If you check an item, it means that transitions for given message types are permitted from initial state to destination state.

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



From/To	closed	new	processing	resolved	verifie
closed	□	□	□	□	□
new	□	□	□	☑	□
processing	□	□	□	☑	□
resolved	□	□	□	□	□
verified	□	□	□	□	□

You can select possible actions for users of available status for each message type. You can use the dropdown list in the header of the table columns to change the values of all elements in the column. The dropdown list at the beginning of a row allows you to change the values of all elements in that row. The dropdown list in the upper left-hand corner allows you to change the values of all elements in the table.



User Status Choose...	Can View Choose...	Can Process Choose...	Receive Cho
administrator Choose...	Sub+Hand	All	All
developer Choose...	All None All Handler Submitter Sub+Hand	All	All
manager Choose...		All	All
tester Choose...	Handler	All	All
viewer Choose...	Submitter	All	All

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 user status can view this message in the transitions history (previously sent messages).
Can Process	All	All users with this user status can do this transition (save this message).

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

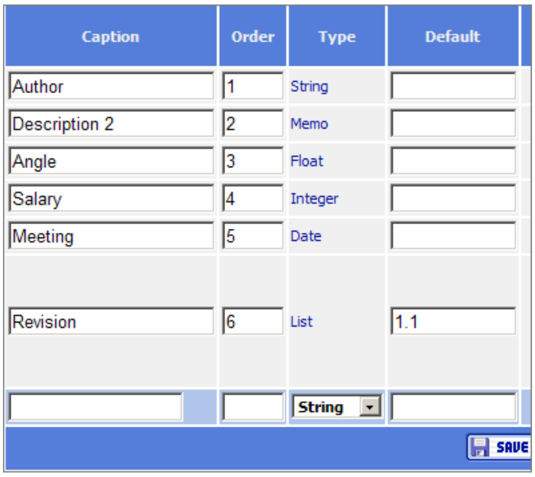
You can have another workflow status where the *Handler* items are selected because you prefer that only one (the handler) can attend a task and solve it (this is a restricted attention mode). Maybe in your development team the members work individually, and the handler of a task is the only one that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



Caption	Order	Type	Default
Author	1	String	
Description 2	2	Memo	
Angle	3	Float	
Salary	4	Integer	
Meeting	5	Date	
Revision	6	List	1.1
		String	

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.
Formula	Pressing this button opens the expression editor for calculated custom fields. A field is considered calculated if there is an expression specified for it. You can get more details about calculated custom fields in the topic Calculated Custom Fields (⧉ see page 98).
List of Values	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

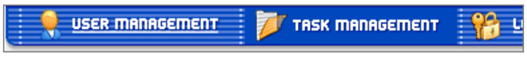
This section describes how to manage users and user groups.

Description

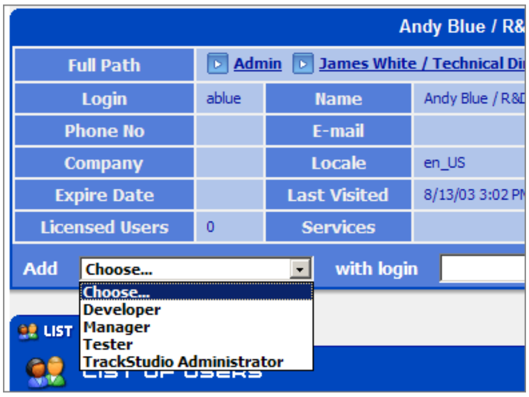
The **User Management** window consists of the following parts.

- The **Main Menu.**
- 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.


Andy Blue / R&

Full Path	Admin James White / Technical Di		
Login	ablue	Name	Andy Blue / R&D
Phone No		E-mail	
Company		Locale	en_US
Expire Date		Last Visited	8/13/03 3:02 PM
Licensed Users	0	Services	

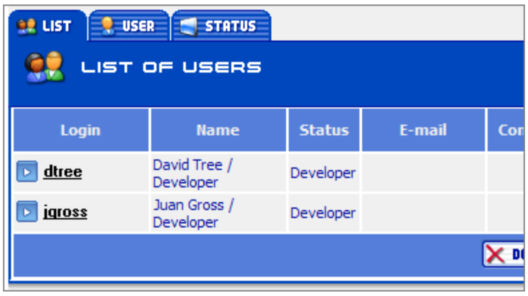
Add Choose... with login

Here you can use the following user properties:

Property	Description	User Header	User Tab
Full Path	In this field information about the position of the current user in the user hierarchy is displayed.	Yes	No
Login	The user's login	Yes	Yes
Status	The user status (a user group). Each user can belong to only one group.	Yes	Yes
Name	Contains the user's name.	Yes	Yes
Phone No	Contains the user's phone number.	Yes	Yes
E-Mail	Contains the user's e-mail.	Yes	Yes

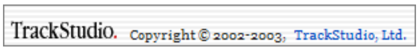
E-Mail for SMS	Here you can specify an additional e-mail. This field is not used at the moment.	Yes	Yes
Services	Contains the services available only for the current user. This property is used only in TrackStudio Host.	Yes	Yes
Time Zone	Contains the time zone of the current user. All the data is displayed according to the specified **Time Zone**.	Yes	Yes
Locale	Contains the locale of the current user. All dates, as well as figures with floating points must fit the specified format.	Yes	Yes
Last Visited	Contains the date when the current user last logged into the system.	Yes	Yes
Expire Date	The date the user's login will expire. The user and his/her subordinate users will not be able to log into the system after the expiration date.	Yes	Yes
Default Project	The project that will be selected right after the user logs in. If the user has no access to the specified project, the access rights will be granted automatically.	No	Yes
Notification type	the type of email notification messages. It can be HTML or plain text. HTML is recommended.	Yes	Yes
Licensed Users	Contains the number of child users available for the current user. When the system checks on this limitation, it checks not only the closest parent, but also all the upper parents.	Yes	Yes

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



Login	Name	Status	E-mail	Cor
dtree	David Tree / Developer	Developer		
jgross	Juan Gross / Developer	Developer		

In the **Status Area** you can find information about the system.


TrackStudio. Copyright © 2002-2003, TrackStudio, Ltd.

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.

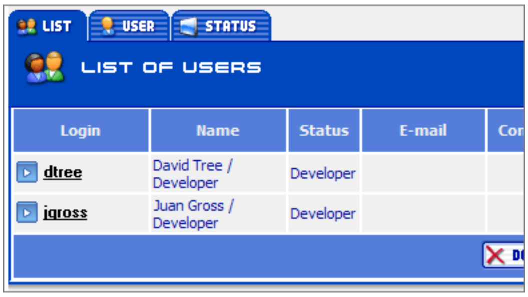
The list of user properties can be found in User Management (⧉ see page 78).

