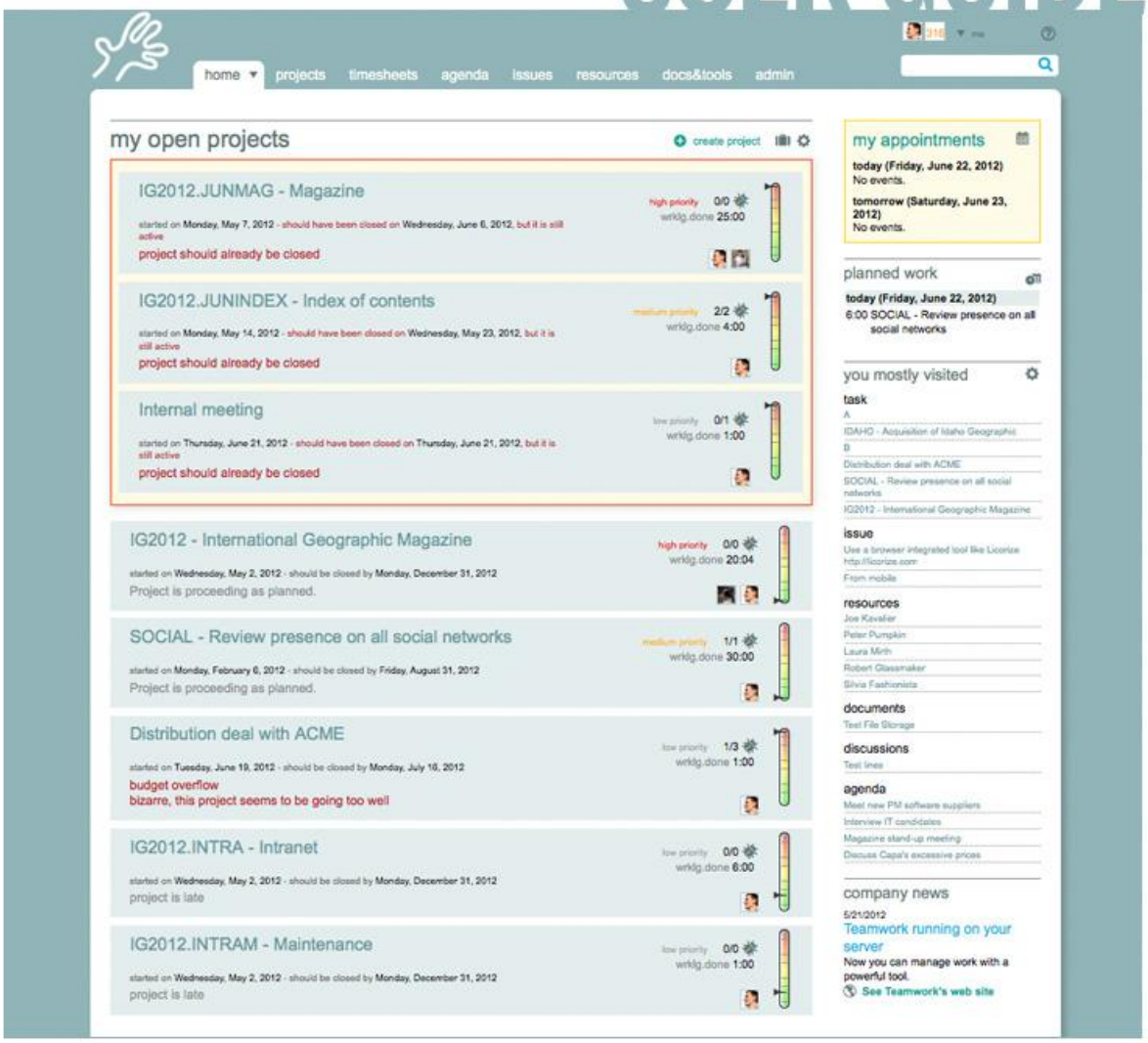
teamwork5

USER GUIDE



User guide - a document from Open Lab

Updated for Teamwork version 5.3

Last modified on Tuesday, October 01, 2013

Teamwork is © Open Lab 2001 – 2013

The quoted products may be registered trademarks of their producers.

Company's HQ

Open Lab
Via Venezia 18b
50121 Florence - Italy

Teamwork Support Services:

Web site: http://www.twproject.com
Support forum and submit/vote feature requests: http://answers.twproject.com/ (958 answered questions – 2013)

Direct contact:

E-mail: info@twproject.com
Phone: +39 055 5522779
Time zone: 'Central European Time' (6 hours ahead of New York)

Summary

What's in here? 12
I. Stories of work management 13
i. Introduction 13
ii. Who uses Teamwork? 14
iii. How do I begin project management? 15
iv. Teamwork's philosophy: a short story 17
v. Managing with lists vs. managing with trees 18
vi. Deep IT integration 19
vii. A critical moment: change and overcoming opposition 20
viii. The problems of estimation 22
ix. Agile methods, SCRUM and Kanban 23
x. Public pages 26
xi. Simplistic cost/benefit evaluations of organizational tools adoption 27
xii. How Teamwork is made with Teamwork 28
xiii. Teamwork and multilinguism 29
xiv. Single and distributed help desks 31
xv. What Teamwork does not do and never will 32
xvi. Thank you for reading these adoption stories 32
1. Begin of user guide: how to start 33
1.1 Basics 33
1.1.1 Login 33
1.1.2 Teamwork header 35
1.1.3 Teamwork sidebars 37
1.2 Profile page 38
1.3 Home pages 39
1.3.1 Most used page: Get Things Done 39
1.3.2 For the project manager – Project manager page 40
1.3.3 For the supervisor and the stakeholders – Supervisor page 41
1.3.4 For the help desk: Help Desk page 42
2. Users as resources 43
2.1 Introduction 43
2.2 Editing 43
2.3 Teamwork' resources with login: users 46
2.4 Company/Department 49
2.5 Search / reporting 51
2.6 Importing contacts 52
3. Projects and tasks 54
3.1 Introduction 54

3.2 Editing 54
Hints - task scheduling: start, duration, end 61
Some remarks on task status change 62
Removing tasks 63
3.2.1 Assignment 65
3.2.2 Expenses 67
3.2.3 Assignment custom fields 67
3.2.4 Assignment cost custom fields 68
3.2.5 Notifications 68
3.2.6 Move assignment 71
3.2.7 Assignment list 71
3.3 Costs 73
3.4 Task tree 75
3.5 Gantt editor 75
3.5.1 Toolbar 77
3.5.2 Textual panel 78
3.5.3 Graphical tree panel 79
3.5.4 Task line editor 80
3.6 Prints and snapshots 81
3.7 Public pages 83
3.8 Import/export 85
3.8.1 Import from Teamwork Gantt 85
3.8.2 MS Project import / export 86
3.8.3 Escape from Basecamp ☺ 88
3.9 Security 90
3.10 Task - other pages 91
3.10.1 Discussions 91
3.10.2 Logs 92
3.11 Business processes 93
Reducing trees to business processes 97
3.12 Search / reporting 98
3.13 Copy/move projects 99
Copy 100
Move 102
4. Issues 103
4.1 Introduction 103
4.2 First issues 103
4.3 Editing 104
Issue re-scheduling 108
4.4 Search / filter / close / order 108

4.5 Bulk management 110
4.6 Organize – Kanban - Planner 111
4.7 Print 113
4.8 Importing and exporting issues 113
4.9 Custom fields on issues 114
4.10 Mobile support 115
4.11 To-dos 115
5. Groupware 117
5.1 Introduction 117
5.2 Workgroups 117
5.3 Boards 119
5.4 Manage by e-mail 120
5.4.1 Adding documents to tasks 120
5.4.2 Adding issues to tasks 121
Using a single e-mail address 121
Using multiple incoming e-mail address 121
5.4.3 Creating TO-DOs 122
5.4.4 Send messages 122
Multiple e-mail, e.g. SMS forwarding 122
6. Work Planning 124
6.1 Introduction 124
6.2 Work load 124
6.3 Resource work plan 127
Example. 128
Example. 128
6.4 Priority 129
6.5 Operator load calculation 130
6.6 Handling future projects by resources as skills 134
6.6.1 Resource hierarchy 134
6.6.2 Future project 135
6.6.3 Running project 136
7. Work time tracking 137
7.1 Introduction 137
7.2 Editing 137
7.2.1 Weekly time sheet 137
7.2.2 Time counters 139
7.2.3 Issue driven 140
7.2.4 From the issue editor 140
7.2.5 From the task editor 141
7.2.6 From the resource editor 141

7.2.7 Subversion 142
7.2.8 Worklog custom fields 143
7.3 Monitoring worklog 145
7.3.1 Check missing worklog 146
7.3.2 Worklog analysis 147
7.4 Validating and billing worklog 148
8. Agenda 151
8.1 Intro 151
8.2 Personal agendas 151
8.2.1 Weekly view 151
8.2.2 Editing 152
8.2.3 Daily view 156
8.2.4 Monthly view 157
8.2.5 Personal agenda configuration 157
8.3 External calendars 157
8.3.1 iCalendar (Outlook/iCal/Google calendar/iPhone calendar…) 157
8.3.2 Google calendar 159
8.3.3 iPhone calendar 160
8.4 Meetings 161
9. Documents 163
9.1 Introduction 163
9.2 File Storages 163
9.2.1 Configuration: file system 163
9.2.2 File system example 164
9.2.3 Using Subversion SVN 165
9.2.4 Amazon S3 165
9.2.5 Usage 165
9.3 Linking a document 166
9.4 Repository 169
9.5 How do I make someone access a folder through a project without giving access to the entire repository? 170
9.6 Document search 171
10. Portal/Dashboards 172
10.1 Introduction 172
10.2 Company news 172
10.3 Dashboard Customization 173
10.4 Portlets 173
10.4.1 Editing portlets 174
10.4.2 Show some external content in your dashboard 174
10.4.3 Create your own portlets 175
10.4.4 Additional portlets in distribution 176

Panic board 176
10.5 Templates 178
10.6 Pages 179
General configuration 180
Personal configuration 180
11. Agile / SCRUM 181
12. Security 183
12.1 Introduction 183
12.2 Areas 184
12.3 Roles 185
12.4 Double area example 187
12.5 Check security 187
13. Tips & Tricks 188
13.1 Search anywhere from everywhere 188
13.2 Custom filters and QBE 189
13.2.1 QBE 189
13.2.2 Date shortcuts 190
13.2.3 Custom filters 192
13.3 Compute in fields 192
13.4 Counters 192
13.5 Internal links 193
13.6 Remove all subscriptions plus defaults on roles 193
13.7 Smart URLs 194
14. Plugins, custom fields/forms, customizations 196
14.1 Task custom fields 196
14.2 Custom forms usage 197
14.3 Create your own 198
14.3.1 Where are custom forms 198
14.3.2 How does it work 199
14.3.3 Plugin dissection 199
14.4 Custom wizards 203
14.5 Customize Excel exports 204
14.6 Custom features 204
14.7 Even more customization options 205
15. Advanced business process 206
15.1 Why extend project management with business processes? 206
15.1.1 Supporting change 206
15.1.2 Flowork and JBPM 206
15.2 Flow administration 207
15.3 Creating your flow 209

16. Configuration 213
16.1 Introduction 213
16.2 Paths, network and security 213
16.2.1 Standard authentication 214
16.2.2 Http authentication 214
16.2.3 LDAP authentication / with fallback 215
16.3 E-mail configuration 215
16.3.1 Configuration of e-mail from Teamwork to users 216
16.3.2 Configuration of e-mail from users to Teamwork 217
16.3.3 Using Gmail as SMTP and POP3 (IMAP) server 217
16.3.4 Debugging e-mail configuration 218
16.3.5 Customizing e-mail 219
16.4 Full-text indexing and ranking 219
Hits management page 220
16.5 Teamwork behavior 220
16.5.1 User defaults 221
16.5.2 Holidays 221
16.5.3 Project defaults 222
16.6 Customization 223
16.6.1 Internationalization 223
16.6.2 Customize labels 223
16.6.3 Translate in a new language 224
16.6.4 Portal/dashboard 226
16.7 Advanced configuration 226
16.7.1 Teamwork security 226
16.7.2 Business process / Flows 226
16.7.3 Custom forms 226
16.8 LDAP/Active Directory 227
16.8.1 LDAP basic parameters 227
16.8.2 Importing users by hand 229
16.8.3 Additional properties 230
16.8.4 Scheduling user import 230
16.8.5 LDAP F.A.Q: 231
17. Monitoring 233
17.1 Introduction 233
17.2 Scheduler 233
17.3 Logging 234
17.4 Messaging system 235
17.5 System check 236
17.6 Tree checks 236
17.7 Licensing 237

17.8 Backup 238
17.8.1 Application backup 238
17.8.2 Database backup 238
17.8.3 Repository backup 238
17.9 Low level administration page 239
17.10 Performance 240
17.10.1 Java memory usage 240
17.10.2 Disable unused languages 241
17.10.3 Enabling more database connections 242
17.10.4 Clustering the database 242
17.10.5 Clustering the web server 243
17.10.6 Monitoring memory usage 243
18. Other sources of information 244
18.1 What is in the blog 244
A complete Gantt editor in the browser 244
Smarter search and recent object functionality 244
Teamwork and multilinguism 244
18.2 What is in the forum 244
18.3 Direct contact 244
19. Installation 245
19.1 System Requirements 245
19.2 Install using the graphical installer 245
19.2.1 Starting it up 245
19.2.2 Installation completed correctly 245
19.2.3 Accessing from other machines 246
19.2.4 Updating 246
19.2.5 Removing Teamwork / Uninstalling 246
How to remove Teamwork's service? 246
19.2.6 Inserting a non expiring license 247
19.2.7 Installation failed 247
Using the installer 247
19.3 Hand/custom install 247
19.3.1 Copying a graphical installation 247
19.3.2 Complete installation by hand 248
19.4 The structure of the obtained web server and application 248
19.4.1 Java 248
19.4.2 The web server 248
19.4.3 The database 248
19.4.4 Optimizing performance 248
Transferring data from one database to another 248

Complete backup 248
19.4.5 Starting and Stopping Teamwork 249
19.5 Platform specific requirements 249
19.5.1 Linux systems 249
19.5.2 Windows Vista/XP/7/8 249
19.5.3 Mac OSX 249
19.6 Database structure 249
19.6.1 Basic task data and dates 249
19.6.2 Task and assignments 251
19.6.3 Issues 252
19.6.4 Agenda 253
19.6.5 Resource 254
19.6.6 What is the logged user? 255
19.6.7 Filter by area? 255
19.7 First access 256
19.8 Upgrading or reinstalling the application 257
19.8.1 Upgrading using the installer 257
19.8.2 Upgrading by hand 257
19.8.3 Upgrading from 3.1 or 3.2.0 versions to 3.2.1 and following 257
19.8.4 Upgrading to 4.5 258
19.8.5 Upgrading to version 5 258
19.9 Appendix: How to ask for help 260
INFORMATION: 260
FOR INSTALLATION PROBLEMS: 260
FOR USAGE PROBLEMS: 260
LOG AND CONFIG FILES: 260
20. Teamwork Mobile 261
20.1 What you can do 261
20.2 Where it works 261
20.3 How to try it 261
20.4 How to buy 262
21. Teamwork from 1 to 5 263
21.1 A very short story of Teamwork releases 263
21.2 Teamwork conceptual evolution leading to version 5 264
Version 3. 264
Version 4. 264
Version 5. 264
21.3 What's new in version 5 264
22. Open Lab products and initiatives 267

23. Index 268

What's in here?

This document is divided in two parts:

1. First section: Stories of work management
2. From chapter one onwards: Teamwork user guide

Over the years we have come to realize that the main objective in adopting new software for work management is not simply software adoption, but also revising the organization of work. And in meeting more and more companies, we realized that the main value in the interaction with us was in consulting about how to structure work organization, more than in learning this or that aspect of the software.

This can be summarized in a principle:

> *No serious problem of work management will be solved by **just** adopting this or that software*

Any real problem will be solved by clarifying what are the intended outcomes and how they can be reached through action. Software can help in collecting data, motivating users, reminding stuff etc., but it is no replacement to clear ideas.

Teamwork is built in such a way to be compatible with the principle above. Hence it is non intrusive, flexible, and different parts may be used in different ways by different users.

We realized that one of the most valuable things we could give to customers and evaluators is our experience in modeling different work situations, using Teamwork in different environments. This is experience that comes from more than ten years in providing "work management" solutions in the whole world (you may check out the customer list on Teamwork site).

Hence in the next section, you won't find instructions on how to use Teamwork, but you'll find how different people in different situations improved their work in various ways, using also Teamwork as a tool.

If you are just looking for a user guide, you can skip this "stories" section and go to chapter *1 Begin of user guide: how to start .*

I. Stories of work management

i. Introduction

In Open Lab we've been developing Teamwork and consulting on project and work management software since 2003, and in previous companies, since the early 90's.

Every time we got in a company with our software, we came with plans to show how the software worked, how to optimize its configuration, how to integrate it with existing applications. But we always ended up actually doing so in the last day of our visit, if ever.

When we got in we always ended up in meetings after meetings where the topic was not specifics of Teamwork, but **questions of work management**, and how to best "map and structure" work management with projects and issues, so that project trees, sets of issues, workflows and so on could be used in a coherent matter.

And we've learned a lot about which mappings work, and which not.

This need for conceptual mappings is teaching something, that is, where is the real value for the customers, what users need.

We progressively realized that the value of software like Teamwork can be simply in the introduction of more structuring, more quality in work. This can be much more than actually using the software – taking it to the extreme, one could simply read this guide, reform practices in the company and forget about the software ☺.

And more people and companies we meet, more we appreciate the way we have structured Teamwork from the very beginning, that is, to be very, very flexible. A group of fashion designers, a company doing production, a team of software developers, an IT group, hardware engineers, event organizers, all these have different needs. And some of these groups may be in the same company, and *different models need to co-exist*.

Meeting diverse needs made Teamwork evolve so that it has become a more powerful mapping tool. Though it is extended, it still is an organically integrated tool.

Modeling company needs for organizing work always implies going beyond the scope of a single application: Teamwork is always used in an applicative context, and so also its openness to external integrations is something that in our concrete experience has made successful adoptions possible.

In this section of the guide you find some of our experiences of "map and structure", some gained by "boot camps" in companies, some by giving web support. You also find the definitions of some concepts related to work management that are not directly linked to software usage.

Some say that all companies need is a to-do list; this is for us just a symptom of scarce experience in what people do at work, of how complex are the needs of a productive company.

We hope that this guide can be of help in understanding which map best fits your organization.

ii. Who uses Teamwork?

Well, actually many different kinds of people and companies. Sometimes evaluators ask: is it used only by software houses? Or only by designers? No: actually software houses are less than 10% of users, and designers even less. It is a diverse public – that is because of the flexibility of the tool. Teamwork is used in more than 50 countries. A partial list of the customers is on the web site:

http://www.twproject.com/customerList.page

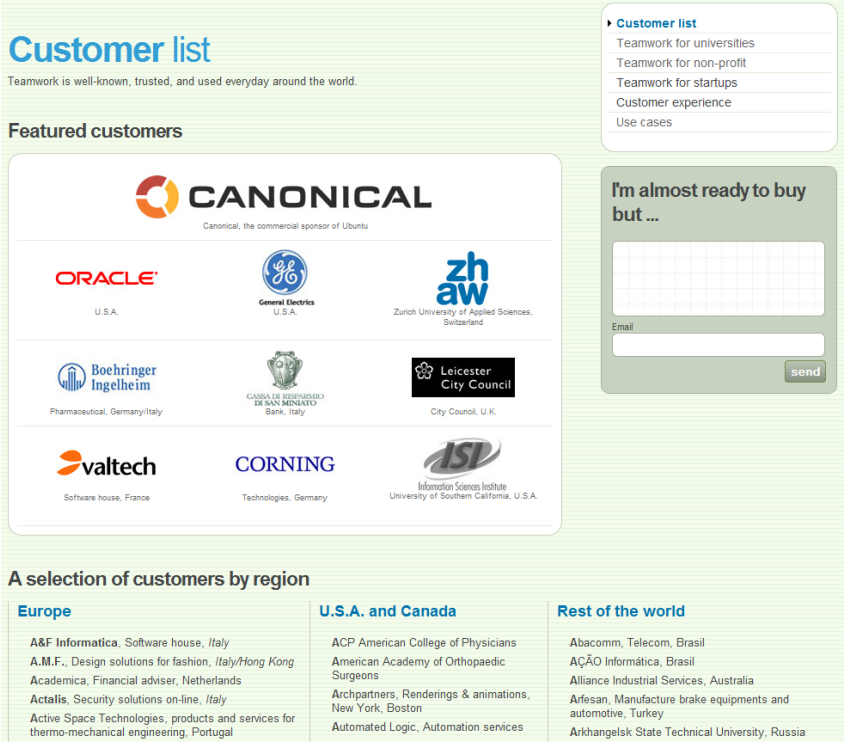


Figure 1 Almost 400 companies are using Teamwork in 2010

These are happy customers: Teamwork is in fact most sold because of our customers talk about it – we don't do any traditional marketing activity.

The companies using Teamwork are quite diverse; but the people that use it share some features: its people that don't work alone, and that know that sharing work and information is not trivial – not trivial to understand what is happening, and how to improve it. It's you, dear reader. You should be called "project manager", but maybe you don't like the sound of it. It is not important: as Scott Berkun (http://www.scottberkun.com) points out, you could be simply named "person thinking about the project at large" – and trying to make it overall successful.

Whether you are building skyscrapers, maintaining tenth-century monasteries, setting up clusters of servers, writing jQuery code or designing cartoons, there is a core of concepts that work management shares.

One can do without management and project managers – as long as things are really simple. It is way of saying that one is doing without any clear authority (maybe gained naturally by the most experienced – no need of formal appointment). In our experience, lack of any authority and organizational means leads very quickly to chaos.

iii. How do I begin project management?

In “Making things happen”, the author (Scott Berkun) states that he assumes that the reader is not stupid, is curious and pragmatic, does not like jargon or big theories, and does not take herself, software, or management too seriously. Well, we do the same in this guide and in Teamwork.

Still, you may have no or little experience in managing team work. Whatever work you will be doing, you may have some requirements, and some dates. if you company has no notion of project / task / issues, you can start this way: list all the things you are doing in your organization. You could separate internal work / external work. In the list obtained, group dependant activities: each group can be called a *project*, and the people that should work on it are the *assignees*.

You may notice that when you are listing “thing that we are doing”, you may also include “things we should be doing”. Notice which of these are stated in the form of concrete actions, like “call X”, or “write Y”, and which are still to be transformed in actions. You should try to transform everything into actions, and get rid of the rest. And still among actions there are simple, brief ones, and others that group many others: you could model the simple ones as issues, and the ones comprising others as tasks (that is, projects which are child of other projects). This is a start of management.

Often we get asked by people evaluating Teamwork:

How do companies really use Teamwork?

In the following you will find several examples. Here are also some good books where to start learning about personal and project management.

Some reference books:

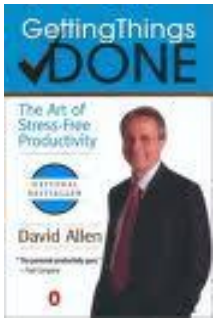
This book by Berkun is our main reference:

Making Things Happen
Mastering Project Management

By Scott Berkun, publisher O'Reilly Media, 2008

http://oreilly.com/catalog/9780596517717

From a personal productivity perspective:



Getting things done

By David Allen

http://www.amazon.com/Getting-Things-Done-Stress-Free-Productivity/dp/0142000280/ref=sr_1_1?ie=UTF8&s=books&qid=1273848829&sr=8-1

On the Agile/Scrum theme:

Agile Project Management with Scrum

By Ken Schwaber

http://www.amazon.com/Agile-Project-Management-Microsoft-Professional/dp/073561993X

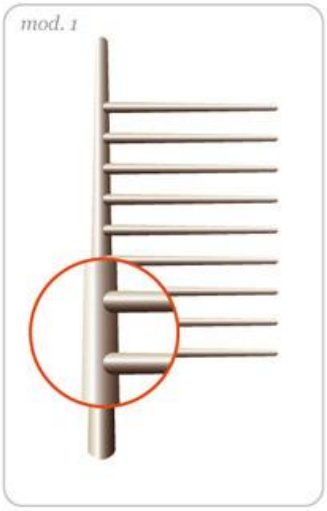
iv. Teamwork's philosophy: a short story

There were once two brothers and a sister, and they were managers at three companies.

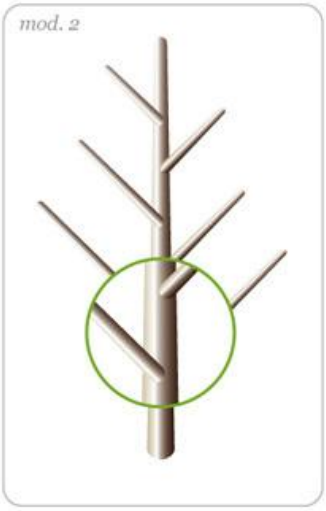
The first brother was called Micro Manager, and he picked the most complex and integrated management tools, which were entrenched in the technical staff IDEs and into all their network activity, so that not a single line of code could be written without it being carefully logged. Not an hour could pass without justifying time spent; not a file could be opened without explaining why. Not a project could be created without designing a 100 leafed Gantt. And after a week everybody hated the system, and then they hated Micro Manager, and everybody was unhappy.

The second brother was called Over Simplify, and he didn't want any kind of management apart from to-do lists. And everybody just had to-do lists, and for the first week everybody was happy. Then many started having long to-do lists, and some started worshipping them, and instead of working, they were compiling longer and longer personal to-do list. And every list was different from any other, and nobody knew what, when, how, and why, and everything was in a mess. And then they hated Over Simplify, and everybody was unhappy.

Their sister was called Reasonable Modesty, and she had minimal goals, had always clear that what matters is how people work and interact, and that software is always secondary, should be flexible and not do too much, and not get in the way. She started evaluating Teamwork.



tabŭla, -ae | *s.f.*



arbor, arbŏris | *s.f.*

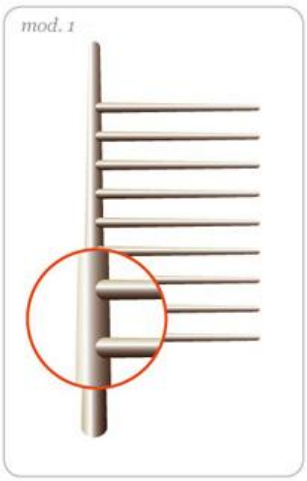
v. Managing with lists vs. managing with trees

The field of "software aided project management", which should by now more aptly named "web based work management" today can be divided by two basically different approaches to management: *list based*, and *tree based*. There are also other approaches, like "let's just use a blog/a wiki", or "e-mail is the way to go", but I believe these to be simply a bit too naive.

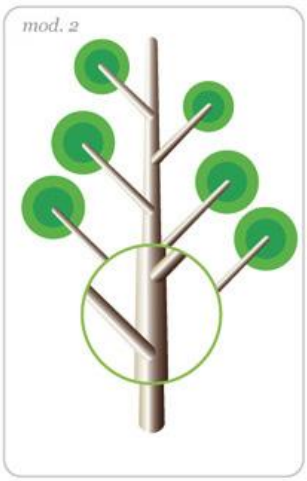
"managing with lists" is the current fashion for web 2.0 applications. "project management 2.0″ is supposedly changing the old approach to the problem: based on the idea of to-do list, with a very very very simple user interface. This in contrast with more classical Gantt-based planning.

Teamwork could hardly be more different from the simple-list model, as we disagree on the basic philosophy: we still think that the good way to model management problems is with the project tree / assignment notion (though not necessarily presented through a Gantt graph), and not with to-do lists. We subscribe the idea that the user interface should be as simple as possible, and that usability concerns should be at the center of development. But we also believe that usability is not necessarily synonymous with poverty of features and integrations.

The difference between lists and tree based management may seem misleadingly small: notice that for example it touches on whether the order of things to be done is just as the order in the list, or is linked to dates. There are far-reaching consequences of this assumption: it is difficult to imagine how ordered lists can be the source of a shared organization, instead of being the result of a shared planning tree of events and dates. These results in completely different applications: a universal dashboard, Teamwork with dates, projects, and different views for different users. And it would be a big mistake to think that one can be somehow transformed in the other.



tabŭla, -ae | *s.f.*



arbor, arbŏris | *s.f.*

You may ask: why can't I have both? In fact, both applications do some of both approaches, but it is a general philosophical choice that has been done: web 2.0 has a minimal modeling structure, Teamwork tries to keep it maximal, giving all options to the users. If you are familiar with Teamwork, "trees" (projects) do indeed "manage" lists (issues and to-do's), but you can't

do without the central notion of assignment, linking “branches” to “leafs” (people).

Teamwork also tries to embrace the existing IT infrastructure (so it can become complex to configure), and hence it is not necessarily an online service: not a purely “web 2.0″ service in this. So, between lists and trees, the choice is yours…

vi. Deep IT integration

In a recent boot camp in Brazil the company that chose Teamwork explained why they chose it. They tried using an open source tool – which was too simplistic. Then a most expensive solution – named like a season, which instead of bringing spring is so convoluted that it brought a post-atomic winter of complications - dropped. They even tried going back to Microsoft Project – but that was obviously a dead end.

Their requirements were:

1. Have a solution running on their Linux server / Oracle database combination, so not to have a further server side technology to maintain
2. Have direct access to the database
3. LDAP (in their case, Open LDAP) integration
4. Supporting HTTPS
5. Integrating with their Linux file server
6. Be reasonably priced
7. Have something web based, capable of modeling their non trivial work flow – without the rigidity of workflow systems

It took their director a month of searches on the web to find Teamwork. They just told us that they could not believe that there was nothing with their requirements. The first 6 are satisfied by Teamwork out of the box. Point 7 was met this way: we created together a “custom wizard": task creation is done from two task templates (see *14.4 Custom wizards*), one for the project initial part, and the second template for the project main part: this creates a new project from the two templates, allowing them to edit every single instance for the particularities of that case, and also in time to update the templates. This is quick and flexible.

So the end result is a low cost and simple solution deeply integrated with their existing IT infrastructure.

vii. A critical moment: change and overcoming opposition

Healthy companies, groups which are self-critical tend to periodically reorganize themselves. It is an opportunity to improve both productivity and quality of work. In these moments, software surveys are done to select new software for project and more in general work management. It is in these phases that sometimes Teamwork is chosen. In the choice what most matters is how the software can "unobtrusively" map to the new organizational practices, and Teamwork in this can be great, because of its flexibility, scope, and the wide fit to the IT infrastructure.

But there may be a problem hindering adoption: user resistance to change.

The great writer and marketer Seth Godin writes in his book *Linchpin*[1] :

" 'Teamwork' is the word bosses and coaches and teachers when they actually mean, 'Do what I say' "

Now our Teamwork software is not at all built in a way to *enforce things*. Enforcing behavior on users is the best way to get them to hack the system. So in Teamwork users can do more or less all they want, the main goal (and the hardest part) being to get users to use the system.

Linchpin deals extensively with change and resistance. For example, at page 168-169: "if their airline started using hidden cameras to push them to do more, they'd actually do less".

A typical reaction to the announcement of the adoption of Teamwork is that there is some user that will feel it as a tool for "controlling" users. But the real world experience is much different: we've never found anyone regularly recording worklog who felt is a way of being controlled. It is feels as something that somehow certifies, stabilizes what I've done. People are giving to the company, and they *like* to testify it. Orderly recording worklog makes you feel *good*. And a management that does the effort of introducing tools like Teamwork is likely a management that is more sensitive than usual to the necessities that improve the quality of work, not the other way around.

