

# ***Track+ 3.1***

Project Management Software





Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this documentation and we were aware of them, the designations have been printed in initial capital letters or all capital letters.

The author and publisher have taken care in the preparation of this document, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information contained in this document or the program itself.

Visit the **Track+** home page on the Web: <http://www.trackplus.de>

**Track+** was designed and written by

Knut Erik Ballestad,  
Mark Birch (Windows installer),  
Maxym Dmytrychenko,  
Clemens Edel,  
Jörg Friedrich,  
Clemens Fuchslocher,  
Alexei Khatskevich,  
and Hartmut Lang.

Translations were provided by

Knut Erik Ballestad (Norwegian),  
Jerry from EMIS (Chinese),  
Jörg Friedrich (German),  
Daniel Kuroczycki (French),  
Rocco Miceli, Cecilia Saltori, Nunzio Bonsignore (Italian),  
Masoud Omidvar (Farsi),  
and Ulf Palmér (Swedish).

Special thanks goes to all the people that submitted good bug reports.

This document was set with the  $\text{\LaTeX}$  typesetting system in Palatino typeface.

Copyright ©2001-2005 by Trackplus

May 2005

## Track+ License

**Track+** comes with two license models: one license for installations supporting not more than ten active users ("Open Source License"), and one license for commercial usage with more than ten users ("Commercial License"). For both license models, the copyright (C) is with Trackplus.

### Track+ Open Source License

For installations with up to ten users **Track+** + has been published as Open Source to facilitate modifications and enable small organizations to improve their project management capabilities without any financial investment.

Redistribution and use of the software described herein in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer listed below in the documentation and/or other materials provided with the distribution.
3. Any derivative work may not use the name and logo of **Track+** or TrackPlus as a product name or to endorse or promote products derived from this work without specific prior written permission by the copyright holders.
4. There are 10 active users or less in the **Track+** database, or it is being used on a significant open source project or in an educational institution or in a charity, or you have significantly contributed to the development of **Track+** over the last two years. Other users should obtain a commercial license (see <http://www.trackplus.de>).

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### Track+ Commercial License

The End User License Agreement for the Commercial License is available as a separate document from Trackplus. In summary, it grants you the rights to use the software based on the valid period and number of users defined in your purchase order. You may modify the software for your own purpose, but any modified software is subject to the same terms as the original.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Target Audience	1
1.2	Example Applications	2
1.3	Features	3
1.4	Key Concepts	3
1.4.1	RACI Method	3
1.4.2	Artifacts	4
1.4.3	Simplicity and Acceptability	5
<b>2</b>	<b>Installation</b>	<b>7</b>
2.1	System Requirements	7
2.2	Setup with Installer	9
2.3	Non-Installer Setup Overview	9
2.4	Step 1: Obtain the Track+ Package	10
2.5	Step 2: Install a Database Management System	11
2.5.1	MySQL Database	11
2.5.2	Firebird Database	12
2.5.3	MS SQL Server	18
2.5.4	Oracle Database	19
2.5.5	HSQL Database	19
2.5.6	IBM DB2 Database	21
2.5.7	PostgreSQL	21
2.5.8	SAP DB Database	22
2.6	Step 3: Install the Java SDK	22
2.7	Step 4: Install the Application Server	22
2.7.1	Tomcat	23
2.7.2	BEA WebLogic 8.1	25
2.7.3	JBoss 3.2	26
2.7.4	JRun 4	26
2.8	Step 5: Configure <b>Track+</b>	29
2.8.1	Configuring the Database	29
2.8.2	Configuring Logging	30
2.8.3	Configuring Timeouts	31
2.8.4	Configuring E-Mail	31
2.8.5	Authorizing E-Mail Domains	32
2.8.6	Configuring LDAP-Authentication	33
2.8.7	Configuring the Attachment Root Directory	34
2.8.8	Installing and Configuring Graphviz	34
2.9	Non-Western Character Sets	34

2.10	Operation in a Proxied Environment	35
2.11	Database Migration Notes	35
2.12	Porting to other Database Systems	36
2.13	Troubleshooting	36
<b>3</b>	<b>Configuration and Administration</b>	<b>37</b>
3.1	The "admin" User	37
3.2	Administering Projects	38
3.3	Assigning Users to Roles	38
3.4	Roles and Access Rights	40
3.4.1	Special Access Keys	40
3.4.2	General Access Key Scheme	42
3.4.3	Multiple Roles	44
3.5	Workflow Configuration	44
3.5.1	Creating a Workflow	45
3.5.2	Changing a Workflow	46
3.5.3	Workflow Reports	46
3.6	Site Administration	46
3.7	Customization	49
3.7.1	Customization Example: Changing a Priority	49
3.7.2	Changing the Report Overview	52
3.8	Adding new Locales	54
3.9	Database Backup	56
3.10	Security	57
<b>4</b>	<b>Using <i>Track+</i></b>	<b>59</b>
4.1	Registering with <i>Track+</i>	59
4.2	Getting Online Help	60
4.3	Creating and Changing Artifacts	60
4.4	Copying Lists from Templates	66
4.5	Quick Report	67
4.6	Reporting and Queries	67
4.7	Exporting Reports	69
4.8	<i>Track+</i> Query Language (TQL)	70
4.8.1	Sorting Criteria	70
4.8.2	Date and Time Functions	71
4.8.3	Attribute Alias Names and Values	72
4.8.4	More TQL Examples	72
4.9	Managing TQL Definitions	73
4.10	Using the Command Line Client	75
4.11	Diagrams	76
4.11.1	Bar Charts	77
4.11.2	Pie Charts	77
4.11.3	Gantt Diagrams	81
4.12	Monitoring Changes	81
4.13	Email Reminder	81
4.14	Linking to the System from a Website	83



<b>5</b>	<b>The <i>Track+</i> Default Process Model</b>	<b>85</b>
5.1	Roles . . . . .	85
5.2	States and State Changes . . . . .	85
5.3	Attributes . . . . .	88
5.4	List Types . . . . .	89
5.5	Email Notifications . . . . .	90
<b>6</b>	<b>Applying <i>Track+</i> to your Projects</b>	<b>93</b>
6.1	The Action Item List . . . . .	93
6.2	Milestone Tracking . . . . .	93
6.3	Risk Management . . . . .	94
6.4	The Issue List . . . . .	95
6.5	Personal To Do List . . . . .	96
6.6	Integration with CVS . . . . .	97
6.7	Generating Release Notes . . . . .	99
<b>A</b>	<b>System Limits</b>	<b>101</b>
<b>B</b>	<b>Frequently Asked Questions</b>	<b>103</b>
B.1	Installation and Administration . . . . .	103
B.2	Usage . . . . .	105
<b>C</b>	<b>Release Notes</b>	<b>109</b>
C.1	Upgrading . . . . .	109
C.2	Release 3.1.0 . . . . .	110
C.2.1	Bugs removed . . . . .	111
C.2.2	New Features . . . . .	111



# List of Figures

1.1	Major project phases	2
2.1	Track+ Installer (here in German)	10
2.2	Creating a new JRun server	27
2.3	Setting the server ports	27
2.4	Launching the newly created server	29
2.5	The site configuration page, first part	31
2.6	The site configuration page, second part	33
3.1	Managing a project via Project Administration web page	39
3.2	Assigning users to roles	40
3.3	Administering roles	41
3.4	Workflow Configuration	45
3.5	Workflow Graphs	47
3.6	Workflow Report	48
3.7	The system status page	49
3.8	Deleting and reactivating users	50
3.9	Managing the department list	51
3.10	Files for localization	54
3.11	Zaval Resource Editor for Localization	55
3.12	Eclipse Plugin Resource Editor	56
4.1	Registration of a new user	60
4.2	Sample help screen and main menu	61
4.3	Artifact overview page	62
4.4	Edit artifact page	63
4.5	Creating a link to a source repository	64
4.6	Modifying the attachment list	65
4.7	Selecting a parent artifact	66
4.8	Copying lists from templates	67
4.9	Sample report page with hierarchie and closed issues	68
4.10	Configuring a report	69
4.11	Customized PDF report	70
4.12	The TQL management page	74
4.13	Bar chart: number of artifacts vs. responsible person	77
4.14	Bar chart: number of artifacts vs. responsible person	78
4.15	Bar chart: number of artifacts vs. responsible person	79
4.16	Pie chart: number of artifacts vs. responsible person	80
4.17	Sample Gantt diagram	81

---

4.18 Registering to monitor changes . . . . .	82
5.1 Sample e-mail notification (localized, in German) . . . . .	91
6.1 States and state transitions for action items . . . . .	94
6.2 States and state transitions for milestones . . . . .	94
6.3 States and state transitions for risks . . . . .	95
6.4 States and state transitions for issues . . . . .	96
6.5 The CVS directory view . . . . .	97
6.6 The CVS file overview . . . . .	98
6.7 The CVS difference between versions . . . . .	98
6.8 The CVS file view . . . . .	99
C.1 The CVS directory view . . . . .	109

# List of Tables

1.1	Key features	3
2.1	System Requirements for the <i>Track+</i> system	8
2.2	Supported Operating Systems	9
2.3	Some code page settings for the jTDS JDBC driver	18
3.1	Example Role Table with Access Keys	44
3.2	Properties for customization of report.jsp	53
3.3	Files for localization	55
4.1	Available comparison operators in TQL	71
4.2	Time units for operators OLDER_THAN and NEWER_THAN	71
4.3	Date formats supported by the DATE-function	72
4.4	All fields supported by TQL	73
5.1	Default roles in <i>Track+</i>	86
5.2	Default states in <i>Track+</i>	87
5.3	Default attributes in <i>Track+</i>	89
5.4	Default list types in <i>Track+</i>	90
5.5	Email notifications	90
A.2	Limits of the <i>Track+</i> system	101



# 1

## Introduction

### 1.1 Target Audience

**Track+** is a method to dramatically increase **efficiency** and **control** in the execution phase of projects. This book, and the **Track+** application is conceived for

- ▶ department managers
- ▶ team leaders
- ▶ project managers
- ▶ people involved in the realization of medium to large scale projects

that want to significantly improve their project management effectiveness. With **Track+** managers will

- ▶ cut project costs by **10 to 50%**
- ▶ cut project delays by a factor of two
- ▶ increase product quality
- ▶ increase customer satisfaction
- ▶ increase employee satisfaction

Even though there is a large user base that uses the **Track+** method mostly for software projects, **Track+** was designed especially for

- ▶ embedded system projects involving hardware, software, and documentation development
- ▶ large software projects

- ▶ installation and commissioning projects of technical equipment
- ▶ multi-cultural projects encompassing world-wide development sites

The tasks of project management in a project typically are attributed to different phases, such as shown in Fig. 1.1. The **Track+** method focuses on the **project execution phase**.

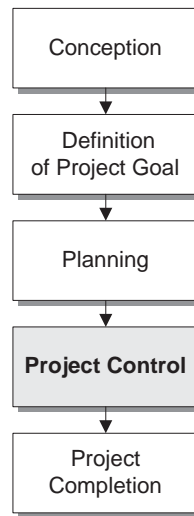


Figure 1.1: Major project phases

In a typical project this is where control is lost and delays occur. This is where standard project planning tools like Microsoft Project can not be used efficiently any more. **Track+** makes the actual state of a project visible in different scopes any time to everybody with the appropriate scope privileges.

For most companies it is a reality that people oftentimes work on different projects in parallel. With its elaborate access control system, **Track+** provides a **matrix view** on all tasks, adapted to the matrix organisation found in many companies. This makes working with the **Track+** method a pleasure for each project employee.

## 1.2 Example Applications

**Track+** is being used in several thousand organisations worldwide. It is therefore difficult to provide a comprehensive overview over the usefulness of the **Track+** method. However, here is an attempt:

**Bug Tracking:** In the meantime, most software producing units use a bug tracker, such as bugzilla or gnats. Those that can afford it might have installed multi-featured dinosaur tools just to notice that the cost of convincing their employees to actually use the tools and the cost of schooling and administration is prohibitive. **Track+** is used very efficiently as a commercial grade bug tracker.

**Quality Gates:** Many companies have a quality assurance system installed with defined quality gates a project must pass in order to proceed. At these quality gates,



checklists are being filled out to not forget any important action. **Track+** can be used to set up these quality gate check lists at the start of a project. The project state can then be seen by looking at the checklist any time, not just at the time the quality gate is supposed to be passed. Since each artifact in **Track+** can be assigned a due date, project delays become visible soon.

**Risk Lists:** Not widely used, risk lists are very important to be able to control a project. Some people coined the phrase that project management is risk management. It is very easy to set up a risk list in **Track+** and watch it change as the project proceeds.

**Contractor Management:** Many projects employ contractors to deliver parts of the work. However, it is not desirable to have the contractors see all issues, bugs, and risks that may be present in the project. With its access control and support of secured extranet access, **Track+** allows contractors to be connected to the project database while being able to access only those issues that concern them.

## 1.3 Features

The **Track+** system offers a comprehensive list of features to support many of the tasks that are encountered during the execution phase of a project. The following table lists the key features.

- 
- ▶ Completely web based, only browser needed
  - ▶ Email notifications and reminders for due tasks
  - ▶ Configurable workflows
  - ▶ Powerful role based access control
  - ▶ Configurable template based reports
  - ▶ Powerful report query language
  - ▶ Arbitrary amount and types of attachments
  - ▶ Pleasant and easy to use
  - ▶ Integrated online help
  - ▶ Supports great variety of operating systems and database systems
- 

Table 1.1: Key features

## 1.4 Key Concepts

### 1.4.1 RACI Method

The RACI matrix is a powerful but simple tool for assigning **R**esponsibility, **A**ccountability, **C**onsulted and **I**nformed roles within an organization, across many functions. Each task is assigned one accountable individual (the buck stops here), one or more responsible individuals (those who do the work) and others who will be consulted or informed

concerning the action. The assignments are displayed in a matrix format on a chart. A facilitated process for creating the RACI chart is very powerful for achieving a common understanding of who does what.

Using the RACI method provides major benefits to organizations:

- ▶ Productivity increases due to well defined accountabilities
- ▶ Rework and scrap is reduced because need specifications are clarified
- ▶ Capacity is increased since overlaps and redundancies are eliminated
- ▶ Project organization structures are streamlined because unneeded layers are collapsed and accountability is placed where it belongs
- ▶ Planning processes are improved because more team members participate as a result of new communication interfaces (consult and inform)

**Track+** provides powerful support for RACI matrices in project organizations. **Track+** softly guides project managers and team members to live with a clear RACI structure, so that at any time everybody concerned and authorized knows who is accountable, responsible, who is being consulted and informed. With **Track+**, this structure is automatically built into a project, without requiring human intervention.

### 1.4.2 Artifacts

Another key concept of the **Track+** method is that of an **artifact**. Example for artifacts are

- ▶ tasks
- ▶ work packages
- ▶ action items
- ▶ mile stones
- ▶ problem reports
- ▶ software bugs
- ▶ feature requests
- ▶ risks

Each artifact has a lifespan between its conception and its death. **Track+** permits you to define your own list of artifact types. You can even define which user role is able to see which artifact types.

A project is made up of many such artifacts of different types and levels of detail, that start at some time throughout a project and that end at some other time. In this area, the art of project management is the art of keeping track of all these artifacts.

One of the key points is that somebody needs to take **responsibility** for an artifact. For any non-trivial sized project, it is impossible that a project manager takes on this responsibility all by himself. Part of this is to make sure that the state of an artifact is accurately

known at any time to all people that have a reasonable interest in it. A **central repository** that is maintained by all people involved in a project and not just by some project managers greatly supports this kind of model. This technique has been used with good success for several years in software defect tracking and change management[6], but can and should be employed as well in tracking any artifact that is part of a project.

For software projects, the *Track+* system delivers a great deal of functionality required to reach CMM and SECM process maturity levels 2 and 3 [7]. For other projects, this system supports adherence to ISO 900x standards.

### 1.4.3 Simplicity and Acceptability

Some tools impose a rather over-engineered process on a project, trying to eliminate as many possibilities for making errors as feasible. However, most of the time it is much more effective to live with a sporadic error than to have fool proof tools. If tools get in the way of people they won't use it.

It is a common pitfall to assume that technology will solve a problem without the involvement or consent of the people that need to work with it. Changing people's work habits to adopt a new technology requires great effort, and the more helpful a tool is perceived the faster this adoption process takes place. Perception of a tool has something to do with its complexity. The more complex a tool is the more time it will take to convince people to use it, particularly those people that need the tools' service only from time to time.

To support the method of managing projects as a bunch of artifacts the *Track+* system was designed. A tool of this kind is as essential to project management as is a word processor to a secretary. The *Track+* system has been developed with the help of many project managers and team members, testing it on real projects in several organizations. The wide acceptance of this system comes not at last from its simplicity and focus. In particular,

- ▶ it is simple to use and simple to administer
- ▶ it is part of the e-mail system
- ▶ it can be made part of the regular information work-flow
- ▶ it reduces workload of a project manager since people can manage their artifacts independently
- ▶ it allows for accurate tracking of project and change states
- ▶ it can be customized to any reasonable process
- ▶ it runs on many hardware and software platforms

The *Track+* system further simplifies the project management process by allowing users to access the artifacts database through a user friendly web interface. The *Track+* system offers the protection of the artifact data by requiring all users to obtain a user name and password.

To reduce the user's potential learning curve, the *Track+* system was designed to have a slim, intuitive interface. Regular users can start working with the system with basically no training. Further features of the *Track+* system include

- ▶ Powerful reporting - the home page dynamically displays all of the active artifacts assigned to the user logged into the system. The user can change the active artifact display according to various criteria. Users can further sort the list of artifacts by almost any available attribute.
- ▶ Document attachment - users can bind and attach a wide array of documents to their artifacts.
- ▶ Closed artifacts - users can assess archived artifacts to view their history. A keyword search feature is available.
- ▶ Web-based administration pages - a project appointed administrator can add user accounts, set their permissions and access levels, and customize many drop down options. This mostly eliminates the need for a system administrator level employee to run the system.
- ▶ User entry control - all **Track+** system pages are user name and password restricted. User accounts can be created by anybody from an authorized e-mail domain. Verification takes place via password sent by e-mail. A project administrator must add access rights to users before they can access any artifacts.
- ▶ Change password - users can change their own password through a Web interface.

The **Track+** system doesn't require any client installation and can be used via a standard HTML browser. Users can register themselves if they have a valid e-mail account. Thereby user administration is kept to a minimum.

The database used for persistent storage requires very little administrative work beyond setting it up once, thereby reducing maintenance costs significantly. Since many organizations have their preferred line of RDBMS, **Track+** supports a wide range of RDBMS such as MySQL, Oracle, Firebird, Interbase, MS SQLServer, and PostgreSQL. With minor modifications, any RDBMS providing a JDBC adapter could be used.

# 2

## *Installation*

The **Track+** application can be installed in two ways:

- ▶ On Microsoft Windows operating systems with an installer
- ▶ In almost any kind of operating system environment using the procedures laid out below.

The installer installs the application in about five minutes with basically no user interaction. It comes with a servlet container (Tomcat), a complete SQL database system suitable even for large installations of 200 or more users (Firebird), the workflow graph visualization software (GraphViz), and the **Track+** application itself.

A complete **Track+** installation consists of five parts:

1. An application server like JBoss, JRun or Tomcat
2. A relational database management system (RDBMS )
3. An SMTP mail server
4. The **Track+** application proper
5. A browser like Mozilla, Netscape, Opera, or Internet Explorer

This chapter describes how to get a **Track+** web server up and running. This work has to be performed once for every server. On the client side there is nothing to do or configure. Users just have to point their browser to the right URL.

### *2.1 System Requirements*

Since the **Track+** system is completely written in Java, it can run on any platform with the requirements shown in Table 2.1. Note that we distinguish between supported configurations and configurations which will most likely work, but are not supported. The difference is that the supported configurations are tested before publication of a release.

Object	Requirement
Main memory	> 256 MByte
Disc space	min. 1 GByte available
Java Virtual Machine	Java SDK (JRE does not suffice!) Version 1.4.x
Database System	MySQL 4.0, 4.1 Firebird 1.0, 1.5, Interbase 6.0 Oracle9i, Oracle 10g MS SQL Server 7 and 2000 HSQLDB 1.7.2 PostgreSQL, SAP DB*, DB2*
Application Server	BEA WebLogic 8.1 JBoss 3.2.x Jetty 4.x, 5.0 JRun 4 Sun ONE Web Server 6.1 Tomcat 4.1 or higher Websphere Application Server 5.1.1
SMTP E-Mail Server	
Browser	Internet Explorer 5.x, 6.x Netscape 7.1 Opera 7.22 Firefox 1.x Mozilla 1.x Safari

Table 2.1: System Requirements for the *Track+* system

The database systems marked with an asterisk have been reported to work with **Track+**, but are not directly supported by the development team. However, database generation scripts are provided for all database systems mentioned in the table.

Operating System	Remarks
Windows NT with SP 6	support will cease
Windows 2000	support will cease
Windows XP	
Linux 2.4	
HP Unix 11	
MacOS X 10.3	

Table 2.2: Supported Operating Systems

The following sections describe procedures on setting up a suitable environment for the **Track+** system.

Due to the three tier architecture of the system the database doesn't have to run on the same server as the application server. However, in the following description it is assumed that application server and RDBMS do share the same machine.

## 2.2 Setup with Installer

The fastest way to set up a completely functional **Track+** system suitable even for large user groups on a computer running the Microsoft Windows operating system is to use the provided installer.

If on first startup you do not get the login screen, there may be another webserver already running on port 80 of your system. You should either shut it down, or deploy **Track+** with it. Alternatively you could change the default port in file `server.xml` of the Tomcat server from 80 to some other number. In this case you have to enter the port number as part of the URL where you access **Track+**, like `http://localhost:7001/track` instead of `http://localhost/track`.

## 2.3 Non-Installer Setup Overview

To install a fully functional **Track+** server follow these steps:

1. obtain the **Track+** package
2. install and configure the relational database management system (RDBMS)
3. install the Java SDK
4. install the application server
5. configure the **Track+** application



Figure 2.1: Track+ Installer (here in German)

It is assumed that there is an SMTP mail server already up and running. All necessary steps are described in more detail in the following sections with a focus on installation for a Linux and Windows system. Installation on other Unix derivatives and with other database management systems should be similar.

## 2.4 Step 1: Obtain the Track+ Package

Download the required files of the **Track+** package from [SourceForge](#) and save it under e.g. `c:\tracktmp` or any other place. At the end of the installation procedure you may remove these files and directories. You need at least these three files (where **v** stands for the version):

- ▶ **track-v-db.zip** (the database setup files)
- ▶ **track-v.war** (Everything else)
- ▶ **track-v.pdf** (You got it if you are reading this)

The source code package **track-v-src.zip** you only need if you want to recompile the application. The client package **track-client-v.jar** you only need if you want to use the client functionality. Please be aware that some browsers add a **.zip** extension to the **.war**-file when downloading and saving it since they recognize it as a compressed file. You may have to manually rename it back to **track-v.war** if your browser has mangled the file name.



## 2.5 Step 2: Install a Database Management System

Before you proceed to the instructions for your specific RDBMS please unzip the **Track+** database package **track-db-v.zip**. The directory where you save the extracted files we call from now on **UNZIP\_DBASE\_DIR**. It has several database specific subdirectories like "MySQL", "Oracle", "Firebird", "MSSQLServer", "Postgres", "hypersonic", "sapdb", and so on.

### 2.5.1 MySQL Database

MySQL is one of the primary supported database systems of **Track+**. Most Linux systems come with a mysql server instance already installed. Please be aware of the license of MySQL. In general, MySQL is only free for non-commercial usage, similar to **Track+**. If you want to deploy **Track+** commercially with MySQL, you need to obtain a MySQL license.

If you need to get the software, you can obtain it from <http://www.mysql.com/downloads/index.html>. Get the latest stable release and install it on your computer.

In the following the procedure is outlined to get the required **Track+** database up and running. In a production environment, it is recommended to set up MySQL with transaction support and take care with regard to access privileges. Particularly, there should be no general GRANT to user "friedj" for all hosts(friedj@"%"), but only to the one the application is running on.

1. Change directory to the place where you unzipped the **Track+** database package **track-db-v.zip** to. This directory we call **UNZIP\_DBASE\_DIR**. It has a subdirectory called "MySQL". Change to this directory, e.g. on Windows

```
cd c:\tracktmp\dbase\MySQL
```

2. Add a new user "friedj" with some password <trackplus\_admin\_password> with the MySQL administration tool **c:\mysql\bin\winmysqladmin.exe** supplied with the MySQL distribution, or some other tool of the many that you may have found useful.
3. Call the mysql command line tool. There should be no space between the p and the password. You are now inside the mysql command line tool, showing the prompt **mysql>**. Don't type this prompt in the following commands!

```
mysql --user=root -p<YourMySQLRootPassword>
```

```
mysql> GRANT ALL PRIVILEGES ON track.* to friedj@"%"
      IDENTIFIED by '<trackplus_admin_password>'
      WITH GRANT OPTION;
```

```
mysql> GRANT ALL PRIVILEGES ON track.* TO friedj@localhost
      IDENTIFIED by '<trackplus_admin_password>'
      WITH GRANT OPTION;
```

```
mysql> FLUSH PRIVILEGES;
mysql> quit;

mysql -ufriedj -p<trackplus_admin_password>

mysql> create database track;
mysql> use track;
mysql> source <UNZIP_DBASE_DIR>/mysql/id-table-schema.sql;
mysql> source <UNZIP_DBASE_DIR>/mysql/track-schema.sql;
mysql> source <UNZIP_DBASE_DIR>/populate.sql;

mysql> quit;
```

The database should now be up and running.

The MySQL JDBC driver closes idle connections after 8 hours. To have the driver automatically reconnected you need to set the option **autoReconnect=true**.

### 2.5.2 Firebird Database

- ▶ The software can be obtained via [http://sourceforge.net/project/showfiles.php?group\\_id=9028](http://sourceforge.net/project/showfiles.php?group_id=9028)  
Select the required server binary depending on your operating system. The instructions for Unix below assume you are using the Linux rpm. For other Unix operating systems, the procedure should be similar. Wherever possible, select the Super Server architecture.
- ▶ Download the InterClient JDBC Driver for Firebird RDBMS from the Firebird site. Don't use any older versions, they have problems.
- ▶ For easy setup and administration of the Firebird Database, and if a Windows PC is available, it is very helpful to have the IBConsole utility at hand which can also be found under :  
<http://www.borland.com/devsupport/interbase/opensource/>

#### Firebird Database on Linux

The installation of the Firebird database software depends on the version you are using. This section mostly describes installation of version 1.0x together with the Interclient JDBC driver. If you are using a newer version such as 1.5, you don't need the Interserver/Interclient components. The required JDBC driver is already included with **Track+**, and installation is much easier. It is therefore recommended to use the actual version of Firebird without the Interserver/Interclient JDBC driver.

Login as user root with superuser rights. Create a group "firebird" and a user "firebird". Be sure to include a password for system security. The home directory should be **/home/firebird**. Make the user a member of group firebird.

1. From the directory where the Firebird rpm file is located run

```
rpm -i FirebirdSS-1.0.0.796-0.i386.rpm
```

or however your Firebird rpm package is called. This will install the Firebird database server at **/opt/interbase** (sic!).

2. Change the owner of /opt/interbase to user "firebird":

```
chown -R firebird /opt/interbase
```

Copy the Interclient tar file from where you downloaded it to location **/opt/interclient**, e.g.

3. 

```
mkdir /opt/interclient
```

```
cp IC2001Linux.tgz /opt/interclient
```

4. Unzip and unpack the InterClient tar archive (e.g. IC2001Linux.tgz):

```
cd /opt/interclient
tar -zxvf IC2001Linux.tgz
```

5. Change to the installation directory and run the install script:

```
cd /opt/interclient/interclient_install_temp_dir
./install.sh
```

Closely follow the instructions. For xinetd.d based systems such as RedHat you may have to manually create an entry "interserver" in that directory. The Firebird Interclient tar file contains such a file.