6.2 List of Users Project Manager

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

Description

The **List** 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, Company, Services, Time Zone, Locale**, and **Last Visited Date**. The purpose of the fields is described in User Management (⧉ see page 78) .



Login	Name	Status	E-mail	Co
dtree	David Tree / Developer	Developer		
jgross	Juan Gross / Developer	Developer		

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).

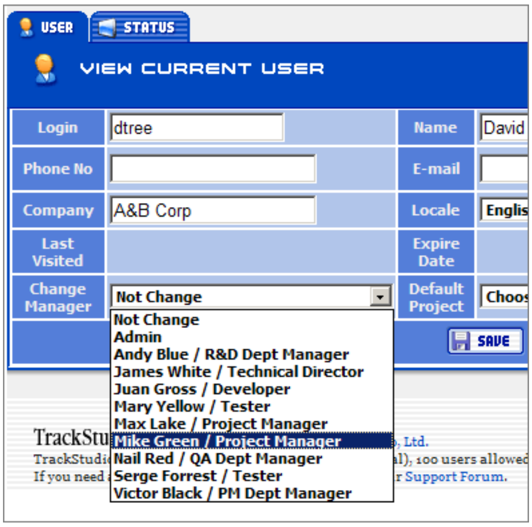
6.3.1 Edit User User

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 the **Add User** button. The detailed description of the user properties can be found in User Management (◘ see page 78).



Login	dtree	Name	David
Phone No		E-mail	
Company	A&B Corp	Locale	Englis
Last Visited		Expire Date	
Change Manager	Not Change	Default Project	Choos

Not Change
Admin
Andy Blue / R&D Dept Manager
James White / Technical Director
Juan Gross / Developer
Mary Yellow / Tester
Max Lake / Project Manager
Mike Green / Project Manager
Nail Red / QA Dept Manager
Serge Forrest / Tester
Victor Black / PM Dept Manager

TrackStu ... , Ltd.
TrackStudi ... l), 100 users allowed
If you need ... r Support Forum.

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.

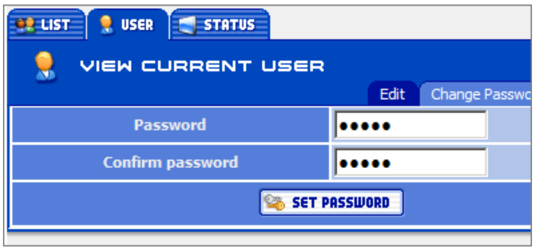
Notes

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



Password	●●●●●
Confirm password	●●●●●

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.

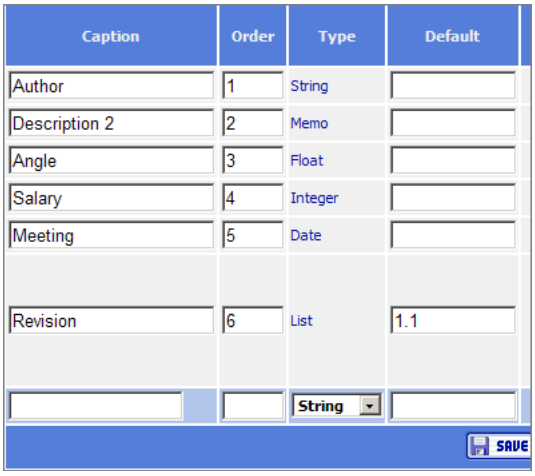
Notes

When you change the password on the **Change Password** tab, the password is changed only in the database, not in the LDAP.

6.3.3 Customize User Project Manager

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

Description



Caption	Order	Type	Default
Author	1	String	
Description 2	2	Memo	
Angle	3	Float	
Salary	4	Integer	
Meeting	5	Date	
Revision	6	List	1.1
		String	

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.
Forumula	Pressing this button opens the expression editor for calculated custom fields. A field is considered calculated if there is an expression specified for it. You can get more details about calculated custom fields in the topic Calculated Custom Fields (see page 98).
List of Values	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.

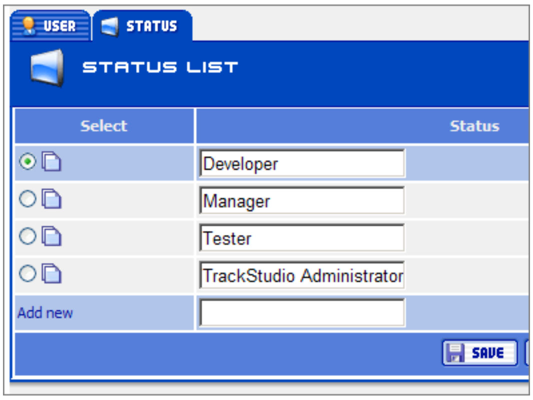
6.4.1 List User Status Project Manager

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** button to select current status. You can view or edit selected user status.
Copy	Use the **Copy** button to copy status.
Status	Status name
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 or selected user status.



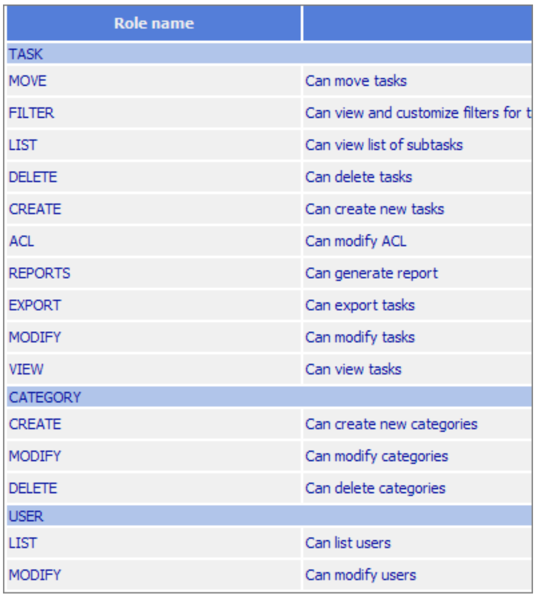
Select	Status
●	Developer
○	Manager
○	Tester
○	TrackStudio Administrator
Add new	

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



Role name	
TASK	
MOVE	Can move tasks
FILTER	Can view and customize filters for t
LIST	Can view list of subtasks
DELETE	Can delete tasks
CREATE	Can create new tasks
ACL	Can modify ACL
REPORTS	Can generate report
EXPORT	Can export tasks
MODIFY	Can modify tasks
VIEW	Can view tasks
CATEGORY	
CREATE	Can create new categories
MODIFY	Can modify categories
DELETE	Can delete categories
USER	
LIST	Can list users
MODIFY	Can modify users

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.
User type privileges	Grant users the ability to view, create, modify and delete user types (groups).
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.
E-Mail import privileges	Grant users the ability to view and modify e-mail submission rules.
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 Advanced Topics

The instructions in this manual are intended for TrackStudio users who have a working knowledge of TrackStudio and would like to use TrackStudio's advanced features.