Conquering users is a non trivial task, and also a crucial point, as the success of Teamwork is directly proportional to the amount of user usage.

For the point of view of the application, we did a lot of work to ease usage: a main example is the introduction of smart and full text smart search – finding things easily and fast is something that users appreciate more and more.

Smarter search and recent object functionality

As usability becomes more and more a crucial feature of applications, helping users with full-text search and recent object lists may still prove insufficient. You may need to go beyond these features, by having a way to keep track of "most used" objects, which will help to:

- guess what you are looking for

- find what you are searching for

Let's see an example.

[1] http://www.amazon.com/Linchpin-Are-Indispensable-Seth-Godin/dp/1591843162

In these weeks you are working on items A, B and C of your favorite web application. Friday, you actually briefly worked on X, Y and Z before going home, as you had these for quite a while in the bottom of your to-do list. Now, you get back to work on Monday, and what you have in your “recent objects” list? Well, X, Y,Z. Useless. But you have full-text search. You search for the name of A, which actually hundreds of other objects share, and which maybe there are far more occurrences than in A, even if nobody has been using them for quite a while, so they fill results on top of your A. Useless. There is no easy way to get back to A: something here is not working.

This is a usability problem; in order to make your application more helpful, you should somehow keep track of what is being used most often by the users. How to do that? A complete answer is not trivial: as often happens in usability problems, what looks simple from the point of view of the user, is actually complex to solve and render. In the end, all complexity should be hidden, but the solution is not trivial.

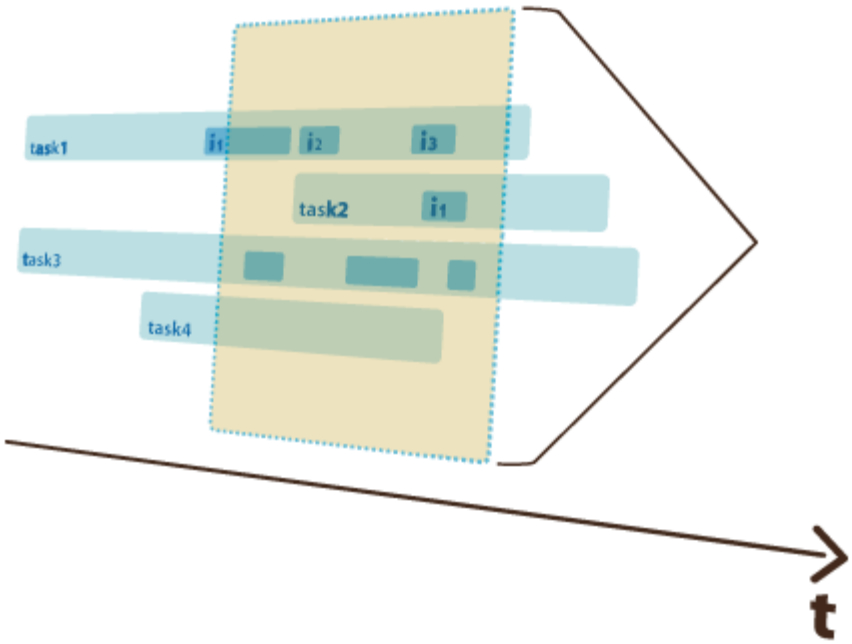
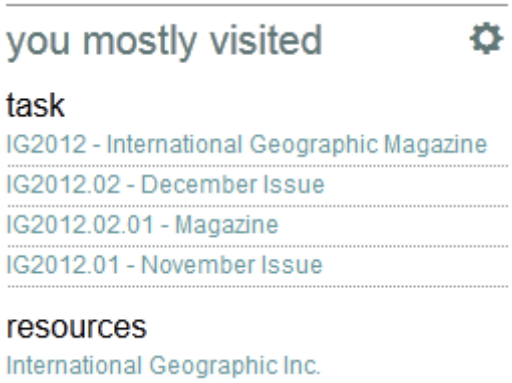


Figure 2 Area of focused interest in time.

What is relevant to you is not just stuff that you occasionally visited, but say projects or documents to which you recently returned to again and again: you need to keep in focus a window of attention. See it in this way: you want the projects or documents to which you are frequently linking to. You need a sort of personal page rank.



This has been implemented in Teamwork by having on all default home pages a “you mostly visited” web part that is surprisingly smart in providing links to what you are most working on.

This example should reassure you that in Teamwork we are taking quite seriously the problem of usability – and hence of lowering user adoption resistance.

viii. The problems of estimation

Teamwork is mainly a tool for collecting work data in real time, while the projects are running. It also provides several tools to *estimate* work to be done.

All modern methodologies share this belief: scheduling and estimations work when they are composed of a detailed planning for a limited period of time. Future evolutions are unpredictable: a little likelihood of getting it wrong on each issue, gives a high likelihood of getting it wrong globally – evil *compound probability* is at work here. But that is not all: there is a snowball effect due to interdependent contributions, so an apparently local "slip" can cause a wider slip elsewhere, and so when locally all is proceeding fine, the end result is far from what planned.

But some (most?) of the value in scheduling is not in getting the dates right, but is simply in commitment. The moment of estimation is and always be present and critical in any organization that is larger than, say, seven people. Because it is also an occasion top meet for people that do not normally work together.

This is why in Teamwork there are no "global" estimations on project: the tool encourages you to do estimations locally: on single assignments, or even more fine grained, issue by issue. And you can do a combination of these.

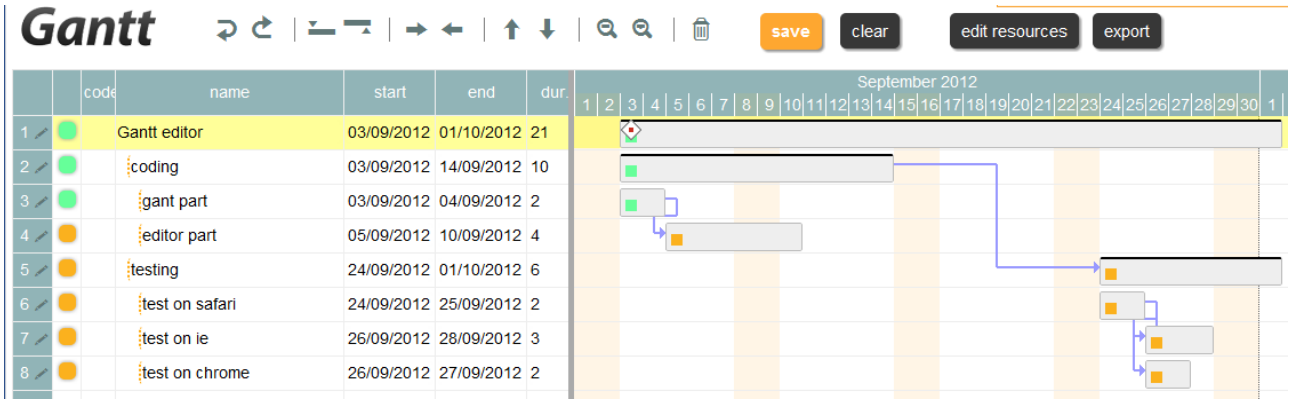
To Gantt or not to Gantt

In almost every company where we went for a Teamwork boot camp, we found something like the Gantt in the picture hanging on the wall: a tattered piece of paper. An abandoned, yellowed Gantt print, left there for about two years.

Reports are useful *if read in the narrow time interval of their validity*. And even simply to be able to get at any time a valid report of the standing activities of a company is a difficult, non obvious result.

There is nothing "evil" in Gantts. Actually, it is a brilliant way of displaying data; only one should realize that it is not a universal solution. It can't be the exclusive focus of your planning. And Teamwork 5 has also a splendid web based Gantt editor ☺ .

So Teamwork both lets you have project trees which are not Gantts – a task may open, pause and close inside the time interval in which it is defined.



ix. Agile methods, SCRUM and Kanban

"Being Agile" is like "being in love". No guarantee that it means the same thing to any two teams.

Seen on Twitter from "jasongorman"

Some context: Scrum (http://en.wikipedia.org/wiki/Scrum_(development)) is a methodology for software development (but not only) belonging to the agile family (http://agilemanifesto.org), which is a set of methodologies which was a direct response to the dominant project management paradigm, waterfall, borrows many principles from lean manufacturing and was formalized when 17 pioneers of the agile methodology in 2001 issued the Agile Manifesto. If you know even a little about Teamwork's approach, and take even a cursory look at the agile principles, you'll feel a family resemblance.

Scrum is not just about software development: see A real-life application of Scrum outside IT http://www.scrumalliance.org/resource_download/548 .

Kanban: On Wikipedia:

http://en.wikipedia.org/wiki/Kanban

Compared with Scrum:

http://www.infoq.com/minibooks/kanban-scrum-minibook

"Kanban uses a visual control mechanism to track work as it flows through the various stages of the value stream. Typically, a whiteboard with sticky notes, or an electronic card wall system, is used"

When we came back to Teamwork version 4 after reviewing some literature on agile methods, and in particular re-considering the Scrum perspective, it became progressively clearer that mapping Scrum ideas to this or that functionality of a software is inevitably a simplification and maybe even a betrayal of the agile philosophy: as these methodologies concern the way you approach problems, and have great variations in detail when it comes to each particular case; see

Agile Project Management with Scrum by Ken Schwaber, Microsoft Press

which is filled of examples.

From this perspective, the main point is not and should not be the management software you are using. We'll get back to this in the final considerations.

We're assuming here familiarity with concepts from agile methodologies and Scrum. Also the examples are tuned to software development, but the line, if valid, is valid in general.

In our experience success and productivity in work are linked to how you deal with two sides of work management:

1. Modeling carefully the complexities of your work environment – and here obviously Teamwork serves you beautifully

2. Bringing this complexity to something easy, light and quickly manageable and

updatable by the single user.

Most of the ideas surrounding “agile” management and “getting things done” revolve around this process.

Mapping examples

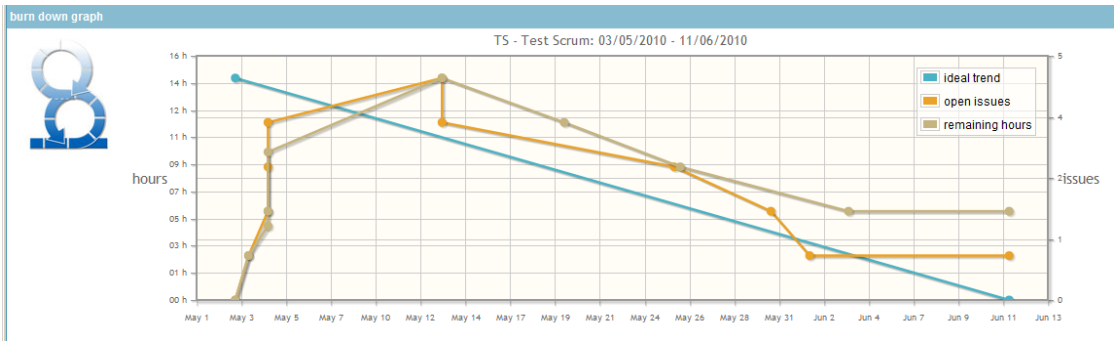


Figure 3 Burn down chart with Teamwork

If one does decide to use management software to manage agile procedures, one should be careful not to take simplistic decisions.

Consider for example backlog: collecting a backlog is the most basic step; how do you collect the backlog? It may be say a shared Google document; so from the PM software point of view, your backlog is a non structured document that the software does not manage. It may be a set of separate entities, say issues? It may be made of tasks with detailed descriptions and work estimations; and so on.

In many examples of teaching Scrum we see cards sticking on boards, and some software just use that idea for the user interface. People seem simply to be missing the fact that a development project may simply have like 800 “cards”. How the heck am I gonna stick those on a board??? It can’t work. You need something more flexible and powerful than a concrete or digital board. Only in some cases a board can be used – and now Teamwork has the Kanban board to do just that – see 4.5 Bulk management, 4.6 Organize – Kanban - Planner.

Stand-up meetings: why is backlogging the subject of management-by-software and meetings aren’t? This is a typical and mistaken hacker’s perspective, because some people focus their management more on their agenda than on their to-do list, if they have such a thing. And you will have projects with both kinds of people (and many more).

Why recording the activity on the assigned tasks has to be done by scaling down hours on the selected items of the backlog? Wouldn’t it be more practical if say one could record activity in the Subversion commits? Or in your Twitter feed? Here too, you need an open ended tool, which collects data from many sources in different ways.

An example: let’s see a sample project and assignment structure: suppose you have a customer, a lead developer, and a set of developers, D1, D2, D3; you want to collect the backlog, and basically your main aim is to let the developers work in quiet conditions and with a stable set of requirements for a month. Well to model that in Teamwork is no big deal, you can support this procedure in several ways; for example you may have a root project ROOT, on which everybody is assigned as “reader”, and lead developer as project manager; you have a child BACKLOG, where the customer is assigned as “worker”, so she can contribute inserting backlog; the backlog is inserted as issues on this BACKLOG task.

You then create a new ROOT’ child task called FIRST SPRINT, move the subset of issues which constitute its effort to it, and assign D1, D2 and D3 as “workers”, so they can edit and close the issues, but nobody externally will change the set of issues. That’s it.

On this structure, Teamwork gives you many, many, many tools to go on, work comfortably, connect this project with others differently structured; it may even be a child of a completely different structured project. You may structure the BACKLOG task itself as a tree; you may have several sprints going in parallel; you may have after the spring a workflow of approval, and again here Teamwork supports you with task as processes. More examples could be made.

Final considerations on Agile

Figure 4 Kanban-like issue management

A most important consideration is that particular methodologies, say Scrum, help solving some class of problems, but will never cover the totality of the working activity of a company, not even the totality of projects of a company. Actually, the original Scrum texts are written in full consciousness of this limitation.

So it would be extremely non-agile to have specific software to follow the “agile” projects, and another one for the “others”. And even the “agile” projects will have so many variations, that they will fit in the agile metaphors at different levels, and hardly fit in a single “Scrum software model”.

So in the end we realized that the mapping between the methodology and the software (or paper) requires great flexibility; agility is in the methodology, not in the software. If you want to use a software, it should be flexible enough to let you map projects, tasks, issues to people and customers, in infinitely many different ways, but so that all data from the different projects and methodologies ends up in the same place.

So no, Teamwork is not yet another Scrum tool, it is a management tool that can help also those that decided to use Scrum for some projects, if they don’t prefer to use paper cards ☺

x. Public pages

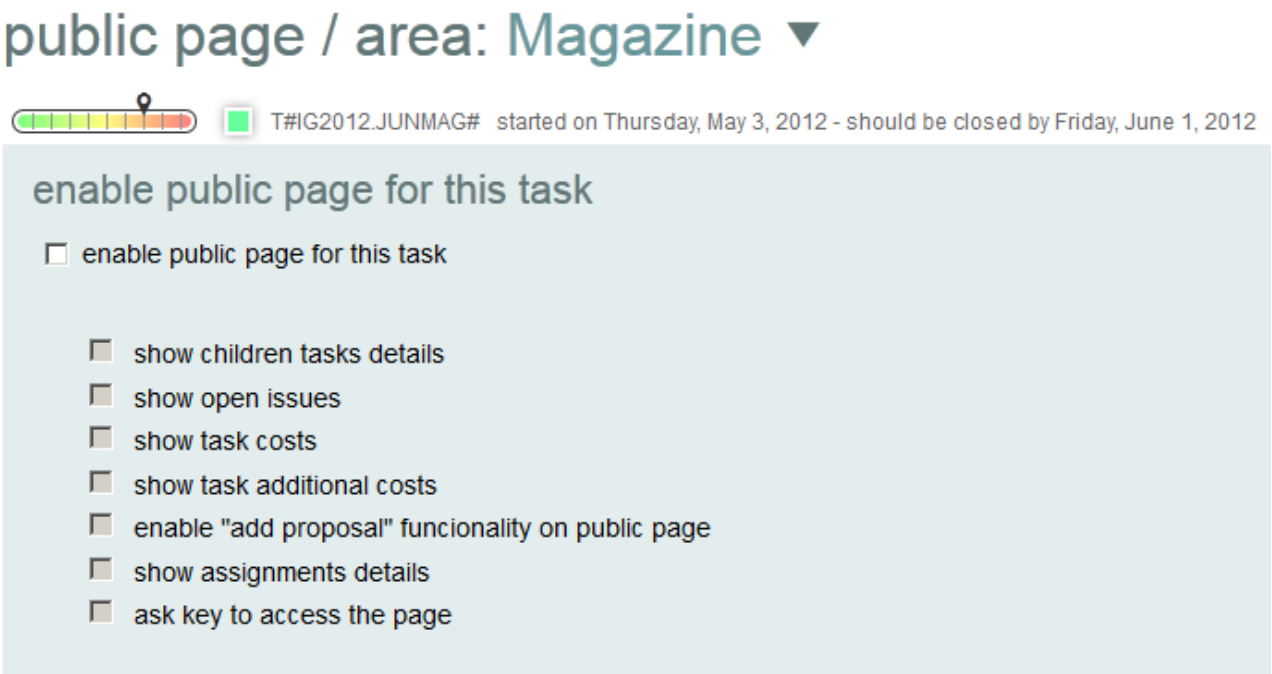
We often got requests from customers and evaluators to give access to some Teamwork data to someone who does not log in Teamwork; data like status, progress and open issues on a project . You have two “classical” options:

1. set the customer as an assignee and give them access to Teamwork with limited rights (which from a Project Management perspective is formally the right choice), or

2. you exported the project data in say a PDF and sent it to the customer

Both choices have their disadvantages: the first one, that you are giving access to a system of which the customer does not know anything about, and the second one is that you have to do the procedure by hand.

Since Teamwork 4.4. we added a third option:



3. public project pages: for any project / task, a public page can be enabled which will be exposed at Teamwork’s address, will not require login, and will present selected information. It is a way to automate distribution of information always updated from Teamwork to people that do not access it.

The pages are available as a URL of the form

[http://your teamwork address]/project/[task id]

Again this eases “talk” between users and third-parties – integrating Teamwork with your extended environment.

xi. Simplistic cost/benefit evaluations of organizational tools adoption

What are the benefits of adopting Teamwork? Can they be quantified? One may rephrase the question as “what is the exact economical gain given by adopting Teamwork”?

Very superficially, this looks like a clear question, which requires an exact answer. Let’s take a closer look.

What does it mean “adopting Teamwork”? If one takes even a cursory look at Teamwork user guide, one should quickly realize that for a tool that can integrate at so many different levels with IT infrastructure, this may mean all sorts of different things: one may be handling just high level projects, sharing them on the web, or one may have integrated it from intranet authentication and certification forms, following every little action in the company.

One may be using the exchange function with Subversion, Google calendars, Twitter and iPhone agendas, so even the boundaries between what is done in the company by Teamwork and what is done by other applications is blurred. So “adopting Teamwork” has different meanings for each adoption process.

But there is an even bigger conceptual mistake that is lingering here, given by the first part of the question, “exact economical gain”: i.e. that taking steps in improving quality of work, by implementing software aided organizational procedures, is a purely economical gain that can be accounted for say is a year after the reorganization. Anybody that has experience in reorganization and working on quality of work and communication knows that consequences cannot be evaluated so simplistically, though they can be great, and span an entire work life.

This said, the benefit that one will have basically depends on the plan and determination of the leader that is introducing innovation, by her/his culture, open mindedness and experience in the field and in human relationship, and the respect that she gets from the team; and we believe that in some cases (not all), Teamwork can be of help for such individuals, more structured help than just a to-do list shared online. But don’t ask us to fool you with numbers thrown at random; you should probably be very suspicious of vendors that promise X% “gains in efficiency” by doing this or that. Our customer list is partly public, the best way is to ask them, and our guess is that everybody will give a different answer.

xii. How Teamwork is made with Teamwork

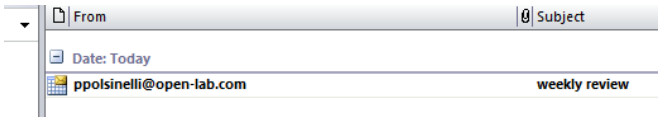
We are running our business with Teamwork – like many other companies. What this ambitious statement says, is that every working activity in Open Lab is a project / issue in Teamwork.

The guys developing Teamwork are indeed using Teamwork for managing work. How we do that? Well, even in our small group, people have different functions and habits. We have two areas, production and accounting; inside production, there are people with different roles, and consequently see and use different data, to which the interface adapts seamlessly. We extensively use the dashboard customization functionalities so that everybody sees what they want.

Teamwork 5 has won the long-standing war with paper in Open Lab (the group producing Teamwork). We have to confess that for some short-lived issues, some of us (including me) were still using post-its and notes on paper as an integration of issues. But Teamwork 5 won: the Ajax issue multi-editor is just too practical. There is no more paper on our desks; add Balsamiq mockups for replacing paper interface drafts, and the coverage is complete.

We cross post issues and bugs, which we get notified thanks to the subscription engine.

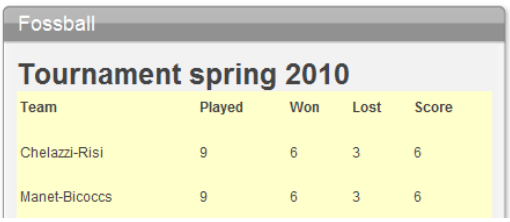
Teamwork worklogs are inserted with help from Subversion logs, which Teamwork 5 does natively.



A section which is widely used is the agenda integrated with meetings, which as it synchronizes with our e-mail clients, is quite practical.

We use boards too, for example to collect notes for our technical meetings. Careful collection of worklogs allows to monitor costs, and also comparison between releases, cost per team etc. .

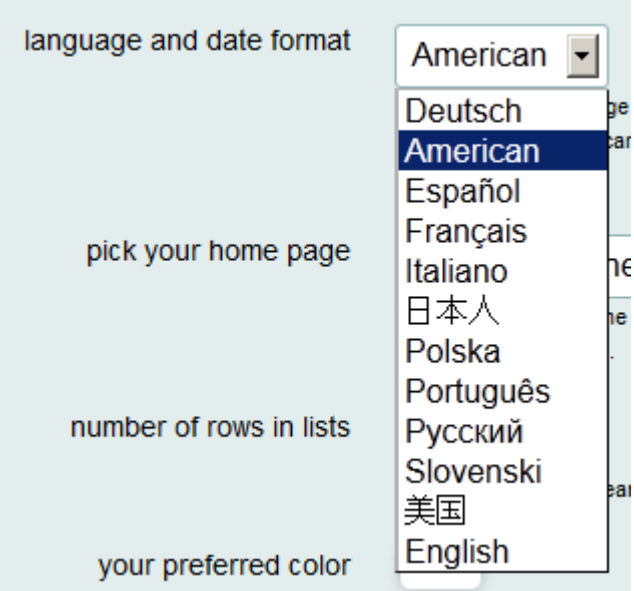
For authentication, our Teamwork is integrated with our Active Directory. As we are "advanced users" :-) , we have added to the scheduled jobs a "SiteAliveTester" custom job which tests that our servers are up and sends e-mail alerts.



We have added some parts to the defaults, such as RSS reader, user voice reader.

Of course we also use news, for example to publish scores of our table-tennis tournament!

xiii. Teamwork and multilinguism



Let's see a bit in detail how we dealt in general with internationalization issues in Teamwork; actually this set of problems will have to be met by any sufficiently powerful web application.

There are several senses in which an application might be said to "support multi-languages", or be "internationalized":

Interface. Labels and messages of the web interface are available in several languages. Teamwork contains a label editor, where you can create a new language and also modify existing labels. Teamwork is used in 43 countries, almost all using it in English; actually some project managers like to have it English as teams are made from people from different countries, so it encourages communication.

But as usual :-) Teamwork does more: it lets you change labels on the fly in the web interface, saving them on the database so that you don't lose customizations on application update.

Data. Data inserted in the application can be inserted in any language. We have been careful about the encoding (always a problem in web applications), so that the full spectrum of UTF-8 supported languages is included, which means also Greek, Cyrillic, Arabic, Japanese… . This also assumes that the database on which Teamwork is running supports Unicode or UTF-8 data. Then you have the further problem that labels and data you have on the interface may need to be channeled on a different mean, e.g. exported in an Excel file, or in a PDF,

and there again you may be plagued by encoding problems.

Search. (This is often forgotten) Full-text search requires multi-language stemming of contents: this is from our technical contribution, which is in the context of Hibernate (an object/relational tool) and Lucene (an indexing engine):

> *You need to know the language in which a document is written, in order to correctly index it; once you know the language, you can instantiate say the Snowball analyzer with the correct language stemmer. To make a practical system, you will need to guess the documents language from its content. We have found a very simple and effective solution [...].*
>
> *In order to make a content "findable" also when searching from a language (say, German) a document in another language (say, English), we actually double indexed the content field, once with the Snowball analyzer and once with the simple StopAnalyzer; so that if you are searching from German and you search "Telefunken", which stemmed would be searched as "Telefunk", will find also "Telefunken" in English documents ? .*

See

http://twproject.blogspot.com/2007/11/using-hibernate-search-with-complex.html

and

http://www.hibernate.org/432.html

So Teamwork's full text search is language-aware. Actually search in Teamwork is much smarter than that, but this is a topic for another post.

Documentation. Documentation may be provided in several languages. In Teamwork's case, as it is by now in 99% of the web applications, it is provided only in English. We also believe that it will be the "power user" of the application that will mostly need documentation, and we assume that she/he can read English.

So how can we evaluate Teamwork w.r.t. all these aspects?

Feature	How it is dealt with in Teamwork
Interface	Available in English, German, French, Spanish, Italian, Portuguese, Chinese, Russian, Slovenian, Japanese
Data	Data in all languages is supported (UTF-8 supported).
Search	Stemming is available for all Lucene analyzers: Teamwork provides out of the box English, Spanish, French, German, Portuguese, Italian, Swedish, Danish, Dutch, Norwegian, Russian, Finnish, but it is easy to include other Lucene extensions.
Documentation	This is provided only in English.

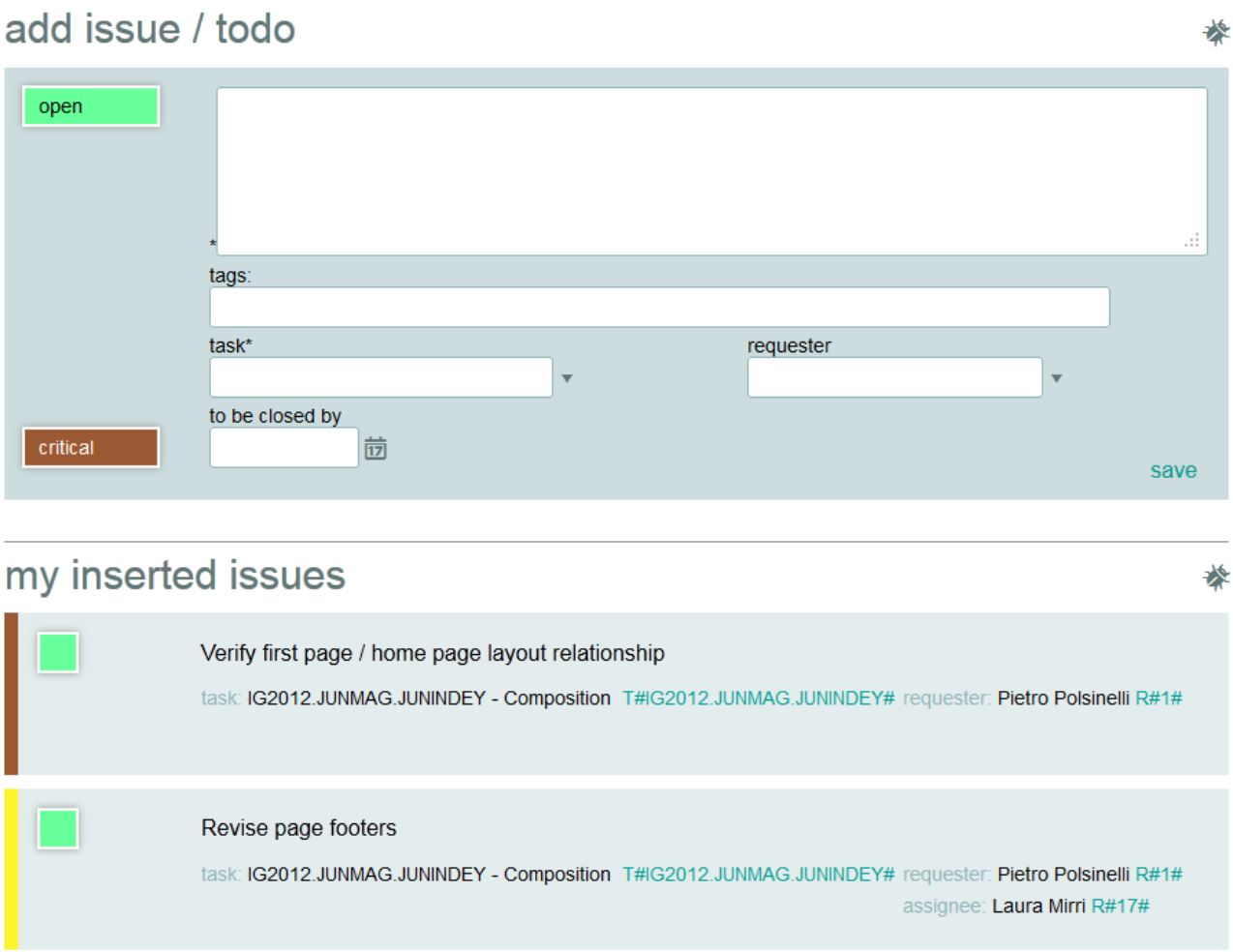
xiv. Single and distributed help desks

Often one of the teams adopting Teamwork works as a *help desk*. There are several functions that may help this kind of work, and since version 4.5 there is a home page template dedicated to help desks built in the default setup.

One of Teamwork customers is a USA newspaper chain which has different offices and web sites. What they did to manage incoming help requests was to create a different project for each help office, and then used a form on each web site (every office corresponds to a distinct local edition and distinct web site) which has as target Teamwork's e-mail address, with a set subject so that the e-mail gets transformed automatically in an issue on the task. Then the project manager gets notified of the new issues, and distributes them along workers. You can also set different automatically receiving e-mail on a *per-project* level.

Often help tasks are separated in levels, some are handled at a "first level2, and some others scale up to a second level: Teamwork keeps track of issue levels and of the passage from an assignee to another, and so models

Last but not least, the home page dedicated to the help desk is particularly simple to use – focused at inserting new issues:



xv. What Teamwork does not do and never will

1. It is not an accounting tool
2. It is not a mind-mapping or “scenario” tool
3. It can’t manage a project of depth 300 or more
4. It can’t manage a project with 5000 nodes
5. It can’t compute critical paths in Pert charts
6. It can’t synch perfectly with Microsoft Project*
7. You can’t “customize” the basic logic of projects
8. It is hostile to micro-management
9. It doesn’t force users to do this or that

This list is meant to save you time: if you don’t like Teamwork’s basic logic, better use another tool. Also if you have a specific application in mind which is not work management, better to search a specific tool rather than “forcing” Teamwork in a non natural direction.

* See section *3.8.2 MS Project* for details.

xvi. Thank you for reading these adoption stories

Thank you for reading these adoption stories. We hope they will help you organizing work and creating your “map” from work practices to Teamwork functions – and back.