6. Depending on your version of Unix or Linux, make sure the interserver process is registered and started with the system. For S.u.S.E distributions or other inetd based systems check if the install script appended the following line to file **/etc/inetd.conf**:

```
interserver stream tcp nowait.100 root
    /usr/interclient/bin/interserver interserver
```

and the following line was added to file **/etc/services**:

```
interserver 3060/tcp #interserver
```

You now have to activate the interserver process by

```
kill -HUP <PID of inetd process>
```

You can do a netstat to check of the installation was successful, e.g.

```
netstat -an | grep 3060
```

and should get a list including the **ESTABLISHED** keyword.

7. Change the owner of **/opt/interclient** to user "firebird":

```
chown -R firebird /opt/interclient
```

8. Add the path to the Firebird binaries to user firebird's path. If you are using S.u.S.E Linux, you add this to **/home/firebird/.bashrc**. You could add it to **/home/firebird/.profile** if you are using some other version of Linux or some other default shell.

Add the following lines to the file:

```
PATH=$PATH:/opt/interbase/bin
export PATH
```

9. Now create the following file **gds.hosts.equiv** in **/etc**:

```
/***** file contents *****/
# This file will allow InterBase to connect to the
# localhost
localhost
+
/***** end of contents *****/
```

10. Check to make sure the port for Firebird has been added to **/etc/services**:

```
grep 3050 /etc/services
```

11. You should get a line like the following:

```
gds_db 3050/tcp # InterBase Database Remote Protocol
```

If you don't, add it to **/etc/services**.

12. Modify the file **/etc/init.d/firebird** and change the line containing **FBRunUser**:

```
FBRUNUser=firebird
```

and the line containing : `${ISC_PASSWORD...}` to

```
:${ISC_PASSWORD:=<new_SYSDBA_password>}
```

13. Make sure you set owner, group, and permissions for the database control file **/etc/init.d/firebird**:

```
chown firebird root
chgrp firebird root
chmod 700 firebird
```

Now only root can read, write, and execute firebird. To manually start and stop Firebird as user root:

```
/etc/init.d/firebird start
/etc/init.d/firebird stop
```

14. For database administration, you may want to use the convenient tool IBConsole. Installation on a Windows system is straightforward. Installation will create an entry in the Services file usually found under **c:\WINNT\system32\drivers\etc** of the form:

```
gds_db 3050/tcp
```

Make sure that this entry is not removed, for instance by an automatic startup script in a company network.

15. Check if the database server is listening:

```
netstat -an | grep 3050
```

should result in a line including **LISTEN** in it.

16. Once the Firebird RDBMS is running, register the server with the IBConsole tool on your Windows client. Alternatively, use the security management tool **/opt/interbase/bin/gsec**. With these tools you will be able to perform user administration for that server (on IBConsole: Tap "Server", "User Security"). Go and change the default SYSDBA password from MASTERKEY to the one set in step 12. Don't forget this password!

```
/opt/interbase/bin/gsec -modify sysdba -user sysdba  
-password MASTERKEY -pw <new_SYSDBA_password>
```

17. Add a new user with name "friedj" and some password:

```
/opt/interbase/bin/gsec -add friedj -user sysdba  
-password <new_SYSDBA_password>  
-pw <trackplus_admin_password>
```

18. The database is created from scratch using the **isql** command line utility (don't type the **isql** prompts) from where you extracted your **Track+** database package:

```
/opt/interbase/bin/isql -u friedj -p tissi  
  
isql> create database "/opt/interbase/db/track.gdb";  
isql> input "dbase/Firebird/id-table-schema.sql";  
isql> input "dbase/Firebird/track-schema.sql";  
isql> input "dbase/populate.sql";  
isql> quit;
```

Make sure the owner and permissions are set right:

```
chown -R firebird /opt/interbase/db  
chmod 700 /opt/interbase/db
```

19. Check if everything works:

```
/usr/lib/java/bin/java -classpath  
/opt/interclient/interclient.jar  
interbase.interclient.utils.CommDiag
```

To access the database via IBConsole, you have to register it in this tool (Menu Database → Register). The file path that needs to be entered is as seen from the server (e.g. **/opt/interbase/db/track.gdb**). The alias is of no importance.

### *Firebird Database on Windows*

The installation of the RDBMS is straightforward using the installation package. The following procedures assume that the database was installed in directory `c:\Programs\interbase`. It can however be placed anywhere else.

1. After installation, set the SYSDBA password from "masterkey" to something more secure:

```
c:\Programs\interbase\bin\gsec -modify sysdba -user
    sysdba -password masterkey -pw <new_SYSDBA_password>
```

Alternatively, use the IBConsole utility if you did download it from the Borland site.

2. Add a new user with name "friedj" and some password:

```
c:\Programs\interbase\bin\gsec -add friedj -user
    sysdba -password <new_SYSDBA_password>
    -pw <trackplus_admin_password>
```

Don't use a different user name since this most likely would lead to trouble later on.

3. Now you can install the **Track+** database itself. Assuming that you had downloaded the **Track+** database package to **c:\tracktmp**, change to this directory and unpack the zip file. You could use a utility program like WinZip for this instead of the unzip command:

```
cd c:\tracktmp
unzip track-v-db.zip
```

You should now have a directory **c:\tracktmp\dbase**.

4. Change to the **Firebird** directory of **dbase** and create the **Track+** database (don't type the **sql>** prompts):

```
cd c:\tracktmp\dbase\Firebird
mkdir c:\data
c:\Programs\interbase\bin\isql -u friedj -p
    <trackplus_admin_password>
sql> create database "c:\data\track.gdb";
sql> input "id-table-schema.sql";
sql> input "track-schema.sql";
sql> input "..\populate.sql";
sql> quit;
```

Of course, you can use any other name instead of **c:\data\track.gdb** for the database file as long as you enter the same name and path into the **...\WEB-INF\Torque.properties** file in the application server webapps directory (see section 2.8.1).

### 2.5.3 MS SQL Server

**Track+** works well with MS SQL Server Version 7 and 2000. We use the jTDS JDBC driver to connect to the database server. The most important point to consider is the correct setting of the character set and collation, so that non-ANSI characters are properly stored and retrieved. This setting should be explicitly set when creating the database. It is not recommended to use the standard settings of the database server.

Region	Collation	jTDS Charset
Chinese traditional (Taiwan)	Chinese_Taiwan_Stroke	CP950
Chinese simplified (PRC)	Chinese_PRC	CP936
Danish	Danish_Norwegian	CP1252
English	Latin1_General	CP1252
Farsi	Arabic	CP1256
Finnish	Finnish_Swedish	CP1252
French	French	CP1252
German	Latin1_General	CP1252
Hebrew	Hebrew	CP1255
Italian	Italian	CP1252
Japanese	Japanese	CP932
Netherlands	Latin1_General	CP1252
Norwegian	Danish_Norwegian	CP1252
Russian	Cyrillic_General	CP1251
Spanish	Modern_Spanish	CP1252
Swedish	Finnish_Swedish	CP1252
	UTF-8	UTF8
	UTF-16	UNICODE

Table 2.3: Some code page settings for the jTDS JDBC driver

Create the database with the Enterprise Manager tool. Make sure you are using the right collation set when creating the database (see Table 2.3). It might be best to add a new user which will be used as owner of the database.

Use the SQL Query Analyzer to run these scripts located in the **dbase** directory of the database package for completely new databases:

1. MSSQL\id-table-schema.sql
2. MSSQL\track-schema.sql
3. populate.sql

The database should now be up and running.

The MS SQL Server JDBC driver needs to be enabled in the **Torque.properties** file located in the **webapps** directory of your servlet container or application server as described below. Don't forget to choose the right character set.



### 2.5.4 Oracle Database

**Track+** runs with Oracle8i, Oracle9i and Oracle 10g databases. This document will not cover how to setup a database in Oracle. Once your database administrator has set up a database and granted you the privileges to create and modify objects in your schema you can use the scripts from the **Track+** database package that you have already extracted to **UNZIP\_DBASE\_DIR** to create the **Track+** tables and indices in your database. We will assume that the database is named "track" and that it is your default database. Your database user name we assume to be "friedj", password "tissi". If you are later on running the **Track+** application as a different database user you would have to provide synonyms for each table, which is something you should try to avoid.

```
sqlplus ``friedj/tissi``
```

```
SQL> @<UNZIP_DBASE_DIR>/Oracle/id-table-schema.sql;
SQL> @<UNZIP_DBASE_DIR>/Oracle/track-schema.sql;
SQL> @<UNZIP_DBASE_DIR>/populate.sql;
SQL> quit;
```

If you use **Track+** with Oracle you need to download the appropriate JDBC driver. Unfortunately, Oracle has a difficult history with their JDBC driver database datatype to Java datatype mapping. As far as **Track+** is concerned, this pertains particularly to the date, time, and timestamp datatype. Here are some hints:

If you are using Oracle 8i and Oracle 9.0 you may be best off with the latest version of the JDBC driver in **classes12.zip**. You have to drop **classes12.zip** into the **Track+** lib directory. Rename this file to **classes12.jar** to have it included into the class path. This is required on some application servers that do not search zip archives for classes (e.g. Tomcat).

If you are using Oracle 9.2 or Oracle 10g and the JDK 1.4, use the JDBC driver provided in **ojdbc14.jar**. You have to drop this file into the **Track+** lib directory.

Some application servers add their own class libraries to the path; with Oracle OC4J there could be a problem that it might give you a different JDK and JDBC driver than what you think. Make sure you have the right JDK and *only* the Oracle driver in the classpath mentioned here.

For more information on possible Oracle related problems you may want to have a look at the Oracle technology support site, particularly at [http://www.oracle.com/technology/tech/java/sqlj-jdbc/htdocs/jdbc\\_faq.htm#08.01](http://www.oracle.com/technology/tech/java/sqlj-jdbc/htdocs/jdbc_faq.htm#08.01) or at the **Track+** forum, searching for "Oracle". If you are upgrading from a previous release make sure that all date related fields for which you like to see the time as well are defined as **TIMESTAMP** in the database.

### 2.5.5 HSQL Database

HSQL is a 100% pure Java relational database engine. It is a continuation of Thomas Müller's closed down Hypersonic SQL Project. HSQL supports a rich subset of ANSI-92 SQL. It is free and open source. You can obtain it from <http://hsqldb.sourceforge.net>.

According to the documentation, the database itself is currently not multithreaded, but it is multithreading-safe. Applications that are multithreading like *Track+* can use the database, but all requests are executed one after the other, one at a time. This means that HSQL is not well suited for hi-concurrency environments with a lot of users accessing the database in parallel.

1. Download the latest **hsqldb\_x.y.zip** file from the HSQL download page to your local computer.
2. Uncompress the files. Any program that can handle zip files should work. The folder where you uncompress it we will call **HSQL\_HOME**, e.g. **C:\hsqldb**.
3. Create or choose a folder where the database files will be stored. Normally this is **HSQL\_HOME\data**. This folder we will call **DB\_HOME**. Copy the following files from you *Track+* database package to **DB\_HOME**:  
`<UNZIP_DBASE_DIR>\hypersonic\id-table-schema.sql,`  
`<UNZIP_DBASE_DIR>\hypersonic\track-schema.sql,`  
`<UNZIP_DBASE_DIR>\populate.sql` and  
`<UNZIP_DBASE_DIR>\hypersonic\sqltool.rc.`
4. You now have to install a suitable Java software developers kit. If you have not done so, look at section 2.6 for instructions. The java command has to be in your path.
5. Run HSQL with:

```
java -cp <HSQL_HOME>\lib\hsqldb.jar org.hsqldb.Server
        -database <DB_HOME>\track
```

If everything went okay you something like this should appear:

```
[Server@1d5550d]: [Thread[main,5,main]]: checkRunning(false)
                                                entered
[Server@1d5550d]: [Thread[main,5,main]]: checkRunning(false)
                                                exited
[Server@1d5550d]: Startup sequence initiated from main()
                                                method
[Server@1d5550d]: Loaded properties from
                        [D:\hsqldb\data\server.properties]
[Server@1d5550d]: Initiating startup sequence...
[Server@1d5550d]: Server socket opened successfully in 32 ms.
[Server@1d5550d]: Database [index=0, id=0, db=file:track,
                        alias= opened sucessfully in 375 ms.
[Server@1d5550d]: Startup sequence completed in 422 ms.
[Server@1d5550d]: 2004-10-09 17:16:36.718 HSQLDB server
                        1.7.2 is online
[Server@1d5550d]: To close normally, connect and execute
                        SHUTDOWN SQL
[Server@1d5550d]: From command line, use [Ctrl]+[C] to abort
                        abruptly
```

6. During startup HSQL will check if the database "track" already exists. If not it will create it.
7. After having created the database, the table structures for **Track+** need to be created. For this, open another console window, change to the **DB\_HOME** directory and run:

```
java -cp C:\hsqldb\lib\hsqldb.jar org.hsqldb.util.SqlTool
      --autoCommit --rcfile sqltool.rc track
      id-table-schema.sql
java -cp C:\hsqldb\lib\hsqldb.jar org.hsqldb.util.SqlTool
      --autoCommit --rcfile sqltool.rc track
      track-schema.sql
java -cp C:\hsqldb\lib\hsqldb.jar org.hsqldb.util.SqlTool
      --autoCommit --rcfile sqltool.rc track
      populate.sql
```

8. The HSQL database should be ready now. Just for information, the default database administrator user for HSQL is "sa" with empty password. This should be set for a production environment. For this you can use the DatabaseManagerSwing utility that comes with HSQL. Don't forget to also change the entry in **Torque.properties**. Using this utility you can also check if you can access the database and see all tables. If so, you should be ready to run **Track+** with this database as well.

## 2.5.6 IBM DB2 Database

This has not been tested, and is not yet supported. However, only minor difficulties are expected.

## 2.5.7 PostgreSQL

There are several users reporting that they have **Track+** running successfully with PostgreSQL. However, this RDBMS is not in the rollout test plans and therefore not officially supported. Here are some hints to get a **Track+** database on a PostgreSQL server. The following assumes that you are currently on the **Postgres** directory of the database package, and that the PostgreSQL utilities are in your path.

```
$createuser --username=postgres --password -d -P -A
                                --sysid=501 friedj
>Enter password for new user: <trackplus_admin_password>
>Enter it again: <trackplus_admin_password>
>Password: <postgres user password>
>CREATE USER

$createdb --owner=friedj --username=friedj --password
                                track "Track+ Database"
>Password: <trackplus_admin_password>
>CREATE DATABASE
```

```
>Password: <trackplus_admin_password>
>COMMENT

$ psql.exe -f id-table-schema.sql track2 friedj
>Password: <trackplus_admin_password>
...

$ psql.exe -f track-schema.sql track2 friedj
>Password: <trackplus_admin_password>
...
```

You have to insert a **BEGIN** in the first line of the **populate.sql** script before you can run it in order to start a transaction. Then execute

```
$ psql.exe -f ../populate.sql track2 friedj
>Password: <trackplus_admin_password>
...
```

### 2.5.8 SAP DB Database

There are several users reporting that they have **Track+** running successfully with SAP SB. However, this RDBMS is not in the rollout test plans and therefore not officially supported.

## 2.6 Step 3: Install the Java SDK

Many systems already come with a Java runtime environment (JRE) installed. For running a **Track+** server the JRE is not sufficient. You need to obtain the Java Software Development Kit (**Java SDK**) version 1.4 as the servlet engine needs the Java compiler to compile the servlets. You can get it from <http://java.sun.com>.

The Windows version comes with an installer. You have to remember the root directory where you install the package (**\$JAVA\_HOME**). This you need later on to configure your application server. Many Linux and Unix distributions come with a Java SDK already installed. Here you have to find out where the root directory of the SDK is located to be able to configure your application server later on.

## 2.7 Step 4: Install the Application Server

To run **Track+**, you have to have an application server or at least a servlet container according to servlet specification 2.2 and Java Server Pages specification 1.1 running.

**Track+** offers the option to render charts and diagrams to get a quick project overview. The charting code uses the JFreeChart library. JFreeChart requires a running windowing system on the server, e.g. the X11 windowing system on Unix systems, to work correctly. If on your server a windowing system is not enabled, you would for example get the following error:

`java.lang.InternalError: Can't connect to X11 window server using ':0.0' as the value of the DISPLAY variable.`

There are several workarounds known:

1. Use JRE 1.4 and set the “headless” property to “true”
2. Virtual Network Computing (VNC)
3. X Virtual Frame Buffer (Xvfb)
4. Pure Java AWT (PJA)

On WinNT and Win2K with JRE 1.3.1, logging onto the console instantiates the window system and chart creation succeeds. You must be logged in as the same user that instantiates the servlet container. Installing VNC and running Tomcat as a service allows normal operation even when logged out.

On Mac OS X 10.1.4, the OS creates an anonymous instance of the JRE. This fails if the machine enters “sleep” mode, but it works during normal screen-saver operation. We have not tried VNC in this environment.

Under Solaris 2.7 with JRE 1.2.2, it has been reported that VNC allows correct operation as long as the same user starts both Tomcat and VNC.

Under RedHat linux 7.1, JRE 1.3.1 and tomcat 4.0.3, it has been reported that vnc-server-3.3.3r2-28 allows correct operation as long as the same user starts both tomcat and VNC. To have this work, edit the file `/etc/tomcat4/conf/tomcat4.conf` to set the `DISPLAY` variable to the same virtual X session created in `/etc/sysconfig/vncservers`. You may need to start VNC before starting Tomcat.

Under JRE 1.4, setting the “headless” property to true allows JFreeChart to render in the absence of an X11 instance. This has been confirmed for RedHat Linux 7.1, 7.1j, 7.2. and on Solaris 2.7.20:

```
System.setProperty("java.awt.headless", "true");
```

In Tomcat 4, the **catalina.sh** script can use an environment setting:

```
setenv CATALINA_OPTS -Djava.awt.headless=true
```

After you have installed your servlet container or application server as described below, and after you have configured the database connection settings **Track+** should be ready to use. For your first logon, there is the system administrator “**admin**” with password “**tissi**” already predefined in the database.

After this prologue, we now describe the installation for some of the more popular application servers.

## 2.7.1 Tomcat

Some Linux distributions already include a Tomcat version; however it is recommended to download a production version 4.1.x or up from <http://jakarta.apache.org/tomcat/index.html> and possibly disable the version that comes with the distribution. Some Linux distributions had Tomcat versions that did not integrate with the Apache web server.

Install the Tomcat package under `/opt/jakarta`. Make sure that the shell files under `/opt/jakarta/bin` are executable. You may want to consult the Tomcat installation instructions to help you with the details. As `$JAVA_HOME` set the path where you installed the Java SDK. Execute command (possibly as user "root"):

```
/etc/init.d/tomcat start
```

Check, if you get the Tomcat default start page under `http://localhost:8080`. Thereafter, shut down the server again:

```
/etc/init.d/tomcat stop
```

You should now have a running Tomcat servlet container. If you have not been successful there are usually two main reasons:

1. port 8080 is already used by another HTTP server
2. you have not set `$JAVA_HOME` to the right path

If you want to have a more professional installation with all security issues considered you should integrate the Tomcat servlet container with an Apache HTTP server. How to do this is well described in the documentation that comes with Tomcat. Depending on how you configured your distribution you may have to disable Zope and the JServ module to avoid conflicts on port 8080.

Installation of the Windows version of Tomcat is straightforward and doesn't require any special skills as long as Tomcat is run stand alone. You may have to set the `$JAVA_HOME` environment variable to the location where you installed the Java SDK, either in the Tomcat startup scripts or in the Windows System Control panel. Sometimes port 8080 is already used by some other server on your computer. Select a different port then for Tomcat, such as 80.

In the following it is assumed that Tomcat has been installed under `$TOMCAT_HOME`, e.g. `/opt/jakarta` for Linux based systems, and `c:\jakarta\tomcat` for Windows based systems. To install the **Track+** application:

1. Copy the `track-v.war` file of this distribution into the `$TOMCAT_HOME/webapps` directory of the Tomcat servlet container. The name of the war file determines under which URL you will later see the **Track+** application. If you want to have it under `http://<yourcomputer>:8080/track` than you need to name the war file `track.war`. If you do not rename the file, you will find the application under URL `http://<yourcomputer>:8080/track-v`, where *v* stands for the version number (e.g. 203).
2. The first time an attachment is added a new directory `trackdata` is created under `$TOMCAT_HOME/webapps/track-v/`. This is where the attachments are being kept. If you are updating a previous installation make sure you save the `trackdata` directory before you install the update. After updating the application you may restore the attachment directory. Starting with release 2.2.0 of **Track+** you should configure the directory where attachments are being kept in the site configuration page. This particularly makes future upgrades more easy.
3. Start the Tomcat servlet container to expand it into directory `track-v`

```
/etc/init.d/tomcat start (on Linux)
```

or use the Windows Tomcat startup icon or batch file.

4. Stop the servlet container again:

```
/etc/init.d/tomcat stop (on Linux)
```

or use the Windows Tomcat shutdown icon or batch file.

5. Modify the **web.xml** and **Torque.properties** files under **\$TOMCAT\_HOME/webapps/track-v/WEB-INF/**. More specifically, for the **Torque.properties** file change the database password **dbUserPassword** to the one chosen as **<trackplus\_admin\_password>**, and set the server local path of the database (**jdbcURL**) to the right network address and database driver.

On the site configuration page, modify the e-mail domain pattern to your allowed e-mail domains, and set the SMTP mail server to your local SMTP server name. How to go about this is described in more detail in sections [2.8.4](#) and [2.8.5](#).

6. Start the Tomcat servlet container

```
/etc/init.d/tomcat start (on Linux)
```

or use the Windows Tomcat startup icon or batch file.

## 2.7.2 BEA WebLogic 8.1

BEA WebLogic is a powerful commercial application server. **Track+** runs on BEA WebLogic with no restrictions. You can deploy **Track+** in either exploded or unexploded mode using the **track-v.ear** file.

To install **Track+** on BEA WebLogic you have to first extract the **track.war** file contained in **track-v.ear**:

```
jar -xvf track-v.ear track.war
```

You now have a **track.war** file in the same directory as the **track.ear** file. Extract file **WEB-INF/Torque.properties** from the **track.ear** file:

```
jar -xvf track.war WEB-INF/Torque.properties
```

Change to directory **WEB-INF** and modify the **Torque.properties** file as described further down in section [2.8](#). Basically, this configures your database connection.

Thereafter you jar everything back together:

```
jar -uvf track.war WEB-INF/Torque.properties
```

```
jar -uvf track-v.ear track.war
```

You can now deploy the application using the standard BEA WebLogic deployment procedure.

You have to set the attachment directory on the site configuration page (see section [2.8.7](#)) when using BEA WebLogic. Otherwise attachments may not work properly or you will loose all attachments once the server goes down.

### 2.7.3 JBoss 3.2

JBoss is the most popular Java applications server with full J2EE 1.3 support. You can obtain it from <http://www.jboss.org> (about 40 MByte).

**Track+** does run successfully with JBoss 3.2 since release 2.1.0. Follow the instructions of installing JBoss. In particular, adapt the `$JAVA_HOME` variable in the JBoss startup scripts to the path where you had installed your Java SDK. Make sure that the ports you want to serve on aren't being used by another (web) server on your machine.

To install **Track+** you have to unjar the `track-v.war` file and modify the `Torque.properties` configuration files as described further down in section 2.8. Thereafter you jar it again to a file `track.war` and copy it to the JBoss deployment directory. If you have properly installed your RDBMS (see section 2.5) **Track+** should be ready to use.

Alternatively, you could use the `track-v.ear` file for deployment. Look at the description for the BEA WebLogic server for instructions on how to do this.

You have to set the attachment directory on the site configuration page (see section 2.8.7) when using JBoss. Otherwise you will loose all attachments once JBoss goes down.

### 2.7.4 JRun 4

Macromedias JRun 4 is another popular J2EE application server which is fully certified to be J2EE 1.3 compatible. There is a free 30 day trial version. In addition, there is a perpetual developers edition limited for low-volume non-production deployments. For a production setup you would have to purchase a license. You can download the software from <http://www.macromedia.com/software/jrun> (about 55 MByte). The following instructions should help you to get **Track+** going quickly under JRun 4 control. If you need more details you should consult the JRun 4 documentation. Furthermore, we strongly recommend to install the latest update for JRun.

#### *Creating a new JRun 4 Application Server*

1. Run JRun 4 admin server (the easiest way to do this is to run the JRun Launcher and start it there)
2. Login to the JRun Management Console (<http://localhost:8000>). Instead of "localhost" you can insert the IP address or domain name of the computer where your JRun was installed.
3. On the right pane, click on the "Create New Server" link. In the newly appearing window (Fig. 2.2) fill field "JRun Server Name" with "trackplus" and click button "Create Server".
4. On the next screen (Fig. 2.3) you can specify ports. Normally the JRun 4 Management Console defaults need not to be changed. Remember the "Web Server Port Number" and click the "Finish" button.