7.1 Full Text Search User

TrackStudio uses Lucene for text indexing, which provides a rich query language that can make constructing full text queries daunting. This document is derived from the Lucene document on Query Parser Syntax.

Description

A query is broken up into terms and operators. There are two types of terms: Single Terms and Phrases. A Single Term is a single word such as "test" or "hello". A Phrase is a group of words surrounded by double quotes such as "hello dolly". Multiple terms can be combined together with Boolean operators to form a more complex query (see below). All query terms are case insensitive.

TrackStudio supports modifying query terms to provide a wide range of searching options.

Wildcard Searches

TrackStudio supports single and multiple character wildcard searches. To perform a single character wildcard search use the "?" symbol. To perform a multiple character wildcard search use the "*" symbol. You cannot use a * or ? symbol as the first character of a search. The single character wildcard search looks for terms that match that with the single character replaced. For example, to search for "text" or "test" you can use the search:

```
te?t
```

Multiple character wildcard searches looks for 0 or more characters. For example, to search for Windows, Win95 or WindowsNT you can use the search:

```
win*
```

You can also use the wildcard searches in the middle of a term. For example, to search for Win95 or Windows95 you can use the search

```
wi*95
```

Fuzzy Searches

TrackStudio supports fuzzy searches. To do a fuzzy search use the tilde, "~", symbol at the end of a Single word Term. For example to search for a term similar in spelling to "roam" use the fuzzy search:

```
roam~
```

This search will find terms like foam and roams

Boolean Operators

Boolean operators allow terms to be combined through logic operators. TrackStudio supports AND, "+", OR, NOT and "-" as Boolean operators . Boolean operators must be ALL CAPS.

OR

The OR operator is the default conjunction operator. This means that if there is no Boolean operator between two terms, the OR operator is used. The OR operator links two terms and finds a matching document if either of the terms exist in a document. This is equivalent to a union using sets. The symbol || can be used in place of the word OR.

To search for documents that contain either "software TrackStudio" or just "TrackStudio" use the query:

```
software || TrackStudio
```

or

```
software OR TrackStudio
```

AND

The AND operator matches documents where both terms exist anywhere in the text of a single document. This is equivalent to an intersection using sets. The symbol && can be used in place of the word AND.

To search for documents that contain "software TrackStudio" and "issue tracking" use the query:

```
TrackStudio AND tracking
```

Required term: +

The "+" or required operator requires that the term after the "+" symbol exist somewhere in a the field of a single document.

To search for documents that must contain "TrackStudio" and may contain "software" use the query:

```
+TrackStudio software
```

NOT

The NOT operator excludes documents that contain the term after NOT. This is equivalent to a difference using sets. The symbol ! can be used in place of the word NOT.

To search for documents that contain "software TrackStudio" but not "japan" use the query:

```
TrackStudio NOT japan
```

Note: The NOT operator cannot be used with just one term. For example, the following search will return no results:

```
NOT TrackStudio
```

Excluded term: -

The "-" or prohibit operator excludes documents that contain the term after the "-" symbol.

To search for documents that contain "software TrackStudio" but not "japan" use the query:

```
TrackStudio -japan
```

Grouping

TrackStudio supports using parentheses to group clauses to form sub queries. This can be very useful if you want to control the boolean logic for a query.

To search for either "software" or "TrackStudio" and "bugs" use the query:

```
(software OR TrackStudio) AND bugs
```

This eliminates any confusion and makes sure you that bugs must exist and either term software or TrackStudio may exist.

Escaping Special Characters

TrackStudio supports escaping special characters that are part of the query syntax. The current list special characters are

+ - && || ! ( ) { } [ ] ^ " ~ * ? : \

To escape these character use the \ before the character. For example to search for (1+1):2 use the query:

\(1\+1\)\:2

7.2 E-Mail Notification User

TrackStudio can send out email notifications when a certain event occurs or at regular intervals. Each e-mail notification type is configured separately. To receive e-mail notifications, the user must subscribe to them or ask the manager to subscribe to them.

Description

TrackStudio can send out e-mail notifications when the following events occur:

- a new task is added to the system or an existing task is modified
- a message is added to the task or other associated events (changing the handler, entering the elapsed or estimated time by the developer, etc .)

You can activate this type of notification on the **Filters->Notify** tab.

Sending out e-mail notifications at regular intervals is performed according to the specified schedule irrespective of the events occurring in the system. You can activate this notification type on the **Filters->Subscribe** tab.

Let us take a closer look at event-based e-mail notifications. Different e-mail notification rules can be active for different projects simultaneously. Each rule is defined using the following parameters:

- **Filter** -- the filter determines for which events e-mail notifications should be sent. For example, a rule can be defined so that a notification will be sent only if the subscriber is the handler of the task (*task handler="current user"*). TrackStudio checks whether the filtering conditions have been met and sends out e-mail messages after the task has been modified. You can specify a filter based on the message parameters for many modifications, such as changes in the status, the resolution, the handler or others. E-Mail notification rules do not depend on the filter currently selected on the subtasks tab.
- **Task** -- Email notification system is activated for a certain task or project. If the system is activated for a project, it affects all sub-projects and the tasks contained within. For example, if a user subscribed to email notifications for the project *TestPro*, it would automatically be activated for all subtask of this project. Also, the rule would be activated both for the existing tasks and for future tasks .
- **User** -- a user is the recipient of email notifications. To receive e-mail notifications, a user should specify his or her e-mail address. If the e-mail address is not specified, e-mail notifications can’t be sent to this user.

When a task is modified or a message is added, only one e-mail notification is sent, even if the change falls under several filters. In such cases the task field which is closest to the modified task in the hierarchy of tasks is used. For example, if we have the following hierarchy of projects *A->SubA->SubSubA*, the subscription to the filter *X* is activated for the task *A*, the subscription to the filter *Y* is activated for the task *SubA*, the email notification will be performed according to the filter *Y* when *SubSubA* is modified. If one level has an activated subscription to a number of filters, the filter is selected randomly.

When defining an e-mail notification rule it is important to differentiate between the following tasks:

1) The parent task for the filter -- this task is displayed in the **Task** column on the **Filters->View** tab. This task determines for which tasks the filter will be visible; it does not directly influence the e-mail notification.

2) The task for which the e-mail notification is activated. This task is displayed in the **Task** column on **Filters->Notify** tab. The email notification will be sent when this task or one of its subtasks are modified. You can activate the email notification for several of the subprojects in each filter.

3) The modified task -- modifying this task results in email notifications.

Workflow settings also influence the e-mail notification. The manager can forbid sending out email notifications to certain user groups on certain transitions. You can find more details about the settings of the workflow in the topic Transitions (↗ see page 73).