If you have one story to tell, write us at info@twproject.com. It may help others too.

1. Begin of user guide: how to start

To start using Teamwork there is no better way than trying the software by installing it or by using the online demo.

Downloads are here:

http://www.twproject.com/download.page

The demos are here:

http://www.twproject.com/onlinedemo.page

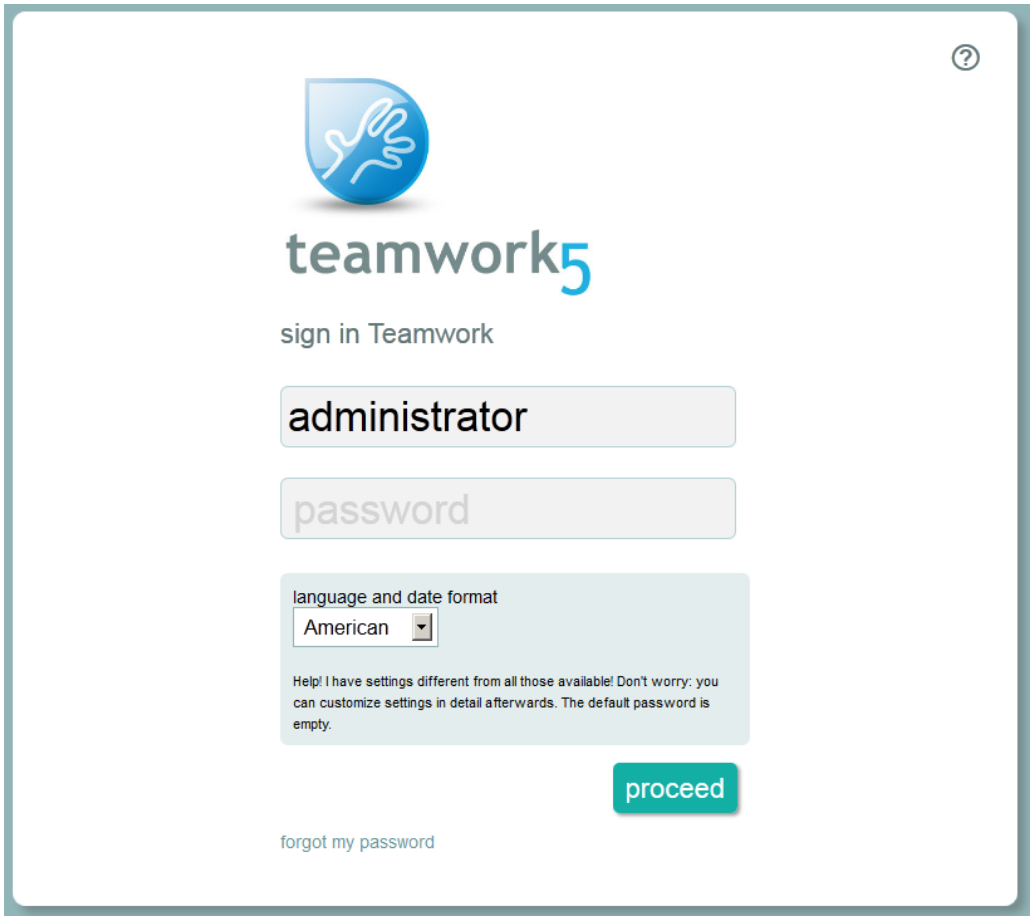
If you did one of these already, so you'll already have some "live" feeling for the application, reading this guide is a good idea.

1.1 Basics

Teamwork is a really powerful environment with lots of features; hence one of our greatest challenges is to create a clear, easy to use and even pretty user interface.

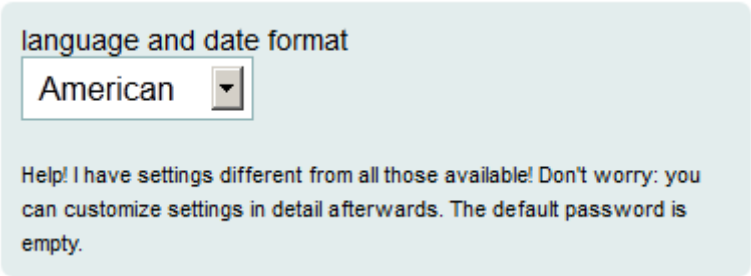
1.1.1 Login

Teamwork welcomes you with the login page:



The first time you log in use "administrator" with empty password; you should set a password as soon as possible.

In case of the very first login, the page appears with an additional field, “language and date format”:

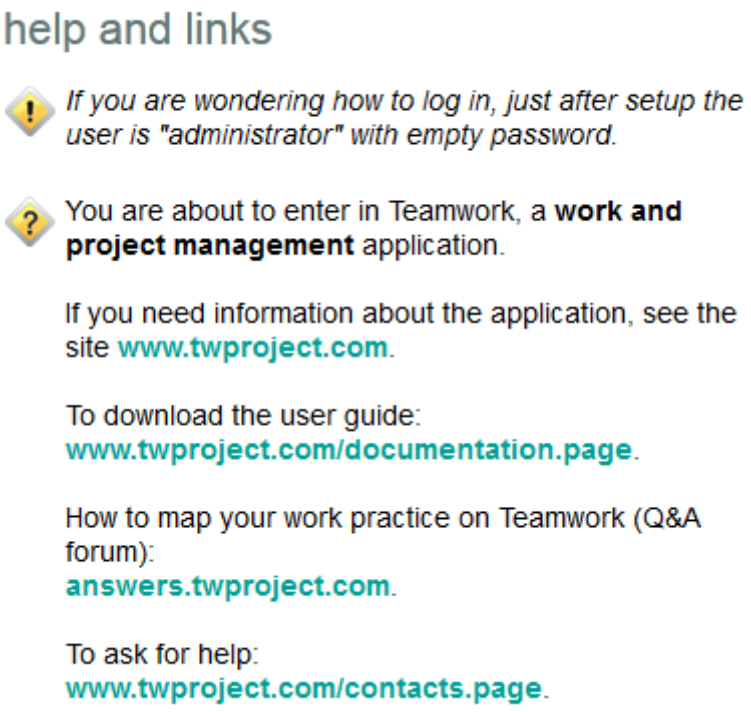


This sets some default localization factors like date format and currency; one can set more refined options once inside the application, in the admin / internationalization section, which will be presented in later chapters:

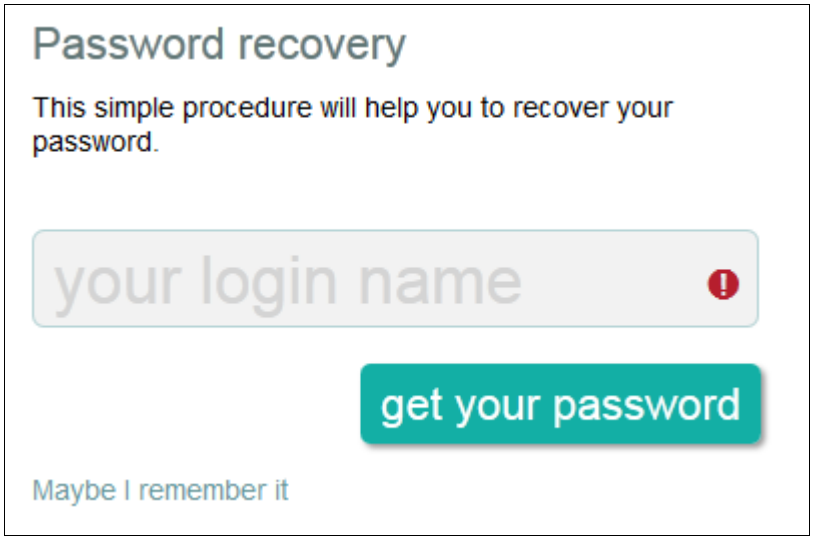
language and date format	American	en
friday is working day	☑	yes
saturday is working day	☐	no
sunday is working day	☐	no
currency format	###,##0.00$	###,##0.00€
print logo	printLogo.gif	printLogo.gif

Different users can choose different user interface languages in their options.

The links opened by ⓘ point to Teamwork support sites:



Below, if you have forgotten your password you can reset it, click on “forgot my password” and you may reset it:



a new password and a link for resetting it will be sent to your e-mail.

After the first login from a browser that has cookies enabled (the default for all “non ancient” browsers), you will log in automatically. If you log off by hand, the cookie will be removed.

1.1.2 Teamwork header

Every page is headed by… Teamwork’s header (ok, there are some exceptions for print pages and some pop-ups, but it is almost always true):

We use here some *de-facto* standards for web sites:

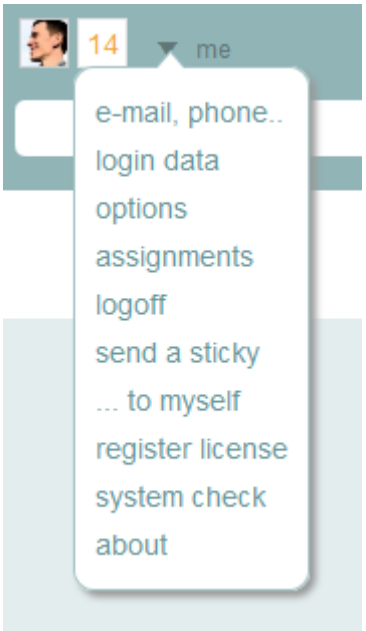
1) Teamwork logo always bring you to your home page

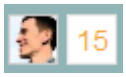
2) The top menu

is always the same. It highlights the section in which you are

3) Search is always available.
It searches text everywhere. Teamwork supports also refined searches for each entity, as tasks, resources or issues. Specific searches will be explained in the following. There are some hidden “tricks” on this field, some secret shortcuts… (solutions in following chapters).

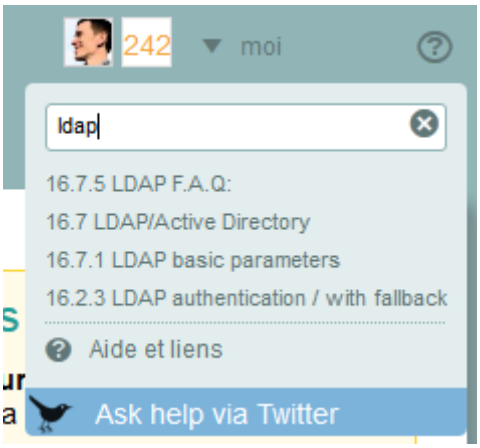
4) Your image: clicking this opens a menu of shortcuts leading to your personal data editor and options: there you can also change your image.



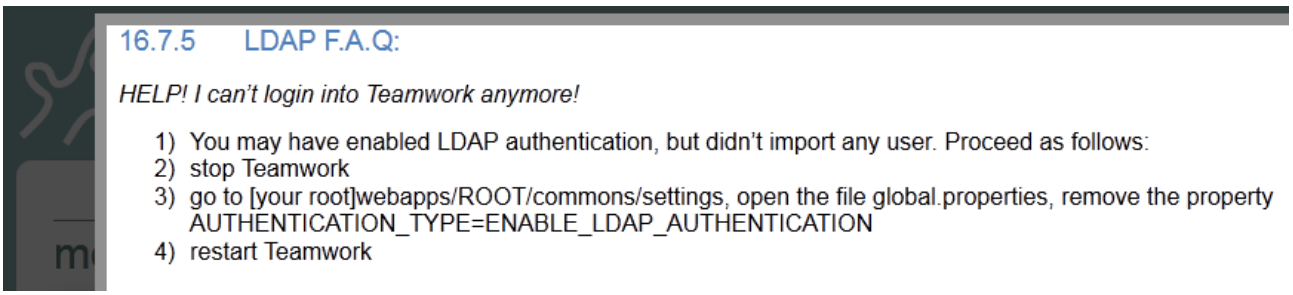
5) Your score 

is always visible. It shows your "Teamwork' loyalty" (amount of usage in the last 30 days): higher score, better user ☺!

6) Search in the user guide: by clicking on ⓘ you open a search filed that will search dynamically as you type in a copy of this user guide distributed with Teamwork:



Clicking on the links will get you to the topics:



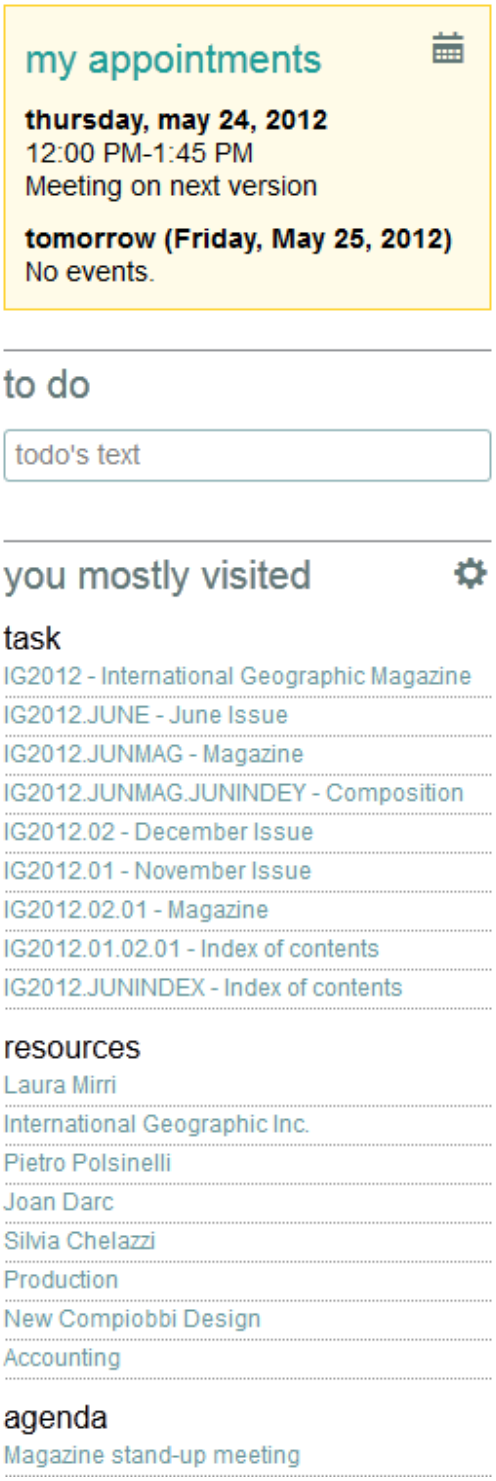
16.7.5 LDAP F.A.Q:

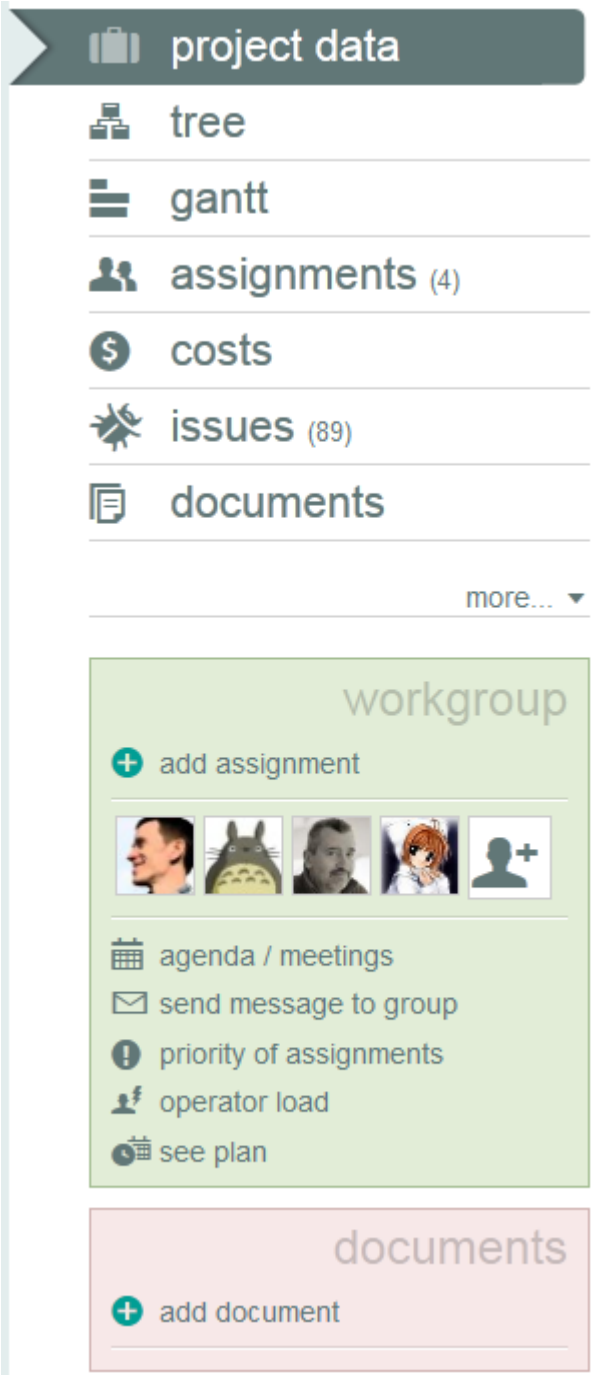
HELP! I can't login into Teamwork anymore!

1) You may have enabled LDAP authentication, but didn't import any user. Proceed as follows:
2) stop Teamwork
3) go to [your root]webapps/ROOT/commons/settings, open the file global.properties, remove the property AUTHENTICATION_TYPE=ENABLE_LDAP_AUTHENTICATION
4) restart Teamwork

1.1.3 Teamwork sidebars

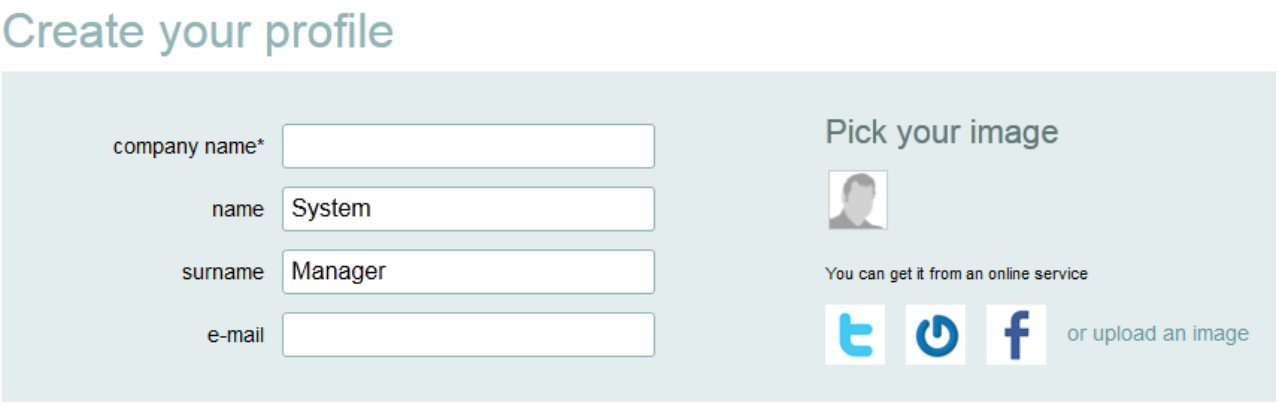
Teamwork 5 is focused on giving more space to information and in refining user defaults. In most Teamwork pages there is a sidebar in which there are links for refining results shown or for moving to functionally connected pages. The content of the sidebars changes dynamically with contents and its state. Below home and project sidebars as examples; for example, the add sub-task button makes sense only in specific contexts.





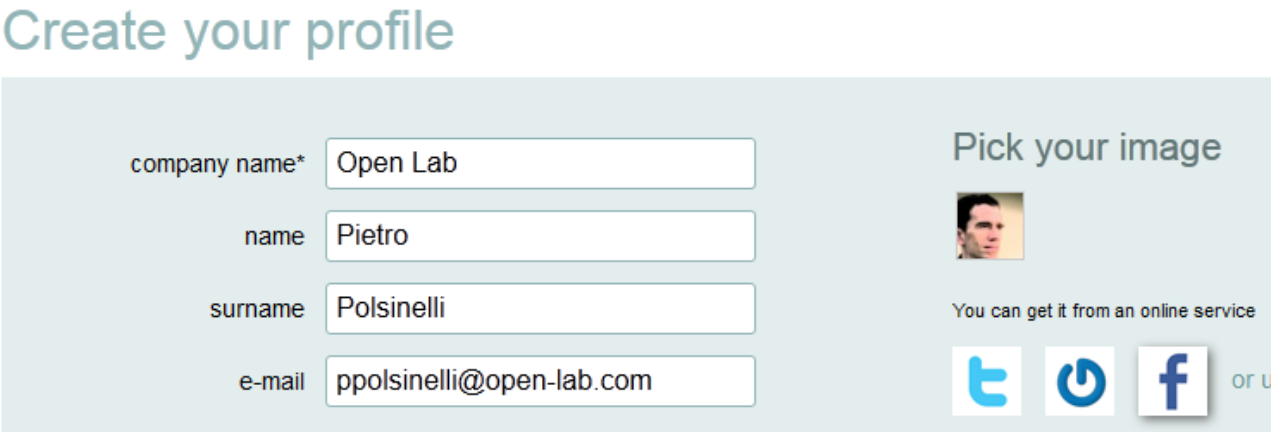
1.2 Profile page

The first user of the application will get this profile page:

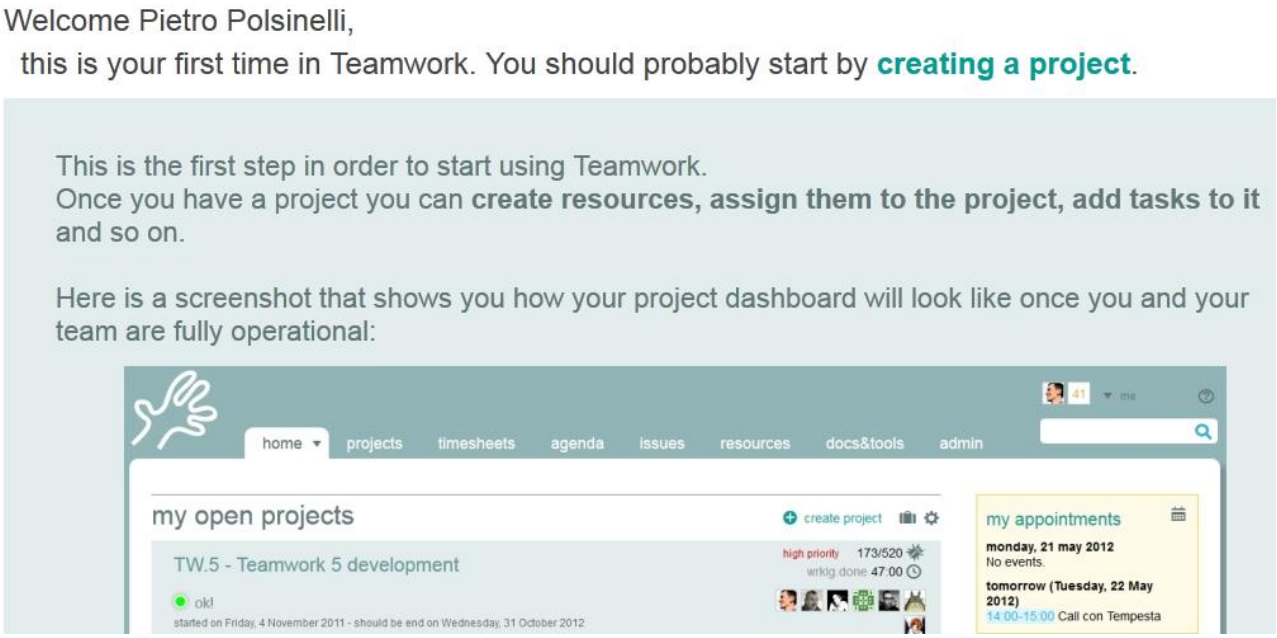


The “company name” will feed a Teamwork company which will be set on the first user. It will also be the name of the default security area (see the following).

After setting an image and filling up the fields:



You get to:



Following users at first login will see the same page without the “company name” field.

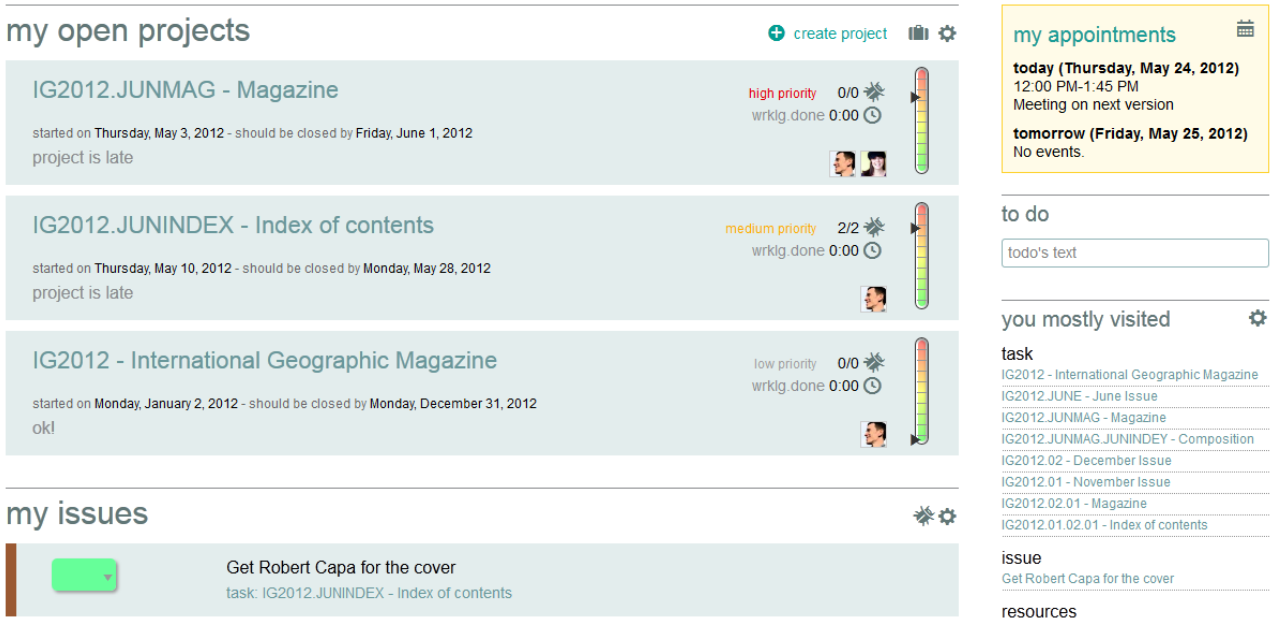
1.3 Home pages

After setting the profile, every time you log in Teamwork shows you the current user home page. Teamwork has several built-in home pages, customizable for different user needs.

Setup creates some hopefully useful sample “dashboards” that reflect common use cases, we’ll see them in detail in the following.

1.3.1 Most used page: Get Things Done

In order to have an idea about how Teamwork can help organizing work’s activity, let’s see an operative page. The “Get things done” page is designed for people that use Teamwork as support for their activities.



You see at first glance what you are working on (“my open projects”), your issues, agenda, to-do’s, your most used entities.

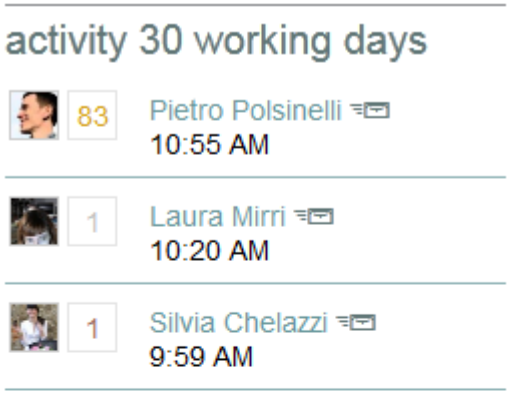
On the bottom part there is a time-bar where your will see the current time (blue vertical line), appointments (green blocks) and milestones (diamonds, if any).

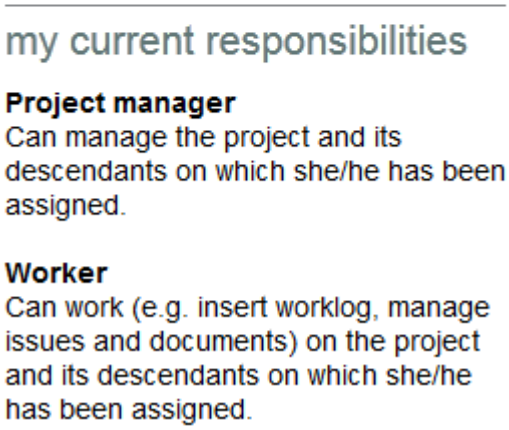
Most of the “boxes” have additional features accessible by clicking on the filter buttons.

For details about each “box” (aka portlet) see “10.4 Portlets”.

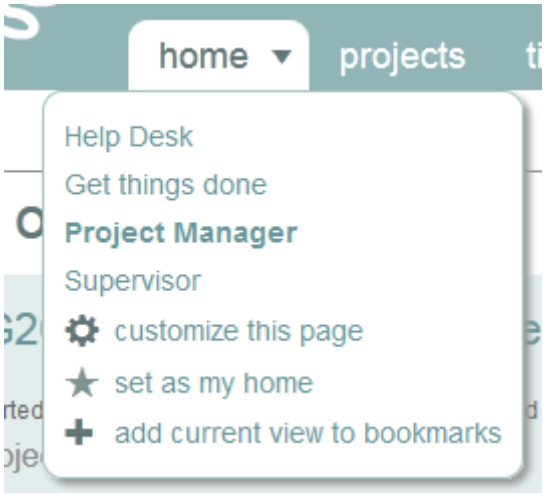
1.3.2 For the project manager – Project manager page

This page has been designed to fit project managers' (PM) needs. The central part is the same as the Get Things Done – after all PMs too need to get things done ☺ , but the right part has more monitoring parts, such as plan, activity and current responsibilities:





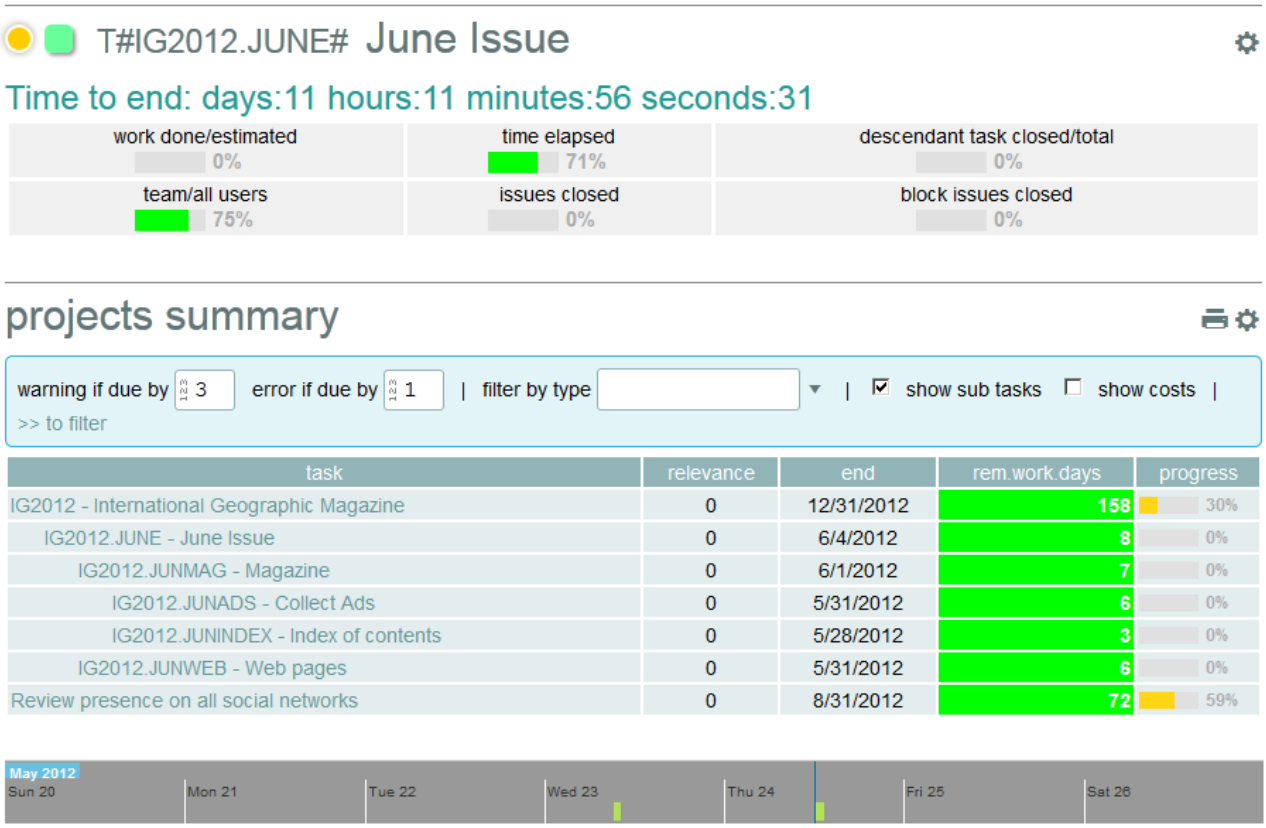
You can choose your preferred page and set it as default by opening the home menu



And pick "set as my home".