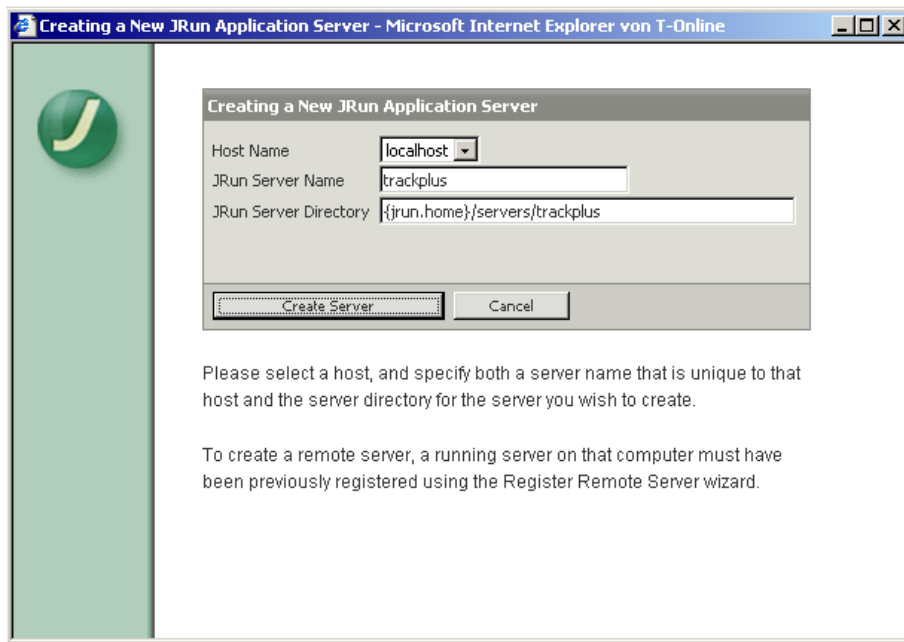


Figure 2.2: Creating a new JRun server

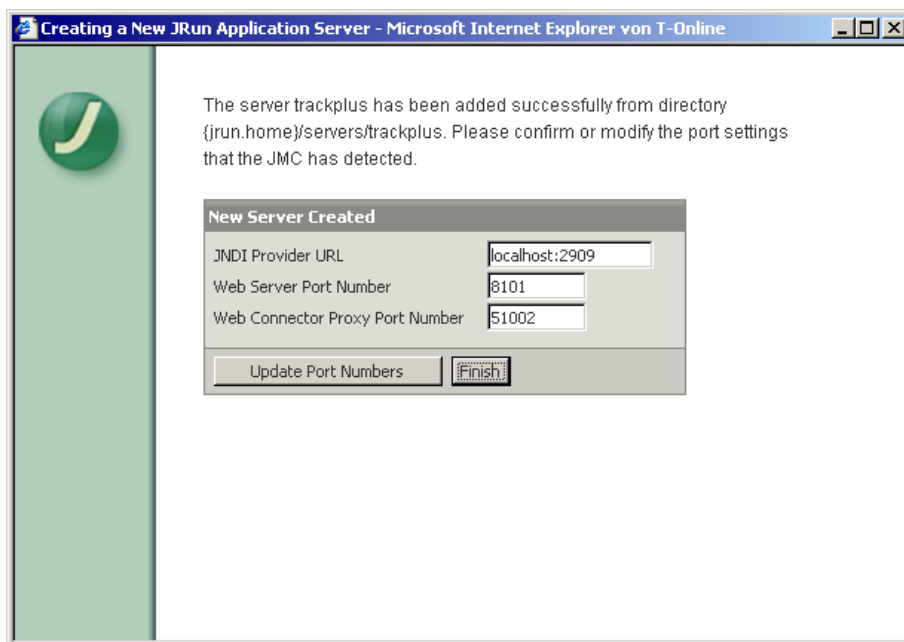


Figure 2.3: Setting the server ports

## Deploying the Track+ Application

1. During creation of a new JRun Application Server the new folder `{jrun.home}/servers/trackplus` was created. This folder should have two subfolders: `default-ear` and `SERVER-INF`. Rename the `default-ear` subfolder to `trackplus-ear`.
2. The folder `{jrun.home}/servers/trackplus/trackplus-ear` is actually an EAR package. It contains three subfolders: `default-ejb`, `default-war` and `META-INF`. Delete folder `default-ejb`, it is not needed by *Track+*. Rename `default-war` to `trackplus-war`.
3. Modify file `{jrun.home}/servers/trackplus/trackplus-ear/META-INF/application.xml`: Change the display name to "Track+", drop the default-ejb module and rename web-uri from "default-war" to "trackplus-war". After these modifications the application element of `application.xml` should look like this:

```
<application>
  <display-name>Track+</display-name>
  <module>
    <web>
      <web-uri>trackplus-war</web-uri>
      <context-root></context-root>
    </web>
  </module>
</application>
```

4. Drop all content of folder `{jrun.home}/servers/trackplus/trackplus-war`, except for the `/WEB-INF/jrun-web.xml` file.
5. Extract the content of previously downloaded file `track-v.war` to `{jrun.home}/servers/trackplus/trackplus-ear/trackplus-war`.
6. Copy `{jrun.home}/servers/trackplus/trackplus-ear/trackplus-war/WEB-INF/lib/log4j-1.2.8.jar` to `{jrun.home}/servers/lib/`. In case folder `{jrun.home}/servers/lib/` doesn't exist you have to create it. This is necessary because JRun 4 uses an older version of log4j inside, and if you don't move this jar, a conflict between the version used by JRun and the version used by *Track+* will occur and *Track+* will not work.
7. Finally we recommend correcting `jrun-web.xml` because by default, it has a configuration not well suited for a production server. But even if without this change *Track+* will work without problems.
8. You may now run your new server using JRun Management Console or JRun Launcher (Fig. 2.4).

Open `http://localhost:8101/` where 8101 is the port number which you have recorded in step 4. If you will get the *Track+* login screen you have successfully deployed *Track+* on your JRun server.

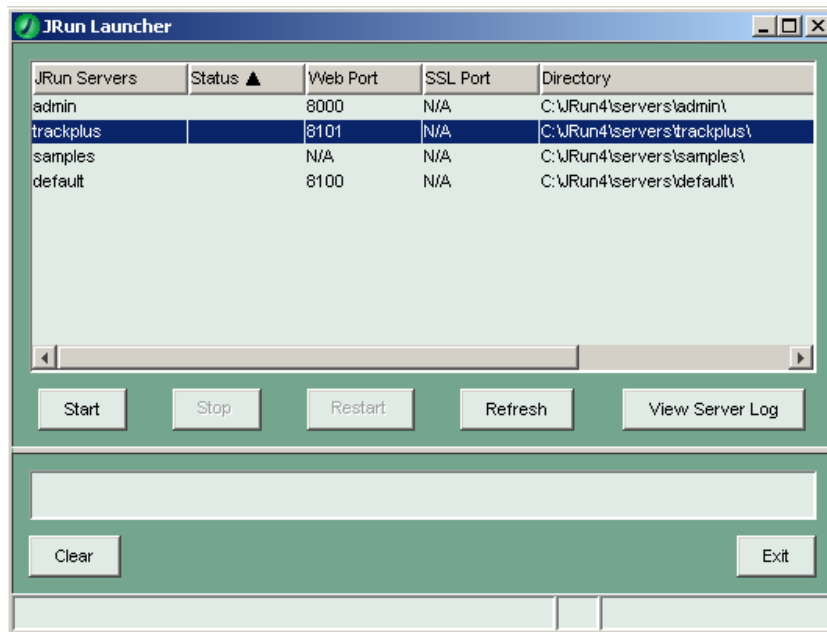


Figure 2.4: Launching the newly created server

9. You have now deployed **Track+**, but to work with it you need to stop the **Track+** server and as a minimum configure the database connection (see 2.8.1), configure e-mail (see 2.8.4) and configure e-mail domains (see 2.8.5). When you are finished with this, start your server again and enjoy **Track+** !

## 2.8 Step 5: Configure **Track+**

The **Track+** application has a certain amount of configuration options, some of which have to be adjusted to a particular installation, and some of which may be used to adapt **Track+** to your requirements. All these configuration options are set via the site configuration page and in file **Torque.properties**, located under directory **\$TOMCAT\_HOME/webapps/track-v/WEB-INF/**.

After changing any of the settings in **Torque.properties** the servlet container Tomcat has to be restarted for the changes to take effect.

### 2.8.1 Configuring the Database

The **Track+** application ships with a default user and password for the database. This default user and password is contained in the file under **\$TOMCAT\_HOME/webapps/track-v/WEB-INF/Torque.properties**. The following is an excerpt and the respective areas are shown.

```
...
# !!! THIS USER WITH THIS PASSWORD HAS TO BE KNOWN BY YOUR
```

```
#          DATABASE SYSTEM !!!
#
torque.dsfactory.track.connection.user = friedj
torque.dsfactory.track.connection.password = tissi
#
#-----
#
# In this section the driver for your database management
# system needs to be enabled. This depends on what kind of
# RDBMS (Interbase, MySQL, etc.) you have and possibly on
# what kind of operating system the database server is running.
# Of course, ONLY ONE DRIVER CAN BE BE ENABLED AT A TIME!!
#
#
# 1) Interbase or Firebird on Windows 32
torque.database.track.adapter=interbase
torque.dsfactory.track.connection.driver =
                                interbase.interclient.Driver
torque.dsfactory.track.connection.url =
                                jdbc:interbase://localhost/D:/data/track.gdb

#
# 1b) Firebird on Window
...
```

If the password is changed in this file it has also to be changed for the database and vice versa. For Firebird and Interbase, the IBConsole utility can be conveniently used for this task. If the user name should be changed as well, it has to be made sure that the new user has complete SELECT, DELETE, INSERT, UPDATE, AND REFERENCE permissions for all tables, functions, and generators of the *Track+* application. The database user acts as a proxy only and doesn't appear anywhere in the user space.

The second item that needs to be configured is the URL of the database and the database driver. Examples are given for various database systems in the `Torque.properties` file. For non-standard character sets have a look at section 2.9.

## 2.8.2 Configuring Logging

In the `Logging.properties` file you can set options for logging purposes. This is usually not necessary and mostly helps if you have problems with the application. If you do not yet have an SMTP server up and running, you can direct e-mail output to get your passwords to the console by uncommenting the line:

```
# log4j.logger.com.aurel.track.util.JavaMailBean = DEBUG
```

In case of problems many other classes may be enabled to generate debug log information by just setting the logging level from **WARN** or **ERROR** to **DEBUG**. For example, in case you are not sure your database is working o.k. you could enable **DEBUG** level logging for the **Torque** package.

### 2.8.3 Configuring Timeouts

In the `web.xml` file a cookie timeout can be set, to allow a user to directly access an issue without having to log on. Usually you would not change this.

### 2.8.4 Configuring E-Mail

**Track+** will send notification messages to involved users for example if certain state changes occur, new issues are created, or responsible persons are being modified. For this feature to work, **Track+** has to be able to connect to an SMTP server. You have to configure your SMTP server name on the site configuration page as shown in fig. 2.5.

The screenshot shows the 'Track+ Server Configuration' page in a Mozilla Firefox browser window. The address bar shows the URL: `http://gandalf/track/editAdminSiteConfig.do?action=load`. The page has a navigation bar with links: Home, Reports, New Issue, Administration (selected), Log off, and Help. Below the navigation bar, there are links for Access Admin, Project Admin, Copy List, Site Admin (selected), Workflow Definition, and My Profile. The main content area is divided into two sections. The top section displays system information: System version: 3.1.0, Database version: 310, License key: EewZwNgU3y2k2KbwNb1Bg27DIHE5IVnDKIucCBaV1rRfOk7NCbv/4tUI+g5EXStdqahv8GWUKCLVynrFyN, License expires on: 4/17/05, and max. no. Users: 1000. The bottom section is for email configuration, with fields for: Track system email (Tracksystem), Mail encoding (Western (ISO-8859-1)), SMTP server name (mail.server.de), SMTP Port (25), SMTP user name, SMTP password, POP server name (mail.server.de), POP port (110), POP user (youraccount@mail.server.de), POP password (masked with asterisks), and Allowed email pattern (^[a-z0-9\.\\_]+@.+)\$).

Figure 2.5: The site configuration page, first part

There are several other options when configuring your e-mail notification service.

**Track+ system email:** here you should define an email account which is being inserted as "send from" by **Track+** in any email the systems sends out

**Mail encoding:** You need to define an email encoding that is being used by **Track+** when sending out emails. You can select one of the list provided, such as "UTF-8", "koi8-r", or "ISO-8859-1". For Western languages, "ISO-8859-1" usually works fine.

**SMTP server name:** For **Track+** to be able to send notifications you need to give it the name of your SMTP server. Some company internal installations do not require authentication before using the server, some do. If authentication is being used, it can be either SMTP authentication or POP before SMTP. In the first case you also have to set the SMTP user name and SMTP password described below. In the second case you have to configure the POP parameters.

**SMTP Port:** Default is 25. Usually, you don't have to change this.

**SMTP user name:** In case you SMTP server requires SMTP authentication, supply the user name here.

**SMTP password:** In case you SMTP server requires SMTP authentication, supply the password here.

**POP server name:** In case of POP before SMTP authentication, supply your POP server here.

**POP port:** Default is 110. You usually do not have to change this

**POP user:** In case of POP before SMTP authentication, supply your POP user account here.

**POP password:** In case of POP before SMTP authentication, supply your POP password here.

**Allowed email pattern:** **Track+** can be configured such that users may register themselves. However, you can restrict registration to specific domains and even user email accounts, for example such that only your customers are allowed to register with the system. The pattern that is allowed to register is a Perl pattern. For example patterns have a look at section [2.8.5](#).

### 2.8.5 Authorizing E-Mail Domains

To prevent users from unverified domains (e.g. hotmail.com, gmx.com) to register, a Perl5 regular expression can be defined so that only e-mail domains matching that expression are allowed to register. You most likely will have to change the example default on the site configuration page.

Example pattern:

```
[^@ \t]@bosch\.com
```

This example permits just users from domain "bosch.com" to register. Some more examples:

```
[^@ \t]@t-online\.de|[^@ \t]@computer\.org
```

matches all users from domains t-online.de and computer.org.  
The pattern

```
[ ^@ \t ]@\w+\.\w+
```

sets basically no restrictions on who can register. However, you have to take special care with Perl reserved characters. These you need to escape with a backslash.

## 2.8.6 Configuring LDAP-Authentication

The number of LDAP-directories used within organizations to allow a centralized management of users and user-accounts is increasing. **Track+** can be configured to use an LDAP-server to authenticate users during logon. This simplifies the handling of login names and passwords for **Track+** users.

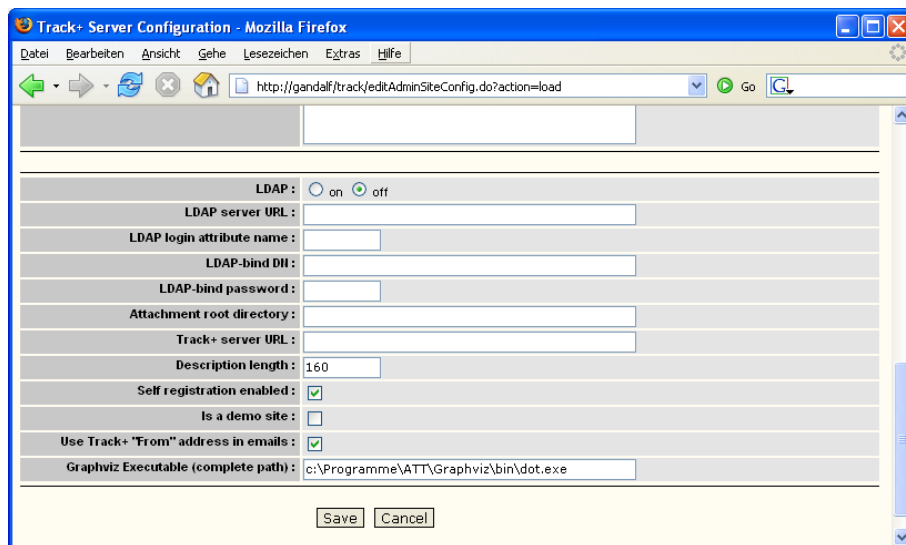


Figure 2.6: The site configuration page, second part

To allow users to use LDAP-authentication you have to activate the LDAP-settings in the site configuration page.

The LDAP server URL is of the form

```
ldap://ldapservers/ou=Main,o=People
```

and the LDAP login attribute of the form

```
uid
```

If searching in LDAP can be done with anonymous bind, keep the **LDAP-bind DN** field empty. Otherwise provide the distinguished name (DN) used for searching.

If searching in LDAP can be done with anonymous bind, keep the **LDAP-bind password** field empty. Otherwise provide the password for the DN you entered above.

LDAP-authentication is disabled by default. When LDAP-authentication is enabled for your **Track+** installation, every user can choose to use it or not (see section 4.1).

### 2.8.7 Configuring the Attachment Root Directory

In releases prior to 2.2.0 all attachments were stored inside the **Track+ webapps** directory. Now you should configure a different root directory where attachments should be stored. If you keep this parameter empty attachments are being stored inside the **webapps** directory. Defining this value will help in future upgrades of the system, since you will not by accident delete all attachments when installing a new version of **Track+**. Furthermore you could have different instances of the **Track+** application use the same directory, as long as they use a common database. For some application servers, e.g. JBoss, you have to set this parameter. Otherwise you would loose all attachments when JBoss goes down.

### 2.8.8 Installing and Configuring Graphviz

In order to have a graphical display of workflows the Graphviz software, a third party product, needs to be installed on the server running **Track+**. Inside of **Track+** you have to enter the complete path to the Graphviz executable, like

```
C:\Program Files\ATT\Graphviz\bin\dot.exe
```

on a Windows machine or something like

```
/opt/att/graphviz/dot
```

on a Unix machine. Temporary files are stored in a subdirectory of the attachment folder, so the same conditions mentioned there apply here as well.

## 2.9 Non-Western Character Sets

**Track+** has been designed to easily support other character sets than the default ISO-8859-1, such as needed for Russian, Chinese, or Hebrew localization. To use appropriate character sets two measures have to be taken:

1. The character set to be used in emails has to be configured
2. The database character set has to be configured.

The configuration of the email character set was described in section 2.8.4 above.

The configuration of the database character encoding very much depends on the database system being used, so no general instructions can be given here. For **Track+** running with MySQL, here are step by step instructions for a Hebrew localization:

1. Find the file `my.ini` (in case of Windows) or `my.cnf` (in case of Linux) for the MySQL installation. In section `[mysqld]` add or modify a line to

```
default-character-set=hebrew}
```

Restart the MySQL server after the change.

2. In file `Torque.properties` change connection URL for MySQL to



```
torque.dsfactory.track.connection.url
= jdbc:mysql://localhost:3306/mytrackplus?useUnicode
=true&characterEncoding=Cp1255
```

You have of course to use the correct database name and server name of your installation.

3. In the site configuration menu change the email character encoding to ISO-8859-8.
4. After these changes restart your application server.

## 2.10 Operation in a Proxied Environment

**Track+** has been designed to operate in a proxied environment. Usually, **Track+** can extract the URLs that are included in notification emails automatically. This automatic detection works fine if there are no proxy servers between the application server and the user clients. If there is a proxy server hiding the application server, the correct server URL has to be configured via parameter **Track+** Server URL on the site configuration page:

```
http://gandalf
```

In non-proxied environments, you can leave this field empty, which will enable the automatic detection feature for the server URL.

## 2.11 Database Migration Notes

Some releases might require a change of an existing database scheme. In such a case the **Track+** database installation package includes a SQL script that should automatically perform the upgrade. To upgrade an existing database following procedure should be followed.

1. Shutdown your currently running **Track+** system release A so that no operations can occur on its database.
2. Make a backup of your database with the tools provided by your database supplier. For instance for Firebird the gsec utility or IBConsole can be used. **This step is very important!**
3. Run the upgrade SQL script on your existing database. The upgrade scripts are usually named like **migrate[r1]to[r2].sql**. For example to upgrade from release 2.1.0 to 3.0.0 on a Firebird database, you can use the isql command line utility or IBConsole (don't type the prompts):

```
/opt/interbase/bin/isql --u friedj --p tissi
                        /opt/interbase/db/track.gdb
isql> input "dbase/Firebird/migrate210To300.sql";
isql> quit;
```

4. Bring your system back up.

Once you have migrated you must not use the previous **Track+** releases any more on this database as this would cause havoc! If you have manually modified entries in the table you may have to check that everything is still okay.

## 2.12 Porting to other Database Systems

The **Track+** system ships with only a few database management systems supported. However, in most cases it is rather easy to adapt to other databases. If your database system is supported by the Torque persistence layer the following should suffice:

1. add a suitable JDBC driver library for your database management system to directory `...WEB-INF/lib`. Use this driver in your `Torque.properties` file.
2. adapt one of the database generation scripts that ship with **Track+** to your database, or run a Torque build process on file `track-schema.xml` specifying your target database.
3. generate the database from your scripts

The number of supported database systems in Torque is growing, and even if you do not find your specific system you could add it by specifying just a few small classes and configuration files to the Torque community.

## 2.13 Troubleshooting

Most problems that will occur when installing the server will be related to the database system software and your application server. It is beyond the scope of this document to deal with these issues in detail. Please have a look at the mailing lists and the newsgroups of **Track+** and the respective software packages. There are good support sites available for these open source softwares which cover most of the problems an administrator will face. Also have a look at the Frequently Asked Questions section of this document. As owner of the commercial version, you will get e-mail and telephone support. Otherwise you will be supported by the public forums.

If you have trouble installing **Track+**, do the following:

1. describe your problem and all error messages you get
2. tell us what application server, database server, Java environment and operating system you are using
3. post or email your `Torque.properties` file

# 3

## *Configuration and Administration*

This release of the **Track+** application supports most commonly occurring administrative tasks via the Web interface. For requirements that go beyond these supported tasks the administrator has to employ the tools that come with the RDBMS, e.g. the "mysql" command line interpreter for MySQL or the "IBConsole" utility for Firebird and Interbase. To support manual SQL commands, the database structure is illustrated at the end of the document. It is beyond the scope of this manual to teach how to issue SQL statements against a database.

The tasks most common to occur will be to add a new project with subsystems, classes, and releases. The tables of a more static nature such as states, categories, etc. can not yet be administrated via the web interface. However, modifying these tables is a task rarely done.

Initially, there are two users and a simple test project already defined in the database.

To login as the system administrator use:

username: **admin**  
password: **tissi**

To login as a regular guest user with a few access rights:

username: **guest**  
password: **trackplus**

### *3.1 The "admin" User*

The **Track+** application comes with a predefined user "admin", password "tissi" installed as the system administrator. The password and e-mail of this user should be changed when first using the system.

The admin user has access to all project administration access management pages. It should be her privilege to add new projects. Each project itself may have several project specific administrators out of the regular user base. These project specific administrators have the same rights as the admin user, except that these rights are limited to specific projects.

It is possible to add more system administrators by manually (via SQL) inserting users into the **TPERSON** table and giving them a user key of less than 100.

Only persons having administrator privileges in at least one project or system wide will be offered access to the project and access management web functions.

## 3.2 Administering Projects

There is a special web page that can be accessed if a person is registered as a system administrator, or has been granted administrator rights for a specific project. When recognized as an administrator, one can modify projects via these web pages. These pages are accessible via the "Administration" tab on the main menu and then the "Project Admin." tab. Initially, only the system administrator "admin" has access to these pages.

Any user can be promoted to administrator for the entire system and all projects by changing her user id to a number below 100. This should only be done to users freshly registered. The regular way of setting up project administrators is described in the next chapter.

The project list contains all projects for which the logged on user is registered as an administrator. Any of Subsystem, Release, and Class can be added by writing in the appropriate field below the lists and pressing the add button. Selecting an item from the list, writing some text in the edit field, and pressing the "Edit" button changes that list entry to the new value. A selected item can also be deleted if it isn't used any more in any issue. Thus, it is not possible for instance to delete entire releases if there are issues that pertain to that release, even if they have been closed.

Deleting entire projects requires to first move or delete all issues pertaining to these projects. This has to be done via manual SQL statements.

For each project, there is a default manager that receives all issues unless the person entering an issue chooses a different person. The default owner should be selected from the list presented under the "Project:" list.

For each project there is the option to include a link to a CVS repository by means of the viewCVS utility. If this option is configured, users can later on enter module and file names they like to refer to in their tasks without entering the project and base URL of the CVS repository.

## 3.3 Assigning Users to Roles

Users registered with the **Track+** application can be assigned to projects in different roles. A user can assume more than one role for a project. Assigning roles to users for a particular project can only be done by a user that has the role of an administrator for that project via the web interface shown in Figure 3.2. The project list contains all projects the logged on user has administrator rights. The role list contains the available roles

Project Administration - Mozilla Firefox

http://gandalf/track/adminProject.do?action=Create

Home | Reports | New Issue | **Administration** | Log off | Help

Access Admin. | **Project Admin.** | Copy List | Site Admin. | Workflow Definition | My Profile

Track+ Administrator

**track+**

Project :	Subsystem :	Class :	Release :
BKMS26 Bauteileabkündigung CTA FAM-AVDC Harun Al Rashid	Datenteil Endsatz HK Funkeinheit Modulator SISA	Mistaken Software Support	1.0

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Add"/>	<input type="button" value="Add"/>	<input type="button" value="Add"/>	<input type="button" value="Add"/>
<input type="button" value="Delete"/>	<input type="button" value="Delete"/>	<input type="button" value="Delete"/>	<input type="button" value="Delete"/>
<input type="button" value="Edit"/>	<input type="button" value="Edit"/>	<input type="button" value="Edit"/>	<input type="button" value="Edit"/>

Project Type :	Default Manager :
Generic Project	Landmann, Eduard
	Landmann, Herbert
	Lederer, Hanspeter

**viewCVS base URL :**

**viewCVS project :**

© 2005 The Track+ Project

Figure 3.1: Managing a project via Project Administration web page

with associated access rights. To manage the roles available in the system see section 3.4.

The two lists in the second row contain the users that have not been assigned to the selected role in the selected project, and those that have. A user with an asterisk (\*) behind its name is not active any more in the system and should not have assigned any new roles for a project. Setting a user to status active or not can be done via the site management page.

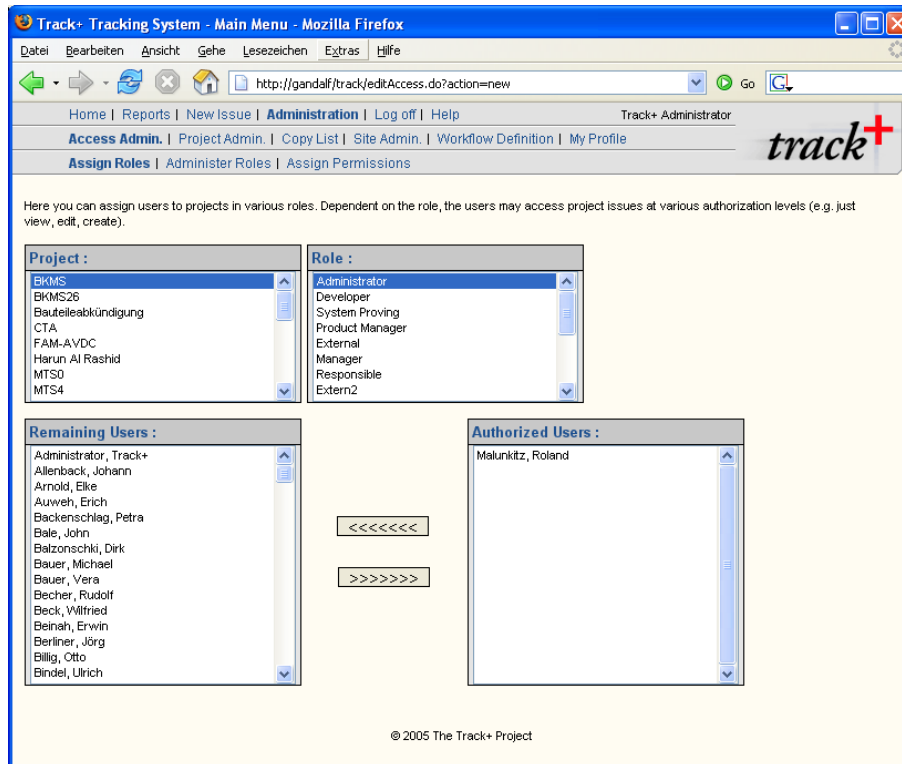


Figure 3.2: Assigning users to roles

## 3.4 Roles and Access Rights

A user has to be assigned roles for each project. The available roles are defined in the menu shown in Fig. 3.3. Each role is assigned an access key with a meaning as explained below and examples are given in Table 3.1. Each role can have limited access to specific list types.

### 3.4.1 Special Access Keys

Some access keys do not follow a general scheme and have a special meaning.

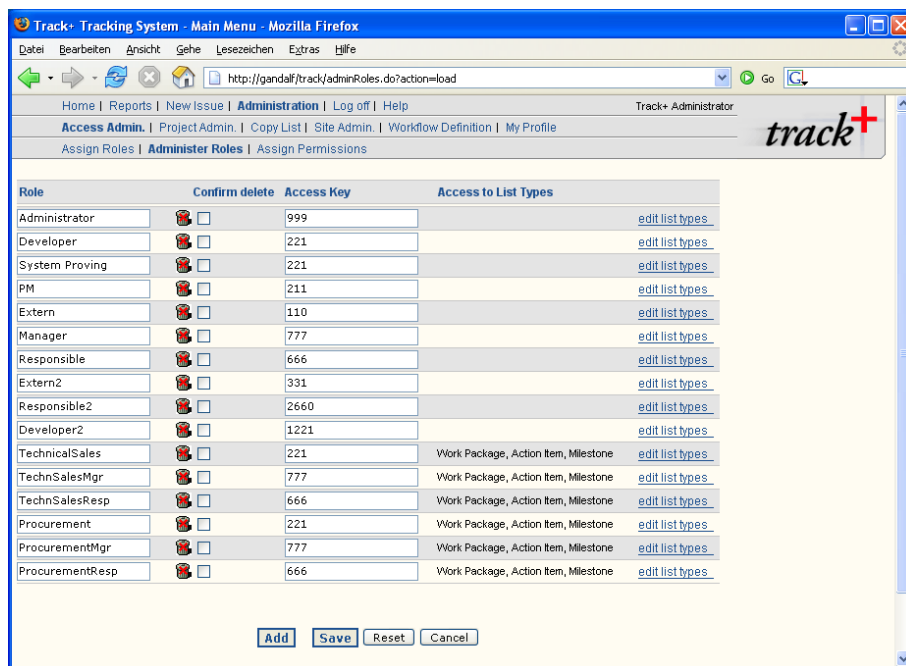


Figure 3.3: Administering roles

### 999: Administrator

An access key of **999** designates a project administrator. A project administrator has special rights for a specific project only, namely the one he is assigned this role to. He may add other persons to the project and assign roles to them. He may also remove persons from projects. He can add, edit and delete subsystems, classes, and releases. He will see the appropriate pages in the web interface that other people will not see. A project administrator has complete access to all issues of a project.