The email notification system checks the filtering conditions a bit different from the usual routine. That is why a filter may be useful for the e-mail notification system, but often cannot be used to filter tasks.

Each filter consists of two parts – one part defines the rules for filtering tasks, while the other part defines the rules for filtering messages. When a new task is created, the notification will be sent if the task meets the filtering conditions for tasks. If there are no filtering conditions specified, the e-mail notification will be sent when any task is modified. When a task is created or modified, filtering conditions for messages are not checked even if they are specified. The following parameters are also ignored:

- **Task/Page**
- **Deep search**
- **Sort order**
- **Column visibility**

Note that when a new task is created, the e-mail notification is sent once the **Save** button is pressed -- not the **Add** button. If the **Save** button is not pressed, the e-mail notification will not be sent.

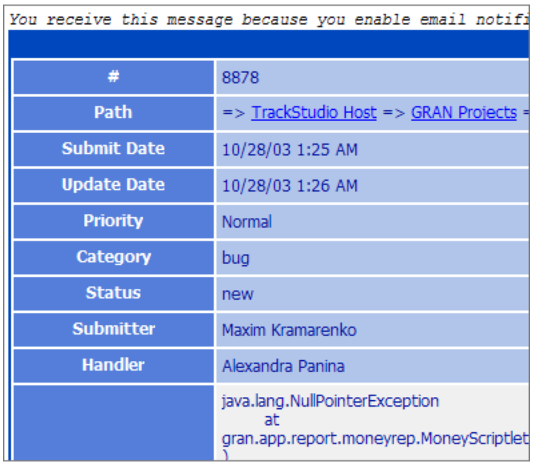
When a message is created, the first thing TrackStudio checks is whether the task meets filtering conditions. Then it determines if the added message meets the filtering conditions for messages. If there are no filtering conditions for messages specified, the e-mail notification is sent when any message is added. Sending e-mail notifications must also be allowed in the settings of the workflow.

TrackStudio does not check the following filter parameters while checking the filtering condition for messages:

- **Task/Page**
- **Deep search**
- **Sort order**
- **Column visibility**
- **View Messages**
- **Filter Messages**
- **Bulk Processing tool**

In the email notification rules, the “current user” means the subscribed user; but when filtering tasks, the “current user” means the user who has modified the task or added a message. For example, if a user performs filtering tasks and makes use of the *“handler=current user”* condition, it will display the tasks of the logged user. If a user customized the email notification system using such a filter, he or she will receive notifications if he is the handler of the modified tasks no matter who is modifying them.

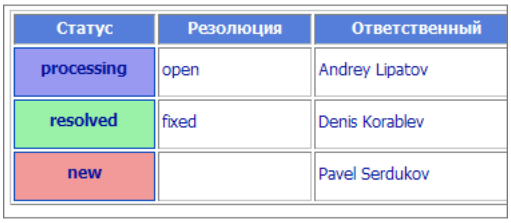
Email notifications are sent out in HTML or text format. At the beginning of each email notification message, there is information about why you are receiving this email notification. After that you will find the information about the task which has been changed.



You receive this message because you enable email notifi

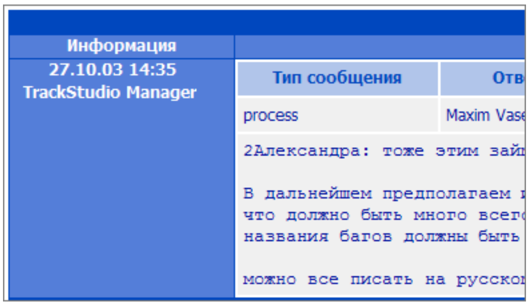
#	8878
Path	=> TrackStudio Host => GRAN Projects =
Submit Date	10/28/03 1:25 AM
Update Date	10/28/03 1:26 AM
Priority	Normal
Category	bug
Status	new
Submitter	Maxim Kramarenko
Handler	Alexandra Panina
	java.lang.NullPointerException at gran.app.report.moneyrep.MoneyScriptlet)

If the task has any subtasks, the list of subtasks is displayed according to the filtering conditions. Note that such parameters as **deep search** or **view messages** are not used to determine the necessity of sending a notification, though they are used to format the e-mail notification. For example, if a filter used for e-mail notification has the **deep search** option on, the e-mail notification will include the list of all subprojects of the modified project.



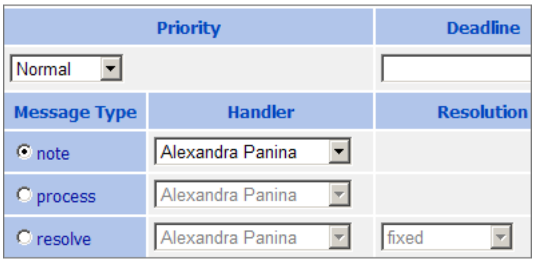
Статус	Резолюция	Ответственный
processing	open	Andrey Lipatov
resolved	fixed	Denis Korablev
new		Pavel Serdukov

Then follows the list of messages for the modified task.



Информация		
27.10.03 14:35 TrackStudio Manager	Тип сообщения	Отв
	process	Maxim Vase
	2Александра: тоже этим зай В дальнейшем предполагаем что должно быть много всег названия багов должны быть можно все писать на русско	

If submitting via email is enabled and the HTML email notification is used, the form for submitting via e-mail is displayed.



Priority		Deadline
Normal		
Message Type	Handler	Resolution
● note	Alexandra Panina	
○ process	Alexandra Panina	
○ resolve	Alexandra Panina	fixed

Example

Here are some examples of using filters for the email notification system.

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 (bug-note).

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" "is" "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.

4) "*(task) handler" "is" "current user" and "message submitter" "is not" "current user"*. If subscribed to this filter, a user will get email notifications when other users add a message to the task the handler of which is the subscribed user.

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 there is a change in the task status or any messages are added.

7.3 E-Mail Submission Project Manager

Users with valid TrackStudio accounts can submit tasks or messages by e-mail as long as their account includes an e-mail address.

Description

TrackStudio supports the following e-mail submission types:

- Task submission
- Plain message submission
- Form-based message submission

If the e-mail submission option is enabled, TrackStudio checks the mailbox at regular intervals. TrackStudio gets an e-mail from the mailbox and tries to import it. If the e-mail meets the requirements of one of the mail submission schemes, the system imports the e-mail as a new task or message. If the e-mail is a reply from a mailer-daemon about a failure while sending the e-mail, it is deleted from the queue. If the message cannot be imported and at the same time it is not a reply from a mailer-daemon, it is either deleted from the queue or forwarded to the specified address.

7.3.1 Task Submission Project Manager

Task submission makes it possible to import e-mail messages as subtasks for the specified project.