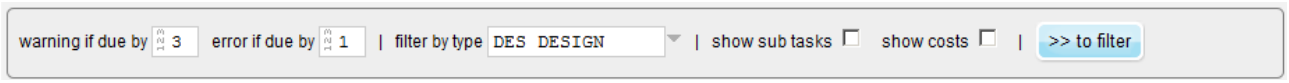
You can also customize all these pages, but for a beginner it's wiser to wait a bit, and get to know Teamwork; if you remove a crucial part just for testing, you may lose sight of relevant data.

1.3.3 For the supervisor and the stakeholders – Supervisor page



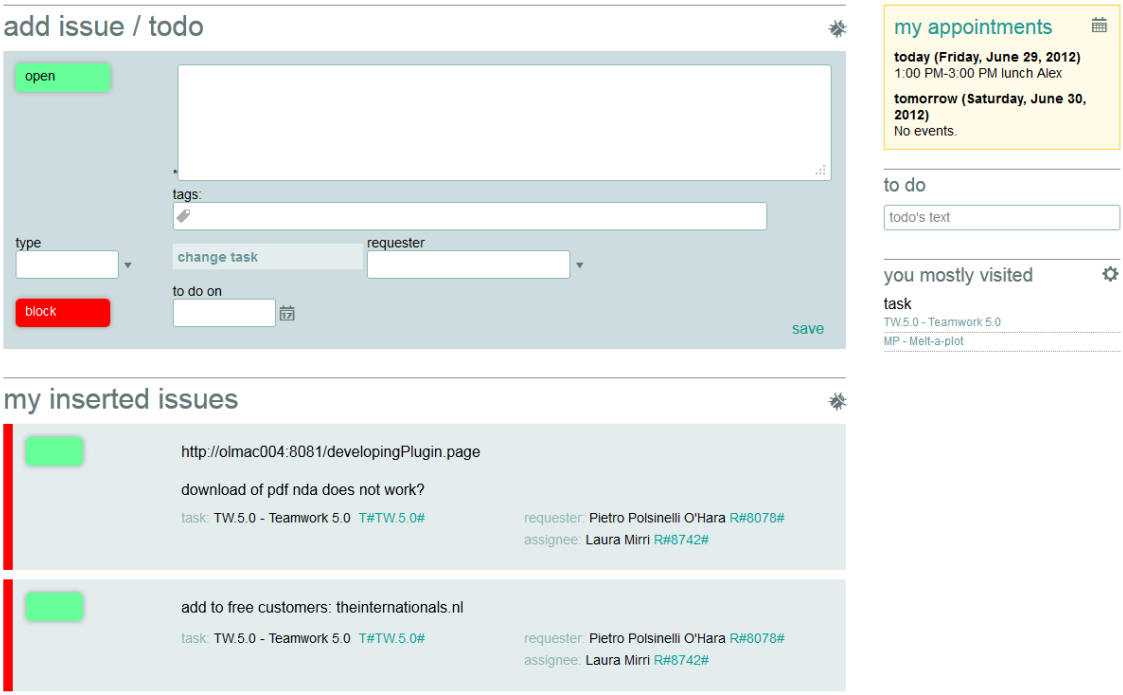
This page contains in particular the “project summary” portlet, which is a quite powerful summary of opens that are almost done or so should be.

If you click on the filter, you’ll get the filtering options:



But don’t be mislead: these ready portlets do just a fraction of the custom data filtering that you will be able to do when you’ll have learned about task filters (see section *2.5 Search / reporting*).

1.3.4 For the help desk: Help Desk page



This is a page with simplified functionality: it basically lets you simply create new issues, which then the project managers may distribute to operators if needed, raise to a second level help etc. .

2. Users as resources

2.1 Introduction

Teamwork helps organize work of groups of people: we usually refer to people as “precious resources”… – well, “resources”.

Using Teamwork you can add and then manage every resource involved in your projects. You can decide to use resource management lightly or in depth by specifying less or more data. If you decide to use it in depth, Teamwork will also keep your contact list, or better your organizational chart or “organigram(me)” (we will see this in detail in the following).

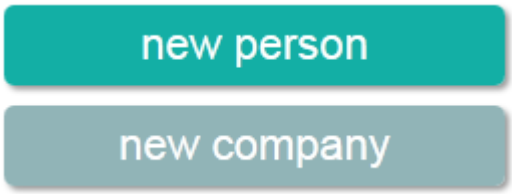
Basically a resource is defined by its name and that’s it, but if you want the resource to be operating in Teamwork, you have at least to define a login name (and a password).

Let’s see how to create a new resource by using the “resource editor”.

2.2 Editing

In order to create a resource you must have permission to do so, so to keep it easy for the moment we’ll assume that you are a super powered “administrator” (which is the default after setup).

By clicking on the “resources” button on the bar you will see your colleagues (if any) and the most important buttons for now, “new person” and “new company”



When you see “my colleagues” as title,. It means that the resources shown are **filtered**: Teamwork shows you the resources that belong to the same company as you do. It is not by default showing all resources you can see, as this may be a large and non significant selection. In order to change what is shown, click on “refine search2”.

Why two buttons? Because Teamwork manages both users and companies, and because these two entities are similar but not identical, for instance a “company” cannot act in Teamwork, cannot login, and a user cannot contain a “department”, but both can be “assigned” (see the following).

Let’s start by creating a new user: press the “new person” button:

courtesy title | name | surname*
code/short name | resource of | area IG Inc.
personal details
location | telephone
address | fax
city | mobile
state | zip | e-mail
nation | url
tags: | manager | hiring date
hourly cost | total working hours per day 7:30 | cost center

The form allows specifying all main data for a person as name (or first name), surname (or last name), e-mail addresses etc. that are common in a basic directory management. If you think this is enough just press save and your first resource will be created. Only the surname field is mandatory.

The code field is not mandatory and it is not unique.

Each mandatory field is prefixed by *. You cannot save a form without filling each mandatory field, in case you try to do so, the field will be enhanced as shown:
surname* and the action stopped.

Let's now examine a form with data inserted:

(id: 22) courtesy title | name Erik | surname* Sunk
code/short name | resource of International Geographic Inc. | area IG Inc.
personal details
home | +39 3334532423434 | esunk@exampleHome.ex
work | +39 055 234 342434 | esunk@exampleWork.ex
tags: | manager Laura Mirri | hiring date
hourly cost | total working hours per day 7:30 | cost center

Notice the three highlighted areas:

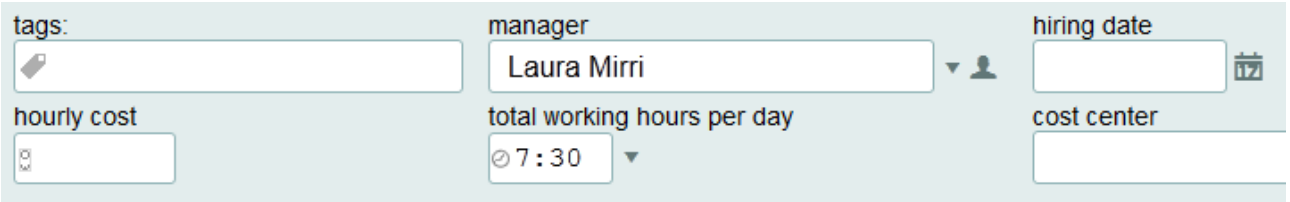
1) resource of International Geographic area this means that “Erik Sunk” is part of the “International Geographic” company. This is how Teamwork constitutes an organigram, as a tree.
Later we will have a look from the “International Geographic” point-of-view.

home

work

2) Teamwork can record more than one personal data. In this box you will see the list of locations. From the list you can remove an entry by clicking the button, or you can sort the entries by using buttons. The order is relevant in the sense that the default resource’ e-mail is the first found fetching the personal data list.

3) manager Graham Greene by specifying the manager you will keep track of resources’ hierarchy.

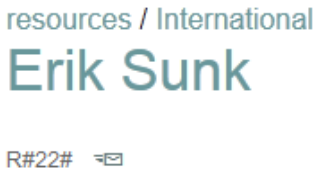


“Total working hours per day” is an important value, as it will determine the load of the user w.r.t. to task and issues assigned.

You can print or get a vCard file (vCard is a standard format for contact exchange) of your resource.

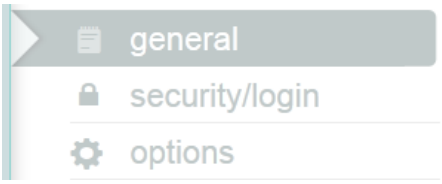
It’s probably easy to guess the meaning of the other fields. Notice only the “area” combo that allows setting your resource’ security environment. For the moment we are setting up a simple mono-area environment. To extend Teamwork’ security model see chapter “12 Security”.

Notice the R#22# which is a resource unique code that allows to link to it from any text in Teamwork (more on this in *13.5 Internal links*).

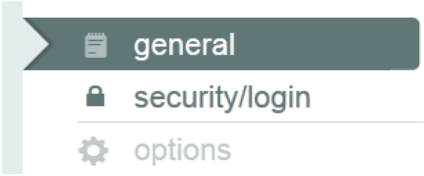


A last note before moving to the security/login view, about the sidebar: when you are creating a resource, buttons change:

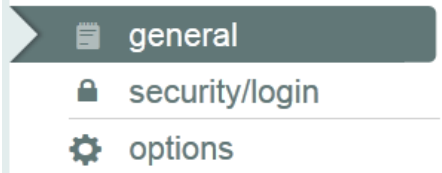
1) Creating



2) Editing a resource without login



3) Editing a resource with login – has options:

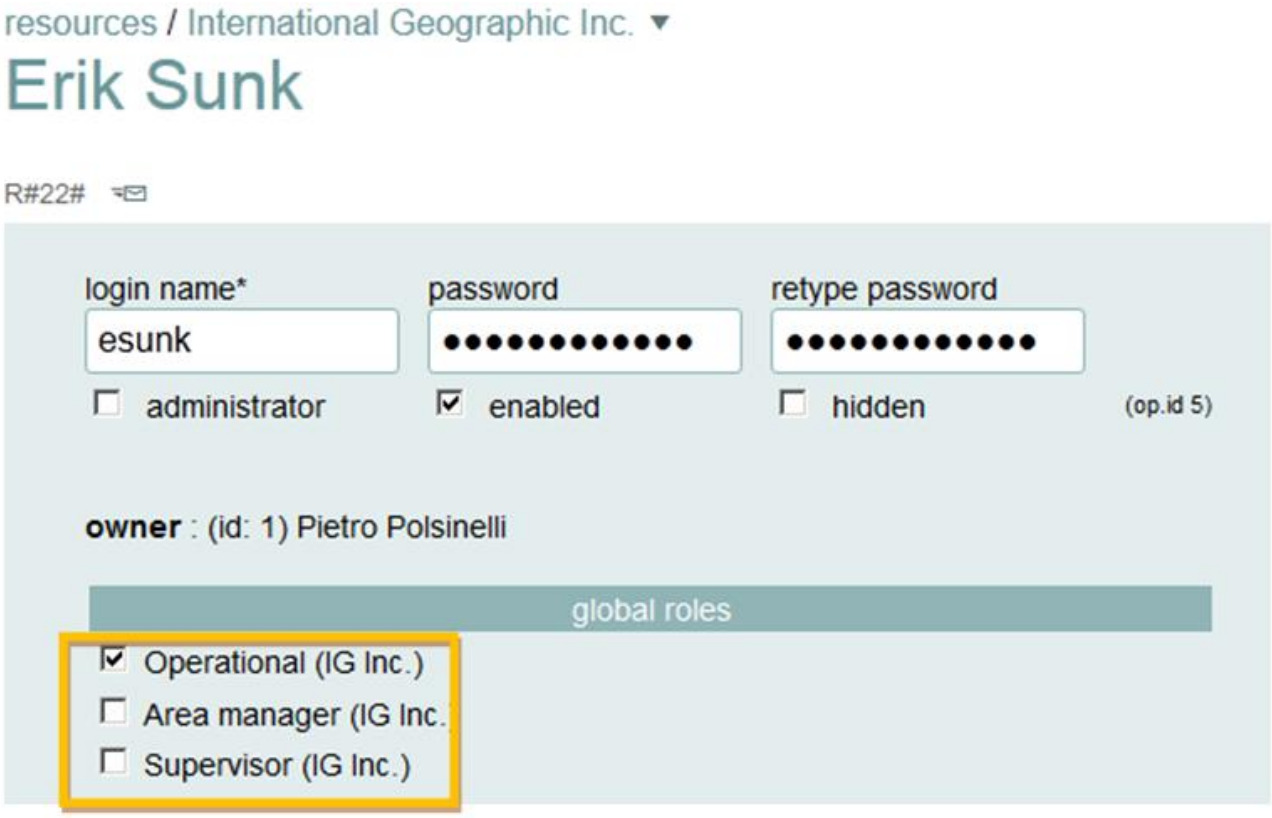


The other buttons will be active once saved. In the following we will examine all links except “documents” and “assignment” that will be treated in their respective section (chapter *9 Document* and section *3.2.1 Assignment*).

2.3 Teamwork’ resources with login: users

If you define a login name and a password, the generic resource becomes a Teamwork’ user and then can login and operate in Teamwork. You may assign resources to projects and issues even if they don’t have a login.

Let’s show the “security/login” tab:



Login name **must** be unique.

A user can be hidden (administrators only can see it) and/or disabled (cannot login anymore). When a user is disabled all data inserted or related to the user is preserved. You can re-enable disabled users when needed.

In the highlighted box above you can choose one or more “global roles”. In the first usage “easy mode” you should check “Operational” on every user you create; this is a role created by default during installation, that provides the minimal set of permissions required to operate comfortably; or you could even more roughly set every user as “administrator” and bypass security in the beginning.

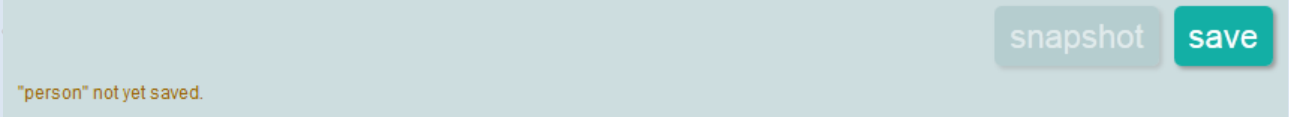
You can define as many global roles as you need but if you want to use the Teamwork’ security in deep, see chapter *12 Security*.

If you have a special security policy regarding password length or expiration times, see *16.7.1 Teamwork security*. If you want to use your LDAP/ Active directory to import/authenticate user see *16.8 LDAP/Active Directory*.

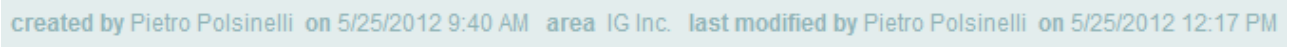
Passwords are not recoverable from the database, so there is no way to recover a forgotten password; eventually, just reset it.

Disabled users are not counted in the license total; so if you have 10 active users and 50 disabled ones (e.g.: temporary employees for specific projects than do not currently work with you), you only need a 10 user license.

When you are creating **a new resource** (actually this works for every important entity) the button bar reports the current status of the object:

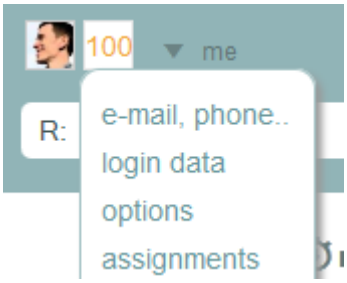


When the object is already saved the status appears slightly different by reporting creation and last change dates:

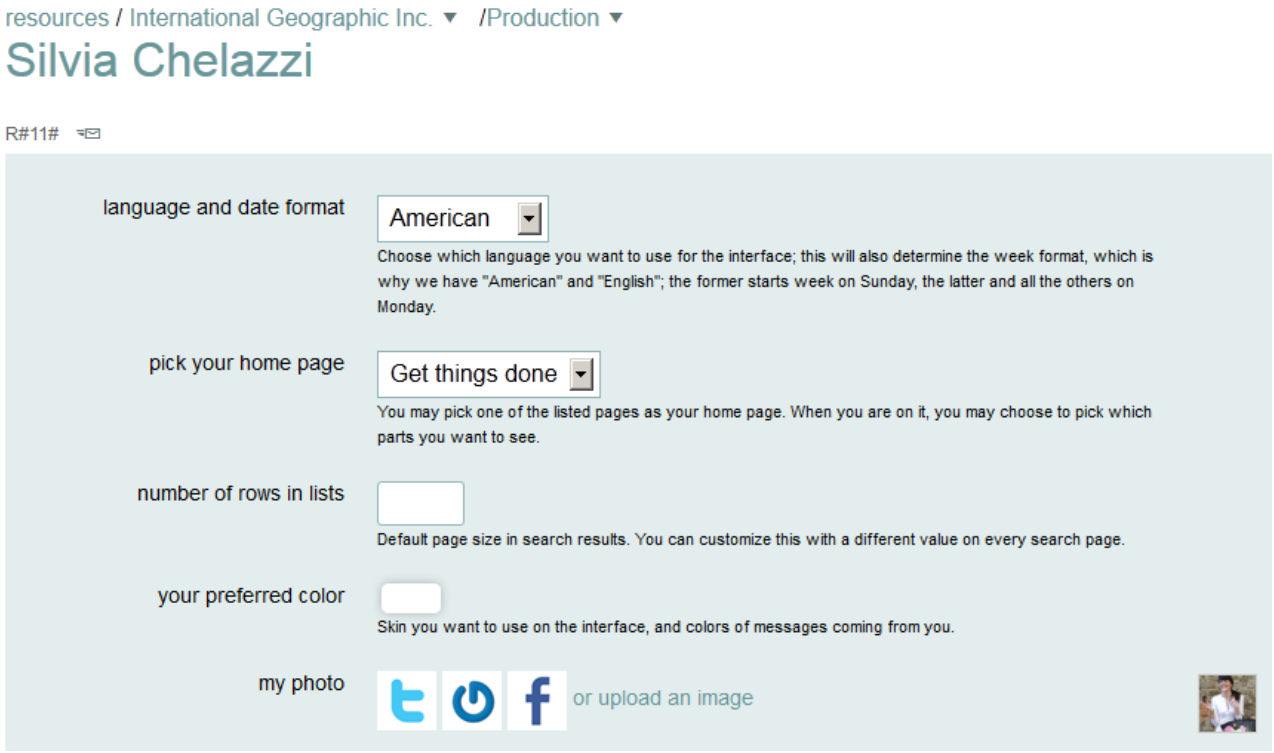


How do I access my own profile????

Just click on the image icon, a menu will appear:



Every operator can customize Teamwork' behavior. Let's have a look at the "options" page:



Every user can customize her/his Teamwork experience by setting image, interface language, her default home page, default color of notes from myself, etc.

Some of this data is relevant for Teamwork's management and we will meet it again later.

> If you are surprised that there are both English and American as language options, this is because in addition to the interface language with this you set the default format for dates, which are different in these two countries.

The meaning of the page "subscriptions" will be clearer later when we will talk about assignments, but for the moment, just remember that Teamwork has an event based subscription engine, so you can be subscribed to listen to a specific event. For example, if you are "subscribed" to a task via e-mail, and the task gets closed, you get a notification e-mail. In this page all your subscriptions are listed.

resources International Geographic Inc. ▼ Production ▼

Pietro Polsinelli

R#1#

		event	media	
1	IG2012 - International Geographic Magazine	task status change	digest,sticky	⊗
2	IG2012 - International Geographic Magazine	task dates change	digest,sticky	⊗
3	IG2012 - International Geographic Magazine	milestone closer	digest,sticky	⊗
4	IG2012 - International Geographic Magazine	task/issue expired	digest,sticky	⊗
5	IG2012 - International Geographic Magazine	added issue	digest,sticky	⊗
6	IG2012 - International Geographic Magazine	closed issue on task	digest,sticky	⊗
7	IG2012 - International Geographic Magazine	issue updated	digest,sticky	⊗
8	IG2012 - International Geographic Magazine	diary change	digest,sticky	⊗
9	Contact Robert Capa for the special images for the web edition	set status "closed"	email,digest,sticky	⊗

You can also remove subscriptions by clicking on the button, and all of them with the "remove all subscriptions" button.

2.4 Company/Department

Companies and departments are both resources and in Teamwork there is no substantial difference between them.

You can create a company with many departments and each department can contain as many resources (yes, resources that can be persons or companies or departments); by structuring your company you get your organigram and some other nice features, like workgroups, that will be described in the following.

Take a look at a company with data fulfilled:

International Geographic Inc. ▼

R#IG#

name* International Geographic Inc.

type Company ▼ add

code/short name IG

resource of

area IG Inc.

personal details

TOPO! Digital Maps
212 Beaver Brook Canyon Road Evergreen
CO 80439
800 010101

Catalog Inquiries
777 South State Road 7 Margate
FL 33068
800 020202
catalog@internationalGeo.ex

tags:

manager

hourly cost 0.00$

total working hours per day 0:00

cost center

job description

note

The International Geographic Society has been inspiring people to care about the planet since 1888. It is one of the largest non-profit scientific and educational institutions in the world. Its interests include geography, archaeology and natural science, the promotion of environmental and historical conservation.

The form is similar to the “person” one, and on the sidebar you get the list of connected resources, list that contains both persons and departments.

An easy way to move across the tree is to use the top navigation bar by “stopping the mouse on” or clicking ▼ :

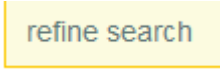
The navigation bar is present whenever the entity is a tree (resources or tasks)

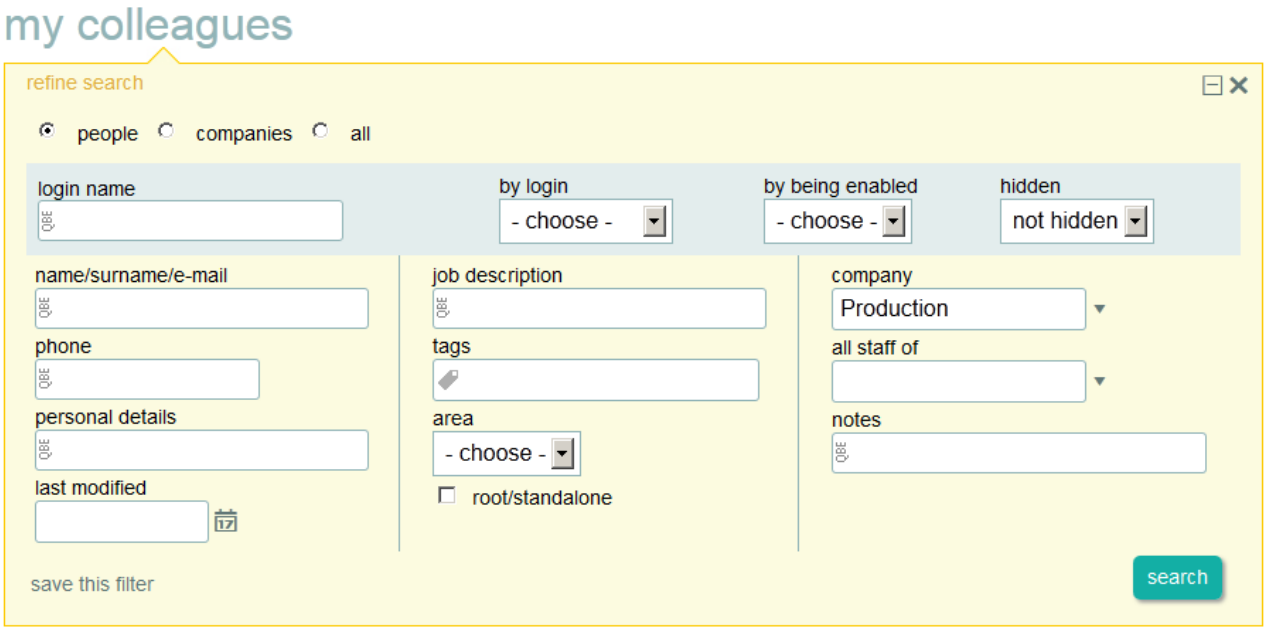
2.5 Search / reporting

In order to retrieve resources just click on the “resources” link in the top bar:

By default you will see your colleagues, but there are already several useful preset filters “Teamwork’ users”, “companies” and “people”.

Not enough? Press “refine search” to get more power:



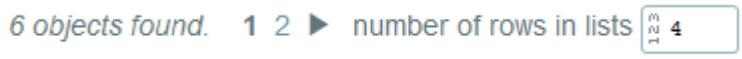


This filter lets you compose refined search conditions – click search to get it going .

Each field is used in logical AND with the others, so for instance if you set the “company” and check the radio “people” you will find only people (not departments) from that company, that in this case means exactly “my colleagues”. In fact every preset filter simply fills the respective fields in the form so you can use them as a guide for composing your own.

Results are usually paged, and you can flip pages by using the paging bar:

Page size can be changed, just click ⚙ then in the field



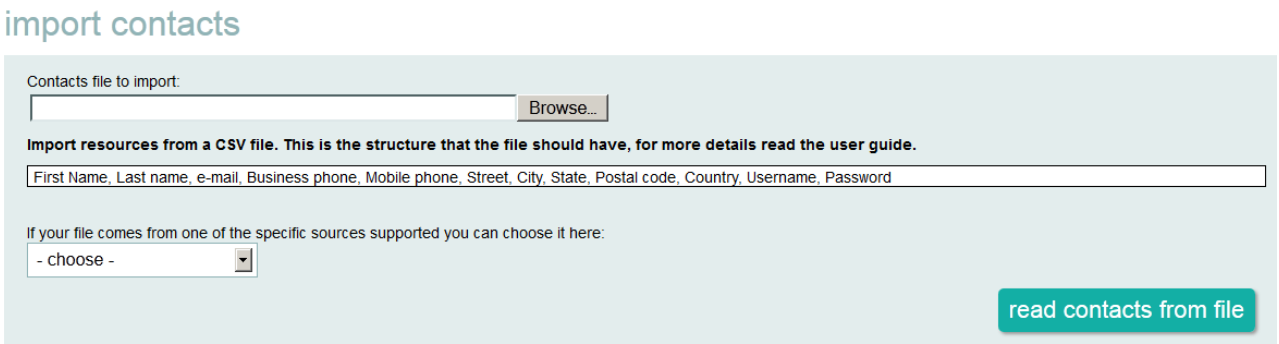
and type a number.

Custom filters are powerful also because teamwork supports Query By Example (QBE) and allows you to store your filters in order to quick repeat complex searches. For more details about filtering and QBE see chapter “13.2 Custom filters and QBE”.

This page, like most Teamwork's list pages, allows printing and exporting to Excel: click in the upper right corner to export in Excel format. The print button will print only the resources shown.

2.6 Importing contacts

Go to resources -> -> import contacts:



Import from a CSV file: the CSV format: you should produce a CSV file with “holes”, as we rely on Thunderbird’s format.

Fields imported and their "column" position:

First_Name -> 1

Last_Name -> 2

E__mail_Address -> 5

Business_Phone -> 7

Mobile_Phone -> 11

Business_Street -> 18

Business_City -> 20

Business_State -> 21

Business_Postal_Code -> 22

Business_Country -> 23

Here is an example line:

Leonardo,Da Vinci,,,davinci@open-lab.com,,055 work phone,,,,333 mobile,,,,,,Via di Vinci 2,,Florence,Tuscany,50100,Italy,,,,,,,,,,,,,

This will parse the file, page the results, and for each line tell you whether an existing resource is found, and you can choose whether to import the line or not, by selecting a checkbox, and choosing the "import selected" button. We had to ask confirmation for each line, in order not to damage existing data. Remember to go through all pages of the result.

You will need to create login data for those users that can log in.

3. Projects and tasks

3.1 Introduction

Projects or tasks? There is no difference in Teamwork. Teamwork lets you break down every project/task in an arbitrary (but reasonably sized ☺) sub-tree of sub-tasks/projects.

We could use the convention of defining “project” to be a top-level task (the root) and simply “task” the others, but in this chapter we will use “task” for both, as it also depends on how you look at a tree, and tasks could become projects and *vice-versa*.

What is the simplest task? In Teamwork a task is simply defined by its name. Of course if you want the task to be useful you have to specify a starting date, duration, and at least a resource that will work/manage/supervise it – in project management terms, a resource assign to the task.

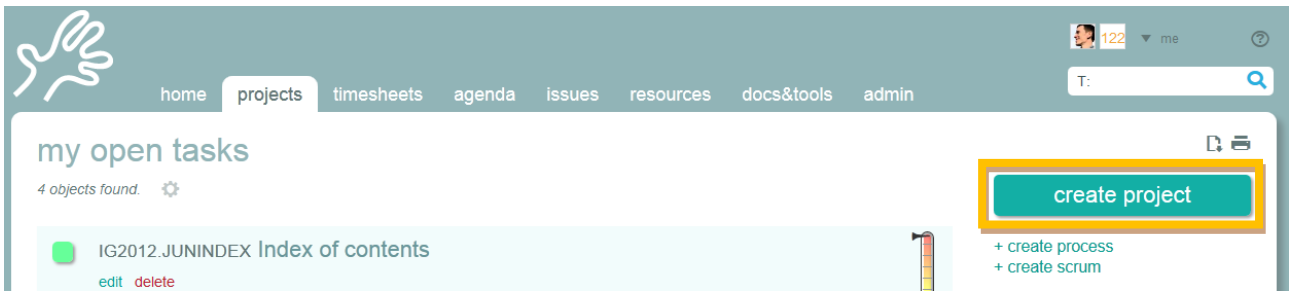
So name, timing and assignments are the main properties of a Teamwork task.

Try it out by creating your first project.