### 777: Manager

An access key of **777** designates a person that will appear in the “manager” list box of an artifact. At the same time, this person will have complete access (create, read, modify, and close) to all issues in this project.

An access key of **770** designates a person that will appear in the “manager” list box of an artifact. However, read, modify, and close rights are only granted for the artifact where this person is the actual manager.

### 666: Responsible

An access key of **666** designates a person that will appear in the “responsible” list box of an artifact. At the same time, this person will have complete access (create, read, modify, and close) to all issues in this project.

An access key of **660** designates a person that will appear in the “responsible” list box of an artifact. However, read, modify, and close rights are only granted for the artifact where this person is actually responsible for. The access keys mentioned above may be extended with a leading “1” or “2”, modifying the right to close an issue as described below.

### 3.4.2 General Access Key Scheme

In general, the access key schema consists of a four digit number where leading zeros are not shown (i.e. a regular positive integer number). Each digit has a meaning, if the number is not one of the special access key described above.

The first, most significant digit controls permission to close an artifact, i.e. to change the state of an artifact to a state with a stateflag marking this state as a “closed” state (see section 5.2). If this digit is “0” or does not exist, complete permission to close an artifact is granted. A “1” means an editor may close this artifact if he is the creator, the current manager or the responsible of this artifact. A “2” means an editor may close this artifact if he is the creator or the current manager. All other numbers are treated as “0”.

The second digit from the left controls read access to an artifact. If this digit is a “0” permission to read an artifact is not granted in this role, not even for the artifacts that this user has created. This may thus not be a very useful configuration. If this digit is a “1” permission is granted to read all artifacts this user has created himself. If this digit is a “2” permission is granted to read all artifacts in this project. All other numbers are treated as “0”.

The third digit from the left or second from the right controls permission to change an artifact. This includes permission to read an artifact. If this digit is a “0” permission to modify an artifact is not granted in this role, not even for the artifacts that this user has created. If this digit is a “1” permission is granted to modify all artifacts this user has created himself. If this digit is a “2” permission is granted to modify all artifacts in this project. All other numbers are treated as “0”.

The least significant digit controls permission to create an artifact in this project. If this digit is a “0” permission to create an artifact in this project is not granted. If this digit is set to “1” permission to create an artifact is granted. All other numbers are treated as “0”.

Some roles have identical access rights. However, since it is possible to define for each role and each project type which lists are visible, this makes sense. The default setup limits access to the lists “work packages”, “action items”, and “milestones” for the roles of Technical Sales and Procurement. All other roles have access to all lists. This setup can be changed via the screen shown in Fig. 3.3.



Ex. Role	Access Key	Meaning
Administrator	999	Project administrator
Developer	221	A powerful developer. May create, read, modify and close everything
Developer 2	1221	May create, read, and modify artifacts, but may only close his own artifacts and those that he is responsible for.
System Proving	221	Same as developer
PM	220	Product management. May read, modify and close, but not create new artifacts
Extern	111	External people, can only see and work with their own artifacts
Manager	777	Manager appears in manager list, can create, read, modify and close everything
Responsible	660	Responsible appears in responsible list, can create, read, modify and close all artifacts she is responsible for. For upgraded installations from releases earlier 3.0.0: used to be "666"
Responsible 2	2660	Can not create artifacts under this role, can only read and modify his own artifacts and those where he is responsible for, may not close an artifact unless he is the creator (under a different role) or the manager.
Technical Sales	221	See "developer" for rights. This role exists so that the visible lists can be different from those of a developer (e.g. Technical Sales people should not see the bug list)
Technical Sales Manager	777	Same as "manager". This role exists so that the visible lists can be different from those of a regular manager (e.g. the technical manager may not be allowed to see the current bug list)
Technical Sales Responsible	666	Same as "responsible". This role exists so that the visible lists can be different from those of a regular responsible (e.g. the technical responsible may not be allowed to see the current bug list)
Procurement	221	See "developer" for rights. This role exists so that the visible lists can be different from those of a developer (e.g. procurement does not need to see the development lists such as problem reports or requirements)

Procurement Manager	777	Same as “manager”. This role exists so that the visible lists can be different from those of a regular manager (e.g. procurement does not need to see the development lists such as problem reports or requirements)
Procurement Responsible	666	Same as “responsible”. This role exists so that the visible lists can be different from those of a regular responsible (e.g. procurement does not need to see the development lists such as problem reports or requirements)

Table 3.1: Example Role Table with Access Keys

### 3.4.3 Multiple Roles

It is quite common that a person is assigned multiple roles. For instance, the project administrator may also be a manager or responsible for some artifacts.

Access rights are handled such that the higher privileges always overwrite the lower privileges. For example, if a person may create issues, but may not read other peoples artifacts in the role “External”, if this person is made responsible for an artifact it will be able to read this artifact.

If a person is designated as “external” and may thus handle only it’s own artifacts, if it is also assigned to a “Developer”, it will have at least all these more extensive rights of a general “Developer”.

There is no way to “subtract” an access permission via a role; access rights always add up.

## 3.5 Workflow Configuration

It has to be clear to all project members what cycles the different artifacts may go through. In a well managed project, these state diagrams are described either in the project handbook or the configuration management handbook.

In larger projects it may be useful to guide project members so they adhere to the defined process. It has to be considered however, that many of such processes do not take exceptional circumstances into account, and may thus block workers from making useful exceptions. In addition, a large multidimensional configuration space is opened up and may lead to configuration errors.

A workflow for *Track+* controls

- ▶ which role (e.g. developer, manager, tester)
- ▶ in which project type (e.g. generic projects, development projects, procurement projects)
- ▶ for which list type (e.g. mile stones, action items, bugs)

- may change an artifact from which state to another state

The **Track+** system thus permits to define state and role based workflows for each project type and list type.

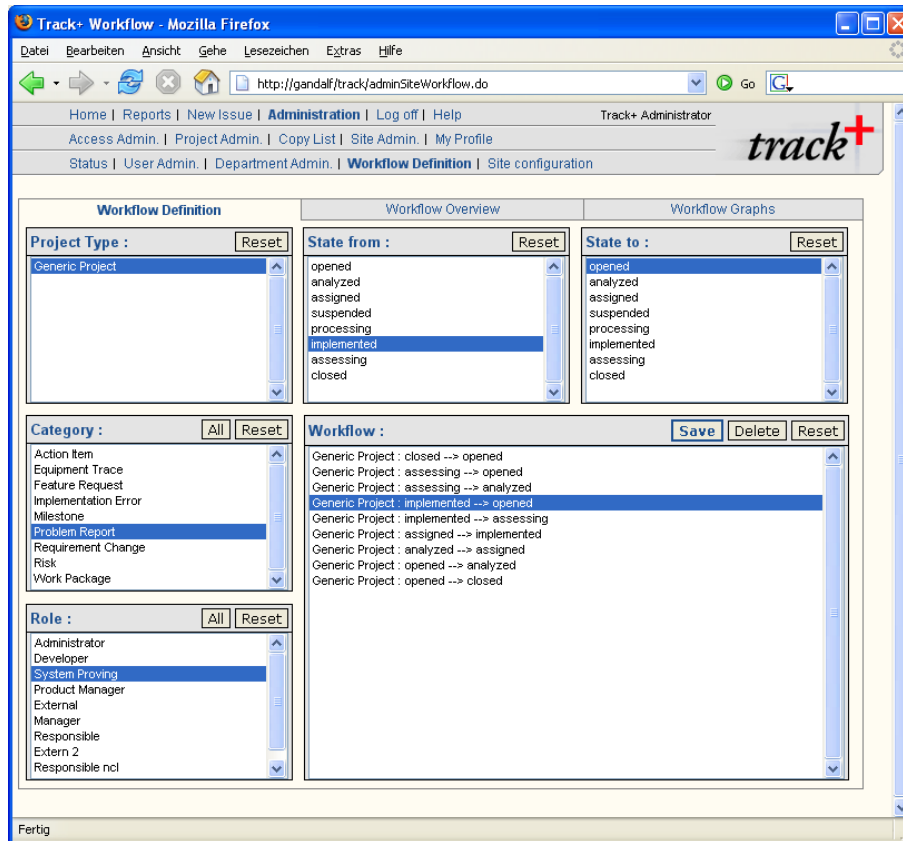


Figure 3.4: Workflow Configuration

The workflow control window consists of six areas. Five areas are used to select project type, the list type, the role, and the state transition. It is possible to select multiple list types and roles at once; it is not possible to select several project types or state transitions in one step.

### 3.5.1 Creating a Workflow

To create a workflow, proceed as follows:

1. select a project type
2. select at least one list type
3. select at least one role
4. select the state transition, consisting of an initial state and a target state

5. push the save button

Thereafter the defined state will show up in the overview area. If you select the defined workflow there, the associated selections in the other boxes will be highlighted.

### 3.5.2 Changing a Workflow

To change an existing workflow you need to select a project type, and then the respective workflow in the overview area. You can now change

- ▶ change list types
- ▶ change roles

You cannot change project types or state transitions for an existing workflow. To change these two attributes, first delete the entire workflow and then reconstruct it with the new attributes.

### 3.5.3 Workflow Reports

Two different views allow to get a good picture on the workflows defined in the *Track+* system.

Probably the most effective view is the graphical state diagram representation as shown in Fig. 3.6. It shows all defined workflows grouped by project and list type.

The second view, as shown in Fig. 3.6 provides a textual list representation of the defined workflows. It lists for each state transition for each project and list type which role can perform this transition.

## 3.6 Site Administration

There is a special web page for several site administration functions that can be accessed by the site administrators. Figure 3.7 shows the introductory site status page. It gives an overview on who is logged in, how many projects are in the database, the total number of issues in the database, the number of active and deactivated users, and a list of users currently logged in. There is the possibility to set the system into maintenance mode. This would prevent any new users from logging into the system, giving them a suitable message. When in maintenance mode, only the site administrator is able to login.

The next screen shown in Fig. 3.8 allows the site administrator to deactivate users so that they are not able to login any more, and so that they do not appear in any drop down lists any more. However, their activity history is still kept in the database, and they may be reactivated again later on. Currently there is no functionality to permanently delete users from the database. Only users that are not deactivated count as “active” users.

The list of departments that can be selected when a user registers with the system can be managed using the screen shown in Fig. 3.9.

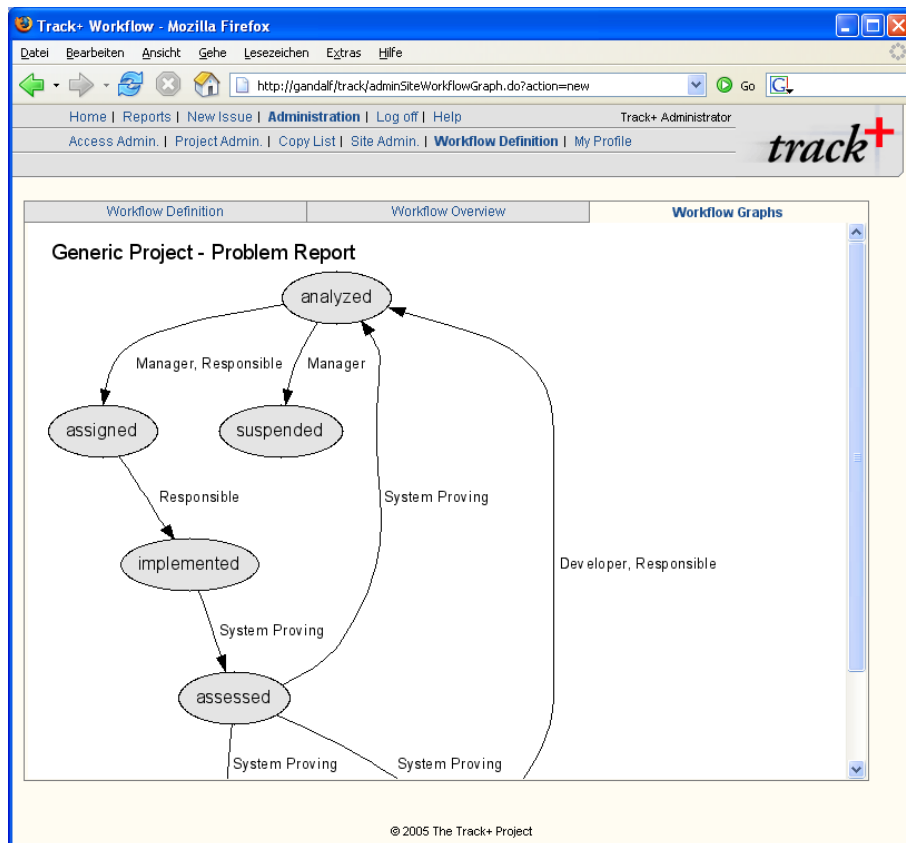
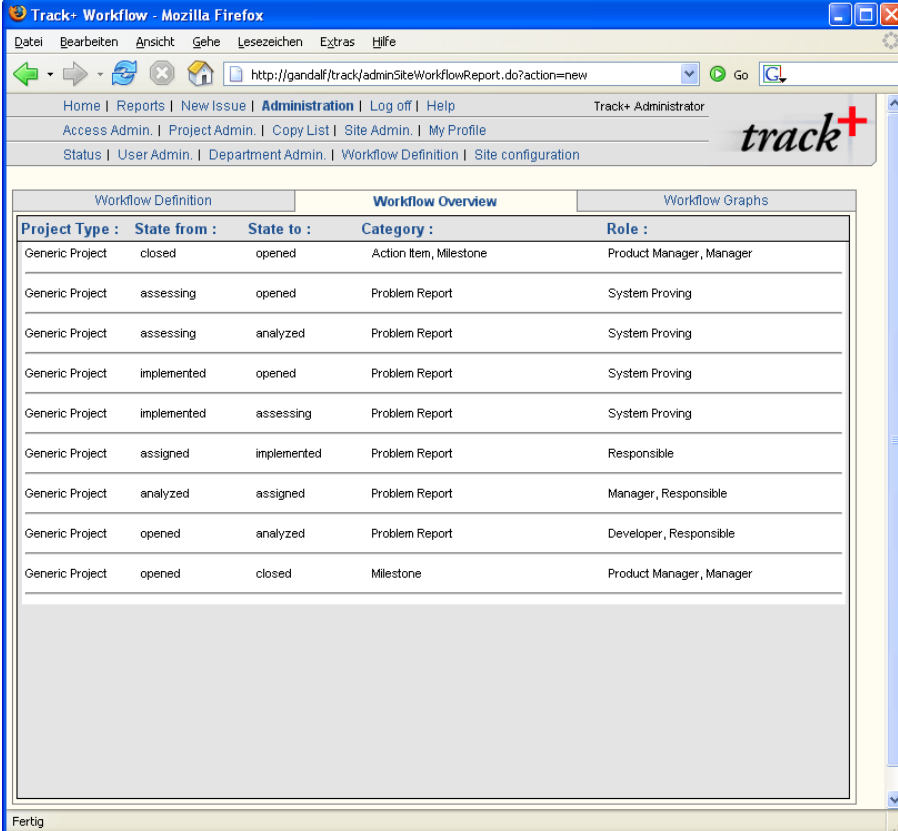


Figure 3.5: Workflow Graphs



Project Type :	State from :	State to :	Category :	Role :
Generic Project	closed	opened	Action Item, Milestone	Product Manager, Manager
Generic Project	assessing	opened	Problem Report	System Proving
Generic Project	assessing	analyzed	Problem Report	System Proving
Generic Project	implemented	opened	Problem Report	System Proving
Generic Project	implemented	assessing	Problem Report	System Proving
Generic Project	assigned	implemented	Problem Report	Responsible
Generic Project	analyzed	assigned	Problem Report	Manager, Responsible
Generic Project	opened	analyzed	Problem Report	Developer, Responsible
Generic Project	opened	closed	Milestone	Product Manager, Manager

Figure 3.6: Workflow Report

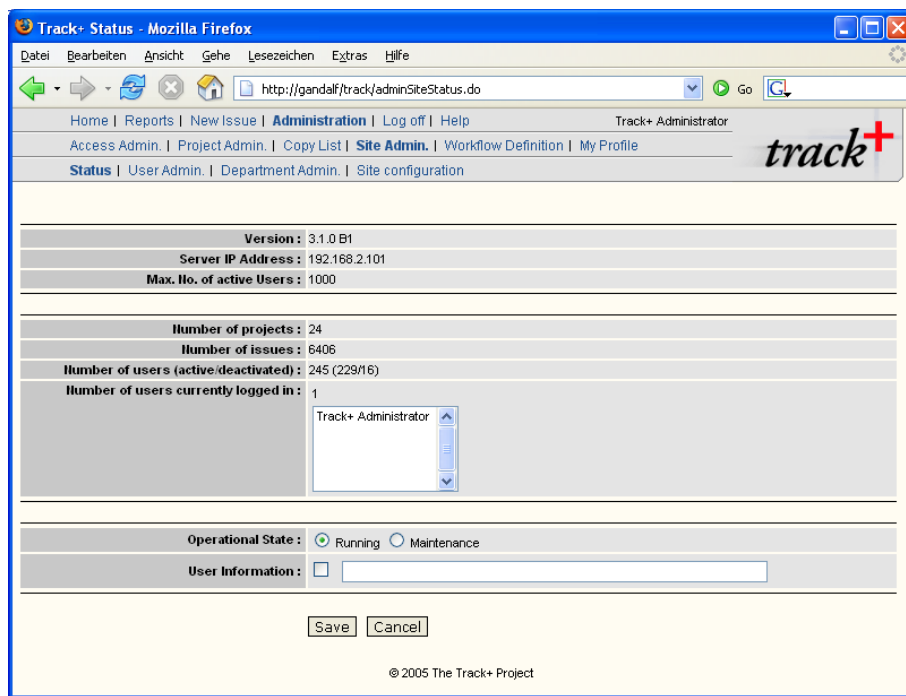


Figure 3.7: The system status page

## 3.7 Customization

For many users the default installation may not match exactly their process requirements. There may be a need to have other priorities and severities defined, or the state model has to be modified. In this chapter it is outlined how to adapt the *Track+* system to these requirements.

Since *Track+* has at its core a database, customization mostly means changing some entries in the database. This can be done as outlined above, with one of the utilities that come with the database. In the following, as an example we go through renaming a priority and changing the name of one priority. Changing the other entities follows the same procedure.

### 3.7.1 Customization Example: Changing a Priority

Entries for the defined priorities can be found in table **TPRIORITY**. The table carries three attributes, **PKEY**, **LABEL**, and **SORTORDER**. **PKEY** has to be a unique integer, and it is recommended to use the lowest available positive integers available. “Label” describes what the user sees unless the entry is being localized. So changing this label usually does not change what the user sees at the interface!

Adding a new priority means adding a new row to table **TPRIORITY**, for instance with a **PKEY** of 4, a **LABEL** of “MyNewPriority” and **SORTORDER** of 0 to have the new priority appear at the top of the list where the user can select it. That would be it unless

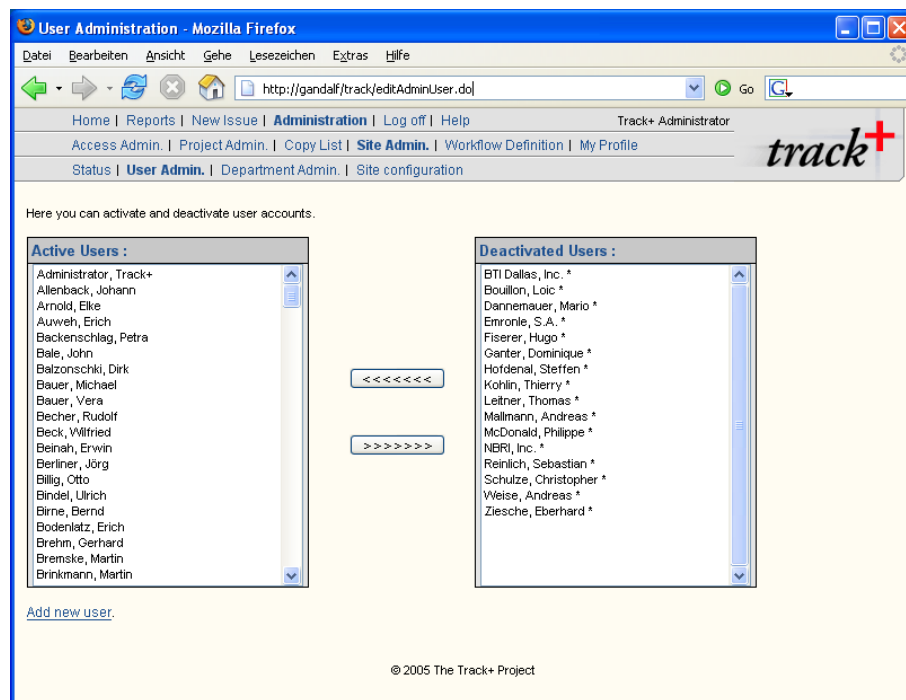


Figure 3.8: Deleting and reactivating users

one wants to localize this label. Localization takes place via files **WEB-INF/classes/resources/BoxResources\_xy.properties**, where xy stands for your locale ("en" for English, "de" for German, "no" for Norwegian, "it" for Italian and so on. Look at your browsers language menu for more).

Here is an excerpt from the standard version of this file:

```
...
TCATEGORY.WorkPackage=Work Package
TPRIORITY=x; occasionally; soon; immediate
TPRIORITY.immediate=immediate
TPRIORITY.occasionally=occasionally
TPRIORITY.soon=soon
TROLE=x; Administrator; Developer; System Proving; PM;
        External; Manager; Responsible
TROLE.Administrator=Administrator
...
```

We need to add a new line which looks like this:

```
TPRIORITY.MyNewPriority=My new Priority
```

If we want to change priority "soon" to "fast" we change this line:

```
TPRIORITY.soon=fast
```



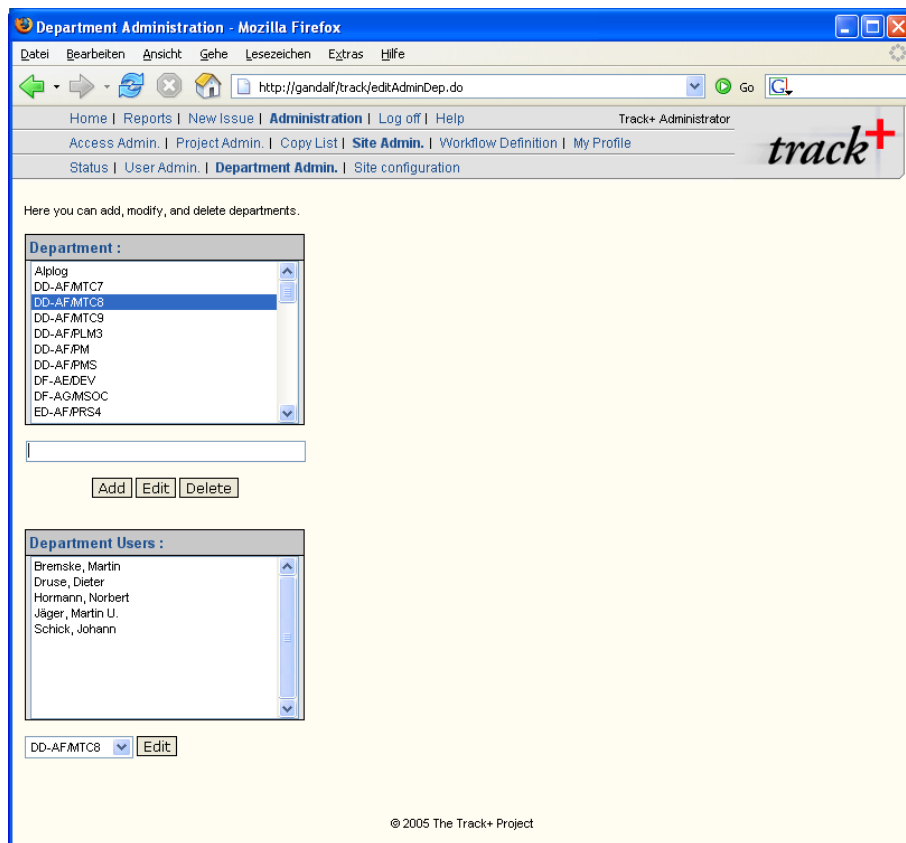


Figure 3.9: Managing the department list

The same procedure holds true for all entries except for the **TSORT** entry. The state table has an additional attribute called **STATEFLAG**. This integer indicates if a state belongs to category "open" (**STATEFLAG**=0) such as "opened", "analyzed", etc., category "closed" (**STATEFLAG**=1) such as "closed" or "suspended", or category "ready for final test or closing" (**STATEFLAG**=2).

The localization of **TSORT** pertains to the sort order box in the report configuration menu; it is not related to any database table. Lines of the form **TXYZ=xyz** are legacy entries. This format should have been deprecated and should not be used any more.

### 3.7.2 Changing the Report Overview

Sometimes you may not be satisfied with the fields shown in the report output. You may need different fields, or you may need them in a different format and order.

The report overview page is defined by the Java Server Pages file **report.jsp**. Here is an excerpt out of this file showing how the column for the responsible is defined:

```
<td width="15%" align="left"
    <%=evenline?"class=\"background-gray\"":""%>
    <logic:equal name="reportRow" property="onPlan"
        value="false">
        <font color="red">
    </logic:equal>
        <bean:write name="reportRow" property="responsible" />
    <logic:equal name="reportRow" property="onPlan"
        value="false">
        </font>
    </logic:equal>
</td>
```

The lines containing the **logic:equal** tags make sure that an artifact that is overdue is shown in red color. The actual value for the responsible of an artifact is shown in the **bean:write** tag.

Here is the example if we had replaced responsible by the originator of the artifact:

```
<td width="15%" align="left"
    <%=evenline?"class=\"background-gray\"":""%>
    <logic:equal name="reportRow" property="onPlan"
        value="false">
        <font color="red">
    </logic:equal>
        <bean:write name="reportRow" property="originator" />
    <logic:equal name="reportRow" property="onPlan"
        value="false">
        </font>
    </logic:equal>
</td>
```

Table 3.2 shows all available properties that may be used in this fashion in **report.jsp**.

Property Name	Explanation
workItemKey	the issue reference number
packageSynopsis	the synopsis
packageDescription	the description
shortPackageDescription	the description, truncated to 160 characters
reference	the reference string
build	the build string
lastEditGUI	the date the artifact was last changed
createdGUI	the date of creation
owner	the manager (lastname, firstname)
ownerID	the object id of the manager
originator	the original author (lastname, firstname)
originatorID	the object id of the originator
responsible	the responsible (lastname, firstname)
responsibleID	the object id of the responsible
projectCategory	the category
categoryID	the object ID of the category
project	the project name
projectID	the project object id
relScheduled	the release for which solution is scheduled
relNoticed	the release where problem was noticed
reqClass	the class
reqPriority	the priority
reqSeverity	the severity
state	the actual state
stateFlag	the stateFlag, for explanation see <a href="#">5.2</a>
startDateGUI	the start date
endDateGUI	the due date
onPlan	is still on plan (boolean)
overdue	is overdue (boolean)
isOpen	artifact is open, not closed (boolean)
superiourworkitem	the reference number of the parent artifact
superioursynopsis	the synopsis of the parent artifact

Table 3.2: Properties for customization of report.jsp

If you want to add or change a column you also have to change the column heading. For this you can either hardcode the heading into the file at the appropriate place, or make it localizable by adding an appropriate entry into file **ApplicationResources.xy.properties**, **xy** being the designator for your locale (e.g. **de** for German or **sv** for Swedish).