Description

To enable task submission, you should open the **Task->E-Mail import** form and set the following options:

- **Enable e-mail import** – enables task submission for the current task.
- **Contains keyword** – allows you to specify the keyword either in the subject or in the body of the message. Keywords make it possible to use one mailbox for importing messages into several projects, therefore you should use different keywords for different projects. If no keyword is specified, the e-mail is imported regardless of its body or subject. The e-mail import rules with the empty keyword list are checked after the rules with a keyword specified. The keyword check is case-insensitive.
- **In** – allows you to specify the field in which to search for the keyword. The **subject** and **body** fields are supported.
- **Category** – the category of the created tasks. The category must be associated with a workflow which has the start state.

If the following conditions are true:

- e-mail submission is enabled;
- the message matches the conditions specified on the E-Mail Import tab;
- the sender’s name or e-mail address belongs to one of the system users; and
- the sender has access rights to the project;

the e-mail message is imported into the system. The fields for the created task are determined in the following way:

Property	Value
Name	E-Mail subject
Description	E-Mail body
Parent task	The task having mail import enabled for it
Category	The category specified in the e-mail import options
Submitter	E-Mail sender
Handler	The handler of the parent task
Priority	Default priority
Status	The default state of the corresponding workflow

Submit date	The data when the e-mail message was processed and imported into the system

If the e-mail message has any attachments, they are added to the created task and can be found on the **Uploads** page. If the task has any custom fields, they are set into the default value.

As soon as the task is created the system sends out an ordinary e-mail notification message to the subscribers.

Example

Joe can use his e-mail account joe@mycompany.com to submit an item. Using his e-mail client, Joe creates a new mail message and sends it to the e-mail address provided by the TrackStudio administrator. For example, the mailbox might be support@mycompany.com. Joe can type a subject, a message and add attachments to the message. If the mailbox has been configured correctly, TrackStudio checks this mailbox for messages periodically. TrackStudio also determines if e-mail import is enabled for any project and if the message meets task submission rules for this project. If everything is valid, TrackStudio imports the task into the specified project.

7.3.2 Plain Message Submission Project Manager

Plain Message Submission allows you to import e-mail messages as notes to tasks.

Description

Unlike Task Submission, you do not have to configure Plain Message Submission for each project separately. When E-Mail Submission is enabled, TrackStudio enables Plain Message Submission for all projects.

If the following conditions are true:

- e-mail submission is enabled;
- the message subject contains a task number specified as #<task number>, e.g. #55;
- the sender’s name or e-mail address belongs to one of the system users;
- the sender has access rights to the specified task; and
- default message type is specified in the workflow;

the e-mail message is imported into the system. The fields for the created task are determined in the following way:

Property	Value
Message body	E-Mail body
Submit date	The date when the e-mail message was imported into the system

Submitter	E-Mail sender
Message type	The default message type specified in the task workflow

If the e-mail message has any attachments, they are added to the created task and can be found on the **Uploads** page. As soon as the task is created the system sends out an e-mail notification to the subscribers.

Example

Joe can use his e-mail account joe@mycompany.com to submit a new bug-note to task 55. Using his e-mail client, Joe creates a new mail message with the subject “#55” and sends it to the e-mail address provided by the TrackStudio administrator. For example, the mailbox might be support@mycompany.com. Joe can type a message and add attachments to the message. If the mailbox has been configured correctly, TrackStudio checks this mailbox for messages periodically. TrackStudio also determines if e-mail import is enabled for any project and if the message meets task submission rules for this project. If everything is valid, TrackStudio imports the task as a subtask of the specified project.

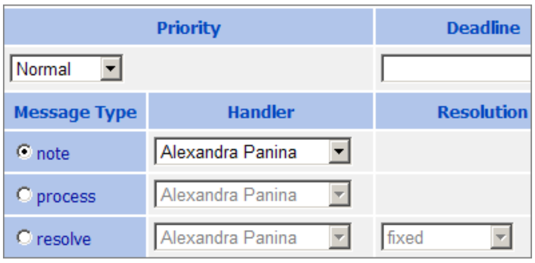
7.3.3 Form-Based Message Submission Project Manager

Form-Based Message Submission makes it possible to import e-mail messages into the system as messages containing the hours spent, the priority, handler, the resolution, etc.

Description

To set parameters for a message, you can use the web form located at the end of an HTML e-mail notification. As with Plain Message Submission, you do not have to configure Form-Based Message Submission for each project separately. When E-Mail Submission is enabled, TrackStudio enables Form-Based Message Submission for all projects.

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.



Priority		Deadline
Normal ▼		
Message Type	Handler	Resolution
● note	Alexandra Panina ▼	
○ process	Alexandra Panina ▼	
○ resolve	Alexandra Panina ▼	fixed ▼

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. Please note that many of the popular Web-based e-mail systems such as Yahoo.com and Mail.com/Email.com intentionally disable JavaScript in messages, and there is no way to re-enable it, so these systems will not work with TrackStudio. Following are instructions for enabling JavaScript in some popular e-mail readers. (Different versions of these readers may be slightly different in the details, but are probably similar.) For more information about these readers, please check the documentation or visit the vendor sites.

Mozilla Messenger

1.From the menu bar, choose **Edit**, then **Preferences**

2.Select **Advanced** from the list of options, then **Scripts & Plugins**

3.Click the box to **Enable JavaScipt for Mail and News**

Outlook Express

1.From the menu bar, choose **Tools**, then **Options**, then **Security**

2.Under **Virus Protection**, then under **Select the Internet Explorer security zone to use**, select **Internet Zone (less secure but more functional)**

Outlook 2000

1.From the menu bar, choose **Tools**, then **Internet Options**

2.Select **Security** Tab

3.Under **Secure Content**, select **Internet Zone**

Problems with E-Mail Stationery in Outlook 2002 (from http://www.scrippy.com/help/mail/outlook2002.htm)

If you have the **Preview Pane** enabled, note that Outlook 2002 will never display stationery as intended in the preview pane. It always has "scripting" turned off for the preview pane and cannot display advanced effects there.

To view a received stationery in Outlook 2002 you must double-click the e-mail to open it in a separate window, and then in the message window click on **View** and then under that on **View in Internet Zone**. (If you do not have this option in your **View** menu, please read the following paragraphs.)

There is also a feature in Outlook 2002 which is supposed to permanently enable the viewing of messages in the **Internet Zone** as above, so that you don't need to manually select this every time. This option is in the main Outlook 2002 window, under **Tools**, **Options...**, the **Security** tab. Under **Secure Content** beside **Zone** you are supposed to be able to select **Internet** instead of the default **Restricted Sites**.

Unfortunately, the Zone setting which is supposed to permit viewing of stationery has a bug and can just cause a worse problem. In the initial release of Outlook 2002, using this setting does not actually put you into the **Internet Zone** but it does remove the **View in Internet Zone** option from the message window's menu. The result is that you then can't view stationery at all! We hope that this bug will be fixed in a service pack. In the meantime, if you try using **Tools**, **Options**, **Security**, **Zone** to select the **Internet Zone**, and the result is that stationery still does not work, we recommend changing that setting back to **Restricted sites** and then using the method described earlier when you view an e-mail stationery message.