3.2 Editing

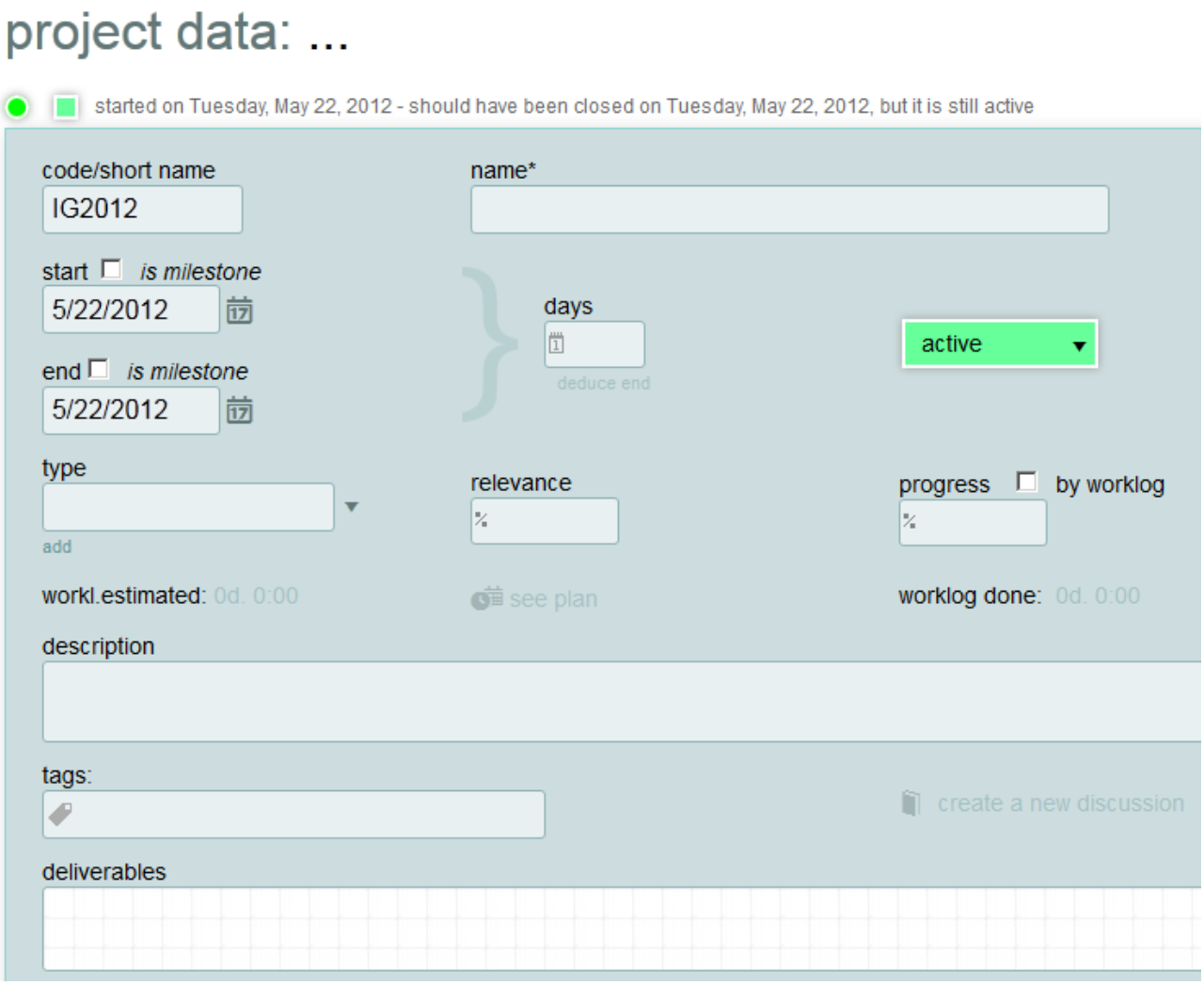
In order to create a task you must have permission to do so, to keep things easy we assume that you are an “administrator” user. Area managers can create projects too; by delegation, project managers need just to be assigned on a root task, in order to be able to create the entire project tree. More on this later.

By clicking on the “projects” on the menu bar you will see by default your open tasks (if any), and you will also have available the “create project” button:



There are several ways to create new tasks, as these cover also business processes and Scrum (an Agile methodology). We will later examine these refined models, for the moment we’ll focus on the simplest way (when you want to go more in depth see chapter 15 “Advanced business process and 11 Agile / SCRUM”).

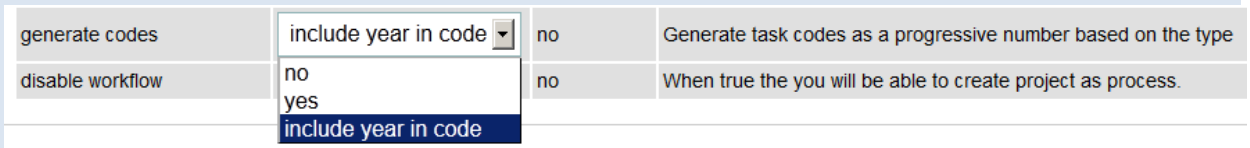
The easiest way to create a task is to press “create project” button. Let’s press it, obtaining:



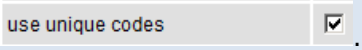
Task' general data summarizes its main properties. In detail:

1) **Code/short name**: this is usually the mnemonic name of the project/task. If your projects are related to external systems (such as an accounting systems) you can use that code. Code is not strictly required to be unique, but having a unique code may be useful for example for referring it by e-mail (see "5.4 Manage by e-mail"). When you are creating sub-tasks the code is proposed automatically by default as code.1, code.2 and so on, but if you change the proposed code with code.a or code.I Teamwork will try to guess the following codes with code.b code.c or code.II and code.III.

- Progressive number guessing is used widely in Teamwork.
- If you change a task code, and the task already has children, the children (and descendants) codes will not be updated, as this may not be what is expected, and Teamwork has no way to tell.
- Teamwork can generate unique codes from task types: if you select a type, didn't type a code, and enabled this options in configuration:



- You can set codes to be unique, and it can be a quite comfortable choice. To set this, go to

admin -> default for projects and enable “use unique codes” 

.

2) **Name**: it is the common/descriptive name of the task. It is mandatory and it is not required unique.

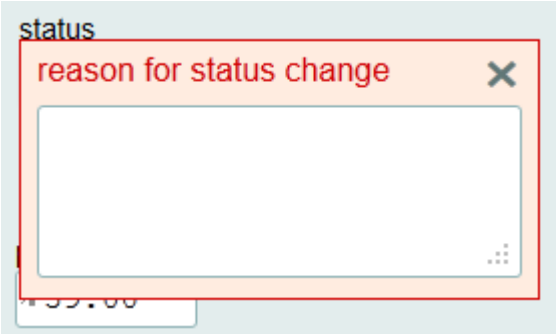
Each mandatory field is prefixed by *. You cannot save a form without filling each mandatory field, in case you try to do it the field will be enhanced as shown and the action is stopped.

3) **Type**: is a rough classification of the task, “production” and “template” are commonly used types. You can add how many types you need by pressing the add button (if you have the permission). Are you wondering what “template” means? Teamwork supports copying a whole project tree with many options, so it is very easy to create a project skeleton as template. But actually any task can be used as “template” (why not?), and be used for “generating” other tasks (see “3.13 Copy/move projects”).

Task types can also be a base for generating task codes: see above and 16.5.3 Project defaults

4) **Tags:** Similarly to the resources case, “tags” is here used to classify your tasks and group them. Tags can be used to filter tasks when searching.

5) **Status:** Task’ status management in Teamwork differs slightly from other project management software in the sense that Teamwork is more… reality driven! To be more explicit, task status and task dates are not necessarily related. In Teamwork it is legal to have an “open” task after the task end date: we believe that this approach is more “real” and practical with respect to closing a task automatically on the end date: the latter may have been fixed a lot of time before, and much may have changed in the meantime. Of course task status must obey some rules related to dependencies, business processes (see “3.10 D”), or status flow, but generally with Teamwork you have more flexibility than with traditional project management software.
If you are changing the task status, a popup will invite you to insert a reason for the change:

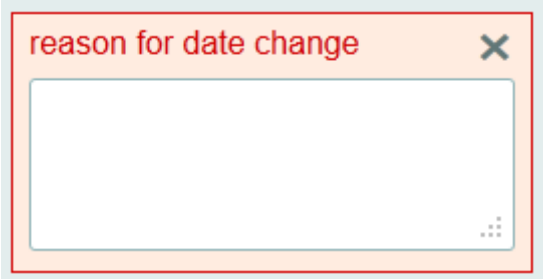


It is not mandatory to fill it. Every change is recorded on the task logs and the event is sent to the notification engine.

6) **Start/duration/end**: as stated before, task timing is one of most relevant information about tasks. Start, end and duration are always consistent; this means that if you specify start and duration the end is computed, or if you specify start and end, duration is computed. Note that duration is computed in “working days”, considering (or not) Saturdays, Sundays (or Fridays) and your company’s calendar

(in order to configure these parameters see section 16.5.2 Holidays).

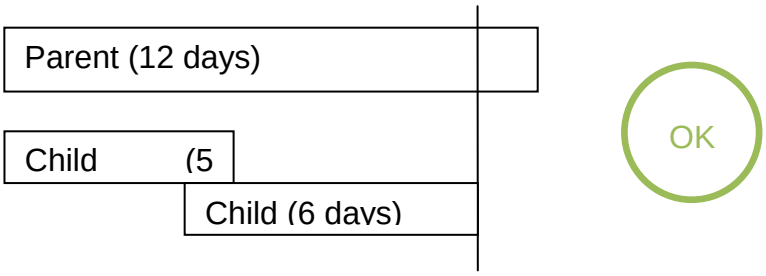
If you are changing dates for an existing task, a popup will propose to insert a reason for the change:

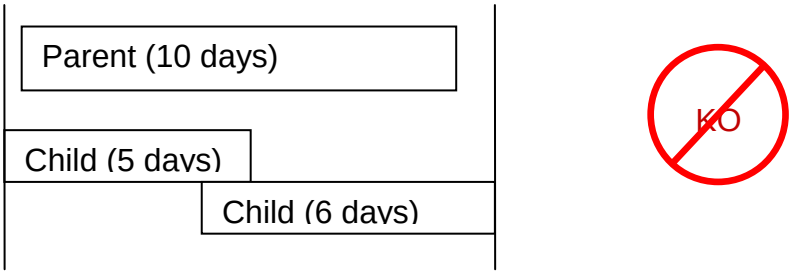


it is not mandatory to fill it. Every change will be recorded on the task logs and the event is sent to the notification engine.

7) **Milestone**: in Teamwork a milestone always matches the start or the end of a task. This because usually a milestone is related to a delivery or a phase kick-off. Of course setting a milestone changes the behavior of the start/duration/end triplet; for instance if you set start and end as milestones, duration is disabled.

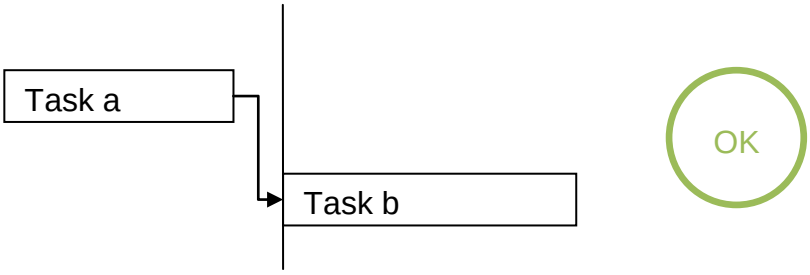
That's all with dates? Not exactly, there are other cases where dates must obey constraints. For instance when a project has children: in this case the parent task duration must be equal or greater than its children duration, and start must be the minimal date.

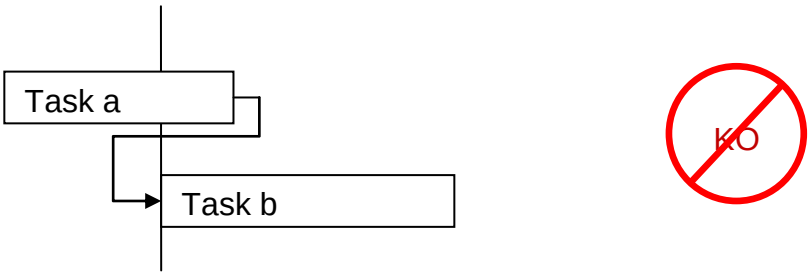




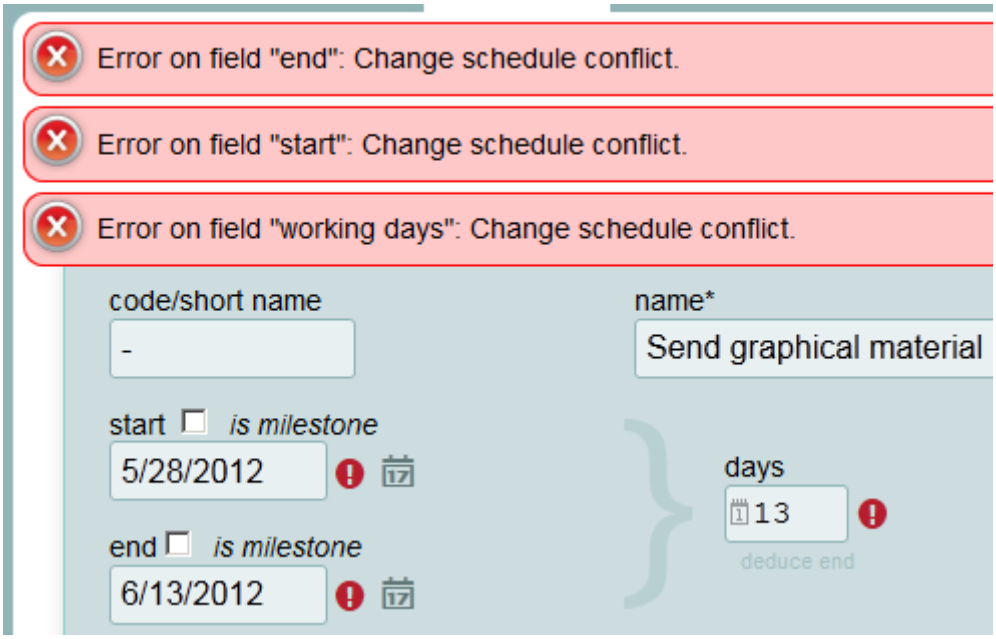
In this case if you try to shrink the duration of the parent, Teamwork automatically sets the minimum duration as compatible with that of the children.

8) Another case is when there are dependencies between tasks: in this case a following task cannot start before the end of preceding one.





in case of dependencies, when you change dates/duration you may cause the propagation of date changes from children to parent along the whole tree.
If your tree in “pinned” by one or more milestones the propagation may fail: in this case you get an error message feedback:



9) **Progress**: is the percentage of completeness. Teamwork usually does not perform **any** computation on this number; for instance if your parent task is composed by two children that are both 40% done, that does not mean that the parent itself should be 40%. You are free to insert any (valid) percentage.
We believe that the PM can assign a significant value to progress only by being “feeling” driven rather than that “mathematically” driven. Of course you can use the statistics box to get a… *better feeling* ☺.

There is one case only when Teamwork computes progress automatically; if you check “by worklog[2]”. In this case progress is computed as estimated effort (in working hours see 3.2.1 Assignment) divided by effective work done.

If you set the option in configuration (admin -> default project role names):

progress 100%	☑	no	Set progress to 100% when closing a task.

progress will be set to 100% when the task is closed, if progress by worklog is not active.

10) **Relevance**: is how important you feel this project is for you/your company. No computation, no constraints.

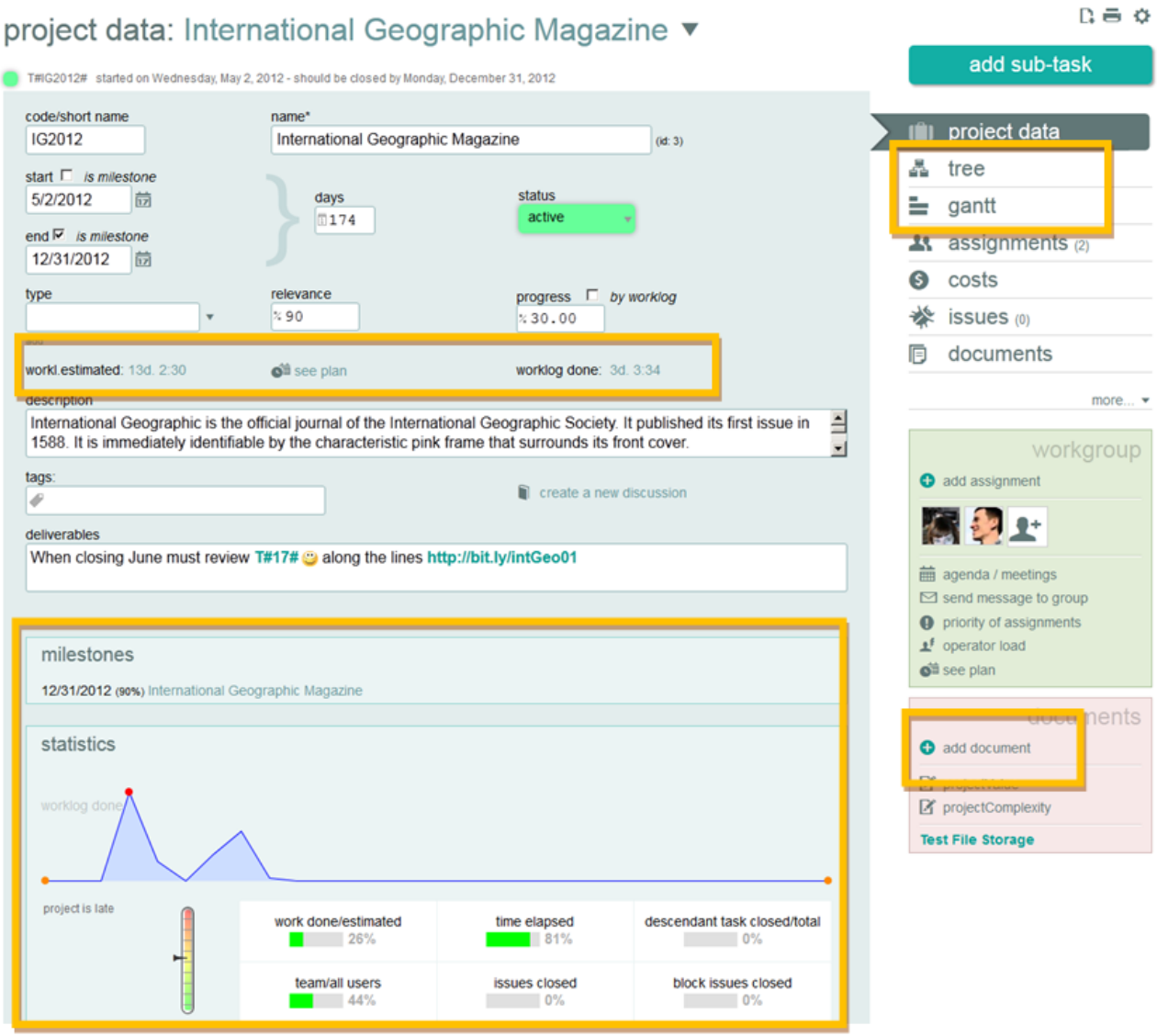
11) **Description/deliverables**: use them for describing projects and delivery. No

[2] “worklog” is a single record of work report. We will often use this term in the following.

constraint on content except size.

Types can be used as key property to activate custom forms or wizards. See “14 Plugins, custom fields/forms” to explore this powerful feature.

Let’s take a look to a saved project form:



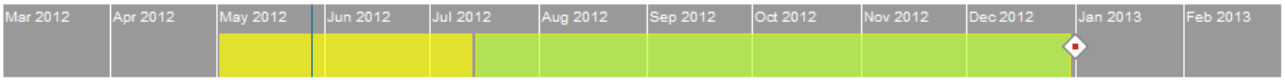
First box contains data about worklog estimated and done. Worklogs are related to assignments (see “3.2.1 Assignment”). From here you can also access to the task plan.

The tree link leads to a simple tree view that allows basic editing; the task Gantt leads to a complete Gantt editor – will see that in a dedicated section “”.

The statistics box summarizes every figure relevant for the task; data in this box will give you a “feel” for the real progress of the task.

Teamwork supports custom forms , the samples provided are “project value” and “project complexity”; if some of these are filled you will see them here, as links (see 14 Plugins, custom fields/forms).

The last information available in the task editor page is the “task summary bar”:



In this bar you will see the task dates scope, eventually the milestones (the diamonds),

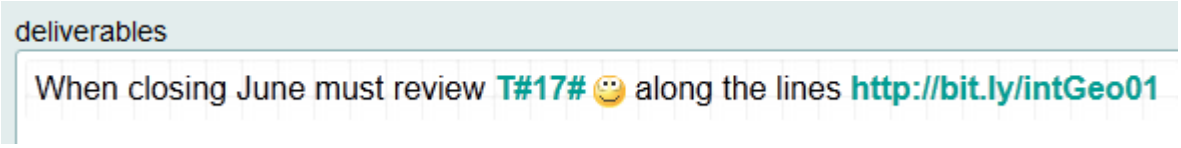
progress (the yellow part), and today (the blue bar).

By comparing progress and today you can see immediately if you are in late or not, just by looking whether the blue line is on green or yellow: in our example we are perfectly in time.

Notice in this top part of the task editor on the left that there a task internal “code”, which every saved task has: T#IG2012#. This code allows you to link to this task editor from every other editor in Teamwork. For example, if on any (other) task editor in the “description” or “deliverables” fields you write such code, this will turn into a link. In this way you can cross link any task to any other, and more – see below.

Cross linking supports links between: tasks (T#CODE#), issues (I#CODE #), resources (R#CODE #), agenda events (E#CODE #), agenda meetings (M#CODE #), (B#ID#) boards, where the value of “CODE” is shown in the editors.

Actually the text in the text area supports also HTTP links, images and smileys: you can get a description like this one:



with active links.

In order to create child tasks just press “add sub-task” button:

But if you want to create a complex tree structure at once see “3.5 Gantt editor”

Hints - task scheduling: start, duration, end

Teamwork supports several strategies for setting dates and durations on tasks. It is a non trivial matter; here we explain how it is modeled in Teamwork.

By duration of a task is meant duration in working days. Whether Saturdays and Sundays are working days is set in global settings (see *16.6.1 Internationalization*).

To set general holidays, go to *16.5.2 Holidays* (you must be administrator).

The duration of a task is considered the reference setting, and a task end is completely determined by start and duration. The duration of a task which has children cannot be shorter than the computed duration of all descendants. Hence the leaf tasks determine the least scheduling of the entire tree, which overall may be wider. In case of dependencies, the "leftmost" tasks determine the start dates of the tree, from that level upwards. Leaf and "leftmost" tasks of dependencies can be thought of as "leading" tasks with respect to setting scheduling, as it is intuitively.

When start and/or end dates should prevail on duration, they should be set as milestones. Before any change of dates or duration is set on a task, a complete verification of the effects of such change is done, navigating the task tree and all dependencies; in case the modification is not possible, as it would alter a milestone or violate a security setting, an alert is shown (as we previously seen).

Scenario. What is currently known are just the durations of the tasks.
Solution. Proceed by setting only durations: Teamwork will propose "today" as start date for the first task: its fine. Any time you can set the real time on the leading tasks, all other tasks scheduling will adapt automatically.

Scenario. The task end date must not be changed.
Solution. Set such date as milestone. No automated changes will reset such date.

Question. Why the start or duration or end of a task is read only?
Solution. Start is read only if the task has dependencies. Duration is read only if the task has children or both start and end are milestones. End is read only if the task has children or there are tasks which depend on the current one.

Question. I want to set start and end by hand but darn duration keeps resetting end!
Solution. Empty the duration field before setting end.

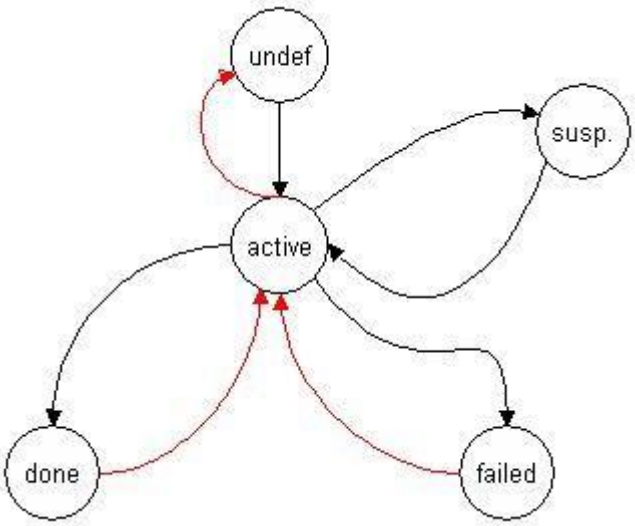
Question. I move the start of a task which has descendants, but it has no effects.
Solution. You should move the start of all the "leftmost" children of such task.

Some remarks on task status change

Status change is a critical point of project evolution; hence they are carefully monitored and consequences exactly defined. Status change behavior is connected to the task tree and eventual dependencies. Status changes are distinct subscribable events on task.

Possible task statuses are:

undefined - active - suspended - done - failed.



In the picture, project statuses: in black "standard" transitions, in red "non-standard" ones.

Standard transitions

undefined -> active

all children become active, if they have no dependencies.

active -> suspended

all active children and their active descendants become suspended.

suspended -> active

sets to active all children and their descendants that have no inhibiting dependencies.

active -> done

may activate dependent tasks, both suspended and undefined. If there is a child not done, the interface alerts before proceeding, and if ok will set to done all descendants.

active -> failed

mark all not done descendants as failed.

Non-standard transitions

active -> undefined

all descendants become undefined.

done -> active

all those that have dependencies must be set to suspended.

failed -> active

nothing happens: child statuses must be reset by hand.

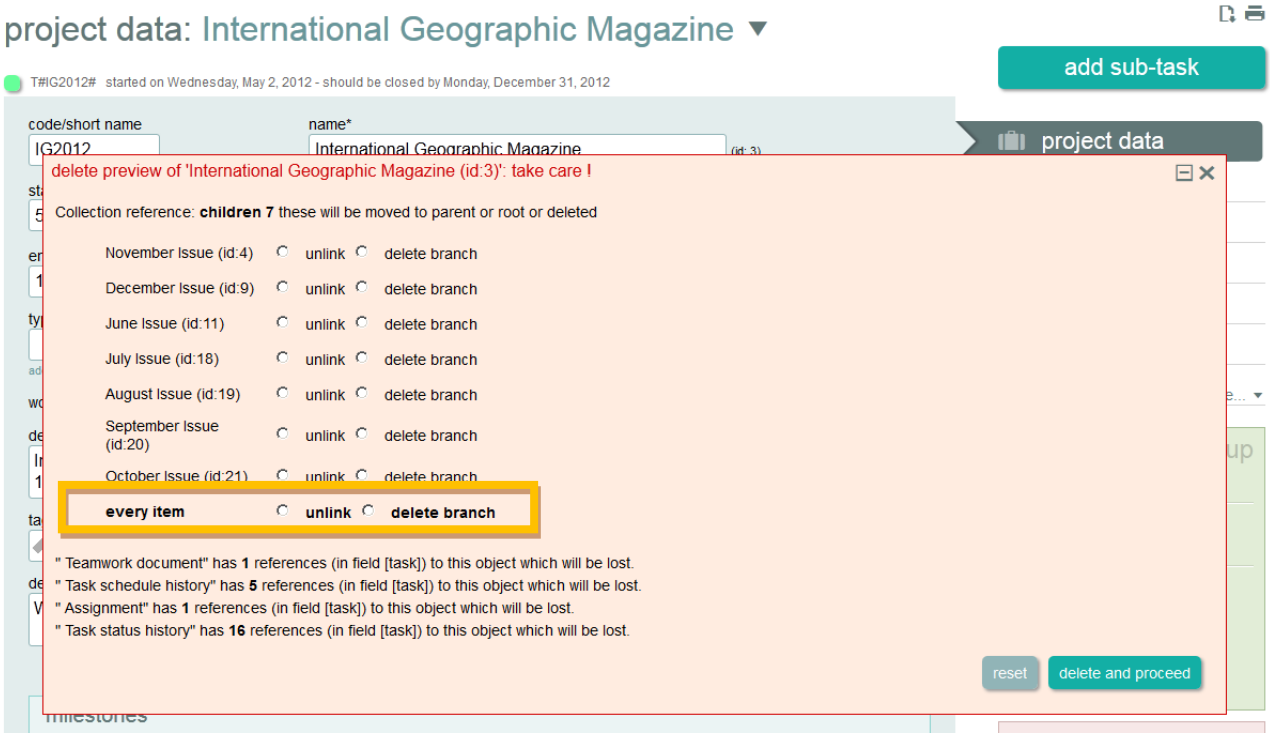
All other combinations are got by composition.

Removing tasks

What happens if I need to remove a project from Teamwork?

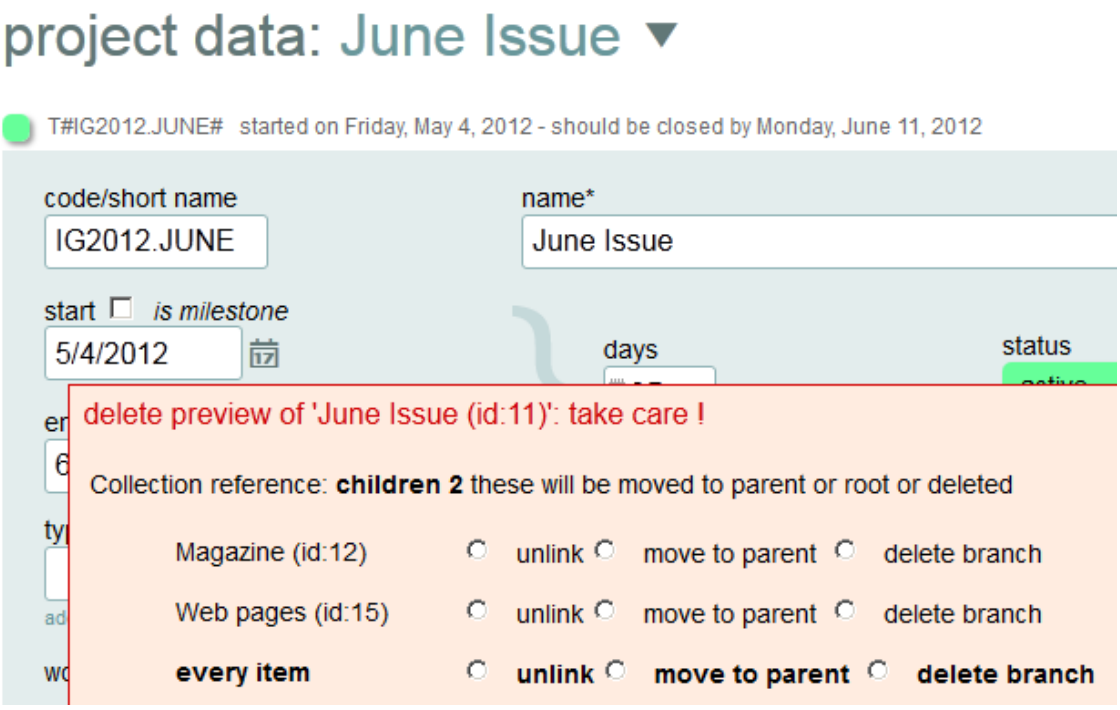
Usually removing complex objects such tasks or resources can have several consequences; most Teamwork' objects are linked to each other (a resource is assigned on a task with a role, a task is part of a tree, there is worklog already inserted, there are other users listening for events and much more…), so before deleting an object Teamwork will check the state and show you the real situation.