## 3.8 Adding new Locales

**Track+** has been designed such that it is easy to localize (adapt to a new language). The **Track+** core team fully supports English and German as official locales, but numerous others already exist, and the list is growing. Here are some instructions how to add a new locale to **Track+**.

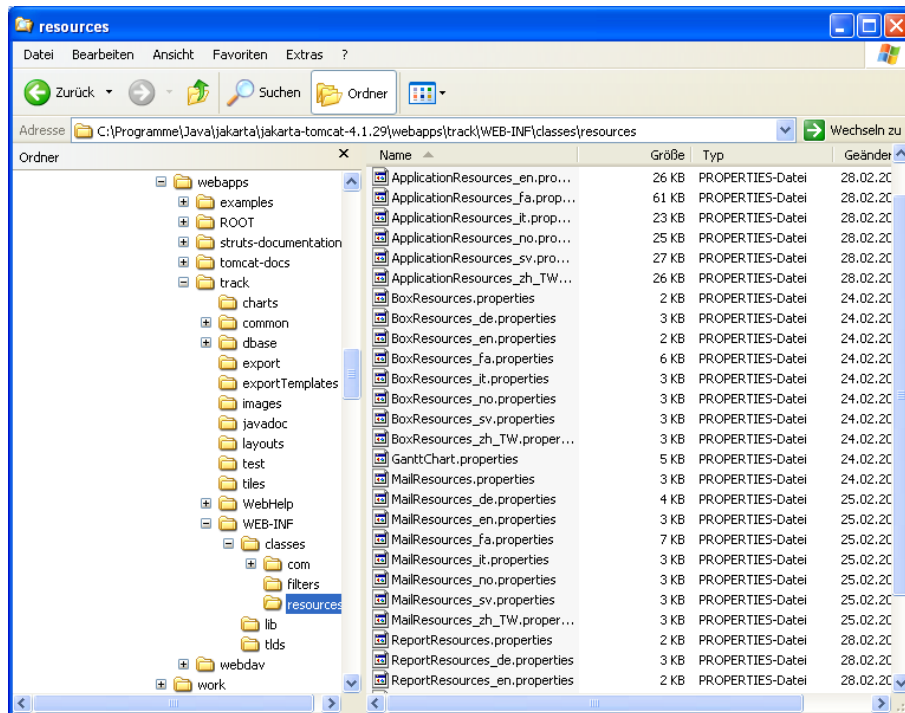


Figure 3.10: Files for localization

The files use for localization are shown in table 3.3. To create a file for a specific locale you need to use appropriate locale postfix and append it to the basename, e.g. **MailResources.de.properties** for a German locale, or **MailResources.fr.properties** for a French locale.

Particularly in the **MailResources** file you find placeholders, shown in curly braces "{0}". These placeholders are replaced by the program at runtime with suitable values, e.g. a number or a text.

It may be useful to employ a suitable tool for editing the resource files. One example is the Zaval Resource Editor which can be freely downloaded from the Internet.

File Name (Base)	Explanation
ApplicationResources.properties	contains most strings that you see at the user interface.
MailResources.properties	contains strings that are used for the emails that are being sent by the <b>Track+</b> system
BoxResources.properties	contains the strings that are displayed in the list boxes of the application
ReportResources.properties	contains strings that are found in customized reports

Table 3.3: Files for localization

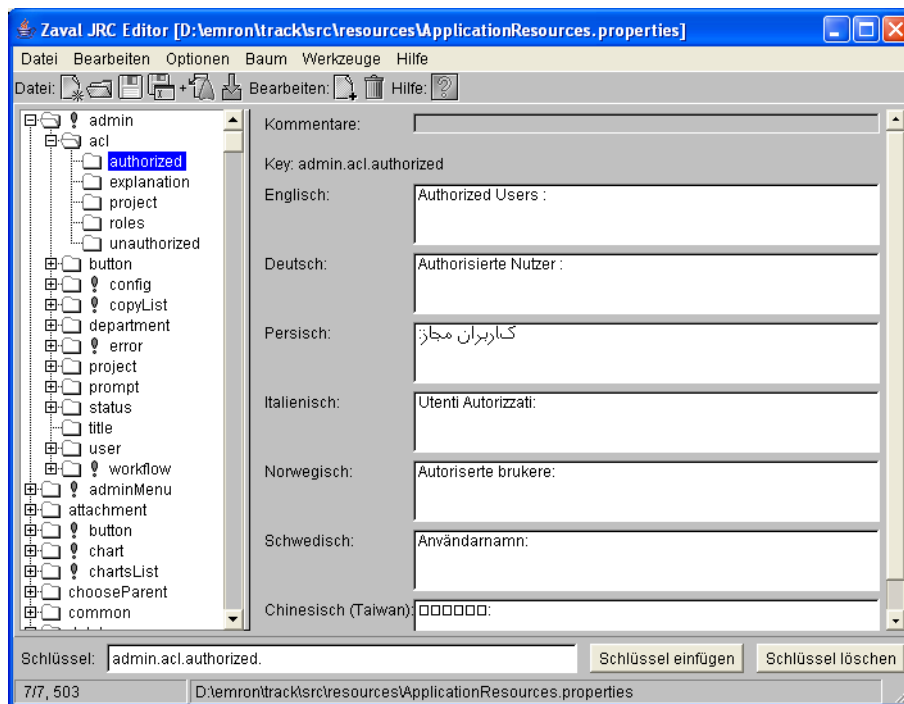


Figure 3.11: Zaval Resource Editor for Localization

The Zaval Resource Editor does not handle Unicode very nicely. A better tool could be a suitable plugin in the Eclipse or NetBeans development environment, as shown in Fig. 3.12.

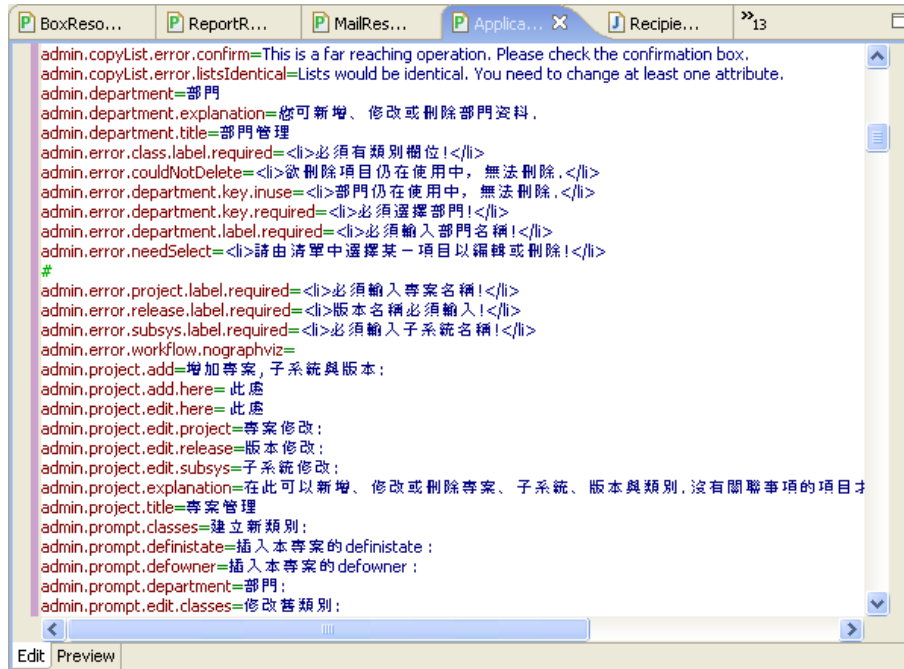


Figure 3.12: Eclipse Plugin Resource Editor

## 3.9 Database Backup

It will be necessary to backup the database of the **Track+** system on a regular basis. For the Firebird RDBMS the suggested procedure is to make a local backup copy of the database using the `gbak` utility and then use the regular backup facilities to backup this file. The backup file is usually much smaller than the regular database since it is compressed. For other RDBMS follow the procedures suggested by the system supplier. On a Unix system, `cron` may be used for the regular backup file creation. In case of a Firebird or Interbase database management system, the following script **backib.sh** installed into `/opt/interbase/bin` will backup the database into a directory `/opt/interbase/backups`:

```

#!/bin/sh
IBROOT=/opt/interbase

$IBROOT/bin/gbak -BACKUP\_DATABASE -USER SYSDBA -PASSWORD XX
$IBROOT/db/track.gdb $IBROOT/backups/track.gbk.

```

It may be installed into `crontab` with

```
crontab --e (as Unix root)
```

to run all weekday nightly at 20:28 (8:28 pm):

```
.....  
# Backup the Track+ database  
28 20 * * 1,2,3,4,5 /opt/interbase/bin/backib.sh
```

## 3.10 Security

If security is an issue, and it should be, the following points should be taken into account:

1. The default passwords supplied with the database and the **Track+** system should be changed during the installation process.
2. The operating system access rights to the database files, backup files, and database utilities should be set restrictive, e.g. read-write-execute permission only to user root or the user the database server is running under.
3. If password snatching over the network is an issue, the connection to the servlet container should be setup as secure with https. It is beyond the scope of this manual to show how to do this, but it is not difficult for someone knowing how to configure Apache or any other application server.

The passwords are stored in the database SHA coded which makes it practically impossible to retrieve them.





# 4

## Using **Track+**

### 4.1 Registering with **Track+**

The **Track+** system can usually be accessed by a link like `http://localhost/track` or `http://localhost:8080/track` if you have it installed on your local computer, or the respective computers name instead of `localhost` if the application is installed on a different computer. Be aware that the first valid URL used to access the system is the one that shows up as link reference in e-mails sent by the **Track+** system.

In order to use the **Track+** system an account is needed. Anybody with a valid e-mail address can register from the login page of the application. The pattern of allowed e-mail domains might have been restricted by the system administrator. By default, all domains are enabled.

After registration, a password is sent to the e-mail address and the user may log on. The password should be changed on first login via the "**Administration** → **My Profile**" tab. The user or login name is case sensitive. The **Track+** system administrator is being informed via e-mail that a new user has registered with the system.

*The freshly registered user doesn't have any access to any project. A project manager or the **Track+** administrator has to add a user to a project so that this user is visible to others in the project and can be assigned issues, and can himself monitor issues in this project.*

A user may be registered by another user of the system, e.g. the project manager, or his line manager. However, etiquette requires that the user to be registered is being informed prior to the registration.

If a user has forgotten his password or login name, from the login page he can have a new password for all his login names sent to his e-mail address.

When LDAP-authentication is enabled for your **Track+** installation (see section 2.8.6) a user can choose to authenticate against his local **Track+** password or against his LDAP password. To use LDAP the user has to check the "LDAP-authentication" checkbox during registering or via the "**Administration** → **My Profile**" tab.

If the users loginname is found on the LDAP-server, LDAP-authentication is turned

Figure 4.1: Registration of a new user

on for this user. The user has now to use his LDAP-password for logon. The LDAP-password is the password the user has to enter, if he wants to bind to the LDAP-server with the “simple user/password” authentication method.

For switching back to local the **Track+** password the user has to uncheck the “LDAP-authentication” checkbox and supply a new local password.

Once the **Track+** server has been installed and configured (see previous chapters), users can register via the procedure described above. Once registered the user may log on to the system and should change her default password to something more secure. Note that passwords are stored in the database in an encrypted (SHA) form, nobody can actually read them.

## 4.2 Getting Online Help

**Track+** comes with a context sensitive online help system installed which you may access by selecting the “Help” hyperlink in the menu bar. Figure 4.2 shows the introductory online help screen when selected from the **Track+** main menu.

## 4.3 Creating and Changing Artifacts

Once you have been granted proper authorization for a specific project you may create and edit artifacts such as feature requests, action items, bug reports or risks. From the main menu you can select the “New Item” link. To edit an existing item you may type the item number into the little box in the menu bar, or get a list of artifacts according to your selection criteria via the “Reports” link. Once you have selected an artifact that

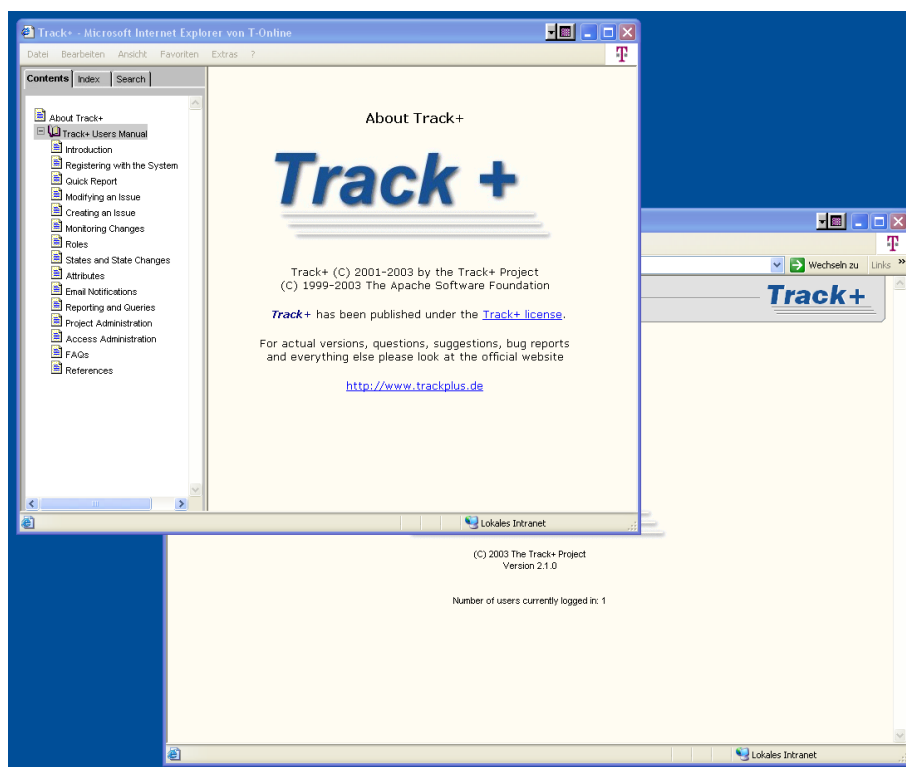


Figure 4.2: Sample help screen and main menu

you have at least read access to the screen as shown in Fig. 4.3 will appear. If you are

**Printable Report for Issue No. 4646 - Mozilla Firefox**

http://gandalf/track/searchItem.do

Home | Reports | New Issue | **Edit** | Administration | Log off | Help

George Sinatra

4646

**Issue No. 4646 : implemented : Verfasser hat keinen Zugriff auf Punkt**

Project: Track+, Rel. 1.3  
 Build: 1.3.1  
 Subsystem: Edit/New Item  
 Class: Software  
 Create Date: 6/26/02 11:26 AM  
 Originator: Balzonschki, Dirk  
 Manager: Sinatra, George  
 Responsible: Sinatra, George  
 List: Requirement Change  
 Priority: occasionally  
 Severity: non-serious

**Description**

Ein von mir geschriebener, inzwischen jedoch einem anderen Projekt (nicht mehr "MDMS - Radio Core") zugewiesener TRACK Report ist von mir nicht mehr einsehbar:  
 "Sie haben keine Zugriffsrechte für diesen Punkt"  
 (Beispiel: [Issue No. : 4445](#))

-> Verfolgung z.B. der angefügten Kommentare nicht möglich  
 -> Ergänzungen sind nicht mehr möglich

**State Change History**

Date	To State	Changed By
1/23/04 9:21 PM	implemented	Sinatra, George
7/9/02 12:02 PM	analyzed	Sinatra, George
7/2/02 9:47 AM	opened	Balzonschki, Dirk

Es wäre hilfreich wenn bei der Anzeige "Sie haben keine Zugriffsrechte für diesen Punkt" auch die Liste der zuständigen Projektverwalter angezeigt wird (ggf. mit Kommentar "Bitte wenden Sie sich an xxx, um sich Zugangsrechte eintragen zu lassen.")

Date	To State	Changed By
7/1/02 3:10 PM	closed	Sinatra, George

Figure 4.3: Artifact overview page

allowed to modify this artifact, you will see an "Edit" link in the main menu. If you have only read access, this link will not be offered to you and you will not be able to modify the artifact.

The overview page shows you all information that is available for this artifact, including all trails for state changes, date changes, and all other changes (audit trail).

If you follow the "Edit" link from the overview page you will get into the screen as shown in Fig. 4.4. Each artifact should have a descriptive synopsis. In the description text you may use simple tags to color, underline, boldface or italicize text, and to create active hyperlinks to other artifacts or external hyperlinks and include source code snippets:

Home | Reports | **New Issue** | Administration | Log off | Help George Sinatra

**track+**

---

Issue No. : **4443**

Create Date : 6/4/02 9:50 AM

Originator : Teufel, Jens

Manager : Sinatra, George

Responsible : Sinatra, George

Project : Track+

Subsystem : Edit/New Item

Class : Software

Rel. Noticed : 1.1

Rel. Scheduled : 1.1

Build : v1.21b1

List : Feature Request

Priority : occasionally

Severity : non-serious

Start Date : 1/2/03

Due Date : 9/1/04

Synopsis : Allow attachments on NewItem

Description : **B I U** **ISSUE** **SRC** [http://](#) **VC** Color:

Currently, attachments can only added after an item is created.

**[b]DESIRED-CHANGE[/b]:**

Allow attachments for "New Item"s, too.

Have a look at [issue 4447/] also.

Fixed with [vc=com/aurel/track/AttachNew.java rev=1.4]AttachNew.java[/vc]

Parent Item **4254** Originator does not get email about state changes Clear

Attachments

Last State Change : 12/5/02 3:17 PM by : Sinatra, George

Current State : **assigned** New State : assigned

**B I U** **ISSUE** **SRC** [http://](#) **VC** Color:

Comment : Text from Mr. Buhmann:

It is not possible to add attachments in the initial dialog for new track report entries. Attachments have to be added in a second (modification) step.

There should be either a note in the entry dialog how attachments may be added or it should be possible to immediately add attachments with the initial setup of a new track report (preferred solution).

Figure 4.4: Edit artifact page

- ▶ `[issue # /]` creates a hyperlink to another artifact. E.g. `[issue 247/]` would create a hyperlink to issue number 247.
- ▶ `[u] [/u]` would underline the text enclosed
- ▶ `[b] [/b]` would give **bold** text
- ▶ `[i] [/i]` would give *italic* text
- ▶ `[src] [/src]` allows you to include source code into the description. It will be displayed in a separate box.
- ▶ `[url=http://...] [/url]` allows you to include active hyperlinks to external web sites. For example, `[url=http://www.trackplus.org]Track+site[/url]` would create an active hyperlink to the **Track+** home page.
- ▶ `[vc=module/file <rev=x.y>]`Description of file `[/vc]` allows you to create an active hyperlink to a CVS repository, assuming you have installed the viewCVS program. The base URL is set once for the entire project this issue belongs to, so there is no need to enter it here. The revision is optional; if set the file in that revision will be displayed. Otherwise an overview page with all revisions of this file is shown in a new browser window.

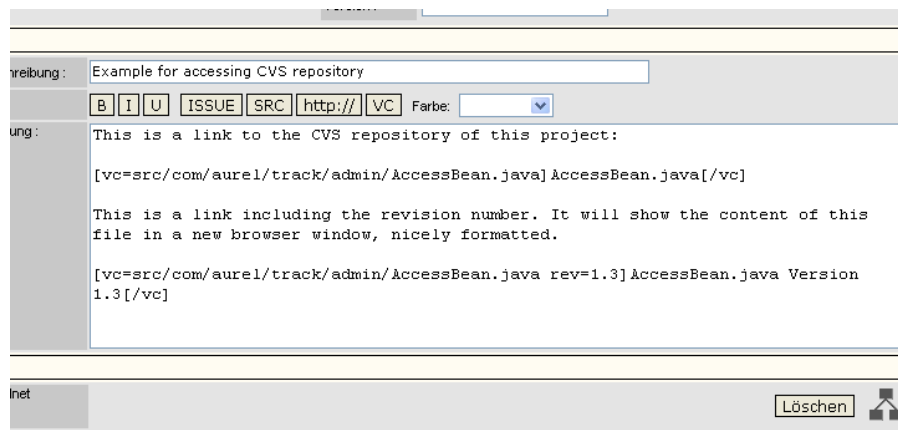


Figure 4.5: Creating a link to a source repository

You may add or delete attachments from an artifact. The attachment list modification takes place via a separate window as shown in Fig. 4.6. You may link an artifact with a parent artifact, to show a hierarchical dependency. To select a parent artifact you click on the organization icon and will get a new window. Here you can filter artifacts by project and subsystem, as shown in Fig. 4.7. If you look at the overview page of an artifact, you will also see all dependent artifacts, that is all artifacts that have this artifact as a parent.

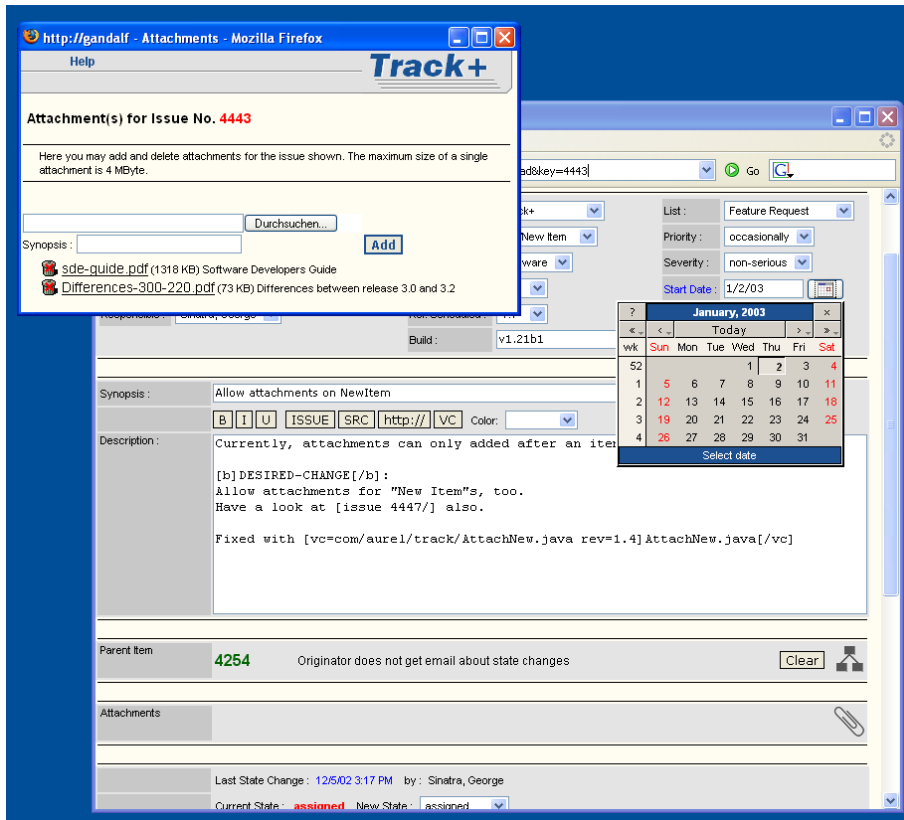


Figure 4.6: Modifying the attachment list

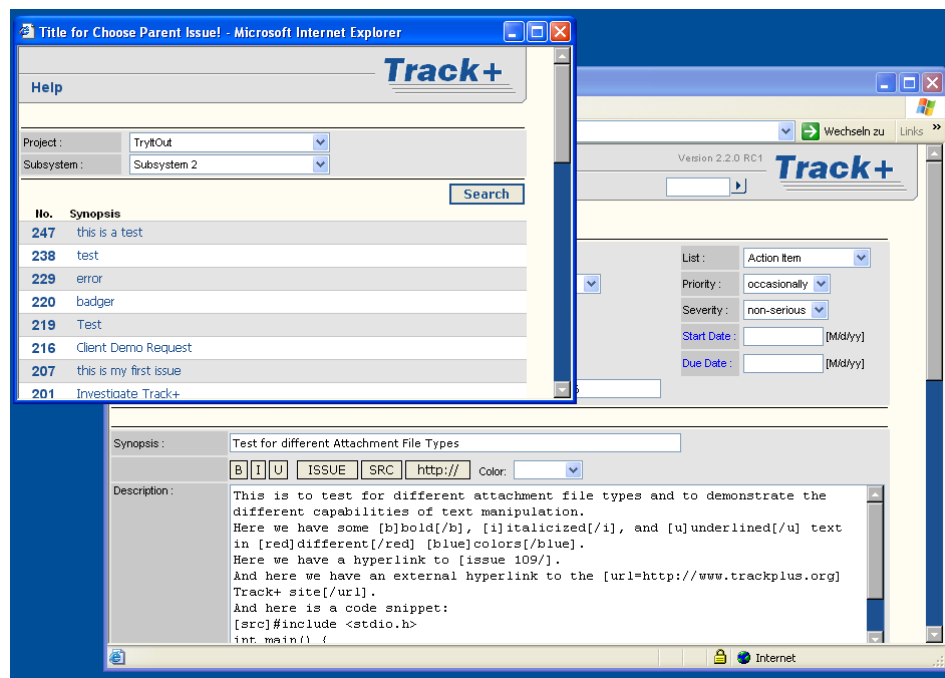


Figure 4.7: Selecting a parent artifact

## 4.4 Copying Lists from Templates

**Track+** offers the possibility to copy a list to a new list, changing some parameters in the process. This can be useful in many ways, e.g. if you have standard quality gate check lists or assessment plans defined in your organization or project. You enter the plan or check list once, and you copy it to a protocol with each development cycle. Then you can work off the protocol until all tasks are closed. Figure 4.8 shows the screen where you can control the copy process. You can copy an entire list to a new one. Each existing item matching the filter criteria on the left is copied to a new list, replacing the existing parameters by the values on the right side. Parent-child relationships are preserved. Links to attachments are copied, but not the attachments themselves.

As an example, let's assume you have a test plan that needs to be gone through with each release of your system. You would enter the plan once, placing all artifacts into the "template" list. Then you would copy the template list into an assessment protocol list, changing the release number to the release to be assessed.

It is recommended to enumerate the synopsis for each template item, to later on get an ordered list by synopsis (e.g. "1.1 My first synopsis").

When copying to a new project, some attribute values may not exist. In such a case a best match is attempted, otherwise default values are inserted into the new list.

You can set the initial state of all new items.

Since this is a far reaching operation, possibly generating a lot of new data, you need to confirm it by marking the confirmation check box.



From List	To List
Project : TryItOut	Track+
Release noticed : 1.0	1.1
Release scheduled : 2.0	1.05
Build : 1	2
List Type : Templates	Equipment Trace
Manager :	Auweh, Erich
Responsible :	Sinatra, George
Initial State :	opened

Check here to confirm : ☐

Figure 4.8: Copying lists from templates

## 4.5 Quick Report

One of the more frequent tasks a user may have to perform is to get an overview over all her unclosed activities. This report can quickly be obtained via the **"My Issues"** menu entry on the navigation bar at the top of the screen. The issue reference numbers in all reports are active links and lead to the **"Edit Issue"** page where the activities and their associated data such as states, owners, descriptions etc. can be modified.

The modification date and the person modifying the issue is recorded in the database. One of the more important attributes of an issue is its **"state"**. Each state change is recorded with date and person who performed the change. The history of all state changes for an issue can be obtained via the **"Edit Issue"** page.