7.4 Calculated Custom Fields Project Manager

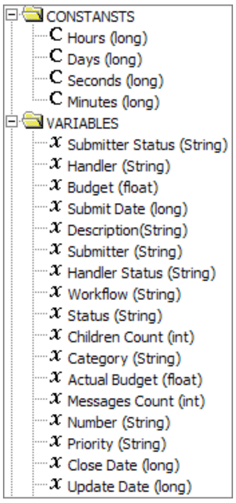
This topic describes how to use calculated custom fields to customize filters, reports and email notification rules.

Description

To create a calculated custom field, you should create a static (non-calculated) custom field and then specify an expression for it. If a custom field has an empty expression, it is considered static. So, if you delete the expression from a calculated custom field, it becomes static. No results can be saved to the database for a calculated field. Calculated custom fields can be created in TrackStudio by those users who have the right to create static custom fields.

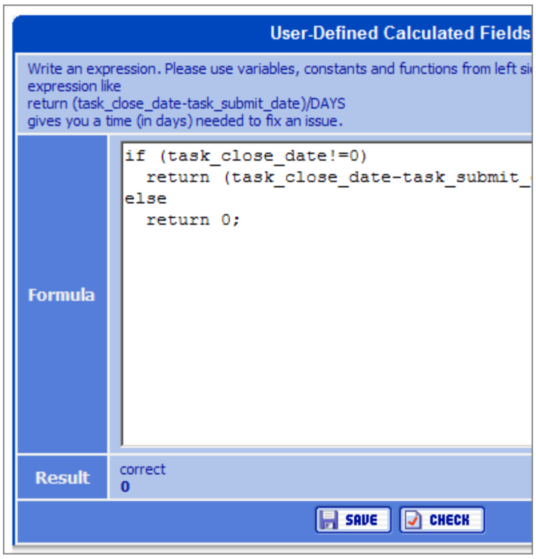
You can specify an expression for calculations in the expressions editor. TrackStudio has a formula editor for editing expressions and checking syntax.

In the left-hand section of the editor’s window, you’ll find a tree with available operations, variables, constants and functions. Left-click on a variable or a constant to add it to the expression.



- CONSTANTS
  - C Hours (long)
  - C Days (long)
  - C Seconds (long)
  - C Minutes (long)
- VARIABLES
  - x Submitter Status (String)
  - x Handler (String)
  - x Budget (float)
  - x Submit Date (long)
  - x Description(String)
  - x Submitter (String)
  - x Handler Status (String)
  - x Workflow (String)
  - x Status (String)
  - x Children Count (int)
  - x Category (String)
  - x Actual Budget (float)
  - x Messages Count (int)
  - x Number (String)
  - x Priority (String)
  - x Close Date (long)
  - x Update Date (long)

In the right-hand part of the window, you'll find a field where you can enter the expression and two buttons – **Save** and **Check**. The **Check** button will check the correctness of the expression by calculating it for the current task. The **Save** button will save the expression to the custom field.



User-Defined Calculated Fields

Write an expression. Please use variables, constants and functions from left si
expression like
return (task_close_date-task_submit_date)/DAYS
gives you a time (in days) needed to fix an issue.

Formula

```
if (task_close_date!=0)
  return (task_close_date-task_submit_
else
  return 0;
```

Result: correct
0

TrackStudio always checks the expressions before saving, but it only checks the current task or user. Sometimes an expression correct for one task can be incorrect for another (for example, it can lead to dividing by zero). If the expression is incorrect or if the calculated value type does not match the field type, the result will be an empty field.

TrackStudio uses Java-like language based on BeanShell to evaluate expressions. It means that you can create not only basic mathematical expressions, but also more complex expressions. (ie. such as, if, for, or while.) You can also use some java classes. For security reasons you can only use following classes in your expressions:

```
java.lang.Boolean
java.lang.Byte
java.lang.Character
java.lang.Class
java.lang.Comparable
java.lang.Double
java.lang.Float
java.lang.Integer
java.lang.Long
java.lang.Math
java.lang.Number
java.lang.Object
java.lang.Short
java.lang.String
java.lang.StrictMath
java.lang.StringBuffer
java.util.*
java.text.*
java.sql.Date
java.sql.Time
java.sql.Timestamp
```

In TrackStudio Enterprise you can use any classes in your expressions (including you own). To do that, you should change the class **gran.tools.ShellClassLoader** and add the classes and packages that you need.

Calculated custom fields and static fields can be created for tasks, users and workflows by using different sets of variables.

For instance, let us create a custom field that will return the number of days since the last update (this field can be used to find out what tasks have been neglected for a long time).

1. Go to the **Task->Customize** tab.

2. Specify **Caption** for the custom field: *since*.

3. Select the field type: **Integer**

4. Click the icon **f(x)** to create an expression.

5. In the left-hand part of the popup window open the **Variables** folder, and select **Update Date**. You can see the type of the variable after its description.

6. As **task_update_date** belongs to the long type and is given in milliseconds, you should get the present time in milliseconds and subtract task_update_date from it. That should give you the following expression:

```
(new Date()).getTime()-task_update_date
```

7. Now you have the difference between the present time and the date of the task update in milliseconds. To convert this number to days, you should modify expression to the following one

```
((new Date()).getTime()-task_update_date)/DAYS
```

8. Press the **Check** button. The expression should be correct.

9. Save the expression using the **Save** button.

TrackStudio allows you to create calculated custom fields of the same types as static custom fields. (i.e. **Integer, Float, String, Date, List**.) You should ensure that the result of the calculation matches the required type and convert the result of the expression to the proper type as needed. For example, if you get the result of date calculation in milliseconds (long), you should convert it to the **Date** type using the constructor

```
new Date(milliseconds)
```

Special attention should be paid to the **List** data type. List is a set of values. In a static custom field you choose one of these values and set it. In calculated fields you should first specify the set of values as usual and then specify the expression which will return the result that exactly matches one of the values (not the keys) from the list.

For example, to create a custom field that will return the day of the week on which an issue was created.

1. Go to the **Task->Customize** tab.

2. Enter **Caption** for the custom field: *weekday*.

3. Select the field type: **List**.

4. Specify the list of possible values. Enter only one value at a time: *Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday*.