Let's try to delete a root task.



In this case Teamwork will tell you that this task has 7 direct children, and you have to choose what to do with them.

In case you go to the delete preview of a task at an intermediate level:



You can:

1. **unlink**: this means that children will be “promoted” as root task
2. **move to parent**: task’ children will become children of task’ parent (children will pulled up one level)
3. **delete branch:** this means that deletion will propagate to the descendants

You can select the radio to apply you action to “every item”.

The highlighted area reports the object linked that will be lost while removing the task. If you are not worried by the message ☺ you can select one option and try to remove the task definitively by clicking on

.

There are some cases where deletion may fail, when the task has “live” references from external objects, references which cannot be removed by default; when this happens, the deletion preview pops-up again with the error message.

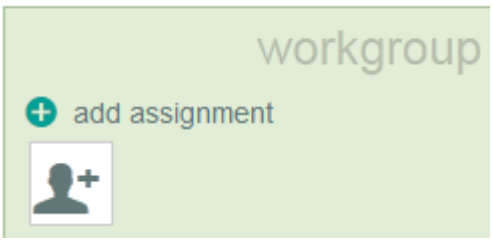
Every time you delete a main object (task, resource, issue, agenda event etc.), Teamwork will show the deletion preview.

3.2.1 Assignment

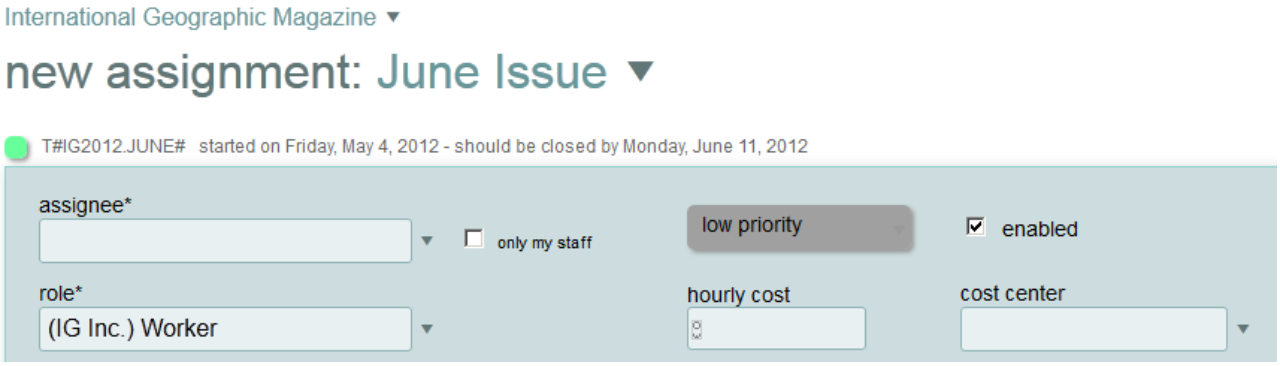
Once a task is created we have to assign one or more person working on it.

Assignments in Teamwork are very important: when you assign a resource you will also implicitly define security settings of the task. In fact every assignment is “weighted” by a “local role” that gives permissions to the assignee (see 15 Security for a complete overview). Permissions are, by default, propagated to descendants, so if you are PM on the root you have PM permissions everywhere, but if you are assigned on a child you can act on the child, not on the parent. This makes Teamwork security very fine-grained, without the stress of managing permissions separately.

The easiest way to do this is to click the “add assignment” button:



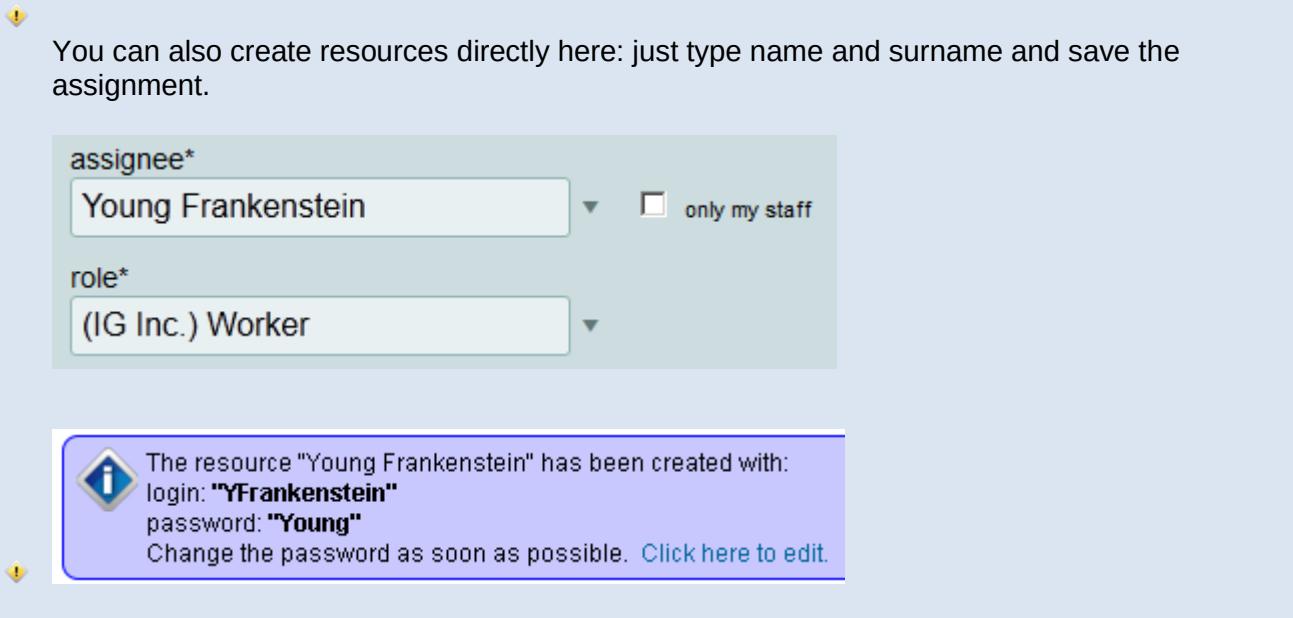
You can choose to add yourself on the task, add a “project manager”, or add someone else:



This is the assignment editor page. Here you have to choose a resource, and the role played in this task; both are mandatory.

> Some basic roles are created during setup, but you can define new ones from the administration page in the security area (see 12.3 Roles).

The resource chosen is the assignee.



Then you can describe the activity, assign a priority (see 6.4 Priority for details), specify an hourly cost and a cost center.

Regarding the estimation of required work, the assignment is the place where it is mostly done; it allows a very fine grained and practical estimation entry point. Notice that directly from this editor you can check the current load of the assignee and check the planned work if you are using that module.

Load by issues: effective when working in an “agile” environment, and in general by issues (see chapter 4 Issues).Click on the link to see issues on which work has been estimated.

There are three more fields that require some explanation:

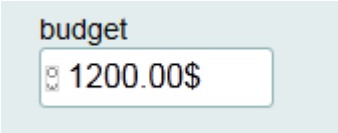
1) **Activity**: could be “all in one interval” or “routine”. Routine activities can be recorded in Teamwork without interfering with the “all-in-one-interval” main ones that usually have more focus and relevance.
2) **Enabled**: in some cases a resource can have long-standing task assigned that don’t require activity for a long time. In this case you can disable the assignment (the only effect is that it will be hidden in your working pages like home page, timesheets or “my tasks”).
3) **Risk**: this is the risk percentage assigned to the resource relatively to this task. This is used mainly when the assignee is at least in some sense a stakeholder, and you have more than one that are sharing the risk.

The **cost** field reports hourly cost for the chosen resource on this assignment. If you create a new assignment, the price set in global settings is first copied in the assignment editor, but as soon as you pick a resource for the assignment which has a cost set, you will see the cost field updated.

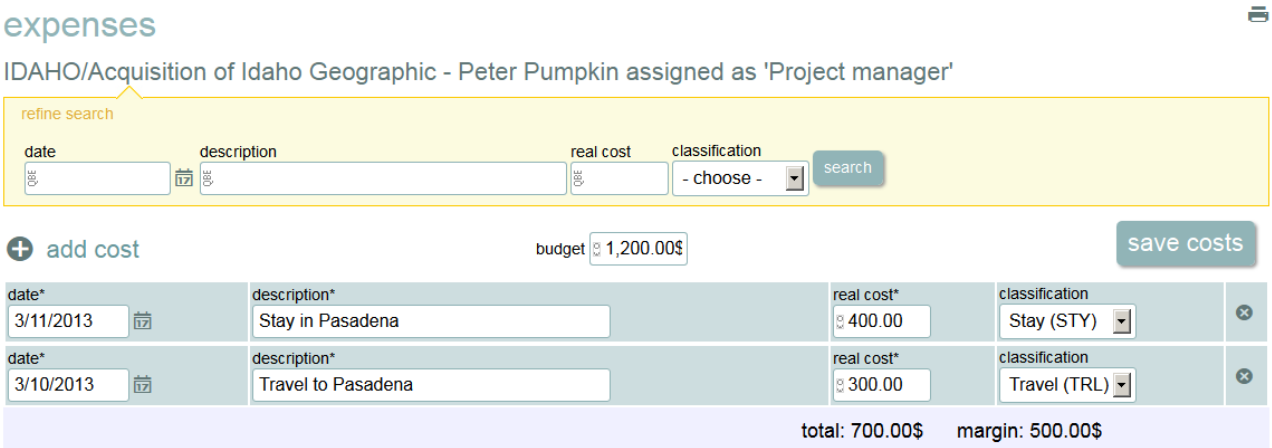
The hourly rate is copied into the assignment at creation, but if you update the general

price of the resource, the already existing assignments will intentionally not be updated, as you may have set that price as an exception. Our idea is that the same resource may have different costs on different assignments, as it is often the case in the real world.

3.2.2 Expenses

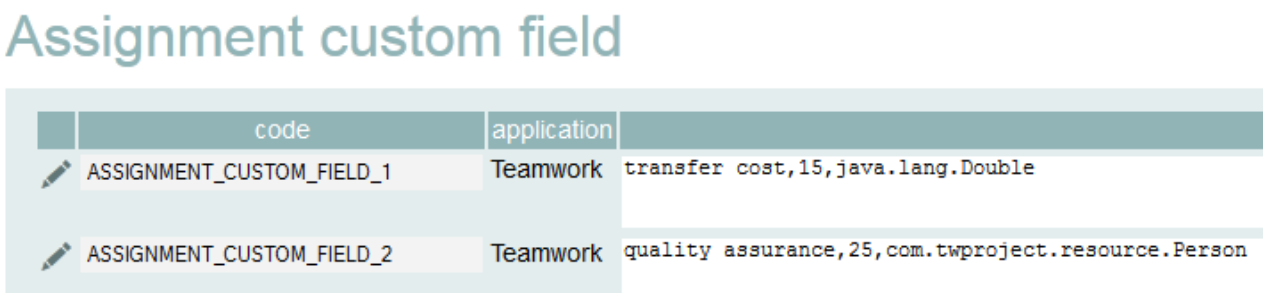


On every assignment you can insert specific “expenses” and a budget. Expenses can be inserted both from the assignment editor and also from worklog week (this link can be disabled by configuration).



3.2.3 Assignment custom fields

Since version 5.2, Teamwork supports custom fields on assignments. To activate them, go to admin -> Customization -> on issues, you’ll get to a dedicated page:

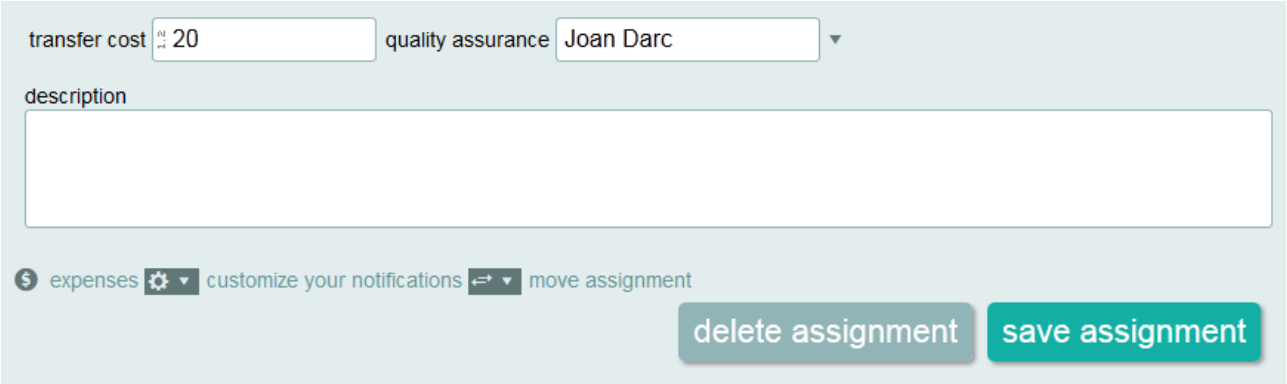


On such page you also have detailed examples and a technical reference, which we do not duplicate here.

Custom Field examples:

- Unused field: `no` -> the field is unactive
- Simplest case: `my code` -> a string
- Lenght defined case: `my code,20` -> a string 20 chars lenght
- Double case: `cost,15,java.lang.Double` (also Integer or Long)
- Date case: `when,10,java.util.Date`
- Teamwork object case: `customer referral,25,com.twproject.resource.Person` -> a Person
- `see this task,25,com.twproject.task.Task` -> a Task
- `attachment,15,org.jblooming.ontology.PersistentFile` -> an uploaded file

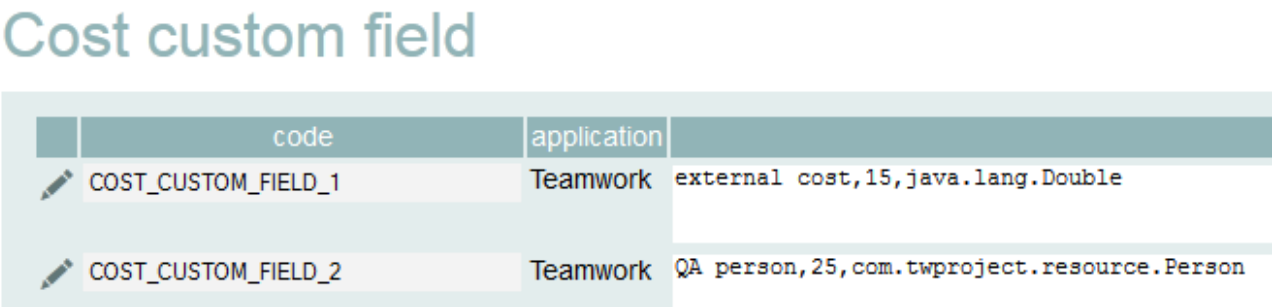
So if for example you add two such custom fields you get on the assignment editor:



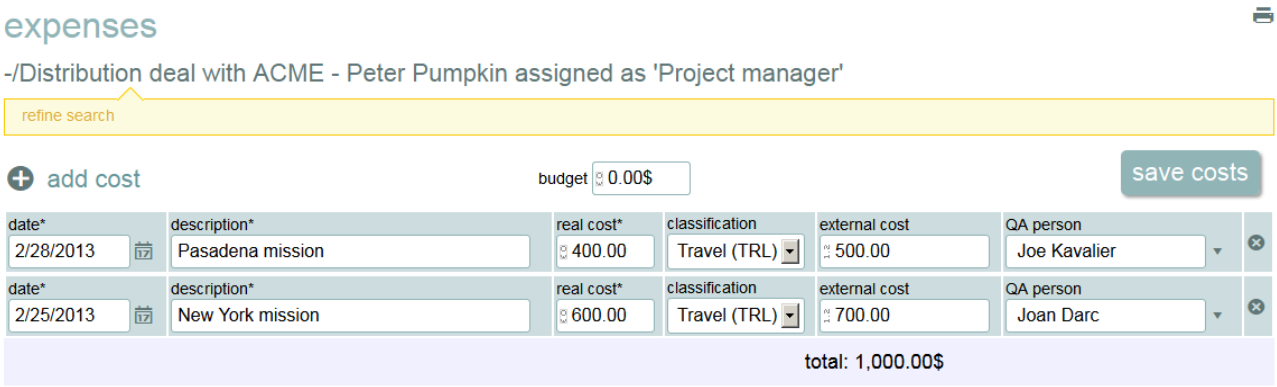
These are typically used for recording external data.

3.2.4 Assignment cost custom fields

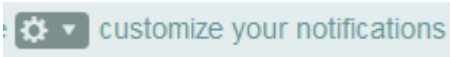
Also expenses or costs on assignments can be customized, adding more fields, analogously as done above. Access from admin -> costs



In this case we added two fields, so we get them when adding costs on assignments:



3.2.5 Notifications



opens the subscription box:

subscribe Pietro Polsinelli to:

☑ notify also for events on subtasks

event type	e-mail ☐	digest ☐	sticky note ☐
creation of this assignment	☐	☐	☐
task status change	☐	☑	☑
task dates change	☐	☑	☑
milestone closer	☐	☑	☑
task/issue expired	☐	☑	☑
added issue	☐	☑	☑
closed issue on task	☐	☑	☑
issue updated	☐	☑	☑
worklog overflow on assignment	☐	☐	☐
budget overflow	☐	☐	☐
worklog misplaced	☐	☐	☐
diary change	☐	☑	☑
child added on task	☐	☐	☐
document added	☐	☐	☐

Here you can subscribe the assignee ("push" subscription) to listen to some events, so to receive messages on the chosen media channel. Default subscriptions are set on roles, and you can change them any time. Also the assignee can edit all her subscriptions.

If you get this message:

subscribe Young Frankenstein to:

The selected resource cannot be subscribed to events as not enabled for login.

It just means that you haven't given a user name and password yet to such resource.

You can subscribe the whole underlying project tree at once by checking this:

☑ notify also for events on subtasks

What does "event type" exactly mean?

event type
creation of this assignment
task status change
task dates change
milestone closer
task/issue expired
added issue
closed issue on task
issue updated
worklog overflow on assignment
budget overflow
worklog misplaced
diary change
child added on task
document added

1) **task status change**: this event is raised when the status of the task changes. Usually this is a crucial point in project management; projects coordinators or production managers should subscribe this event.

2) **task dates change**: this event is raised when the task' schedule changes.

3) **task milestone closer**: when a milestone is approaching this event is raised. By default the alert is 3 days before the milestone, but you can change it from the configuration pages (see 16.5.3 Project defaults)

4) **task issue/expired:** the end / closing date for an issue on the task or the task itself has expired and the status is still open.

5) **added issue:** an issue was created on this task.

6) **closed issue on task:** an issue was closed on this task.

7) **issue updated**: this event is raised when an issue on the task is updated.

8) **worklog overflow on assignment**: when the work done on the task exceeds estimations, this event is raised.

9) **budget overflow:** expenses for the assignment are beyond budget.

10) **worklog misplaced**: Teamwork allows every user to add worklog (of course only users assigned to a task) whenever they want. This means the you can add worklog on closed tasks or when outside the task dates scope (e.g. your task starts 1st January and ends 1st February and someone insert worklog on the 3rd of February). This is heresy in classical PM software, but it something that in real life happens all the time. So the alternative is between having rigid software with incomplete/fake data or flexible software and real data; we prefer the latter. In order to stay informed in case of "strange" worklog recordings, Teamwork raises this event.

11) **diary change**

12) **child added on task**

13) **document added**

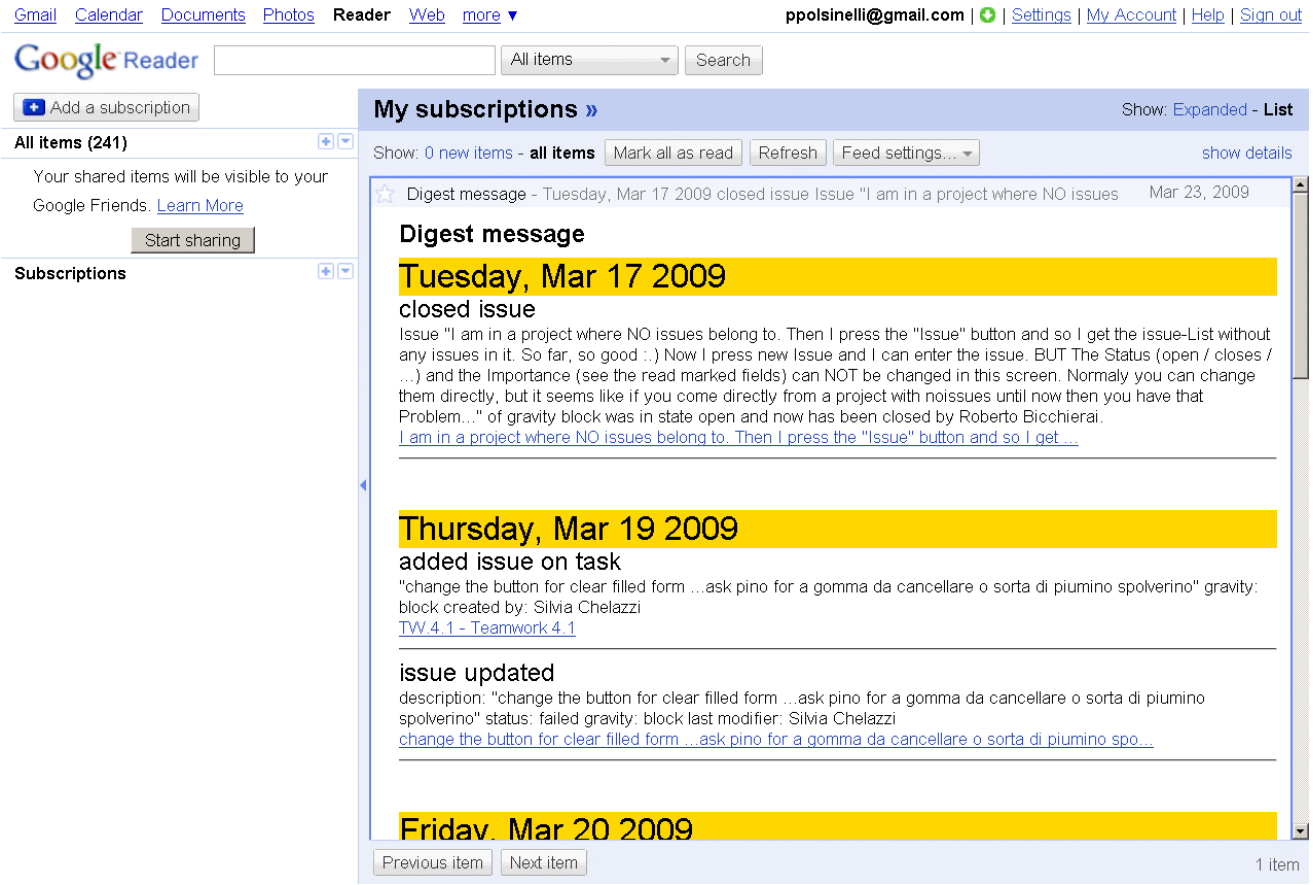
You can subscribe the whole list of events by checking the top column checkbox. And you can subscribe also to any event on any descendant task by checking ☐ notify also for events on subtasks.

When creating an assignment, choosing a role determines the default subscriptions that will be proposed on the assignment; the configuration of default by role is done on the role editor, going to the tab default subscriptions:

A nice "channel" on which you may subscribe say a task tree, is the "digest" one, which will send you by e-mail or on your RSS reader the digest of the events on the project, depending on your options:

where do you like to receive digest messages? ☐ by e-mail ☐ by RSS

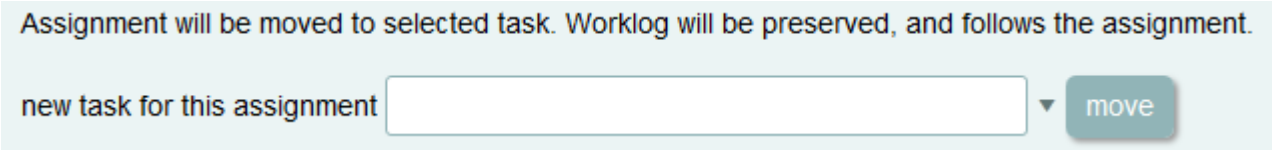
Here what you'll get on your reader:



So consider that the behavior of subscriptions is also regulated by your options, and subscription events come not only from tasks, but also from issues, worklogs, and agenda.

3.2.6 Move assignment

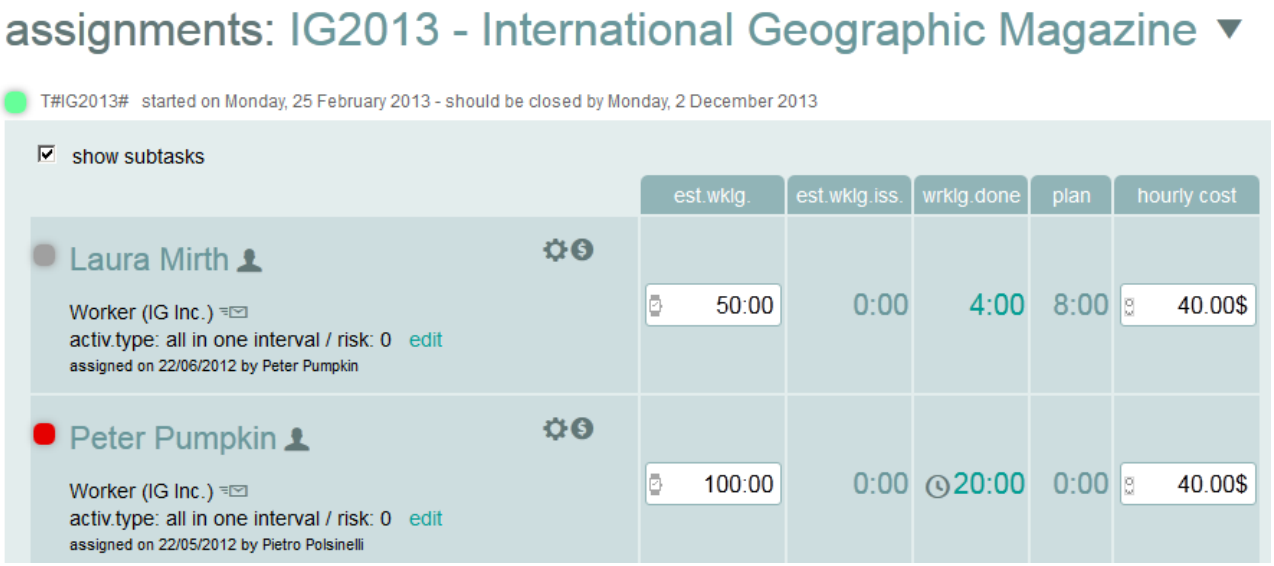
The assignment concept makes Teamwork an extremely flexible tool, so when restructuring project trees you can also move single assignments:

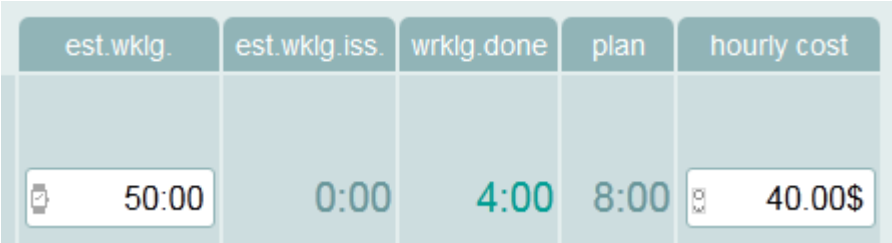


All data linked to the assignment will “follow” it e.g. worklogs, estimations, expenses, subscriptions. Issues are linked to resource / task so they will not follow – use bulk issue move to move those too..

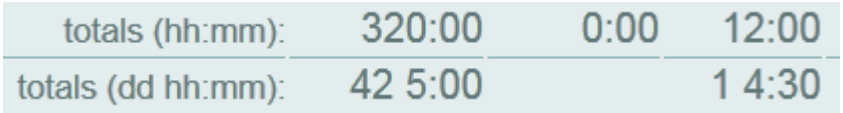
3.2.7 Assignment list

By saving the assignment you will get back to the assignment list:

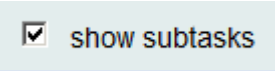




Here you see priority, assignee, role, description; in the columns estimated work from assignment, estimate work from issues, estimated work from plan, work done and hourly cost; totals on the bottom.



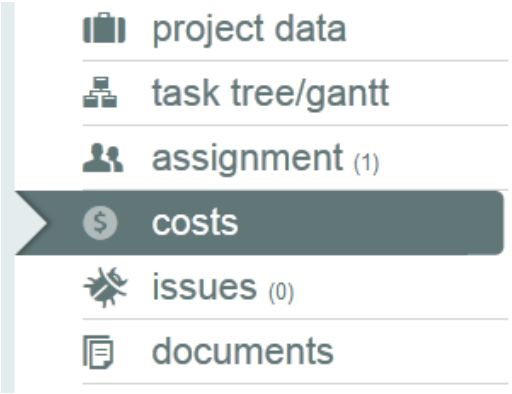
You can change priority, estimation and cost directly here. By clicking on the work log done total you will access the work log detail.

By checking 

you will see here also the assignment on descendant tasks.