Each issue in this report has a little status icon associated with it. A construction site icon designates issues that haven't been closed or suspended, and that are not late. A warning icon (exclamation mark) designates issues that are not yet closed, and are late. Furthermore, the entire line is marked red for late issues. A green checked off sign indicates issues that have been closed on time, a red checked off icon indicates issues that are closed, but have been closed late.

## 4.6 Reporting and Queries

Beside entering an artifact and changing its states over the life-cycle, the most common task in project and change management is getting a view on all important issues one is concerned with. For this purpose, the **Track+** application currently provides two entry points:

No.	Project	Subsystem	Synopsis	State	Responsible	Last Edited
105	Track+, 1.3	Reports	Report configuration persistence	opened	Sinatra, George	21.10.02 14:11
123	Track+, 0.5	EditNew Item	Support of Parent/Child relation	opened	Sinatra, George	15.04.05 21:16
4254	Track+, 0.5	Documentation	Originator does not get email about state changes	implemented	Sinatra, George	15.04.05 21:17
4443	Track+, 1.1	EditNew Item	Allow attachments on NewItem	assigned	Sinatra, George	15.04.05 21:18
4646	Track+, 1.3	EditNew Item	Verfasser hat keinen Zugriff auf Punkt	implemented	Sinatra, George	23.01.04 21:21
4701	Track+, 0.5	Reports	No email sent to person editing an issue	opened	Sinatra, George	03.07.02 15:56
5221	Track+, 1.3	EditNew Item	New filter criteria for 'creator' ('Verfasser')	opened	Sinatra, George	15.04.05 21:18
4846	Track+, 1.3	EditNew Item	The 'Synopsis' line is too short even for some val	analyzed	Sinatra, George	01.02.03 16:24
5098	Track+, 0.5	Documentation	Projektmanager in Manager/Bearbeiter unsichtbar	implemented	Sinatra, George	15.04.05 21:24
5394	Track+, 1.3	Reports	Track+ verliert Rückwärtsreferenz	opened	Sinatra, George	15.04.05 21:25
5420	Track+, 1.3	EditNew Item	Changehistory shows wrong User	opened	Sinatra, George	15.04.05 21:26
5205	Track+, 1.3	EditNew Item	'Subsystem' added to subject of mail notification	analyzed	Sinatra, George	05.12.02 15:25
5441	Track+, 1.3	Documentation	Track+ verliert Kommentare	opened	Sinatra, George	12.12.02 13:27
5444	TryOut, 2.1	Documentation	Ein kleiner Testeintrag	processing	Edel, Clemens	30.12.02 14:30

Figure 4.9: Sample report page with hierarchie and closed issues

- A quick overview on "My Issues"
- A more complex configurable query to get artifacts of interest

The "My Issues" link provides a list with all issues where the user logged in has been noted as either \$ORIGINATOR, \$OWNER, or \$RESPONSIBLE, and with states in category "open" (e.g. open, implemented, assigned, etc.). This should serve the need of most regular users. The default sort order is by issue number.

For more complex queries, the user can assemble a "filter" via the "Complex Reports" link that restricts the output of the query. While this may not cover every possible need, it should serve most requirements.

When defining a filter, only one project at a time can be queried. Each selection as shown in Figure 4.10 restricts issues to those that match the selected criteria. In the example, all issues of project "Caliph++" that are not closed will be reported. If a criteria is not selected it is not being applied. Thus it is the same if one selects all criteria of a certain class or none at all.

For the text filter, only the fields "Synopsis", "Description", and "Build" are considered. Pattern matching is done via regular expressions as defined for the Perl language version 5. Pattern matching is not case sensitive. Here are some examples:

- "word" would filter out all records that do not have "word" or "Word" or any other variant in any of the Synopsis, Description or Build fields.
- "word.\*filter" would filter out all records that do not have "wordfilter" or "wordXYZfilter" or any other such variant in any of the relevant fields.
- "word|anyw" would filter out all records that do not have "word" or "anyw" or its upper and lower case variants in any of the relevant fields.

The screenshot shows the 'Report Configuration' window in Mozilla Firefox. The browser address bar shows the URL: `http://gandalf/track/reportConfig.do?action=create`. The page has a navigation bar with links: Home, Reports, New Issue, Administration, Log off, Help. The user is logged in as 'George Sinatra'. Below the navigation bar, there are tabs: Custom, TQL, My Issues, Managers Issues, Responsibles Issues. The main area contains several filter sections, each with a dropdown menu and a list of items:

- Project :** All, Harun Al Rashid, SDMX 155E16, SRDU 511 T, Track+, TryItOut
- Subsystem :** All, Analysis, Business Logic, Capacity Monitor, Database, Doc Architecture
- Class :** All, Dev. Doc., Mistaken, Software, Support, User Doc.
- Release :** All, 1.6, 1.5.2, 1.5.1, 1.5, 1.0
- Manager :** All, Auweh, Erich, Becher, Rudolf, Fuchs, Andreas, Gleisser, Martin, Mattes, Christoph, Meier, Walter, Reise, Timoteus, Reutterle, Norbert, Sandow, Rudolf
- Responsible :** All, Becher, Rudolf, Fuchs, Andreas, Gleisser, Martin, Mattes, Christoph, Meier, Walter, Reise, Timoteus, Reutterle, Norbert, Sandow, Rudolf, Stab, Horst
- List :** All, Templates, Action Item, Equipment Trace, Feature Request, Implementation Error, Milestone, Problem Report, Requirement Change, Risk
- Current State :** All, opened, analyzed, assigned, suspended, processing, implemented, assessed, closed
- Date Filter :** None
- Contains Text :** (regular Perl5 pattern)
- Priority :** All, occasionally, immediate, soon
- Severity :** All, non-serious, serious, critical

At the bottom right, there are buttons: Reset, Cancel, and Execute Query.

Figure 4.10: Configuring a report

- "word\s\*\w\*\s\*anyw" would filter out all records that do not have "word" followed anywhere in the rest of the text by "anyw" or its upper and lower case variants in any of the relevant fields.

The date filter allows to filter issues according to the criteria shown in the selection box. The report may be clicking on the head labels of each column in the report output. If there are hierarchical dependencies between different entries of the report set, they will be displayed as shown in Fig. 4.9.

## 4.7 Exporting Reports

With **Track+** it is possible to create customized reports and graphically visualize report data by using the report export function. The export can write all relevant data of a report to new formats such as a comma separated value (CSV) file, XML file, HTML file, or PDF file. These files may be further processed for example by a spread sheet program to change sort orders, perform counts, etc.. The templates for the export function are deployed for the system in folder `webapps/track/exportTemplates`. The format of the files is given by the JFreeReport specification.

Fällig	Nr.	Projekt	Subsystem	Kurzbeschreibung	Zustand	Bearbeiter	Letzte Änderung
	105	Track+	Reports	Report configuration persistence	opened	Sinatra, George	102/102 2:11 PM
	123	Track+	EditNew Item	Support of Parent/Child relation	opened	Sinatra, George	4/15/05 9:16 PM
	4254	Track+	Documentation	Originator does not get email about state changes	implemented	Sinatra, George	4/15/05 9:17 PM
9/1/04	4443	Track+	EditNew Item	Allow attachments on NewItem	assigned	Sinatra, George	4/15/05 9:18 PM
8/30/02	4646	Track+	EditNew Item	Verfasser hat keinen Zugriff auf Punkt	implemented	Sinatra, George	123/04 9:21 PM
9/8/02	4701	Track+	Reports	No email sent to person editing an issue	opened	Sinatra, George	7/3/02 3:56 PM
	5221	Track+	EditNew Item	New filter criteria for 'creator' ('Verfasser')	opened	Sinatra, George	4/15/05 9:16 PM
	4846	Track+	EditNew Item	The "Synopsis" line is too short even for some val	analyzed	Sinatra, George	2/1/03 4:24 PM
11/7/02	5098	Track+	Documentation	Projektmanager in Manager/Bearbeiter unsichtbar	implemented	Sinatra, George	4/15/05 9:24 PM
	5394	Track+	Reports	Track+ verliert Rückwärtsreferenz	opened	Sinatra, George	4/15/05 9:25 PM
	5420	Track+	EditNew Item	Changehistory shows wrong User	opened	Sinatra, George	4/15/05 9:26 PM
	5205	Track+	EditNew Item	'Subsystem' added to subject of mail notification	analyzed	Sinatra, George	12/5/02 3:25 PM
	5441	Track+	Documentation	Track+ verliert Kommentare	opened	Sinatra, George	12/12/02 1:27 PM
	5444	Track+	Documentation	Ein kleiner Testeintrag	processing	Edel, Clemens	12/30/02 2:30 PM
Gesamtzahl Einträge:		14					

Figure 4.11: Customized PDF report

## 4.8 *Track+ Query Language (TQL)*

A well known and widely used query language for accessing database systems is the *Structured Query Language* (SQL). Via its **WHERE**-clause SQL offers a powerful and easy way to define criteria which limit the result set.

With the *Track+ Query Language* (TQL) the *Track+* system offers a facility similar to the SQL **WHERE**-clause to define which criteria a result set of artifacts has to fulfill. A TQL expression is made up of a number of simple comparison terms that are related by the boolean operators **AND** or **OR**. Example:

```
attribute == value AND attribute >= value
OR attribute <= value
```

The **AND** operator has a higher precedence than the **OR** operator. This behaviour can be overwritten with parentheses:

```
attribute == value AND (attribute >= value
OR attribute <= value)
```

Table 4.1 shows all possible comparison operators and their meaning.

### 4.8.1 *Sorting Criteria*

The sort order of the result set of a TQL query can be defined by a sort criterium. The sort criterium is represented by the attribute to be used for the sort. The sort criterium is defined by the key word **ORDER BY** followed by the attribute name and an optional

Operator	Description	Example
<code>==</code>	equal	<code>attribute == value</code>
<code>!=</code>	not equal	<code>attribute != value</code>
<code>&gt;=</code>	greater than or equal	<code>attribute &gt;= value</code>
<code>&gt;</code>	greater than	<code>attribute &gt; value</code>
<code>&lt;=</code>	less than or equal	<code>attribute &lt;= value</code>
<code>&lt;</code>	less than	<code>attribute &lt; value</code>

Table 4.1: Available comparison operators in TQL

keyword **ASC** for ascending sort order or **DESC** for descending sort order. The **ORDER\_BY** expression has to precede the TQL expression proper and has to be separated from it by a semicolon:

```
ORDER_BY state DESC, Project; state == open OR state == implemented
```

## 4.8.2 Date and Time Functions

Some attributes of an artifact represent dates or timestamps. To facilitate the selection of artifacts based on these attributes, operators are available in addition to the comparison operators shown in Table 4.1. These date related comparison **OLDER\_THAN** and **NEWER\_THAN** can be used with a variety of time units, as shown in Table 4.2.

Time Units	Example
HOURS	<code>NEWER_THAN 55 HOURS</code>
DAYS	<code>OLDER_THAN 34 DAYS</code>
WEEKS	<code>OLDER_THAN 21 WEEKS</code>
MONTHS	<code>NEWER_THAN 13 MONTHS</code>
YEARS	<code>OLDER_THAN 8 YEARS</code>

Table 4.2: Time units for operators **OLDER\_THAN** and **NEWER\_THAN**

For example, the following TQL expression would return all artifacts that have been created at most 8 weeks ago and not less than 13 days ago:

```
Created OLDER_THAN 13 DAYS AND Created NEWER_THAN 8 WEEKS
```

The operators **OLDER\_THAN** and **NEWER\_THAN** permit to define a point in time relative to a point in time of an attribute, as shown above.

To work with absolute points in time, the **DATE** function can be used. The **DATE** function requires as its parameter a date and time in a specified format, as shown in Table 4.3. The code **YYYY** stands for the year, **MM** for the month, **DD** for the day, **hh** for the hour, **mm** for the minutes, and finally **ss** for the seconds. The following example shows how to use the **DATE** function:

Date Format	Example
YYYY-MM-DD	2004-01-01
YYYY-MM-DD hh	2004-01-01 12
YYYY-MM-DD hh:mm	2004-01-01 12:12
YYYY-MM-DD hh:mm:ss	2004-01-01 12:12:12

Table 4.3: Date formats supported by the DATE-function

```
LastEdited > DATE(2004-03-10) AND
LastEdited < DATE(2004-06-14 24)
```

This limits the result set to those artifacts that have been modified between March 10, 2004, 00:00:00 and June 14, 2004, 24:00:00.

### 4.8.3 Attribute Alias Names and Values

Each artifact in **Track+** carries a number of attributes that may be used as selection criteria and for sorting. TQL recognizes the attributes given in Table 4.4 exactly as written there (watch upper and lower case letters).

Values in a TQL expression have to be in the right format. Some values are localized, and you have to enter the value that corresponds to your current locale and that you see in all other places of the application as well. For example, take the **State** attribute. One possible state may be “opened” in English, or “geöffnet” in German. If you see the user interface in German, your preferred locale as sent to the **Track+** server by your browser is German. Thus you need to enter “geöffnet” as value. If you see the Chinese user interface, you would have to enter the Chinese value for “opened” (good luck if you are not Chinese).

Date and time functions are not localized; you have to use the format given in Table 4.3.

### 4.8.4 More TQL Examples

Below some examples of TQL expressions are listed to demonstrate how to use the different features of this language.

- ▶ State != opened
- ▶ Lastedit < DATE(2003-01-01)
- ▶ Project NOT\_LIKE "Foo%"
- ▶ ORDER\_BY OriginatorFirstname DESC ; ItemNo > 0
- ▶ (Created OLDER\_THAN 20 DAYS) AND State == geöffnet
- ▶ Project == "Southpark" AND ManagerLastname == "Cartman"
- ▶ (ManagerLastname == "Cartman" OR ManagerLastname LIKE "Mc%")  
AND (Lastedit OLDER\_THAN 2 DAYS)

Field Name	Comments
Build	
ChangedByFirstname	
ChangedByLastname	
Class	use localized values
Created	format see Table 4.3
Description	
EndDate	format see Table 4.3
ItemNo	
Lastedit	format see Table 4.3
List	use localized values
ManagerFirstname	
ManagerLastname	
OriginatorFirstname	
OriginatorLastname	
Priority	use localized values
Project	
ReleaseNoticed	
ReleaseScheduled	
ResponsibleFirstname	
ResponsibleLastname	
Severity	use localized values
StartDate	format see Table 4.3
State	use localized values
Subsystem	
Synopsis	

Table 4.4: All fields supported by TQL

```

AND Project == "Southpark"
AND (Created > DATE(1999-12-24)
    AND (State == opened OR State == suspended))

```

## 4.9 Managing TQL Definitions

With TQL, users can run powerful queries against the artifact database. Some TQL expressions might be used over and over. For that reason it **Track+** provides the possibility to assign a name to a TQL query and store it in a repository.

For this purpose, **Track+** offers three repositories:

- ▶ a personal TQL repository
- ▶ a public TQL repository

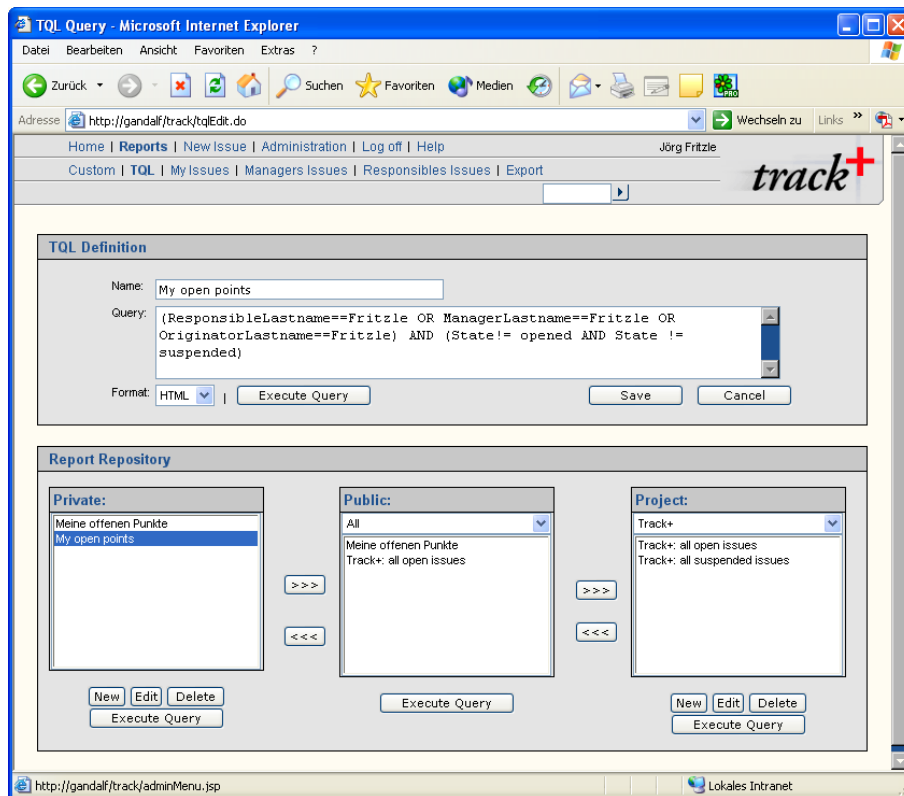


Figure 4.12: The TQL management page



- ▶ a project specific TQL repository

Each user of the **Track+** system has his own, personal TQL repository. Here he can store, modify and delete TQL expressions that he has developed for himself and found useful. Users can copy some of their own, personal TQL expressions into the public repository and make them available to all other users of the system, that share at least one project with them. This is a convenient way to interchange TQL expressions between users in order to later on personalize them. From the public repository, each user can copy queries into his own personal repository to modify them there. Each user can delete his own published queries from the public repository, but not those owned by anybody else. There is a filter available to filter for queries a specific user has published.

A user that has project administrator rights in at least one project may add, modify and delete TQL queries from the project repository. Members of a project may use project specific queries and can copy them into their own, personal repository for modification. Entries in the project specific repository list can be filtered by project name.

All access rights are handled such that only authorized persons can access, modify and delete TQL query definitions in the repositories.

To execute a query from a list, select it and push the “Execute Query” button. To modify one of your own queries, select it and hit the “Edit” button. Correspondingly, you can delete a query with the “Delete” button after selecting the entry to be deleted. To create a new TQL query definition, press the “New” button, define the TQL expression, assign a unique name to it, and press the “Save” button. To move queries between repositories use the buttons marked with “>>>” and “<<<”.

There is no limit on the number of queries a user can store in his repository. There is also no practical limit on the length of the TQL expression. However, the maximum length depends on your database server capabilities. For example, for Oracle 9i the maximum length would be 4000 characters.

## 4.10 Using the Command Line Client

**Track+** provides primarily a web based user interface. There is no need to install any client software, keeping the effort for system administration low. Nonetheless, **Track+** comes with a command line client that can be used to access some part of the systems functionality. Future versions will expand the capabilities of the client even more.

The **Track+** -client accesses the **Track+** + server by http-requests and analyzing the responses. It thus uses the same ports used by a web browser and does not require to reconfigure existing firewalls. It also observes all access control configurations, i.e. it is not possible to get any information via the client that would be impossible to get via the web interface of the server.

The **Track+** -client is written in Java and requires a suitable Java Run Time environment (JRE) or Java Software Development Kit (JDK) is installed on the computer where it is to run. The following listing demonstrates how to access the client, and the corresponding output of the clients `--help` command.

```

1 $ java -jar track-client.jar --help
2
3 usage: java -jar track-client.jar [options]
4
```

```

5 Options:
6  -ef,--export-format <format>  Set the export format.
7  -ph,--proxy-host <host>        Host of the HTTP proxy.
8  -pp,--proxy-port <port>        Port of the HTTP proxy.
9  -pri,--private-report <id>    Request a specific private report.
10 -pro,--project-report <id>     Request a specific project report.
11 -pub,--public-report <id>     Request a specific public report.
12 -bu,--base-url <url>          Location of the Track+ installation.
13 -h,--help                      Print this help message.
14 -p,--password <password>      Use this password for login.
15 -q,--query <tql>              TQL statement which should be
16                               executed.
17 -u,--user <username>          Use this user name for login.
18 -v,--verbose                   Turn on verbose output.
19
20
21 Get all open issues which are older than 50 days:
22
23 $ java -jar track-client.jar --user stan --password \
24     --base-url http://www.example.org:8080/track-tql/ \
25     --query 'State == opened AND Created OLDER_THAN 50 DAYS'
26
27
28 Get all available export formats:
29
30 $ java -jar track-client.jar --user cartman --password \
31     --base-url http://www.example.org:8080/track-tql/ \
32     --export-format list
33
34
35 Get all available private reports:
36
37 $ java -jar track-client.jar --user kyle --password \
38     --base-url http://www.example.org:8080/track-tql/ \
39     --private-report list
40
41
42 Request a public report:
43
44 $ java -jar track-client.jar --user kenny --password \
45     --base-url http://www.example.org:8080/track-tql/ \
46     --public-report 42 --export-format XML

```

Listing 4.1: The command line client help command

## 4.11 Diagrams

**Track+** offers the possibility to present the results of reports in form of pie or bar charts and Gantt diagrams. The charts are always based on the result set of a report. To generate this result set, use one of the techniques described above. Artifacts that are not part of

a result set are not considered in drawing the diagrams.

### 4.11.1 Bar Charts

The first type of bar charts is shown in fig. 4.13. It visualizes the number of artifacts vs. responsible person, separated by state. This gives a quick overview on the workload for each person.

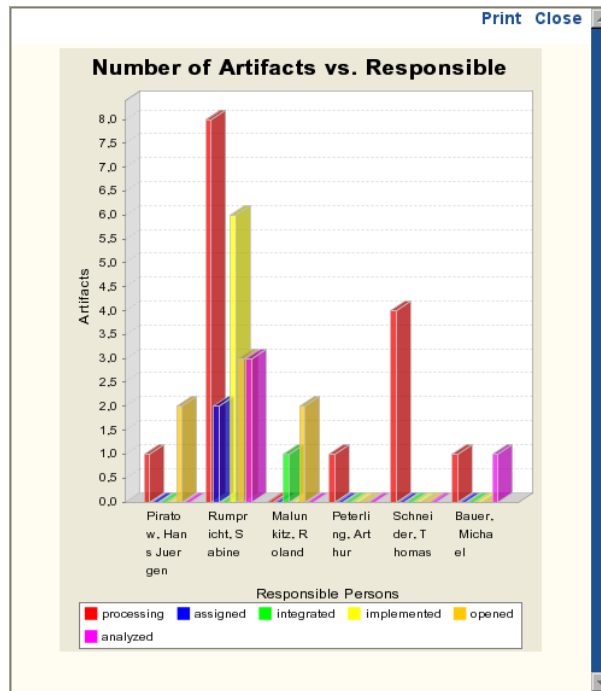


Figure 4.13: Bar chart: number of artifacts vs. responsible person

The second type of bar chart is shown in fig. 4.14. It gives a quick overview on total number of artifacts for several projects, separated by states. If closed artifacts are included, this chart gives a first impression on the maturity of a project.

The third type of bar chart is shown in fig. 4.15. It permits the visualization of the number of opened items over the last four weeks. This diagram can help in monitoring testing activities and, in conjunction with the amount of testing performed allows to draw conclusions on the maturity of a project.

### 4.11.2 Pie Charts

The pie chart shown in fig. 4.16 depicts for each user the number of artifacts for all projects included in this report. This way a manager can get a quick overview on the workload of each person included in the report.

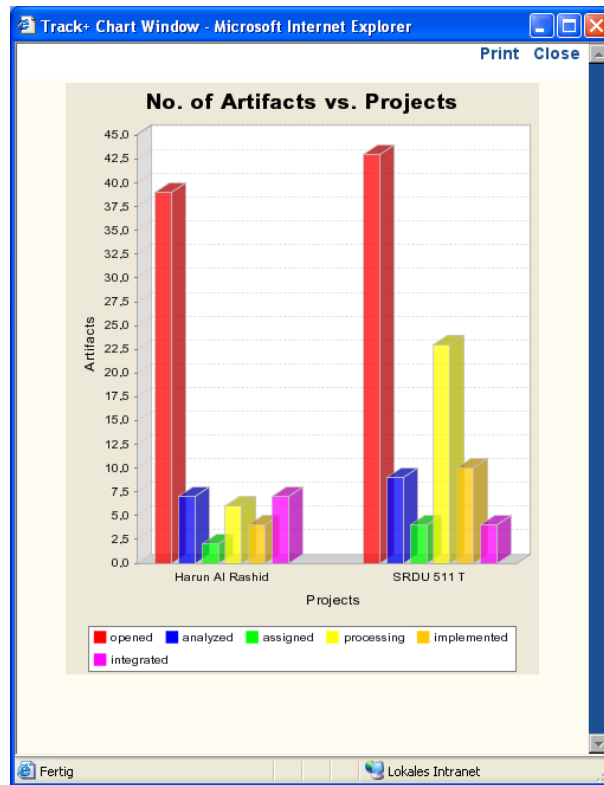


Figure 4.14: Bar chart: number of artifacts vs. responsible person

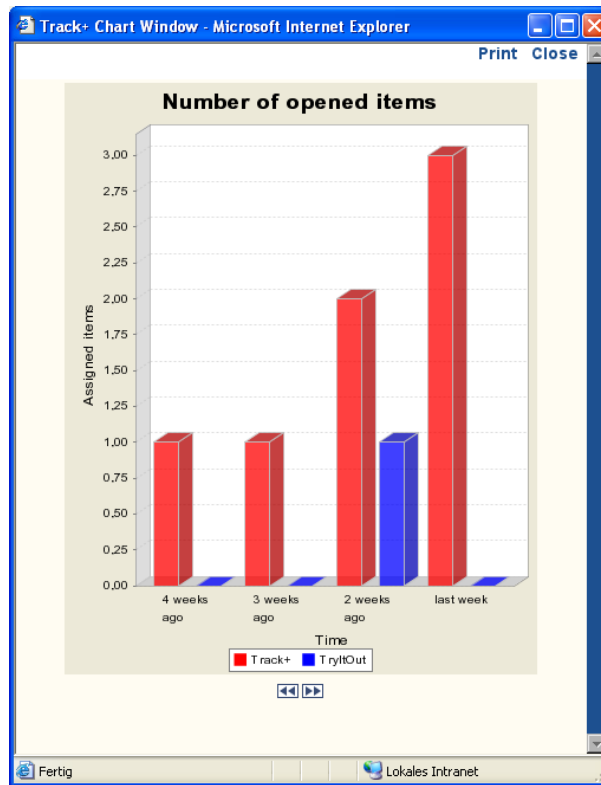


Figure 4.15: Bar chart: number of artifacts vs. responsible person

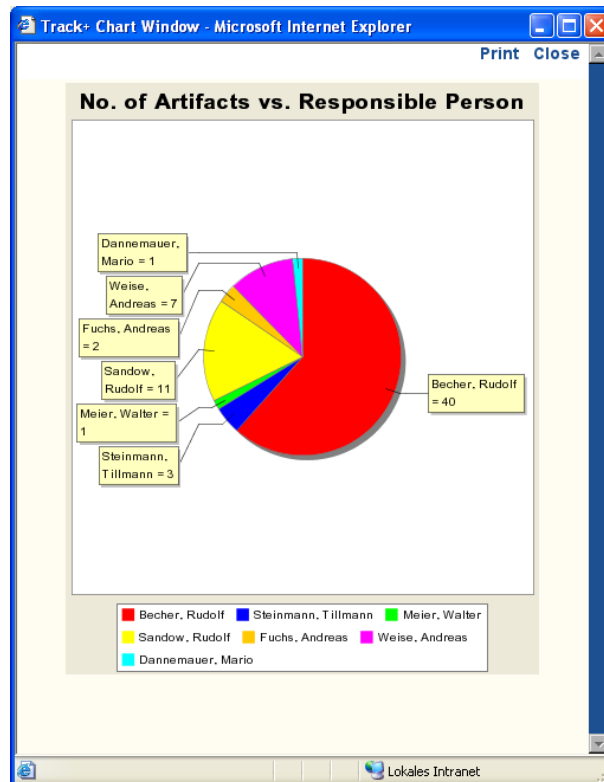


Figure 4.16: Pie chart: number of artifacts vs. responsible person

### 4.11.3 Gantt Diagrams

For artifacts including start and end dates it is possible to draw Gantt diagrams. Artifacts that have neither a start date nor a due date are disregarded for Gantt diagrams. Artifacts which have just a due date are displayed as milestones. Parent artifacts are drawn as super tasks.