5. Click the icon **f(x)** to create an expression.

6. Type the following expression:

```
Calendar ca = Calendar.getInstance();
ca.setTime(new Date(task_submit_date));
int day = ca.get(Calendar.DAY_OF_WEEK);
switch (day){
     case Calendar.SUNDAY: return "Sunday";
     case Calendar.MONDAY: return "Monday";
     case Calendar.TUESDAY: return "Tuesday";
     case Calendar.WEDNESDAY: return "Wednesday";
     case Calendar.THURSDAY: return "Thursday";
     case Calendar.FRIDAY: return "Friday";
     case Calendar.SATURDAY: return "Saturday";
}
return null;
```

7. Press the **Check** button. The expression should be correct.

8. Save the expression using the **Save** button.

TrackStudio allows you to create custom fields of all types (**List, String, Date**). When custom fields specified for a task or a workflow is calculated, the following variables are defined:

Variable	**Type**	**Description**
task_budget	float	task budget (in hours)
task_category	String	task category name
task_priority	String	task priority name
task_actual_budget	float	task actual budget (in hours)
messages_count	int	messages for the task
task_submit_date	long	task submit date (in msec)
task_workflow	String	task workflow name
task_deadline	long	task deadline (in msec)

task_handler	String	task handler name
task_close_date	long	task close date (in msec)
task_status	String	task status (state) name
task_update_date	long	task update date (msec)
task_children_count	int	amount of subtasks
task_resolution	String	task resolution
task_name	String	task name
task_number	String	task number (#)
task_description	String	task description
task_submitter_status	String	submitter status name
task_handler_status	String	handler status name

When custom fields specified for a User is calculated, the following variables are defined:

Variable	Type	Description
user_company	String	user company name
user_email	String	user e-mail address
user_children_count	int	amount of subordinated users
user_expire_date	long	user expire date (in msec)
user_lastvisited	long	user last visit date (in msec)
user_locale	String	user locale
user_login	String	user login
user_manager_login	String	login of user's manager
user_manager_name	String	name of user's manager
user_name	String	user name
user_status	String	user status name
user_phone	String	user phone number
user_timezone	String	user time zone

Also you can use the following constants:

Constant	Type	Description
DAYS	long	msec/day
HOURS	long	msec/hour
MINUTES	long	msec/minute
SECONDS	long	msec/second

7.5 Data Analysis Project Manager

“Distribution reports” allow statistical analysis of the data distribution.

Description

To design this kind of report, you must specify a filter that filters out the required tasks; the data that should be displayed on the X and the Y axes; which parameter should represent the data; and which function should be calculated for the selected parameter.

The X and the Y axes’ parameters can be any parameter having a fixed amount of values, including custom fields of the **List** type. The data can be any numeric field, including custom fields of the **Integer** and **Double** types. The following functions are available: **max, min, avg, sum**. Note that not every possible parameter combination can make sense.

When constructing the report the system finds subtasks of the current task which satisfy the filter condition. After that the data is grouped based on parameters specified for the X and the Y axes, and for each group the value of the target function is calculated.

Example 1

Let us assume you have chosen the following parameters:

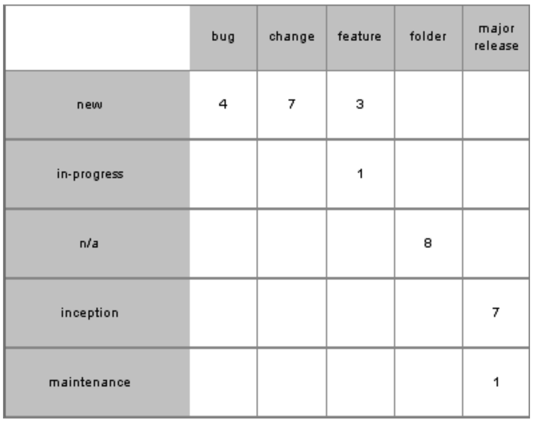
```
X: Category
Y: Submitter
Value: Actual Budget
Function: Sum
```

In this scenario the system will determine the the worktime required to solve the task from the task submitter and the category of the task. The system will also calculate the total time expenses on each category, on each submitter and the total time spent on all the tasks.

Example 2

Let’s design a report demonstrating the distribution of the number of tasks depending on the category and the status:

```
X: category
Y: status
Value: Task Amount
Function: Sum
```



	bug	change	feature	folder	major release
new	4	7	3		
in-progress			1		
n/a				8	
inception					7
maintenance					1

Example 3

Create the calculated custom field *'submitdate'* with the List type accounting the following formula:

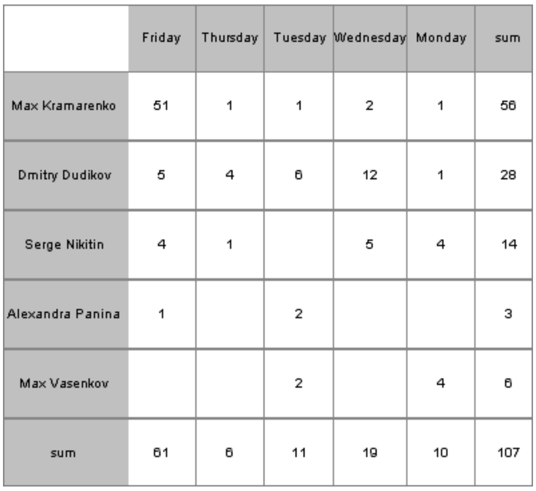
```
Calendar ca = Calendar.getInstance();
ca.setTime(new Date(task_submit_date));
int day = ca.get(Calendar.DAY_OF_WEEK);
switch (day){
     case Calendar.SUNDAY: return "Sunday";
     case Calendar.MONDAY: return "Monday";
     case Calendar.TUESDAY: return "Tuesday";
     case Calendar.WEDNESDAY: return "Wednesday";
     case Calendar.THURSDAY: return "Thursday";
     case Calendar.FRIDAY: return "Friday";
     case Calendar.SATURDAY: return "Saturday";
}
return null;
```

The list of the possible values of the elements in the list must correspond to the list of possible results of the calculation of the formula. This calculated field returns the name of the day of the week when the task was created.

In order to find out how many tasks are created and by whom on different days of the week, create the following filter:

```
X: submiday
Y: submitter
Value: Task Amount
Function: Sum
```

For **Task Amount** only the **Sum** function makes sense as the maximum and average value will be always equal to 1. Design the report in any format convenient for you.



	Friday	Thursday	Tuesday	Wednesday	Monday	sum
Max Kramarenko	51	1	1	2	1	56
Dmitry Dudikov	5	4	6	12	1	28
Serge Nikitin	4	1		5	4	14
Alexandra Panina	1		2			3
Max Vasenkov			2		4	6
sum	61	6	11	19	10	107

From this example of the report it is clear that the greatest amount of tasks are created by *Max Kramarenko*. It is also clear that the tasks are created predominantly on *Fridays*, and that the fewest amount of tasks are created on *Thursdays*.

7.6 Internationalization Developer

This section describes how to localize your TrackStudio interface.