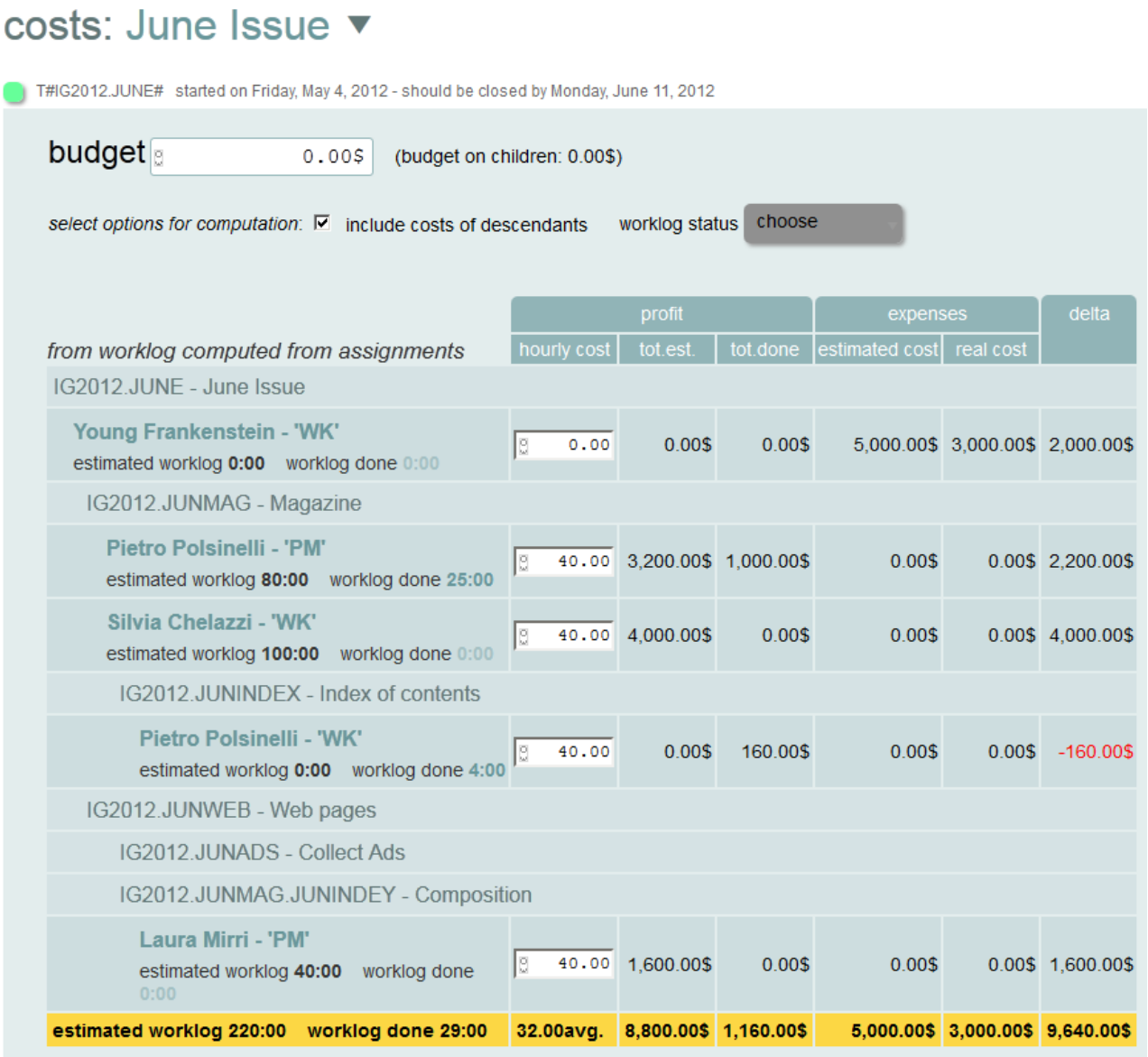
3.3 Costs

Cost monitoring is a relevant aspect of project management. In Teamwork cost management is mainly focused on tracking work costs, but has also some additional features that let PM's form a complete overview. Click on costs:



To access and modify the cost page you need specific permissions on the task; usually only PMs can see it.

Costs are tracked for all tasks, and from a root it is easy to get an overview of the overall costs for the entire tree or for a branch.



costs: June Issue

T#IG2012.JUNE# started on Friday, May 4, 2012 - should be closed by Monday, June 11, 2012

budget 0.00$ (budget on children: 0.00$)

select options for computation: ☑ include costs of descendants worklog status choose

from worklog computed from assignments	profit			expenses		delta
	hourly cost	tot.est.	tot.done	estimated cost	real cost	
IG2012.JUNE - June Issue						
Young Frankenstein - 'WK' estimated worklog **0:00** worklog done 0:00	0.00	0.00$	0.00$	5,000.00$	3,000.00$	2,000.00$
IG2012.JUNMAG - Magazine						
Pietro Polsinelli - 'PM' estimated worklog **80:00** worklog done **25:00**	40.00	3,200.00$	1,000.00$	0.00$	0.00$	2,200.00$
Silvia Chelazzi - 'WK' estimated worklog **100:00** worklog done 0:00	40.00	4,000.00$	0.00$	0.00$	0.00$	4,000.00$
IG2012.JUNINDEX - Index of contents						
Pietro Polsinelli - 'WK' estimated worklog **0:00** worklog done **4:00**	40.00	0.00$	160.00$	0.00$	0.00$	-160.00$
IG2012.JUNWEB - Web pages						
IG2012.JUNADS - Collect Ads						
IG2012.JUNMAG.JUNINDEY - Composition						
Laura Mirri - 'PM' estimated worklog **40:00** worklog done 0:00	40.00	1,600.00$	0.00$	0.00$	0.00$	1,600.00$
estimated worklog 220:00 worklog done 29:00	**32.00avg.**	**8,800.00$**	**1,160.00$**	**5,000.00$**	**3,000.00$**	**9,640.00$**

First of all notice that each task can have a “budget”, costs from worklog, and eventually also additional costs.

Costs and budgets from children are collected to the parent, so you can decide whether to insert them as split on children or aggregated on the parent.

The additional costs/budgets set on assignments are here also shown and considered in totals.

By clicking on the checkboxes you can change the behavior of the page and, accordingly, the values of sums.

> Teamwork handily remembers user choices across sessions, so when you check an option and you login again that check will be considered the default.

Beyond the worklog part, you can change hourly costs for you assignment, and add new “additional” costs by using [add cost] button. Additional cost data can be modified and will be saved by clicking on [save costs]. You can remove additional cost by clicking on ⊠:

additional costs	classification	estimated cost	real cost	
IG2012.JUNE - June Issue				⊕
	- choose -	0	0	⊗
IG2012.JUNMAG - Magazine				⊕
IG2012.JUNINDEX - Index of contents				⊕
IG2012.JUNWEB - Web pages				⊕
IG2012.JUNADS - Collect Ads				⊕
IG2012.JUNMAG.JUNINDEY - Composition				⊕
		0.00$	0.00$	

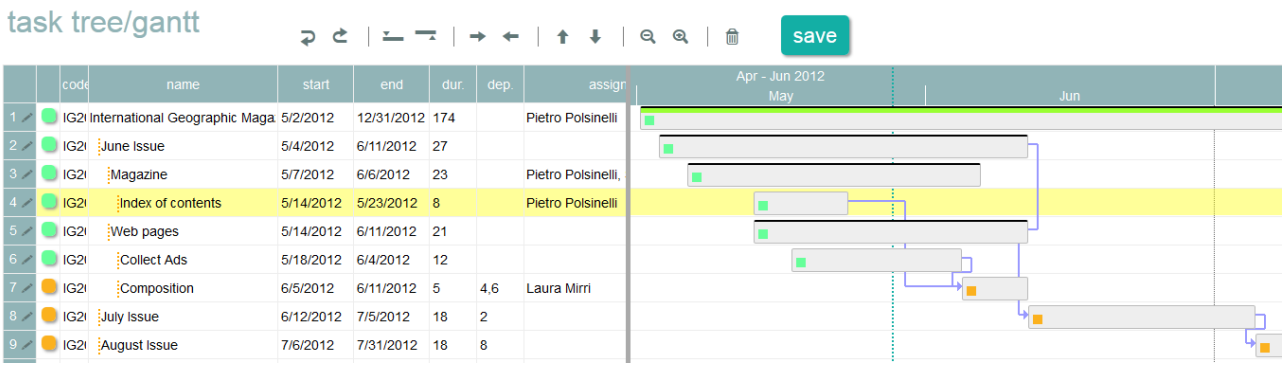
Finally the bottom line with the totals:

	tot.est.	tot.done	delta	delta on budget
results for selected options	13,800.00$	4,160.00$	9,640.00$	-4,160.00$

3.4 Task tree

Since Teamwork 5.1 we introduced a simple project tree view alongside the powerful Gantt editor (see below). This allows a quick view / navigation of the project tree and basic data editing (code / name).

3.5 Gantt editor



Teamwork focus is capturing work done in real time. To do this we built a tool that can model real time situations and easily change in time. This is in contrast with traditional project management methodology where projects are defined in advance in all details – not realistic for most work situations.

So Teamwork's model of project, task trees, dates and task states is different from that of classical Gantts: tasks can be open beyond their due dates, task can be suspended inside their time scope, and so on. Also the automation due to dependencies has been somewhat simplified.

Another difference between Teamwork's model and classical Gantt is that in Gantts there is always a hidden "mother project" from which everything depends; in Teamwork instead we can many – even thousands – open root projects.

Still in time we found that users would find it useful to have a tool like those usually provided for Gantt planning – a graphical environment where sub task can easily created,

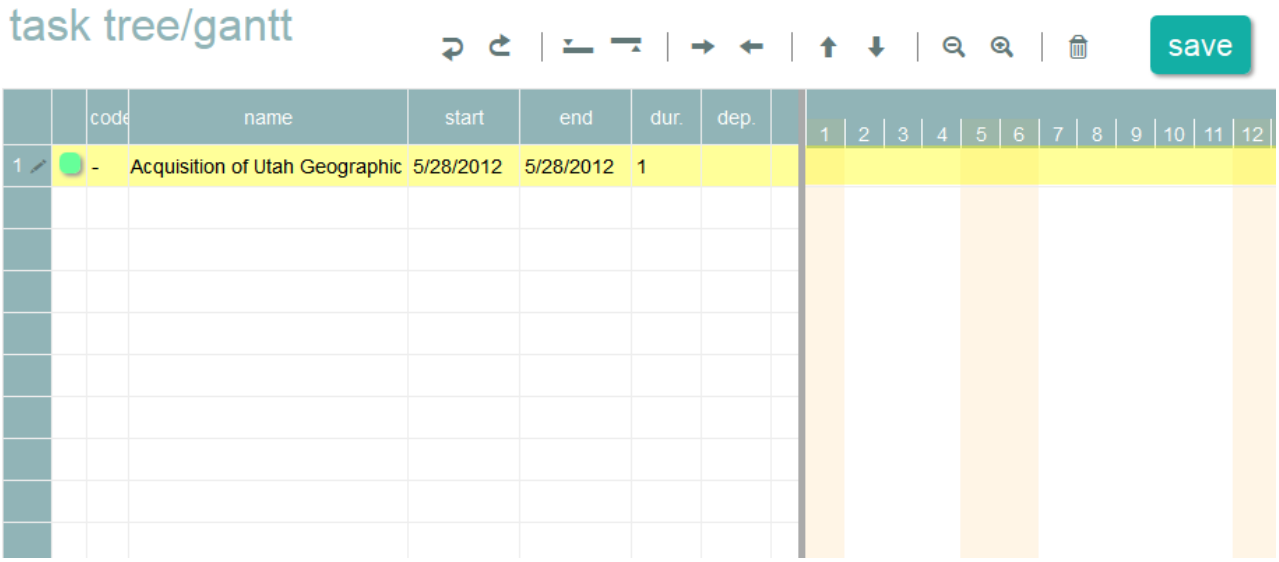
time can be set, dependencies created, and tree branches moved by dragging. To access this quite powerful tool just click on the

button.

> When you access the tree of a task, the tree shown is always from the root project. Still if the user is project manager from a certain node downwards, the higher part of the tree will not be shown. This may make some operations impossible, as they would propagate day changes up the tree to regions that are not editable by the logged user.

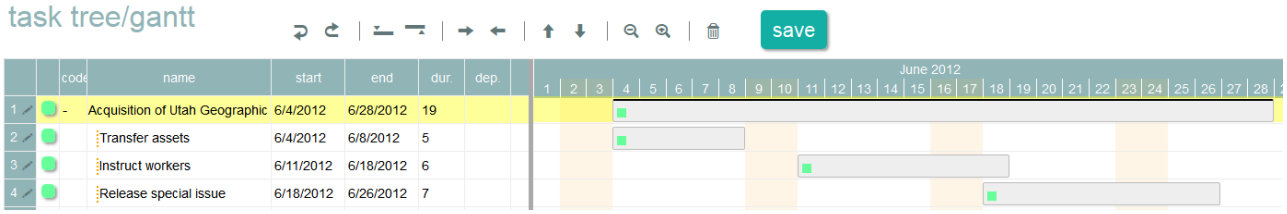
You changes in the tree editor will be saved only when you click the “save” button. This way you can “play” with the tree without destroying the original project structure.

In case of a brand new root project, you can use the tree editor to build quickly a full tree structure; from

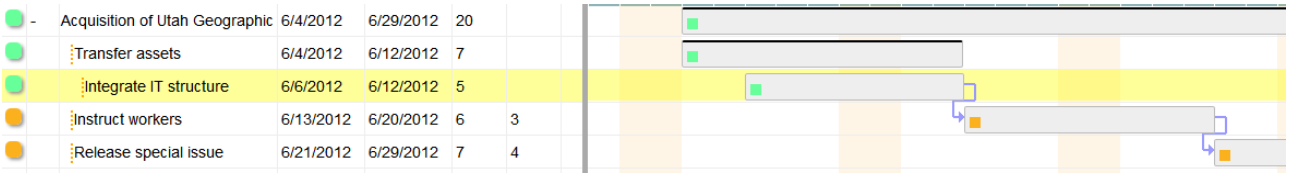


To create child tasks, you just need to write their name in the “name” column:

Then you set start and durations either by dragging or by writing dates by hand



It is even easier if you set dependencies between tasks, using the dep. Column: just write a row number there. You can set multiple dependencies by separating row numbers with comma.

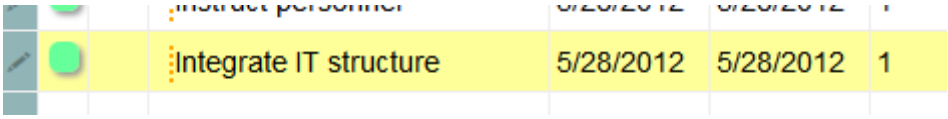


Notice that task statuses are updated correspondingly.

3.5.1 Toolbar

The ↩ ↪ are the usual undo – redo buttons; consider that their scope is only for the current task tree window after the last "save" clicked.

Each task corresponds to a line, and for each line when selected – the yellow background



there are a set of operations possible by using the buttons in the button bar above

: inserts a line above the current one, at the same project tree level – a "brother" of the current one.

: inserts a line below above the current one, as a child of the current project tree level.

➡ : indents the selected line, making it a child of the line above it.

⬅ : out-dents the selected line, promoting it as brother of its current father and as child of the current father's father.

⬆ : exchanges the order of two tasks – they must be at the same level.

⬇ : exchanges the order of two tasks – they must be at the same level.

: these are the usual zoom in – zoom out buttons.

🗑 removes the focused line; here too consider that the removal is relative to the task tree seen, and that it will not be effective on the previously saved data until the "save" button is pressed.

🖶 Prints both side of the Gantt: remember to set the browser print settings to vertical and to enable printing background images. Also consider zooming the Gantt graph part in order not to have scrollbars.

> The task tree root line cannot be removed from this view.

3.5.2 Textual panel

The task tree view is divided in two parts, separated by a vertical line; the two panels separator can be dragged left or right depending where you are operating. The left part is the textual panel:

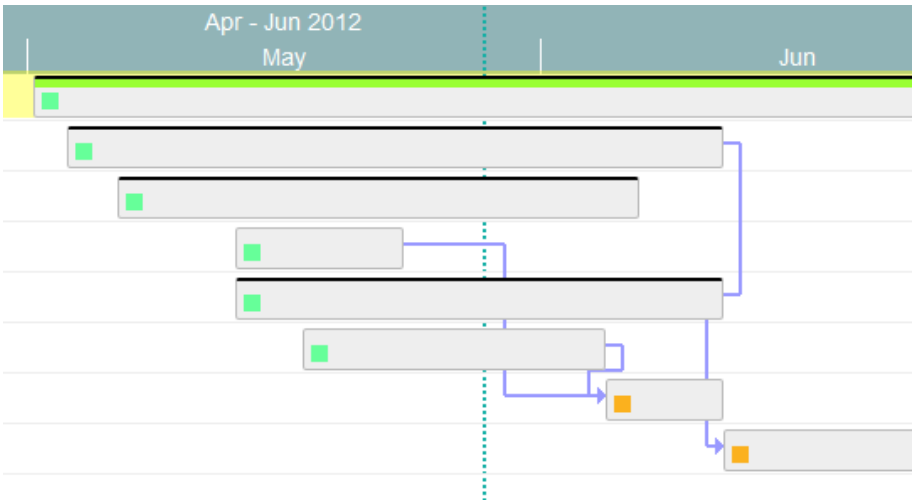
		code/short name	name	start	end	dur.	dep.	assignees
1		IG2012	International Geographic Maga	5/2/2012	12/31/2012	174		Pietro Polsinelli
2		IG2012.JUNE	June Issue	5/4/2012	6/11/2012	27		
3		IG2012.JUNMAG	Magazine	5/7/2012	6/6/2012	23		Pietro Polsinelli, Silvia Chelazzi
4		IG2012.JUNINDEX	Index of contents	5/14/2012	5/23/2012	8		Pietro Polsinelli
5		IG2012.JUNWEB	Web pages	5/14/2012	6/11/2012	21		
6		IG2012.JUNADS	Collect Ads	5/18/2012	6/4/2012	12		
7		IG2012.JUNMAG.JUNI	Composition	6/5/2012	6/11/2012	5	4,6	Laura Mirri

Columns can be expanded by dragging, and all contents (code, name, dates and dependencies) can be edited. Important for modeling are dependencies, which are introduced by simply writing the row number in the “dep” column. You can set multiple dependencies by separating row numbers with comma.

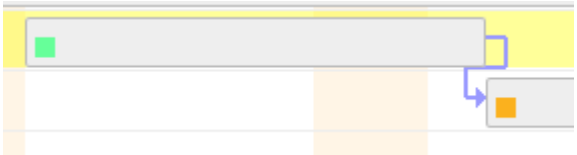
> A modeling novelty introduced with Teamwork 5 is that you can have dependencies between any two tasks – not just between brothers.

Among data that cannot be inserted from here are assignments – but see below.

3.5.3 Graphical tree panel

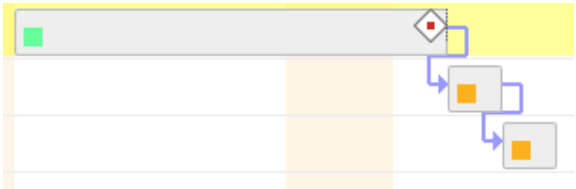


On the right panel, where task' durations are shown graphically, you can move tasks in time by dragging them, and all dependants will be moved accordingly – if possible.



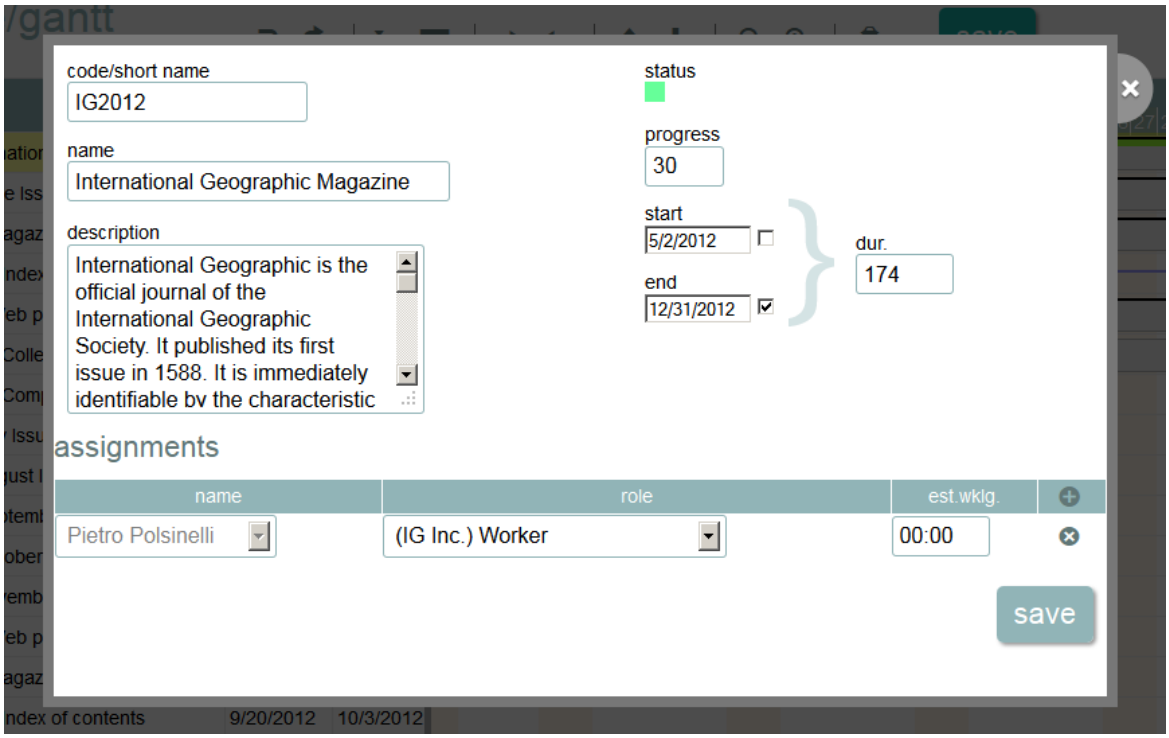
If you stop with the mouse at the start or end of the task, the mouse cursor will become a bi-directional arrow and it will be possible to expand / contract the task duration by dragging. All dependant task dates will again be changed accordingly if possible.

All operations are checked with the project constraints and done only if possible. If for example you have a milestone set, it will not be possible to change dates by drag and drop.



3.5.4 Task line editor

When in the Gantt you can click on ✎ for every line, getting a compact task editor:

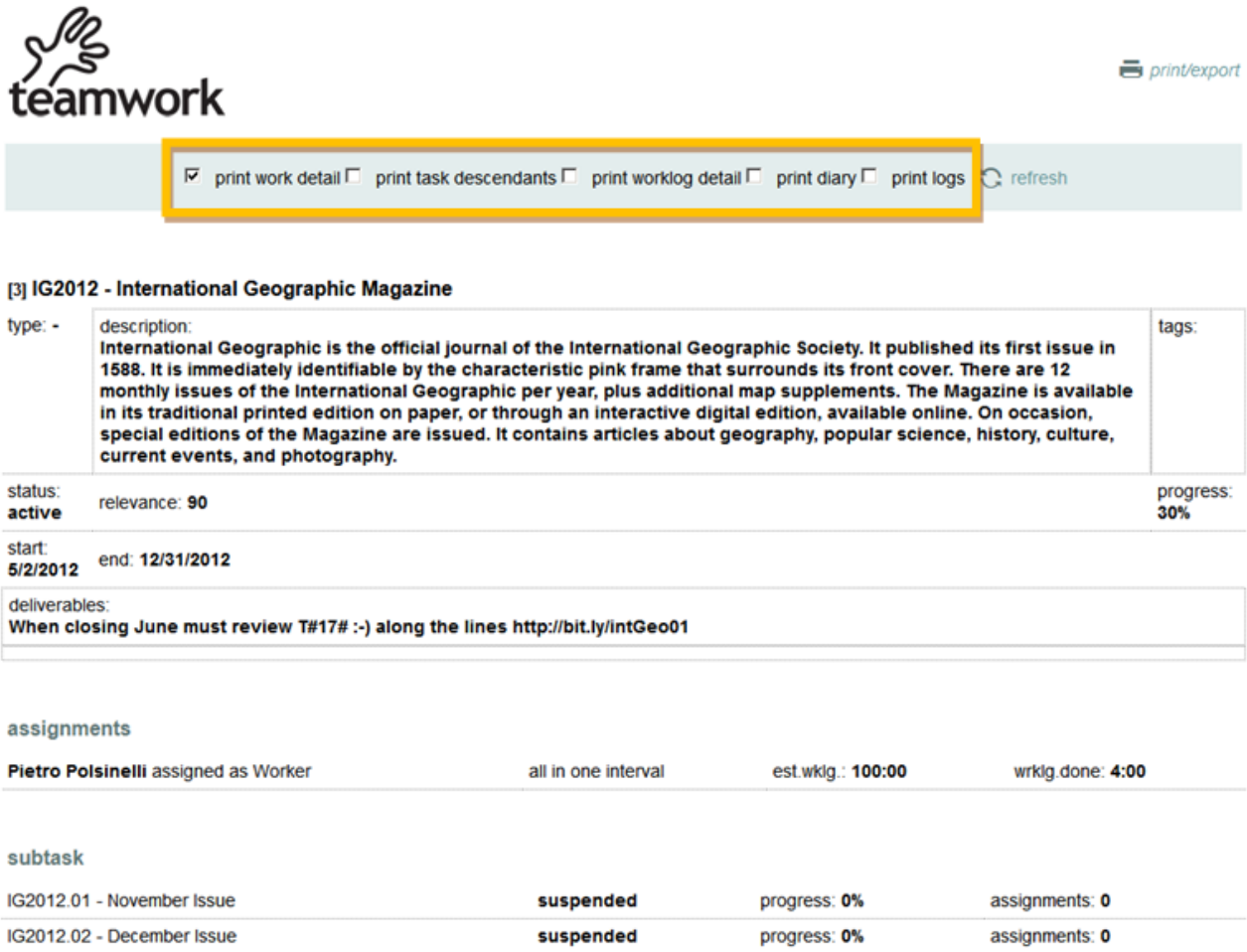


which is a practical way to add say milestones and assignments.

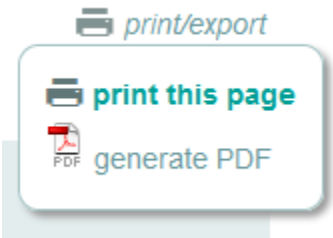
> Everything you do – assignments included - in the Gantt tree is not saved until you press the big "save" button on the main window.

3.6 Prints and snapshots

From every task editor you can access print a report of your task – in the upper right part of the page 🖶. By clicking on “print / export report” you get



And again hovering on the 🖶 button you get to print or export the page as PDF.



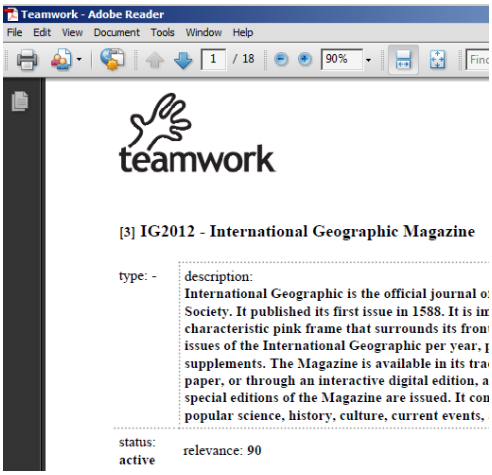
You can change which data gets printed by selecting the print

These options are always available on every Teamwork’ report page.

If you pick “snapshot” instead of printing, a new PDF document will be generated and attached to the task

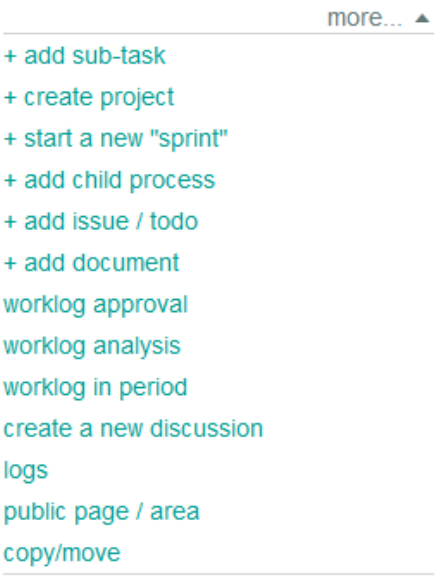
which contains a snapshot to the current task data. Snapshots are an easy way to “freeze” a project report in time.

You can customize the logo printed on reports from the administration page.



3.7 Public pages

When in the task editor, try clicking on the “more” link: a further set of features becomes visible.

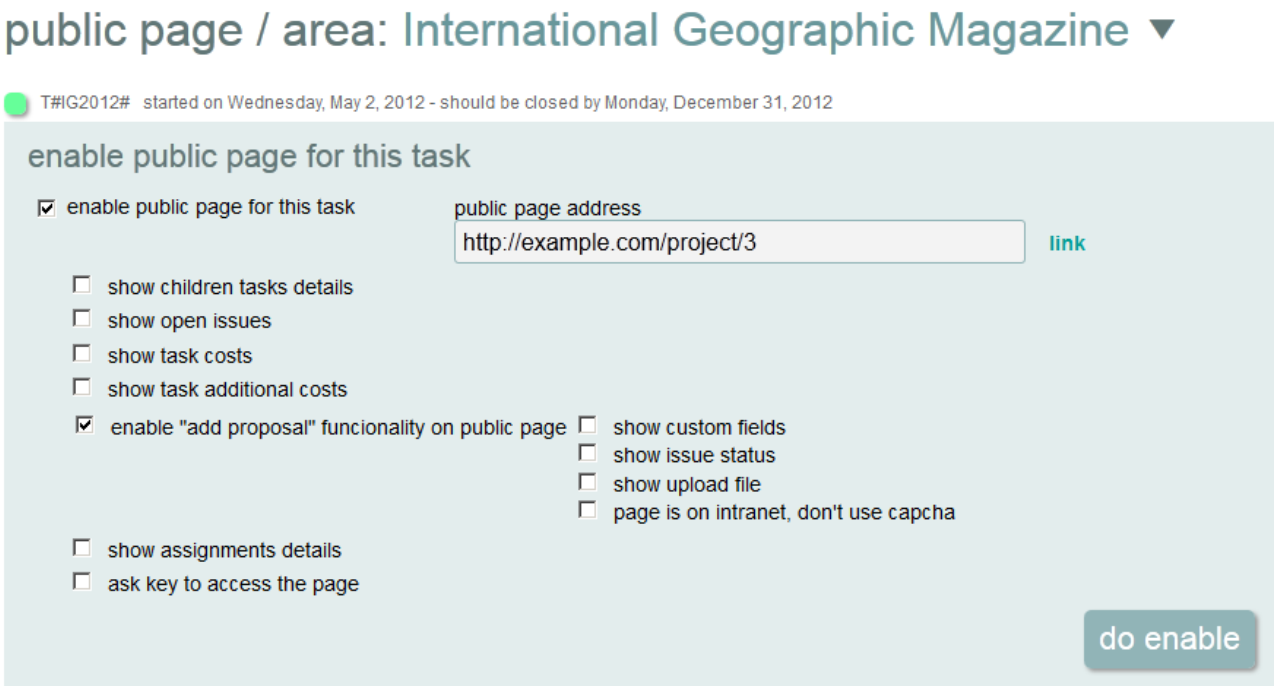


Among these, click “public page / area”: for any project / task, a **public page** can be enabled which will be exposed at Teamwork’s address, will not require login, and will present selected information. It is a way to automate distribution of information always updated from Teamwork to people that do not access it.