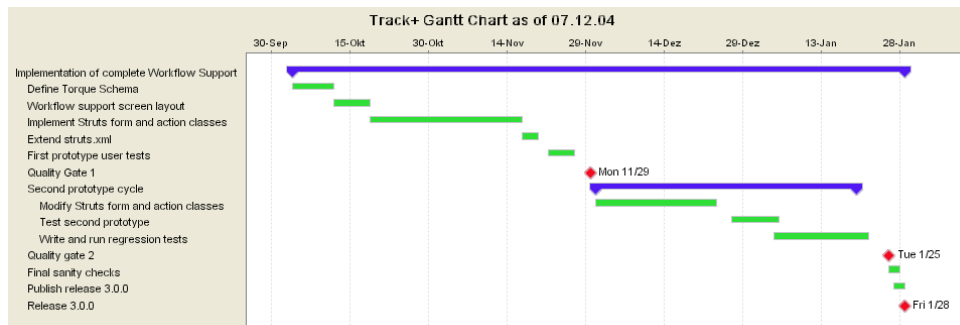


Figure 4.17: Sample Gantt diagram

## 4.12 Monitoring Changes

Usually, only the users involved in an artifact are being notified of changes to that artifact. However, everybody can register to be informed by e-mail of changes to classes of artifacts. Access restrictions are observed by the system, thus no information is sent if the user has not at least read rights for an artifact.

Via the "Administration → My Profile" menu one can access the page shown in Figure 4.18. In the example shown, the user is being notified via e-mail if changes to issues that belong to project TCProject0 and subsystems TCProjectCategory\_10, TCProjectCategory\_20, and TCProjectCategory\_30 are entering states "opened", "suspended" or "closed", or any changes that are made to such issues while they are in these three states. The user can modify this list any time at his own discretion.

## 4.13 Email Reminder

Artifacts can be assigned a start and a due date. The **Track+** system can automatically send reminders to users that tasks are about due or overdue. Each user can configure in his personal profile

- ▶ in what kind of role he wants to be reminded. For example, a user might care only about tasks due that he is responsible for, but not for those that he has originated or is manager of.
- ▶ the time buffer in days before the due date that the first reminder email is sent. A user may want to know a few days before the due date that something is coming up.

**Edit Registration for the Track+ Tracking System - Mozilla Firefox**

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe

http://gandalf/track/editRegistration.do?action=Edit

Home | Reports | New Issue | **Administration** | Log off | Help Jörg Fritze

Access Admin. | Project Admin. | Copy List | **My Profile**

**track+**

### Edit User Profile

<b>Username :</b> fch2bk	<b>No. of reminders per week :</b> 5
<b>Last Login :</b> 2/28/05 10:47 AM	<b>Days lead :</b> 10
<b>Password :</b> <input type="password"/>	<b>Remind me as originator :</b> <input checked="" type="checkbox"/>
<b>Repeat Password :</b> <input type="password"/>	<b>Remind me as manager :</b> <input checked="" type="checkbox"/>
<b>First Name :</b> Al	<b>Remind me as responsible :</b> <input checked="" type="checkbox"/>
<b>Last Name :</b> Rashid	<b>Minimum priority level :</b> occasionally
<b>Email Address :</b> harunAlRashid@imman.de	<b>Minimum severity level :</b> non-serious
<b>Phone :</b> 3409	<b>Preferred email type :</b> plain
<b>Department :</b> GF-KF/SQW1	<b>Preferred Language :</b> English
<b>Do not send emails to me :</b> <input type="checkbox"/>	

**Save** **Reset** **Cancel**

**Project :**  
Harun Al Rashid  
SDMX 155E16  
SRDU 511 T  
Track+  
TrytOut

**Subsystem :** All  
Analysis  
Business Logic  
Capacity Monitor  
Database  
Doc Architecture  
Doc Requirements  
GIS Viewer  
GUI  
Installation

**State :** All  
opened  
analyzed  
assigned  
suspended  
processing  
implemented  
assessed  
closed

**Add to Issue Watch List**

### My Issue Watch List

Delete	Project	Subsystem	State
	Track+	Documentation	opened, suspended, closed
	Track+	Installation	opened, suspended, closed

Fertig

Figure 4.18: Registering to monitor changes



- ▶ the frequency in times per week with which he wants to receive the reminder emails.
- ▶ the priority and severity level a task must at least have to be considered for a reminder email. This way trivial tasks can be excluded from the reminder facility.

## 4.14 *Linking to the System from a Website*

It is possible to create HTML links on any webpage pointing to the **Track+** system. Two types of links are supported:

1. a link to a specific artifact, referenced by its number. The format is  
**http://yourserver.com/track/printItem.do?key=<key>**  
You have to give the artifacts key number here instead of “key”.
2. a link to enter a new artifact for a specific project, referenced by project name. The format is  
**http://yourserver.com/track/editItem.do?action=create?project=<project name>**  
You have to supply the proper project name here.

If the user activating the link is already logged onto the system, she will directly be guided to the proper screen. If she is not logged on, she will be first asked to log on. After proper authorization she will be automatically guided to the proper screen.



# 5

## The **Track+** Default Process Model

### 5.1 Roles

The generic default roles in the context of the **Track+** system are described in Table 5.1. These roles are being mapped to the tangible roles in the **Track+** system. These roles are suggestions only, and may be modified depending on the needs of a specific organization. For instance, if **Track+** is being used to manage a sale-order-commissioning process, the roles of \$TESTER and \$INTEGRATOR may not be needed.

### 5.2 States and State Changes

States can be categorized into two classes: “open” and “closed”. Most states belong to the “open” category; the default configuration sets only state `closed` and `suspended` to category “closed”. This allows for easy queries such as: “give me all open issues”, meaning all those that are not in states `closed` or `suspended`.

Table 5.2 describes the states that are delivered with the **Track+** system as a default configuration. The class of each state is shown in brackets. These states may be renamed, extended or deleted to suit customer specific needs.

Currently there is no enforcement of valid state transitions as this has shown to be not too helpful in most organizations. Enforcement of state transitions will however be part of the next release. Certain state transitions can already be controlled via permissions assigned to roles, e.g. one can define roles that are not allowed to close an artifact.

All important state transitions (e.g. from an open state to a closed or suspended state) are being e-mailed to the people concerned anyway. Thus the risk of losing an issue is rather small.

<b>\$ORIGINATOR:</b>	the originator of an issue. This can be basically any person in the company. Customers are represented by one person in Product Management and one person in System Proving.
<b>\$CCB:</b>	the change control board. It meets on a regular basis and decides which changes have to be executed. During development and prior to integration, the CCB may be represented by an \$EXECUTOR or the \$PM.
<b>\$CM:</b>	the person responsible for configuration management in the project. It decides on versioning issues and proper archiving of published versions.
<b>\$PM:</b>	the project manager
<b>\$TESTER:</b>	the tester of a resolved issue. He ultimately decides if the issue solution is accepted.
<b>\$INTEGRATOR:</b>	the person that integrates subsystems or modules that have been changed and generates a single, testable release
<b>\$EXECUTOR:</b>	the person that executes the issue. It receives the change order by the \$CCB.
<b>\$MANAGER:</b>	the owner of a submitted issue. Typically, this will be the \$PM or the \$ORIGINATOR, or some person responsible for a sub system.
<b>\$RESPONSIBLE:</b>	the person responsible for carrying out actions related to this issue. \$RESPONSIBLE for example should take care that a fix is implemented in case of a bug report.

Table 5.1: Default roles in *Track+*

initial	the initial state of every issue if no other state is assigned. This means the issue has been filed. This state is optional and may be super-ceded by opened [open]
opened	this means the issue has been filed and the person or group responsible for it has been notified [open]
analyzed	the issue has been examined and it has been evaluated whether a solution has to be pursued. Useful for e.g. problem reports and requirement changes [open]
assigned	the issue has been assigned to a \$EXECUTOR [open]
processing	the issue is being processed by the \$EXECUTOR [open]
implemented	the solution has been implemented and is ready for integration by \$INTEGRATOR or \$TESTER [open]
integrated	a solution has been found and undergone a first assessment at the support site. It has been sent to the party who reported the problem; that party is validating the solution [open]
assessing	the solution is undergoing final verification before it is being closed [open]
closed	the solution has been confirmed by the party which reported it, or the \$TESTER [closed]
suspended	in some cases, it may be necessary to suspend work on an issue; in this case, its state changes to suspended rather than closed [closed]

Table 5.2: Default states in **Track+**

## 5.3 Attributes

Each issue carries attributes as shown in Table 5.3.

<i>Class</i>	should be custom configured, e.g. {System, Software, Hardware, User Documentation, Electrical Installation, Roof, etc.}. The available classes are project dependent.
<i>State</i>	as described in the previous chapter
<i>Priority</i>	can be any of {immediate, occasionally, soon} and describes how fast a solution is needed
<i>Severity</i>	can be any of {non-critical, serious, critical} and describes how seriously the user of a system is effected as long as the issue isn't being completed
<i>List</i>	describes the type of list, for example Problem Report, Action Item, Feature Request, Risk, Work Package etc..
<i>Originator</i>	the \$ORIGINATOR of the issue or his representative as described previously in the role definitions, usually the person that created this artifact
<i>Manager</i>	the manager or owner of the issue; can be the \$PM, or the person responsible for the subsystem, or the person responsible for the issue
<i>Responsible</i>	the person currently responsible for the issue (\$RESPONSIBLE), as described previously in the role definitions
<i>Create Date</i>	the date the issue was created.
<i>Last Modified</i>	the date the issue was last modified. State change modifications dates and due date changes are recorded separately. Textual modifications are traced but the changes themselves are not being recorded. Attribute changes are being traced and recorded, including date of change and person who carried out the modification.
<i>Project</i>	the project to whom this issue belongs. A single issue has to belong to exactly one project.
<i>Subsystem</i>	the subsystem within a project to whom this issue belongs. A single issue has to belong to exactly one subsystem.
<i>Release Noticed</i>	the release of a project to whom this artifact pertains to. A single artifact may be related to two releases. If for instance a problem report covers more than two releases it has to be copied. This facilitates release independent bug fixing (e.g. a bug is fixed in a later release but remains in a previous release)

<i>Release uled</i>	<i>Sched-</i>	the release where an artifact is supposed to be closed. The possible distinction between “release noticed” and “release scheduled” permits to clearly A single issue has to belong to exactly one release. If for instance a problem report covers more than one release it has to be copied to the several releases. This facilitates release independent bug fixing (e.g. a bug is fixed in a later release but remains in a previous release)
<i>Build</i>		some string with up to 25 characters representing the build number.

Table 5.3: Default attributes in **Track+**

## 5.4 List Types

Besides keeping track of start and due dates, one of the strengths of **Track+** is the ability to manage several lists for a project. Lists can be categorized according to their type. Table 5.4 gives a few examples.

<i>Problem Report</i>	Somebody has a problem with something, rather unspecified. May be a know how problem, a user interface problem, or a bug.
<i>Requirement Change</i>	The agreed on requirements need to be changed to achieve the desired functionality.
<i>Requirement</i>	A list with requirements. This permits for requirement engineering and tracing. Since attachments can be added to each artifact, you can add figures, source code, or other documents to a requirement.
<i>Implementation Error</i>	There was some problem found in the implementation, i.e. the requirements are correct, but the implementation is not.
<i>Work Package</i>	In a well managed project there exists a work breakdown structure with many work packages. It is very tedious for a project manager to keep track of the status of such work packages. Using this list and having each \$RESPONSIBLE take care of the status of her work package the project manager can without a lot of effort obtain the actual status of his project and look for instance for late packages.
<i>Action Item</i>	There may be use for a rather unspecific list of action items at the project management level.
<i>Milestone</i>	Keeping a list with important milestones could be the shortcut version of keeping a list with work packages.
<i>Risk</i>	Project management without a current list of relevant risks is not really complete.

<i>Release Notes</i>	Once a requirement has been implemented, or a bug has been fixed, you can copy it from the bug list to the release notes list. Once a release is assembled, you can extract a report filtering for the release notes list. This way you can make sure you have all changes included in your release notes.
<i>Templates</i>	This list permits to define templates that may be copied to new lists, changing certain parameters in the copy process. This is very helpful for defining assessment plans and quality gate check lists, that can be reused and copied to a protocol list. The protocol list may be worked off for each project, release, and build.

Table 5.4: Default list types in **Track+**

Note that for the different list types, not every state out of the set of possible states may make sense. It is up to the definition of a specific project how to use the possibilities the **Track+** system offers. Starting with release 3.0.0, it is possible to limit the set of states for each project type and list type. For example, for milestone lists two states might suffice: opened and closed. To activate this feature and have a different setup than the default one, you have to delete the respective entries from table TPSTATE using your favourite database management tool.

## 5.5 Email Notifications

The **Track+** application sends email notifications when state changes occur. Table 5.5 lists the default notifications for state changes. In addition, \$MANAGER and \$RE-

From state	To state	notified
any	closed	\$ORIGINATOR, \$MANAGER, \$RESPONSIBLE
any	assessed	\$ORIGINATOR, \$MANAGER
any	suspended	\$ORIGINATOR, \$MANAGER, \$RESPONSIBLE
any	any other except closed or suspended	\$MANAGER, \$RESPONSIBLE

Table 5.5: Email notifications

SPONSIBLE are being informed in case

- ▶ \$MANAGER
- ▶ \$RESPONSIBLE
- ▶ any of project, subsystem, release, class, category, and priority are changed.

In case \$MANAGER or \$RESPONSIBLE are changed, the old entry as well as the new entry are being informed.



\$MANAGER and \$RESPONSIBLE are being informed in case the start or due date has been changed.

All email notifications contain a direct link to the issue. If the user has logged into the system during the past seven days (configurable by the system administrator), she can directly access the respective page if cookies are enabled in the users browser.

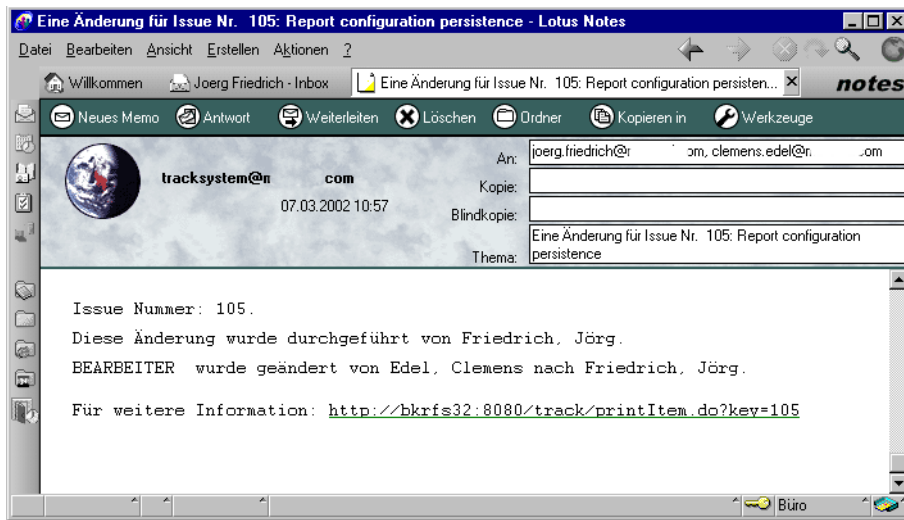


Figure 5.1: Sample e-mail notification (localized, in German)

Each user can switch off all email notifications to himself entirely in his personal profile. However, for obvious reasons this does not concern the password reminder facility.



# 6

## *Applying **Track+** to your Projects*

### *6.1 The Action Item List*

The action item list is the most general in the **Track+** system. It was designed and developed to help facilitate communication within groups as small as members of a project or as large as members of an entire organization. In today's world, action items levied against members of an organization are created from several different sources through various avenues (either by email, voice mail, in meetings, or by word of mouth) resulting in actions not being completed because either the action was forgotten, misplaced, or miscommunicated. The intent of the action item list is to:

- ▶ centralize and organize all action items for a particular organization or project
- ▶ make the system available through the Internet, and notify individuals via email when an action is assigned to them or is nearing or has surpassed the due date.
- ▶ allow for a concise and up to date overview on a projects status

Not all states the **Track+** system has to offer are helpful when dealing with action items. Fig. 6.1 shows the state subset that could be used for action items. An action item is created and initially set to state "opened". Action items should always carry a due date and a person to which they are assigned (\$RESPONSIBLE). Once work on an action item has started the state is being set to "processing" by \$RESPONSIBLE. This permits other people to see that something is actually happening on this thread. The thread can be closed by \$RESPONSIBLE or a manager once all tasks associated with this action have been carried out.

### *6.2 Milestone Tracking*

In a well managed project, there will be lots of "binary" milestones, those that are either passed or not passed. It is an often observed project sickness that milestones are not

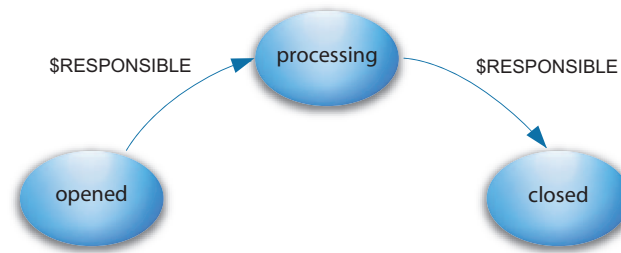


Figure 6.1: States and state transitions for action items

defined in a “binary” fashion. You will hear people coming up with terms like “conditionally passed”, “90% done”, “almost there” and so on. If you hear such phrases you know you are in trouble.

It is in the nature of a milestone that you have either passed it or not. It may take some effort to define project milestones, but it pays well. The best milestones are connected to assessable, 100% complete results. For example, a good milestone would be “Customer Requirements reviewed with customer and frozen”. A bad example would be “Customer Requirements 50% finished”. A milestone has to have a due date associated with



Figure 6.2: States and state transitions for milestones

it. It can be either “opened” or “closed”. The state transition from “opened” to “closed” marks the passing of the milestone.

It is quite common that the milestone passing dates, as envisioned in the first project plan, have to be moved. In a badly managed project such movements become visible only at the milestone due date, or worse, at the next quality gate. With **Track+** it is no problem to move a due date, but the change is recorded and you can watch how a milestone slips. Looking at the database, you can draw reports of milestone slippage in your project, telling you quite clearly where you stand.

## 6.3 Risk Management

Planning a project and then executing the plan is based on estimates. Unless you have a magic crystal ball you are making assumptions that could or could not be true. One definition for an estimate calls it a prediction that is equally likely to be above or below the actual result[8]. Estimate uncertainty occurs because an estimate is a probabilistic assessment of a future condition[9].

The many events that could lead to your estimate being too optimistic I call risks. There could be a risk that you won't be able to meet the schedule, or the budget, or the expected quality level. You may not be able to get the right people at the required time. A supplier you are counting on falls short to deliver on his contract. You may depend on weather conditions, and they turn out to not being in your favor. It's up to your imagination what could happen.

Managing a project to a good extent means managing risks. It is very helpful to always have a list with the ten most prevalent risks in your project. You should have a strategy available on how to react if any of these risks materializes.

The **Track+** system helps you to manage a list of risks. There may be more than just ten



Figure 6.3: States and state transitions for risks

risks. Enter all you know, and assign them to the different priority and severity levels. This permits to draw reports with those risks that are most likely to occur, or that have the greatest impact. Use the priority to describe the probability of occurrence, and the severity to describe the possible impact. The most critical risks are those that have the highest probability of occurrence and the largest impact.

There are only two states for a risk: it is either "opened" or "closed". State "closed" means the probability of this risk occurring in the future is zero. Reasons to close a risk could be that there is no future for your project, because it has been closed, or that the risk already occurred, or that you absolutely can rule out that the risk could ever come up again.

## 6.4 The Issue List

In the software business this is also called the "bug list". Unlike milestones and action items, issues are usually not planned, but they will occur. For large projects, and particularly in software projects, from the beginning to the end there will be many issues. From my experience, for an embedded system with about 100.000 non commented lines of code we had to handle about 2500 issues before we were able to ship the system. Since it is close to impossible to control such an amount of issues without any tool support, bug tracking systems have been on the market for quite some time.

The **Track+** system is well suited to act as an issue or bug tracker. Compared to other systems it supports well large organizations with many projects, due to its technology and access control features. Except for access control the **Track+** system imposes very few restrictions on its users. This leads to fast adoption and intense usage even without managerial pressure.

The life of an issue starts with state "opened". Anybody with appropriate permission in

a specific project can create a new issue. There is a default person assigned as manager of an issue who is responsible to evaluate and possibly assign it to somebody for further investigation or resolution. If there is a time gap between the decision to further pursue

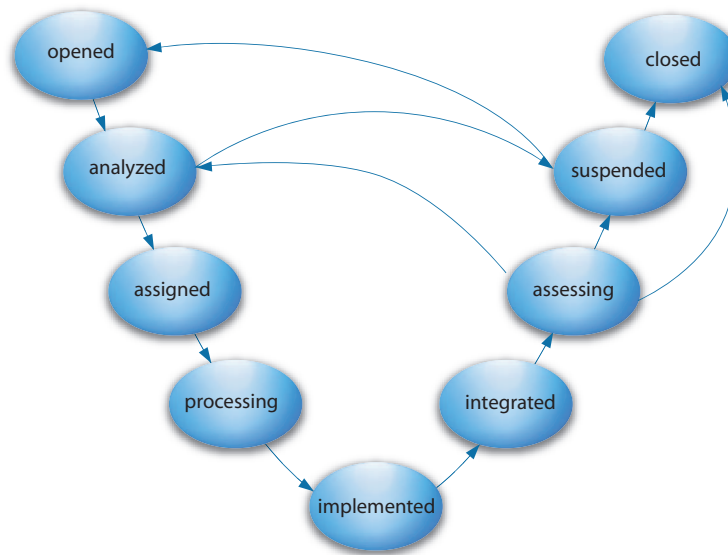


Figure 6.4: States and state transitions for issues

this issue and the time it can be assigned to somebody for resolution it is possible to mark an issue as “analyzed”. If the analysis shows that resolution of an issue should be deferred to some later time the issue can be set to “suspended”. From this state it can always be reopened, or directly be assigned to some person for further processing.

To permit management some insight when an issue is actually being worked upon by the person responsible for it, state “processing” is available. This state marks the period during which resolution of an issue is pursued. While most other states describe the end of a period, this state describes the process. The end of this period is marked by state “implemented”, which means that resolution has been completed, but no integration and tests have been carried out.

The integrators and testers can use states “integrated”, “assessing”, and “closed”. If the integration and assessment process reveals no further problems with the implemented issue resolution, the issue can be closed. In any other case, the issue is being set back to “assigned”.

As with state “processing”, state “assessing” defines a process rather than the end of a process. This again is to enable management visibility when the assessment process, which could take some time, has actually been started.

## 6.5 Personal To Do List

It is possible to configure **Track+** such that each user has his personal to do list that is not visible to others. There are two basic ways how to achieve this.

On solution is to define a new project, e.g. named “Personal”, and assign each user the “Extern” role. This permits her to create, modify and close artifacts that only she can see.

The second approach is to define a new list type “personal list” and define via table **TROLELISTTYPE** that only role “Extern” can access this list.