Description

Internationalization is the process of changing the format of dates and numbers to the one used in your region and translating user-interface text to your language. For example, the 12th of January 1990 will look like 01/12/90 in the American date format and 12.01.1990 in the German one. The date format in TrackStudio is used when displaying and inputting information.

Note: all input fields which require the date to be entered are marked with the icon, pressing it will open a popup calendar where you can select the date. This calendar already takes into account the appropriate date format..

Similarly, some user-interface text like *Your Name* in English can be translated to *Ihr Name* for German locales.

User locale is specified separately for each user in the user settings (**User Management->User->Edit**). There is a drop-down box with the list of available locales on this page. The list is determined by the settings of Java Virtual Machine on which TrackStudio is running.This list likely contains the locale that you need, but there may be no resources in TrackStudio for some locales. The resources are language files containing the translation of all text messages in TrackStudio into your language and a set of images that determine the appearance of TrackStudio. All the necessary resources for the English è Russian locales are already included in the standard TrackStudio. If you want to localize

TrackStudio, consult the section Localizing the Interface (⧉ see page 106).

If you select a locale for which there are no resources, the date format will change to the selected language, but the interface will remain English (by default).

7.6.1 Character Encoding Developer

This section describes character encoding.

Description

Character Encoding is specified for the entire TrackStudio instance. If you are using Enterprise Server Manager, specify the character encoding on the **General** tab. Another way to specify character encoding is to edit the parameter **trackstudio.encoding** in the file **trackstudio.properties**.

```
trackstudio.encoding UTF-8
```

7.6.2 Localizing the Interface Developer

To localize the interface, you should do two things: translate the text part of the interface into the language you need and edit the set of images.

Description

Translating the text interface

The entire TrackStudio text is stored in resource bundles. A resource bundle is a file containing key/value pairs. TrackStudio loads its text using keys while the correct values are retrieved based on the user’s locale settings.

For example, the key/value pairs for the English locale may look like this:

```
UserTitle.inc.SERVICES=Services
UserTitle.inc.TIME_ZONE=Time Zone
UserTitle.inc.LAST_VISITED=Last Visited
UserTitle.inc.LOCALE=Locale
```

Making your own translation involves copying the English resource bundle, renaming the file, and translating its contents. To do that, find the file language_en.properties in the directory TrackStudio/webapps/TrackStudio/WEB-INF/classes and copy it to a new file. The last two letter in the name of the new file must be a valid ISO Language Code.

These codes are the lower-case two-letter codes as defined by ISO-639. You can find a full list of these codes at a number of sites, such as: http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt

International Characters

When translating from English, you may need to use special characters for you language.

Unfortunately, all resource bundle files must be saved in ASCII format which doesn't allow for many international characters. It's recommended you work on your translation in a text editor that supports all characters in your language. After finishing your translation, use the **native2ascii** tool to convert international characters to the ASCII format. Here's how you use the native2ascii tool:

```
native2ascii -encoding XXX
             my_translation.properties language_YY.properties
```

where *my_translation.properties* is the input file that use national character encoding and *language_YY.properties* is the output file.The **-encoding XXX** parameter is optional. If you don't specify it, Java will use the default encoding value, taken from the system property **file.encoding**.After you translate the file, save it to the directory **TrackStudio/webapps/TrackStudio/WEB-INF/classes**.

Creating your own skin.

A skin consists of a set of GIF images which are elements of the TrackStudio interface and a set of CSS files (cascading style sheets). These sets are located in the directory **TrackStudio/webapp/skins** in folders with names like defaultSkin.XX (**defaultSkin.en, defaultSkin.ru**). The first part of the folder name is the name of the skin, while the second part is a two-letter locale abbreviation.

A folder containing a skin has the following structure:

style.html – a CSS file containing styles

style-print.html – a CSS file to format the print version in TrackStudio

style.js – a JavaScript file containing dynamic effects

cssimages/ - the folder with images used in style.html

images/ - the folder with the elements of the TrackStudio interface.

The elements of the TrackStudio interface are named in a special way. For example,

```
arw.time.gif
but.addusr.gif
but.addusr.h.gif
ico.copy.gif
mmn.logout.gif
mmn.logout.h.gif
mmn.tskmgm.a.gif
tab.acs.gif
tab.acs.a.gif
txt.mgs.gif
```

The explanation of the elements in the filenames:

but – button

arw, ico – icon

mmn – menu element

tab – tab element

txt – graphical text element

The index ‘h’ (hover) means that this image is replaced by its index-free pair when the mouse cursor is over it, the index ‘a’ (active) means that this image replaces the main one when the element is active (selected).

To create your own skin, copy the contents of the folder defaultSkin.en to a folder with a different name according to the rules stated above. Then use your image editor to create the necessary images and replace the ones that are in the folder. DO NOT CHANGE the filenames. Copy the folder with the new skin in the directory **TrackStudio/webapp/skins** and restart TrackStudio.

8 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.

8.1 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.

8.2 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.

8.3 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 62

Adapter Development 109

Adapter Structure 110

Advanced Topics 87

Authorize Through the LDAP 23

B

Building 113

C

Calculated Custom Fields 98

Categories 67

Change Password 81

Character Encoding 106

Concepts 1

Configuration Files 19

Creating a New Bug 34

Creating a New Filter 36

Creating a New Project 32

Creating a New User 31

Customize Task 57

Customize User 82

Customize Workflow 76

D

Data Analysis 103

Database Configuration 16

DB2 Configuration 18

Debugging Errors 28

E

Edit Category 68

Edit Filter 47

Edit Task 56

Edit User 80

Edit User Status 84

E-Mail Import 58

E-Mail Notification 89

E-Mail Submission 93

Enabling E-Mail Notification 37

Export 59

External User Self-Registration 24

F

Filter Subscription 53

Filters 45

Firebird Configuration 19

Form-Based Message Submission 96

Full Text Search 87

G

Getting Started 30

H

How to Install an SSL Certificate. 13

HSQLDB Configuration 16

I

Importing and Exporting the Database 14

Installation 7

Internationalization 105

L

List of Users 80

List Reports 65

List User Status 83

Localizing the Interface 106

M

Message Types 72

Messages 60

MS SQL Configuration 18

N

Notify by Email 54

O

Open API 109

Oracle Configuration 17

P

Plain Message Submission 95

PostgreSQL Configuration 17

Priorities 71

R

Report Types 63

Reports 63

Requirements 7

S

States 71

Status 83

Subtasks 43

T

Task 55

Task Concepts 40

Task Management 39

Task Submission 94

TrackStudio Server Manager 9

TrackStudio/SA Configuration 8

TrackStudio/WAR Configuration 16

Transitions 73

U

UNIX-Specific Notes 26

Update from 2.7 27

Uploads 61

User 80

User Concepts 79

User Management 78

V

View Filter 46

View Reports 66

W

What's New 4

Windows Service Management 12

Workflow Management 67

Workflows 70