## 6.6 Integration with CVS

The Concurrent Version System (CVS) is a free and widely used professional source code management system. **Track+** can be used to relate code changes in a CVS repository to any artifact by means of the **viewCVS** utility. The link can be to a file in general, or to a specific version of that file. To make the link, the project administrator has to

track/src/com/aurel/track/attachment

Current directory: [\[trackplus\]](#) / [track](#) / [src](#) / [com](#) / [aurel](#) / [track](#) / [attachment](#)  
Files shown: 10

File	Rev	Age	Author	Last log entry
<a href="#">AttachFileBase.java</a>	<a href="#">1.1</a>	8 months	friedj	First checkin for post 2.2.0 release
<a href="#">AttachFileDb.java</a>	<a href="#">1.3</a>	6 weeks	langh	attachment handling (deleting/download) improved
<a href="#">AttachFileSimple.java</a>	<a href="#">1.1</a>	8 months	friedj	First checkin for post 2.2.0 release
<a href="#">AttachLinkTag.java</a>	<a href="#">1.7</a>	11 days	langh	URLEncode for attachment file-names fix for bug#1173741
<a href="#">AttachmentException.java</a>	<a href="#">1.2</a>	8 months	friedj	First checkin for post 2.2.0 release
<a href="#">AttachmentForm.java</a>	<a href="#">1.2</a>	8 months	friedj	First checkin for post 2.2.0 release
<a href="#">AttachmentPersist.java</a>	<a href="#">1.5</a>	6 weeks	langh	attachment handling (deleting/download) improved
<a href="#">DownloadAttachmentAction.java</a>	<a href="#">1.6</a>	6 weeks	langh	attachment handling (deleting/download) improved
<a href="#">EditAttachmentAction.java</a>	<a href="#">1.2</a>	8 months	friedj	First checkin for post 2.2.0 release
<a href="#">SaveAttachmentAction.java</a>	<a href="#">1.6</a>	6 weeks	langh	attachment handling (deleting/download) improved

Show files using tag:

[CVS Administrator](#) Powered by [ViewCVS 0.9.2](#)

Figure 6.5: The CVS directory view

define the URL of your viewCVS server and the project name in the CVS repository. This permits to easily move entire projects to a new server without having to change the **Track+** database. In the description section of the artifact you can use the version control (vc) tag to describe the file or module you want to point to. If you click on the link rendered in the overview page you will directly see the module or file in a new browser window. To get an overview what has been changed in a certain period (time-line), you can use the **Track+** reporting facilities and define a filter on the LastEdit and State fields, and extract all files or modules using the reportClient. To further increase tightness of integration between **Track+** and CVS, in one of the next versions of **Track+**

**CVS log for**  
**track/src/com/aurel/track/attachment/AttachLinkTag.java**  
 Up to [\[trackplus\]](#) / [\[track\]](#) / [\[src\]](#) / [\[com\]](#) / [\[aurel\]](#) / [\[track\]](#) / [\[attachment\]](#)  
[Request diff between arbitrary revisions](#)

---

Default branch: MAIN  
 Bookmark a link to: [HEAD](#) / [\(download\)](#)

---

Revision [1.7](#) / [\(view\)](#) - [annotate](#) - [\[select for diffs\]](#), *Thu Mar 31 18:00:59 2005 UTC* (11 days, 21 hours ago) by *langh*  
 Branch: [MAIN](#)  
 CVS Tags: [HEAD](#)  
 Changes since 1.6: +24 -10 lines  
 Diff to [previous 1.6](#)

URLEncode for attachment file-names  
 fix for bug#1173741

---

Revision [1.6](#) / [\(view\)](#) - [annotate](#) - [\[select for diffs\]](#), *Sun Feb 27 16:08:17 2005 UTC* (6 weeks, 1 day ago) by *langh*  
 Branch: [MAIN](#)  
 Changes since 1.5: +5 -2 lines  
 Diff to [previous 1.5](#)

attachment handling (deleting/download) improved

---

Revision [1.5](#) / [\(view\)](#) - [annotate](#) - [\[select for diffs\]](#), *Mon Oct 18 18:06:56 2004 UTC* (5 months, 3 weeks ago) by *langh*  
 Branch: [MAIN](#)  
 CVS Tags: [REL301](#), [REL300RC2](#), [REL300RC1](#), [REL300](#)  
 Changes since 1.4: +8 -2 lines  
 Diff to [previous 1.4](#)

bytes and kbytes

---

Revision [1.4](#) / [\(view\)](#) - [annotate](#) - [\[select for diffs\]](#), *Fri Oct 8 18:38:06 2004 UTC* (6 months ago) by *langh*

Figure 6.6: The CVS file overview

[Return to AttachLinkTag.java CVS log](#) Up to [\[trackplus\]](#) / [\[track\]](#) / [\[src\]](#) / [\[com\]](#) / [\[aurel\]](#) / [\[track\]](#) / [\[attachment\]](#)

**Diff for /track/src/com/aurel/track/attachment/AttachLinkTag.java between version 1.6 and 1.7**

version 1.6, 2005/02/27 16:08:17	version 1.7, 2005/03/31 19:00:59
<b>Line 38</b>	<b>Line 38</b>
package com.aurel.track.attachment;	package com.aurel.track.attachment;
import java.util.List;	import java.io.UnsupportedEncodingException; import java.net.URLEncoder; import java.util.List;
import javax.servlet.http.HttpServletRequest;	import javax.servlet.http.HttpServletRequest;
<b>Line 125</b>	<b>Line 127</b>
<pre> /**  * private String getDownloadUrl(AttachFileBase attachFile)  * {  *     return DownloadAttachBaseURL +  *  *     "?attachKey=" + attachFile.getAttachKey() +  *     "&amp;itemKey=" + attachFile.getItemKey() +  *     "&amp;attachName=" + attachFile.getAttachName();  * }  */ </pre>	<pre> /**  * private String getDownloadUrl(AttachFileBase attachFile)  * {  *     String url = "";  *     try {  *         url = DownloadAttachBaseURL +  *  *         "?attachKey=" + attachFile.getAttachKey() +  *         "&amp;itemKey=" + attachFile.getItemKey() +  *         "&amp;attachName=" +  *         URLEncoder.encode(attachFile.getAttachName(), "UTF-8");  *     } catch (UnsupportedEncodingException e) {  *         ;  *     }  *     return url;  * }  */ </pre>
<b>Line 138</b>	<b>Line 146</b>
<pre> /**  * private String getDeleteUrl(AttachFileBase attachFile)  * { </pre>	<pre> /**  * private String getDeleteUrl(AttachFileBase attachFile)  * { </pre>

Figure 6.7: The CVS difference between versions



```

*/
package com.aurel.track.attachment;

import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.tagext.TagSupport;

import org.apache.struts.taglib.TagUtils;

/**
 * Generate a URL-encoded hyperlink to the specified URI, with
 * associated query parameters selecting a specified property.
 */
public class AttachLinkTag extends TagSupport {

    private static char AMPERSAND = '&';
    private static char QUESTION = '?';
    private static char TAG_FINISH = '>';

    private static String QUOTE = "\"";
    private static String KEY = "key=";
    private static String EMPTY = "";
    private static String NESP = "&nbsp;";
    private static String CLASS = "class=";
    private static String HREF = "href=";

    private static String TABLE_START = "<table border=" + "0" + " cellpadding=" + "0" + " cellspacing=" + "0" + ">";
    private static String TABLE_FINISH = "</table>";
    private static String TR_START = "<tr>";
    private static String TR_FINISH = "</tr>";
    private static String TD_START = "<td>";
    private static String TD_FINISH = "</td>";

    private static String A_START = "<a ";
    private static String A_FINISH = "</a>";

```

Figure 6.8: The CVS file view

there will be an interface for CVS clients to automatically enter all CVS information into the description area of a state change and automatically induce a state change.

## 6.7 Generating Release Notes

With **Track+** it is easy to generate release notes, i.e. to get a list with all issues that had been closed for a specific release. For this purpose, use TQL and export the query result to XML or a CSV file for further processing. The following TQL expression can serve as an example:

```
State == closed AND ReleaseScheduled == 3.0.1
```

Of course, further restrictions can be applied to filter out artifacts that should not occur in the release notes. Using this process it can be made sure that no changes are missed.





## System Limits

The following limits apply to the current version of the *Track+* system.

Object	Limit
Size of issue description	32.000 characters, depends on database system. E.g. Oracle allows only 4000 characters.
Size of issue synopsis	80 characters (brevity is very good, where we are or are not understood)
Size of state change description	10.000 characters, depends on database system.
Size of login name	10 characters
Size of first name	25 characters
Size of last name	25 characters
Size of e-mail address	60 characters
Session timeout	60 minutes <sup>1</sup>
Size of attachment	4 MByte

Table A.2: Limits of the *Track+* system

---

<sup>1</sup>The session timeout can be configured in the `WEB-INF/web.xml` file. If there has been no browser request for that long the session is terminated by the server and one has to log in again.



# B

## Frequently Asked Questions

### B.1 Installation and Administration

**Q:** *Why is the installation procedure so complicated?*

**A:** There are two reasons why we can't offer a simple one click solution yet. First, the **Track+** system needs some infrastructure to be able to run, namely a fully featured database management system and a servlet container (something like a web server). Most of the "complicated" steps in the installation procedure refer to getting these two items up and running. Secondly, it is not yet standard that systems have the required Java run time environment installed. Thus we have to make sure that it is available.

**Q:** *Why isn't everything packed into one rpm file for installation?*

**A:** Most of the installation deals with installing the prerequisite package for the **Track+** system, such as the Java Developers Kit, the Tomcat servlet container, and an appropriate SQL database management system. Some of these packages come with support for installation, and some don't. The **Track+** application proper needs very little configuration, and an install script would not make much of a difference in terms of ease of installation.

**Q:** *I can't get Tomcat started with the **Track+** system. What could be the reason?*

**A:** There are two main reasons why the system might not come up or work properly:

1. The database server and interserver JDBC adaptor are not properly installed or the WEB-INF/Torque.properties does not point to the right location of the database file. You should get a message giving you a hint towards database problems when trying to log on.

2. There is a port conflict between Tomcat and an Apache server running JServ on the same port. You need to disable the JServ module in Apache if you want to run Tomcat (on the same port). This does not only concern the working port (e.g. 8080) but also the administration ports (e.g. 8007 and 8008). Any serious installation integrates Apache with Tomcat, disabling JServ inside Apache. Even though it is not difficult to do, it would be beyond the scope of this manual to go into details.

**Q:** *Many service providers offer servers with php installed, and almost nobody supports Tomcat or some other servlet container. Why didn't you write the system with php or some other scripting language?*

**A:** It is definitely right that there aren't many providers supporting servlet containers. There are systems out there similar to **Track+** written in php or Python. Some of them mix the presentation layer with the logic, so you have to be at the same time a good page designer and php programmer, which doesn't always match. You may have a lot of php in your page design, or a lot of page design in your php. A good framework like the J2EE and Struts greatly enhances the separation of the various pieces of such a system, thereby improving maintainability and scalability.

**Q:** *The mail notification feature doesn't seem to work. What could be the reason?*

**A:** The **Track+** system needs access to an SMTP server defined in the server configuration menu. You can even check with telnet if your SMTP server responds under the address you think it should. Some servers require that you have logged in and tried to retrieve some mail before allowing you to use the SMTP server (many ISPs handle it that way). You can configure **Track+** to provide POP3-before-SMTP authentication or SMTP authentication.

**Q:** *I want to run the **Track+** system on my Windows box just to try it out, but I can't log on since the password isn't being sent to me. What should I do?*

**A:** If you do not have access to an SMTP server, you can retrieve the password by enabling debug mode for the "JavaMailBean" class and looking at the application server console output. The mails are written to that console, and you can get the password from there. Look at the section of the file where logging is configured. There is a line commented out with a description regarding this issue. You have to restart the application (or your application server) after the change to make this effective.

**Q:** *When I look at my Linux processes I see a large number of interserver, ibserver, and Tomcat processes. Isn't that eating up my entire system memory and performance?*

**A:** Under Linux, threads are shown as processes when using the ps command. There is a thread for each user, and Tomcat and Firebird open a certain number of threads even if no user is actually logged on. However, all these threads occupy the same memory space and there shouldn't be any performance issue because of that.

- Q:** *I have problems getting up **Track+** with Firebird or Interbase using JDK 1.4. What's wrong?*
- A:** With JDK 1.4 you may have to add `-noverify` to the Java process running Tomcat to prevent the Interbase Interclient software from throwing `java.lang.Verify` errors.
- Q:** *I have problems getting the Interclient/Interserver software to run. I am getting a message saying that the InterClient and InterServer use incompatible client/server protocol versions. What's going on?*
- A:** Try `chmod+x` to the `/opt/interbase/interclient/interserver` file.
- Q:** *What are the most common problems when trying to install **Track+** ?*
- A:** The single most common problem is to get the Interclient/Interserver software to run when using the Interbase or Firebird database. This is the JDBC adaptor between the Firebird RDBMS and the **Track+** application. Usually, once this is successfully tested as described in this manual, the application will be running with no problems.
- Q:** *I can't get **Track+** to connect to my MySQL database. What is wrong?*
- A:** There may be an issue with your MySQL privileges.
- On some operating systems, there seems to be a weird interaction between the JVM, DNS resolution and the MySQL driver where a JDBC URL pointing to "localhost" will resolve as "localhost.localdomain" and will prevent the connection to MySQL from authenticating correctly because most people configure MySQL for "localhost". One way to get around this is to use IP addresses in both the MySQL ACL as well as in the JDBC URL.
- There have been reports that using "127.0.0.1" instead of "localhost" resolved a "Server configuration denies access to data source" connection issue.
- Q:** *In my e-mails I get as hyperlinks a link to localhost instead to the correct address of my application server. Where can I configure this?*
- A:** In order to minimize the amount of configuring the system installer has to do **Track+** automatically derives it's own URL from the first call it gets after the application server has been started. Thus, you should not test your installation with `localhost` but rather with the real network name or IP address.

## B.2 Usage

- Q:** *Does the **Track+** system use a database?*
- A:** Yes, the **Track+** system uses a relational database management system. Supported are currently Interbase 6.0, Firebird, and MySQL. All database management systems supported by the Torque persistence layer <http://db.apache.org>

should work. These are currently to just name a few Postgres, Oracle, SAPDB, Sybase, and MS SQL Server. Others such as DB2 have been implemented but aren't verified yet.

**Q:** *Are there any license costs associated with the **Track+** system?*

**A:** The **Track+** system is being developed as an Open Source project on Sourceforge (<http://trackplus.sourceforge.net>) and governed by its own license. If you use **Track+** commercially with more than 10 users you need to obtain a commercial license. You may modify **Track+** for your own purposes, but you may not publish the results outside of your company or organizational unit.

The required infrastructure can be configured such that there are no further license costs associated with it. Free servlet containers or application servers are for example Tomcat, Jetty, or JBoss. Free full size database systems include Firebird and Postgres. You can use MySQL, but it requires similar to **Track+** a commercial license if you use it commercially and not in an Open Source environment.

**Q:** *What is needed to set up a working **Track+** system and how difficult is it?*

**A:** Basically, to get a running **Track+** system you have four prerequisites:

1. You need a running Unix or Windows box where Tomcat and/or Apache runs on. The tests have been carried out on S.u.S.E. Linux 7.2 and 8.0 and Windows XP, but basically any Unix or Windows supporting Java JDK 1.3 or up should do.
2. You need a running database server which can be the same box as mentioned above. For some popular RDBMS such as MySQL or Firebird, the software you can download from the Internet as described in this documentation.
3. You have to make sure that a Java JDK 1.4 or up is installed on the server. This software can be downloaded from the Internet as described in this documentation.
4. You have to have the appropriate rights on your machines to install software in the desired directories. Since running the Tomcat container and the RDBMS should commence when the system is started, this may require root access rights to install the respective boot scripts and add a database administrator to the system user database.

Is it difficult to install the system? Well, you may not want to give it to your secretary, but anyone who has a little knowledge on getting a Windows or Unix system up should have no trouble to get **Track+** going without much hassle. The most difficult part is getting the database and application server to run. The installation of the **Track+** application itself is a one step procedure.

**Q:** *Was it necessary to write yet another bug tracking tool?*

**A:** The **Track+** system is not a bug tracking tool but a tool to keep track of all sorts of lists that have to be visible to more than one person. For simple personal lists



you may want to stick with Excel or Lotus To Do Lists. If you just want to do bug tracking for a few projects and small groups you may be happy with GNATS or Bugzilla.

Of course, there may be some tools out there that could do the same or even much more, so let us know if you find one that is

- ▶ free for up to 10 people
- ▶ inexpensive for large installations
- ▶ easy to use
- ▶ able to handle 1000 projects
- ▶ and slick ;-)

**Q:** *Can the **Track+** system be localized?*

**A:** The **Track+** system can easily be localized by editing a few text files, one for the e-mail messages, two for the drop-down boxes, and one for the rest of the texts. It comes already with German, English, and Norwegian localization and adapts to the users browser. Thus, the same server may serve in German or English or Norwegian depending on the users browser settings. Anybody willing to support other languages (Italian, Spanish, French, Polish, Chinese, Portuguese) is welcome to contribute.

**Q:** *I have tried to register but get a message that my email address isn't valid. What am I doing wrong?*

**A:** Since the **Track+** system is designed such that user administration is carried out mostly by the users themselves it is possible to restrict access to the system to certain e-mail domains. This prevents people from registering that have no authorized email account inside controlled domains and try to login from uncontrolled domains such as hotmail.com. Your system administrator sets up these restrictions.

**Q:** *On my Unix workstation, the letters of the texts are very small and basically not readable. Can I increase the font size?*

**A:** **Track+** makes use of style sheets and the smallest font is set to 11 pt. Using points should work on any system, and if you get very small letters it is most likely that on your system and browser the font mapping is not configured in the right way. You may have to modify your X11 resources file and add something like

```
Netscape*documentFonts.xResolution*iso-8859-1: 100
Netscape*documentFonts.yResolution*iso-8859-1: 100
```

as an example if you are using a Netscape browser.

**Q:** *I tried to upload an attachment, but it didn't work. What's wrong?*

- A: **Track+** has a built in file size limit for the uploaded files of 4 MByte to protect the server from malicious or inadvertent overloading. Thus, if your file is larger than that the upload won't be carried out.
- Q: *How can I change the locale of the **Track+** menus and resources? Do I have to change something in a configuration file?*
- A: **Track+** automatically adapts to the users browsers preferred language setting. You can set the preferred language for example in Internet Explorer via the Extras → Internet Options → Languages menu. In other browsers there are equivalent configuration menus.



## Release Notes

### C.1 Upgrading

Most of what follows refers to files in your servlet containers (e.g. Tomcat, Jetty, Websphere etc.) **webapps** directory. Under this directory you should find a directory **track** if your **Track+** system was accessible via an URL like **http:yourServer.com/track**. Under the **track** directory you find all files belonging to the **Track+** application. This

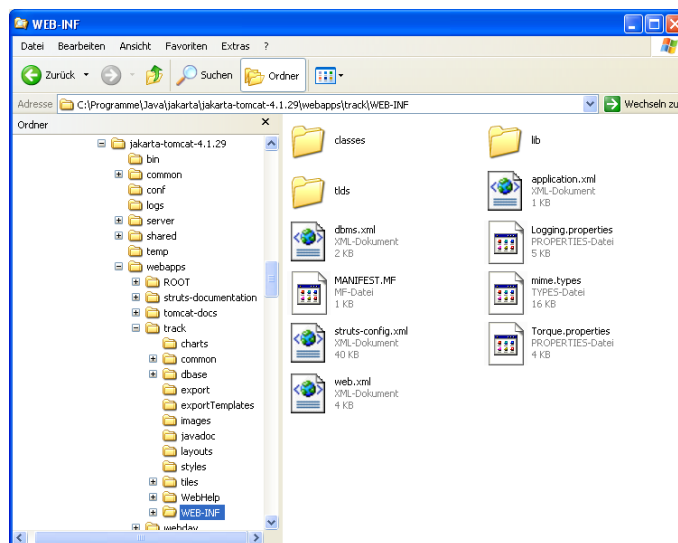


Figure C.1: The CVS directory view

having said, the procedure to upgrade **Track+** is as follows:

1. Save all configuration files that you may have modified to some safe directory. These files are `Torque.properties` and `web.xml`, and possibly the resource files.
2. Make a backup copy of your entire database.
3. Run the database upgrade script provided with the new release. If you try to upgrade over several releases, run *all* upgrade scripts in chronological order. It is possible that there are release which did not require a database update, so you may have to run a migration script from an earlier version to the current version.
4. Save all attachments to a safe directory. The attachments are located either in the `trackdata` context or, starting with **Track+** release 2.0.6 under `webapps/track-v/trackdata/attachments`. Starting with release 2.1.0 you could configure the directory where attachments are stored. It is strongly recommended to store them outside the application context.
5. Drop the new `track-v.war` to the appropriate place in your containers `webapps-directory`. You may have to rename this file for instance to `track.war` if you want to replace your previous installation. In some application servers you should unzip the war file in this directory, since they do not automatically do this.
6. Restart your server or reload the application via your containers administration interface.
7. Stop the server.
8. If you are coming from release 2.1.x or 2.2.x, copy the previously saved file `web.xml` to your new installation. Modify the configuration file `Torque.properties` according to the settings you had in your previous installation. **DO NOT JUST COPY THIS FILE, SINCE IT'S STRUCTURE HAS CHANGED.** The upgrade from release 2.1.x or greater to 3.0 automatically copies all settings from your `web.xml` file into the database. If you upgrade from a release later than 3.0, you do not have to change `web.xml` any more.
9. If you have modified the resource files, you have to reapply the changes. Do not just copy the old resource files since they have changed.
10. Restart the server.
11. Go to the site configuration page and check your site configuration settings.

## C.2 Release 3.1.0

With release 3.1.0 several new features have been added, as described below. Since the **Track+** database structure has been extended to provide room for future features, a database migration script has to be applied to existing databases.

The previous release has proven to contain only few bugs and to be very reliable. Practically all errors reported from release 3.0.0 could be resolved with release 3.1.0.

### *C.2.1 Bugs removed*

The following bugs have been fixed with release 3.1.0:

#### *1173741: Attachment name is not UTF encoded*

The attachment name is now UTF-8 encoded as well.

#### *1160778: Error in "migrate210to300.sql"*

Has been fixed.

#### *1118610: Security problem*

Two action classes and did not check if the user was logged on. This security problem was fixed (thanks to Marcin Slusarczyk).

#### *1118573: Copy List crashes*

This was not exactly a bug, but it was not nice either. This should not be possible any more.

#### *1100672: Attachment not saved*

Attachments were not saved when added to a freshly created new item on database systems that enforce foreign key constraint checks (like Oracle).

#### *TQL error with Interbase/Firebird*

Due to a problem in the Torque layer with Date/Timestamp fields, queries including date fields as selection criteria from TQL could lead to null results. Has been fixed by a Torque patch and by switching all date fields in the database to timestamp.

### *C.2.2 New Features*

The following new features have been implemented with release 3.1.0 compared to release 3.0.1:

#### *Hierarchical Report Display and Sorting*

The report overview now offers the possibility to display the hierarchical dependency of artifacts. Sorting can be accomplished by clicking on column headers.

#### *Configurable Report Layout and PDF-Reports*

**Track+** now has a powerful report configuration engine which can directly output in various formats like HTML, PDF, or Excel. The report layout can be configured by XML based templates, including company logo etc.. Some standard layouts are provided

with the **Track+** distribution. You are welcome to contribute your own to the **Track+** community.

### *Workflow Support*

**Track+** now supports the definition of workflows and state diagrams. State transitions can be configured based on project type, list type, and roles. The workflow configuration can be graphically displayed.

### *CVS Integration*

It is now possible to create links from Track+ issues into a CVS repository. Each project can configure its own repository.

### *Email Reminder*

**Track+** now offers the ability to send email reminders to users that like it for tasks that are about to be due or are overdue. Each user can configure the time buffer and email frequency in his personal user profile.

### *Email Switch Off*

Each user can switch off emails from the **Track+** system.

### *Improved Access Control Management*

New screens have been added to facilitate access control management.

### *Farsi Localization*

Farsi resource files are available. This is interesting in so far as Farsi is a right to left language. The resource files are not maintained by the **Track+** core development team.

### *Support for Firebird 1.5*

The JayBird JDBC driver running on Firebird 1.5 is now supported.

# Bibliography

- [1] BROOKS, FREDERIK (1995) *The Mythical Man Month*, Addison-Wesley.
- [2] ED. THAYER, R.H. (1997) *Software Engineering Project Management*, IEEE Computer Society.
- [3] GILB, TOM (1988) *Principles of Software Engineering Management*, Addison-Wesley.
- [4] BLACK, RON (2000) *Complete Idiot's Guide to Project Management with Microsoft Project 2000*, Que.
- [5] PROJECT MANAGEMENT INSTITUTE (2001) *A Guide to the Project Management Body of Knowledge*, Project Management Institute Publications.
- [6] Overview on Change Management Tools, partly with user evaluations.  
<http://www.daveeaton.com/scm/PMTools.html>
- [7] *Engineering Capability Maturity Model (SECM)*. EIA/IS-731.1, proposed standard
- [8] DEMARCO, TOM (1982) *Controlling Software Projects*, Prentice Hall.
- [9] KITCHENHAM, B. LINKHAM, S. (May/June 1997) Estimates, Uncertainty, and Risks, *IEEE Software Magazine*, No. 3, 69–74.

# Index

## Symbols

<b>DATE</b> function .....	71
TQL .....	71
<b>NEWER_THAN</b> operator .....	71
TQL .....	71
<b>OLDER_THAN</b> operator .....	71
TQL .....	71
<b>ORDER_BY</b>	
TQL .....	70

## A

access key .....	40
administrator .....	41
general schema .....	42
manager .....	41
responsible .....	41
special .....	40
access rights .....	40
account .....	59
accountability .....	3
action item .....	89
active user .....	46
adding locale .....	54
admin user .....	38
administration	
projects .....	38
site .....	46
system .....	37
administrator .....	38
privileges .....	38
application server	
installation of .....	22
artifact .....	4
artifacts	
changing .....	60
creating .....	60
assessment plan .....	90
assessment protocol .....	90
attachment .....	110
configuration root directory ....	34
attachments .....	64

attributes .....	72
TQL .....	72

## B

bar chart .....	76
bugs removed .....	111

## C

changing artifacts .....	60
charts .....	76
check list .....	90
client .....	75
CMM .....	5
command line .....	75
configuration .....	29
Graphviz path .....	34
of attachment root directory ....	34
of database .....	29
of e-mail .....	31
of LDAP .....	33
of logging .....	30
of timeouts .....	31
reminder email .....	81
system .....	37
workflow .....	44
context sensitive help .....	60
creating artifacts .....	60
CSV export .....	69
customization .....	49
example .....	49
priority .....	49
report overview .....	52
CVS .....	97

## D

date .....	71
TQL .....	71
deactivation	
user .....	46
default	
manager .....	38
owner .....	38



- description tags ..... 62
- diagram
  - Gantt..... 81
- diagrams ..... 76
- E**
- e-mail
  - authorizing domains ..... 32
  - configuration..... 31
- editing artifacts ..... 60
- email notification ..... 90
- email reminder ..... 81
- examples
  - TQL..... 72
- export ..... 69
- F**
- filter ..... *see* report filter
- Firebird RDBMS
  - installation ..... 12
  - installation on Linux ..... 12
  - installation on Windows..... 16
- G**
- Gantt diagram ..... 81
- graph
  - doesn't show ..... 22
- Graphviz ..... 34
- H**
- help ..... 60
  - context sensitive..... 60
- I**
- installation ..... 7–36
  - application server ..... 22
  - Firebird RDBMS..... 12
  - SDK..... 22
  - Tomcat ..... 23
- ISO 900x ..... 5
- L**
- LDAP
  - authentication..... 60
  - configuration..... 33
- license ..... iv
- linking
  - from website ..... 83
  - to create issues ..... 83
- to issues ..... 83
- Linux
  - installation of Firebird ..... 12
  - installation of MySQL RDBMS.. 11
- list
  - action item ..... 89
  - assessment plan ..... 90
  - bugs ..... 89
  - issues ..... 89
  - milestone..... 89
  - problem report ..... 89
  - quality gate ..... 90
  - requirement ..... 89
  - requirement change ..... 89
  - risk..... 89
  - templates..... 90
  - work package ..... 89
- localization ..... 54
- logging configuration ..... 30
- M**
- matrix view ..... 2
- milestone ..... 89
- modifying artifacts..... 60
- monitor artefacts..... 90
- monitoring changes ..... 81
- MS SQL Server RDBMS
  - installation on Windows..... 18
- multiple roles ..... 44
- my profile ..... 81
- MySQL RDBMS
  - installation on Linux ..... 11
  - installation on Unix ..... 11
- N**
- new features ..... 111
- O**
- obtaining
  - application server ..... 23
  - Tomcat ..... 23
- obtaining Track+ ..... 10
- online help..... 60
- Oracle ..... 19
  - JDBC driver ..... 19
  - problems..... 19
- P**
- password ..... 59

- LDAP password ..... 59
- PDF output ..... 69
- personal ..... 96
- pie chart ..... 76, 77
- project control ..... 2
- project execution ..... 2
- projects
  - administration ..... 38
- Q**
- quality gate ..... 90
- queries
  - TQL ..... 73
- query ..... 68
- query language ..... 70
- R**
- RACI ..... 3
- RDBMS ..... 7
- RDBMS, supported ..... 6
- registration ..... 59
  - by manager ..... 59
- release notes ..... 90, 99
- reminder email ..... 81
- report
  - configuration ..... 68
  - customization ..... 52
  - export ..... 69
  - TQL ..... 70
- report filter ..... 68
- reporting
  - charts ..... 76
- reports ..... 67
- repository
  - TQL ..... 75
- requirement change ..... 89
- requirement list ..... 89
- responsibility ..... 3, 4
- roles ..... 38, 40
  - access rights ..... 40
  - multiple ..... 44
- S**
- SDK
  - installation of ..... 22
- SECM ..... 5
- setup overview ..... 9
- site administration ..... 46
- sorting
  - TQL ..... 70
- sorting criteria ..... 70
- spread sheet ..... 69
- state ..... 85
  - closed ..... 85
  - open ..... 85
  - suspended ..... 85
- system administration ..... 37
- system configuration ..... 37
- system requirements ..... 7
- T**
- tables ..... 37
- templates ..... 66
- time ..... 71
  - TQL ..... 71
- timeouts
  - configuration ..... 31
- to do list ..... 96
- Tomcat
  - obtaining ..... 23
- TQL ..... 70
  - definitions ..... 73
  - examples ..... 72
  - queries ..... 73
  - repository ..... 75
- Track
  - configuration ..... 29
- troubleshooting ..... 36, 103
- U**
- Unix
  - installation of MySQL RDBMS .. 11
- upgrade ..... 109
- upgrading ..... 109
- user deactivation ..... 46
- user roles ..... 38
- W**
- website
  - linking from ..... 83
- Windows
  - installation of Firebird ..... 16
  - installation of MS SQL Server RDBMS  
18
- work package ..... 89
- workflow configuration ..... 44

workflow diagram ..... 34

## **X**

X11

    Can't connect ..... 22

XML output ..... 69



Trackplus  
Eugen-Ruoff-Str. 30  
D-71404 Korb  
Germany

Fax: +49 (0) 7151 93 79 41  
Tel.: +49 (0) 7151 93 79 40  
E-Mail: [support@trackplus.de](mailto:support@trackplus.de)