The pages will be available at URL of the form

[http://your teamwork address]/project/[task id]

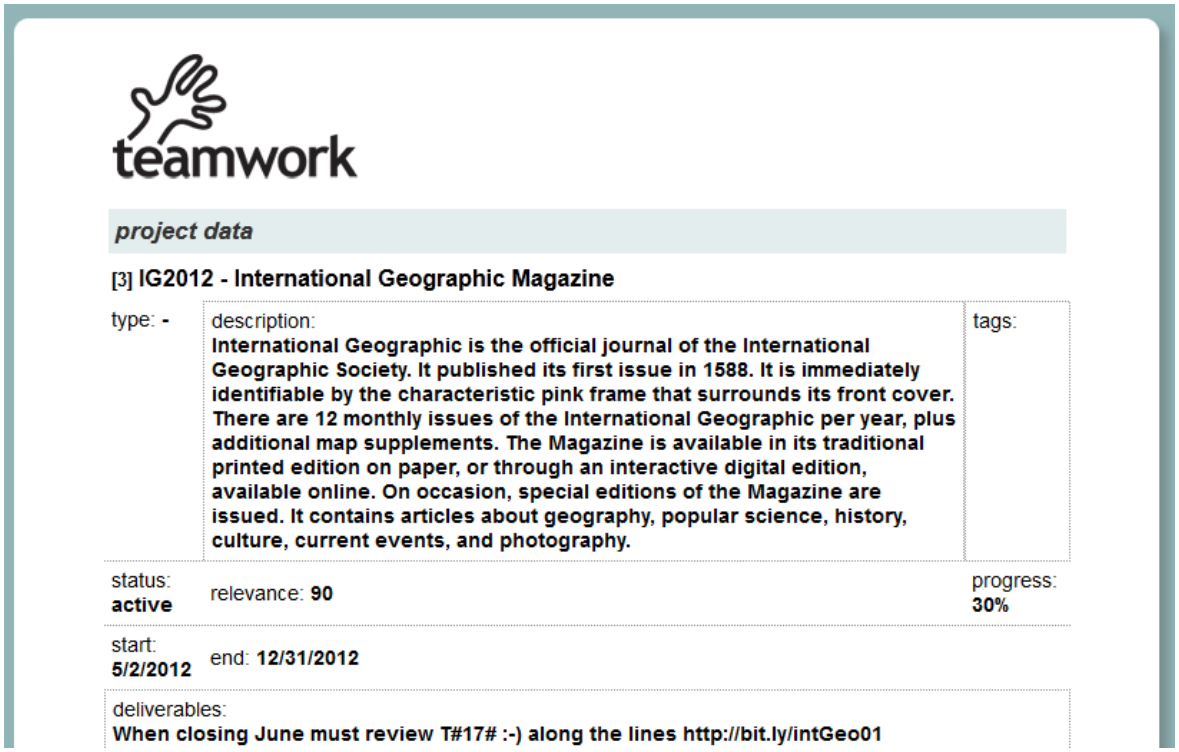
When enabling such pages, there are several options available:



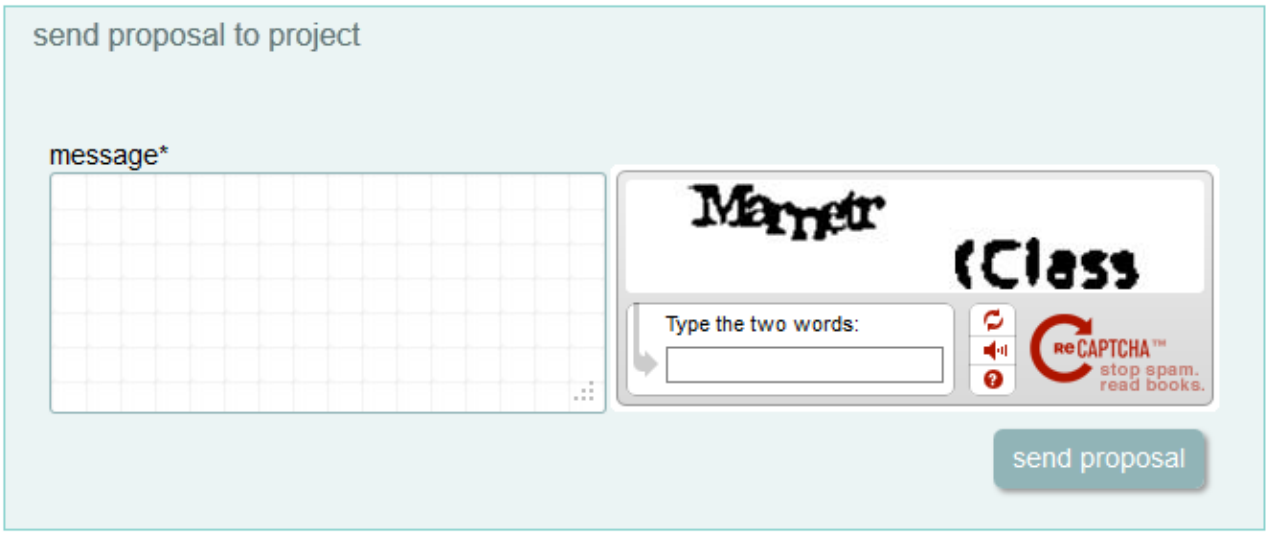
You first have to enable this page and save before getting the options enabled. Most options are self explanatory, but:

- “enable add proposal”: if you enable this option, visitors to the page will be able to propose issues / features on such task: the proposal will be saved as an open, unassigned issue on the task.
- “ask key to access the page”: access to the page will be protected by a key which you define and will be requested on access.

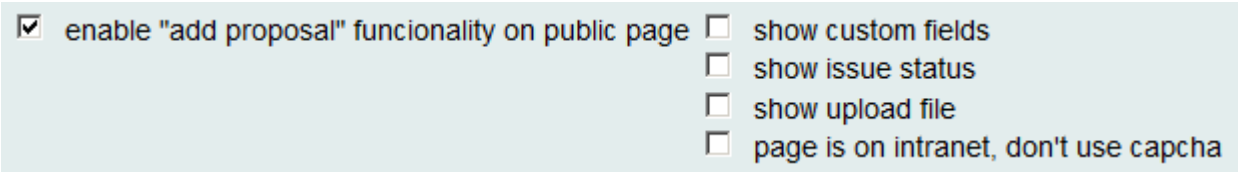
This is an example resulting page:



We see here that from the public page people can send proposal to the project:



And if you enable “add proposal” more options are available:

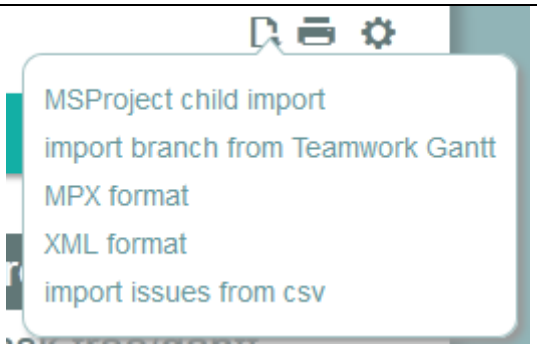
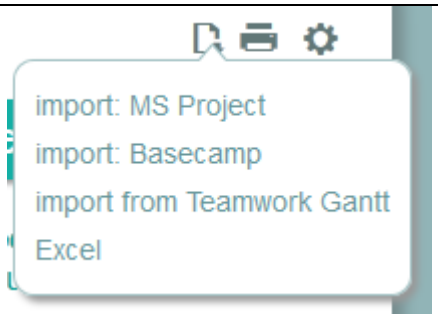


3.8 Import/export

Teamwork supports several forms of interactions with third-party services and applications, which is by now a must-have requirement of any complex application. You may need to migrate projects handled in other applications into Teamwork, or to export a project in a different format, compliant to corporation standards, and so on. Here we will see imports from two popular applications, and export to one of these, but the Teamwork's open data structure allows many other forms of import/export.

From any editor or list, just hove on the button.

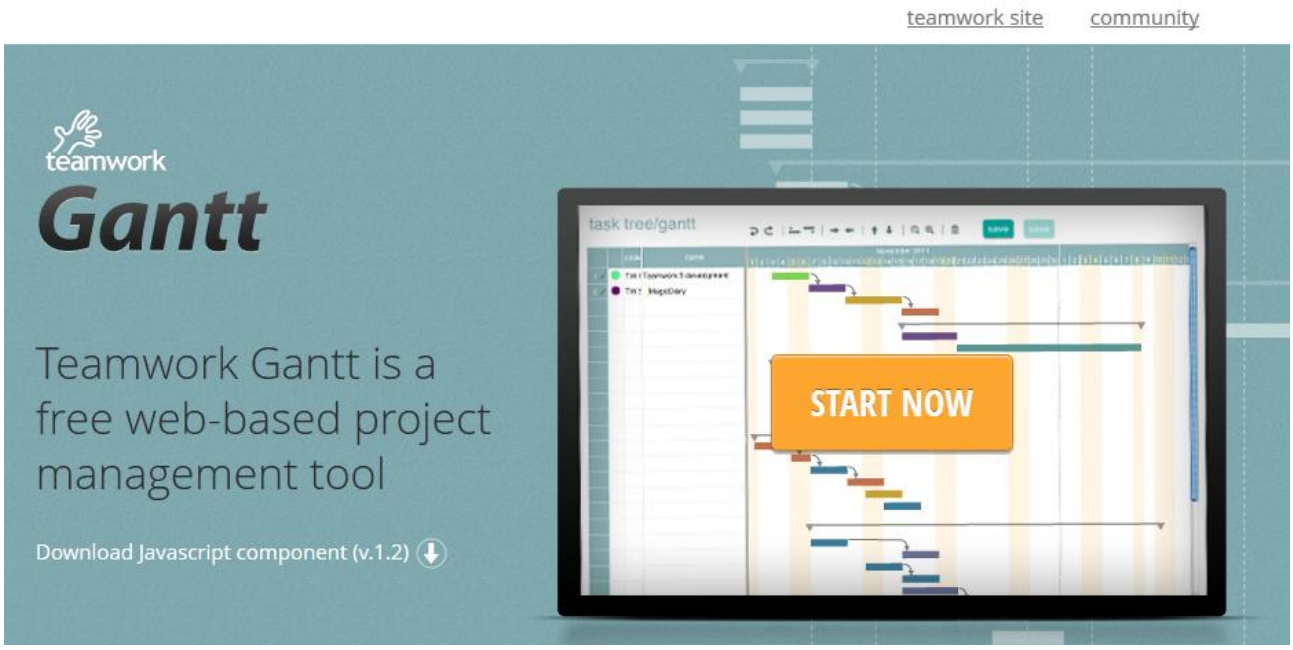
> You can both do import/export operations from the project lists, in which case projects will be created as new roots, or from the context of a project node, in which case projects will be imported as child of such node.

From a task editor	From the project list
	

3.8.1 Import from Teamwork Gantt

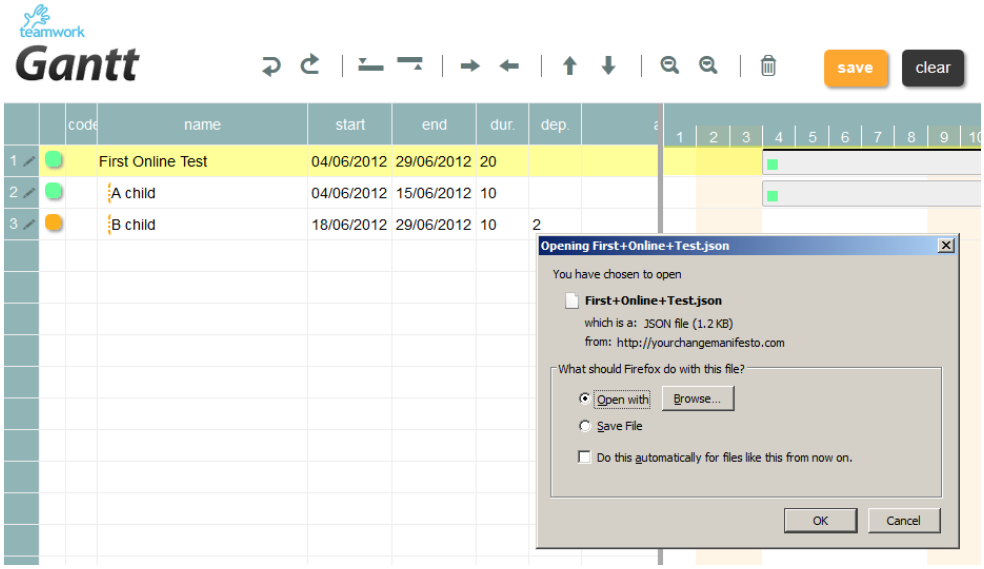
Teamwork's team has created a free online service where users can create Gantt charts on the web – Gantts in the extended sense of teamwork's model:

http://gantt.twproject.com

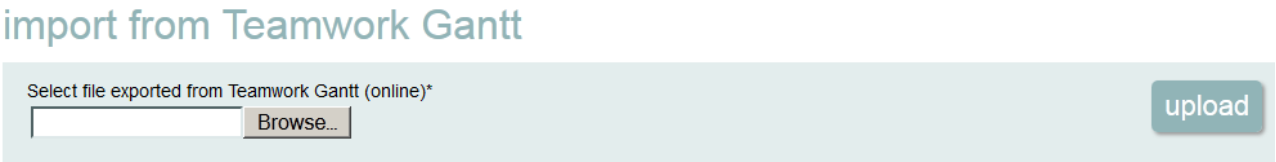


These Gantts can be exported from the web and imported in your Teamwork instance, by

exporting the online Gantt content as a file (will have “json” extension)



and then importing it in your Teamwork instance – just click on “import from Teamwork Gantt”.



Resources will be mapped to existing ones by name or created as new if not found.

3.8.2 MS Project import / export

Microsoft Project™[3] (from here on MSP) is one of the most (un?)popular project management tools, and Teamwork supports both export and import to/from it.

> Actually, the same mpx format can be used to import/export with GanttProject, the free client, see http://ganttproject.biz

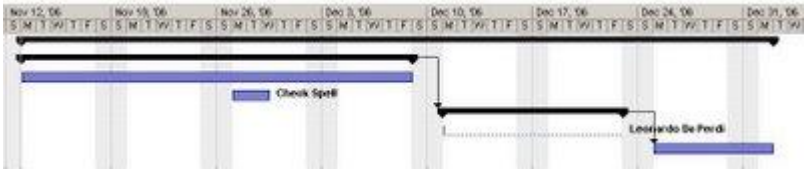
In order to *export* a project from Teamwork go to the project editor of your root (or even a branch) and just press MPX format button.

A .mpx file will be created and your browser will ask what to do with it.

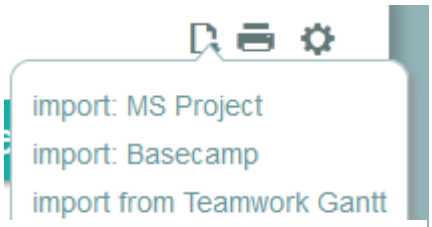
	Task Name	Duration	Start	Finish	Predec	Resource Names
1	International Geograph	36 days	Mon 11/13/06	Mon 1/1/07		Walter Scott,Andy Whirl
2	Contents	28 days	Mon 11/13/06	Fri 12/8/06		Super 0'Admin,Check Spell,Graham Greene
3	Collect periodics	20 days	Mon 11/13/06	Fri 12/8/06		
4	Collect and check	2.5 days	Mon 11/27/06	Wed 11/29/06		Check Spell
5	Graphics	10 days	Mon 12/11/06	Fri 12/22/06	2	
6	Main development	10 days	Mon 12/11/06	Fri 12/22/06		Leonardo Da Perdi
7	Print	6 days	Mon 12/25/06	Mon 1/1/07	5	

[3] Microsoft Project is a trademark of Microsoft Corporation.

The scheduling work done in Teamwork gets exported exactly as it was:



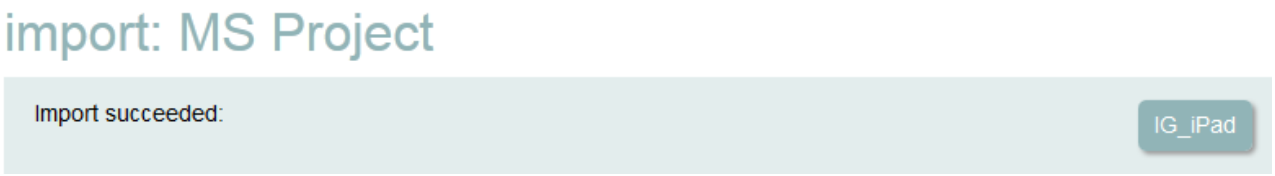
If you want to *import* a root task go to “projects” then hover on and select “import: MS project". This page will be displayed:



Then you have to select the file by browsing on your local disk and then press “go” button.

In case of import from .mpx file you have to choose the language in which the file is written.

If everything goes well:



you can then follow the link (the same name of the task on MSP), and check out the structure on the Gantt editor:

		code	name	start	end	dur.	dep.
1		-	IG_iPad	6/4/2012	6/14/2012	9	
2		5	Yearly finantial report	6/4/2012	6/7/2012	4	
3		6	New tech funding	6/4/2012	6/7/2012	4	
4		1	Ipad app for magazine	6/6/2012	6/14/2012	7	
5		2	Feasibility study	6/6/2012	6/6/2012	1	
6		4	Meetings	6/6/2012	6/13/2012	6	
7		3	Do not do meeting	6/8/2012	6/13/2012	4	5,3

You can also import a MSP task under an existing Teamwork project, simply by starting the import from the project’s editor.

What actually happens when you are importing/exporting projects from/to MSP?

First of all, consider that the two applications have a different underlying model of “project”. Teamwork has a notion of status distinct from duration, and MSP doesn’t.

Another difference concerns resources. MSP resources are relatively poor objects, so when you are exporting from Teamwork you are losing most resource data, except for names, and when importing you will get an almost empty resource record. When importing, Teamwork tries to guess if the resource exists already by using name and surname (the only data that MSP export is sending).

The absence of a task status in MSP implies that when you are exporting the status will be lost and when importing the status will be calculated using task dates.

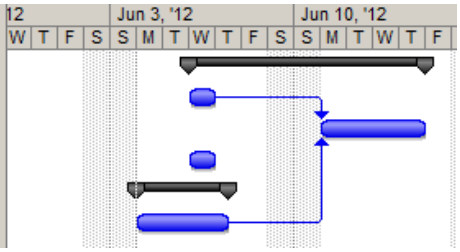
We strongly suggest revising all the task and resources created after import.

This module will satisfy people that use MSP or other clients basically as a Gantt-drawing tool. MSP power users will never be satisfied, as the two applications have a deeply different model; same for Teamwork power users. This functionality can be of use also for those who are forced to have a MSP output for corporate and/or certification purposes (we know people that use Teamwork and export to MSP only when they have to send the project structure to review staff).

Summing up, in exporting and importing we always lose some data, but these functions may still be quite useful.

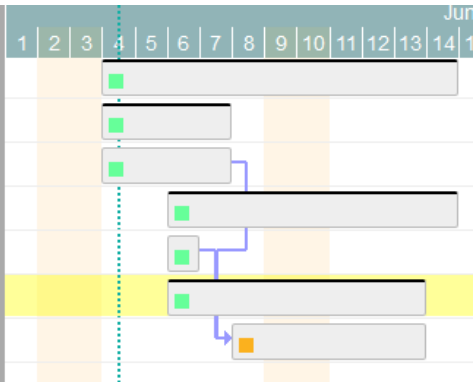
Since Teamwork version 5, dependencies can be set also between non brother tasks. So from MS project:

	Task Name	Duration	Start	Finish	Predecessors
1	**Ipad app for magazine**	**7 days?**	**Wed 6/6/12**	**Thu 6/14/12**	
2	Feasibility study	1 day?	Wed 6/6/12	Wed 6/6/12	
3	Do not do meeting	4 days?	Mon 6/11/12	Thu 6/14/12	2,6
4	Meetings	1 day?	Wed 6/6/12	Wed 6/6/12	
5	**Yearly finantial report**	**3.5 days?**	**Mon 6/4/12**	**Thu 6/7/12**	
6	New tech funding	3.5 days?	Mon 6/4/12	Thu 6/7/12	



You get:

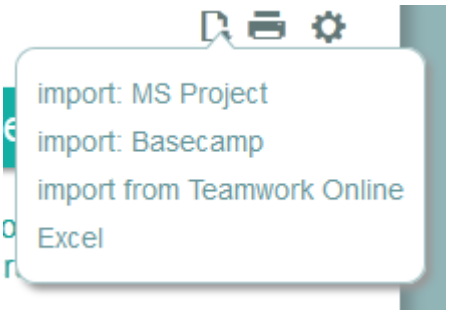
	code	name	start	end	dur.	dep.
1	-	IG_iPad	6/4/2012	6/14/2012	9	
2	5	Yearly finantial report	6/4/2012	6/7/2012	4	
3	6	New tech funding	6/4/2012	6/7/2012	4	
4	1	Ipad app for magazine	6/6/2012	6/14/2012	7	
5	2	Feasibility study	6/6/2012	6/6/2012	1	
6	4	Meetings	6/6/2012	6/13/2012	6	
7	3	Do not do meeting	6/8/2012	6/13/2012	4	5,3



with the correct status change automation in place.

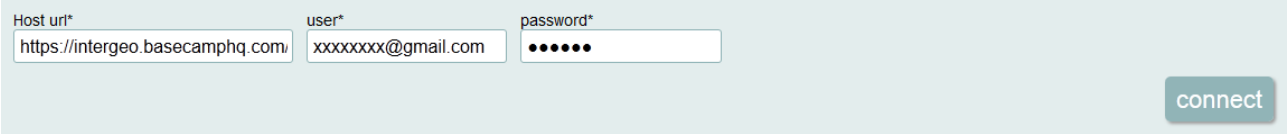
3.8.3 Escape from Basecamp ☺

Basecamp™[4] is a basic todo-list management tool. If you are reading this section probably you decided to change management tool: well, Teamwork supports importing projects and resources from a Basecamp account.



To start importing, go to the "projects" page and click on and select import: Basecamp.

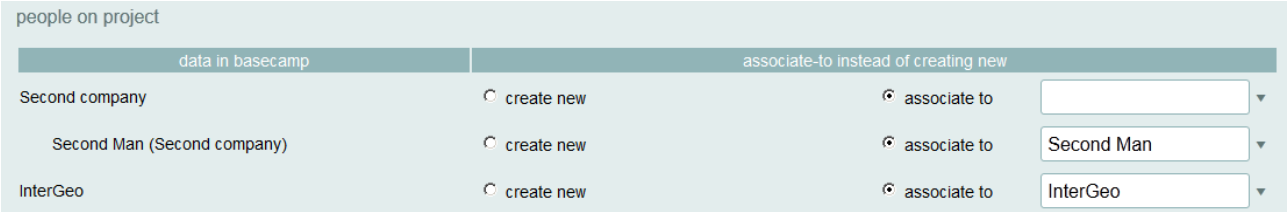
Then insert your connection data:



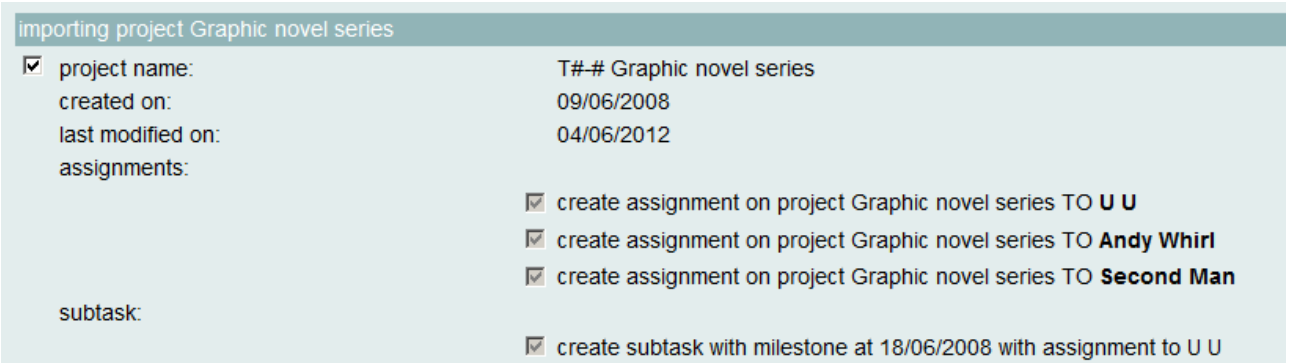
Press "connect" and wait for a Basecamp response. After a while, if connection can be established, something like this will be displayed:

In the first block you have some hints on how Basecamp entities will be mapped in Teamwork.

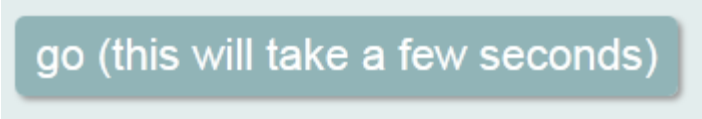
Then there is the "resource" import part. Here for every Basecamp "resource" you can decide whether to create a new resource or simply map it to an existing one.



The last part is relative to projects import.



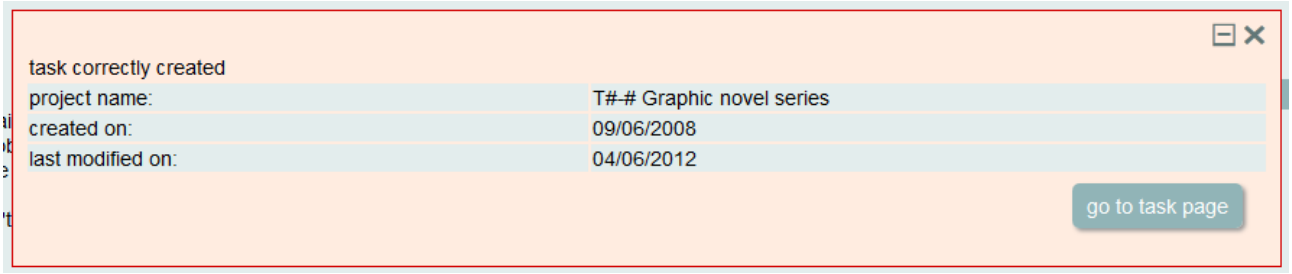
You should check the task that has to be imported. Then by pressing on "go" button you will proceed with the real import (it may



[4] Basecamp is a trademark of 37signals.

take a while – don't refresh the page).

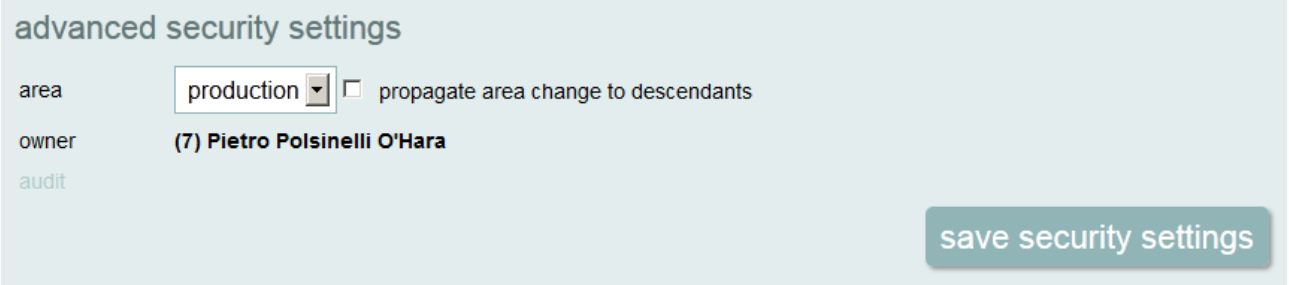
A message will detail the import result and there will be a link to the imported project:



We strongly suggest revising all the task and resources created after import.

3.9 Security

In the public page / area there is also the control of the "advanced security settings": we advise users not to change anything here unless it is strictly necessary.



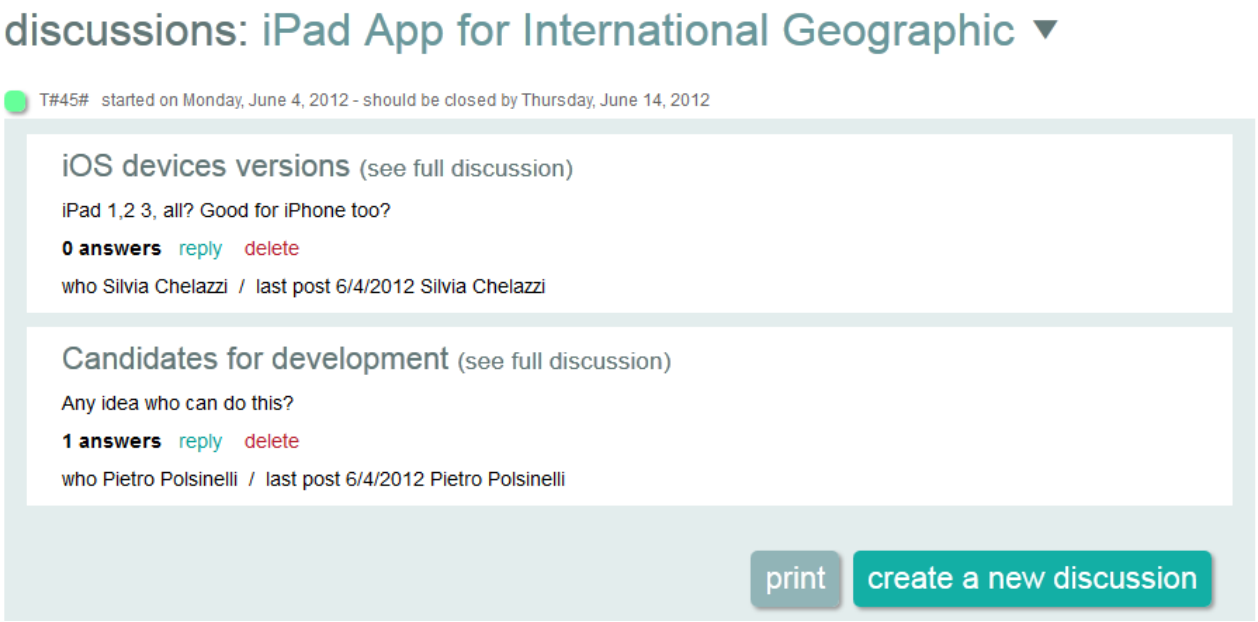
This allows you to change the area and owner of the task. This will be relevant only if you are using a multi-area environment, which is a use case for advanced users. For detail see *12 Security*.

3.10 Task - other pages

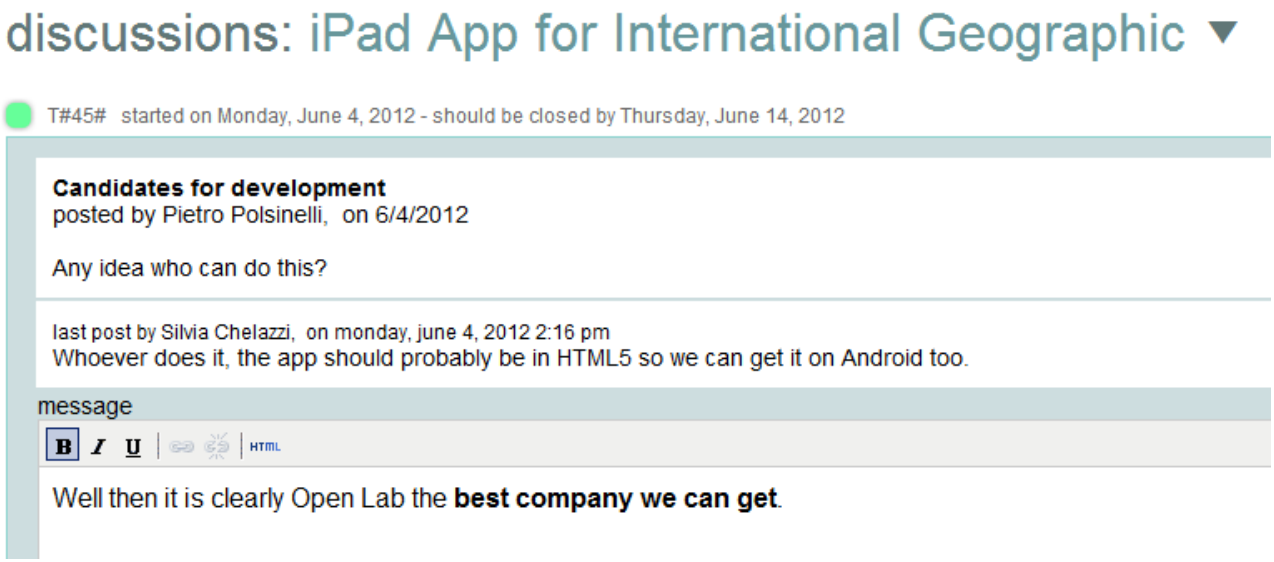
Teamwork tasks have further functionalities.

3.10.1 Discussions

Every task can be associated to a multi-threaded discussion/forum. You can create one more entries (thread) ore reply to existing ones.



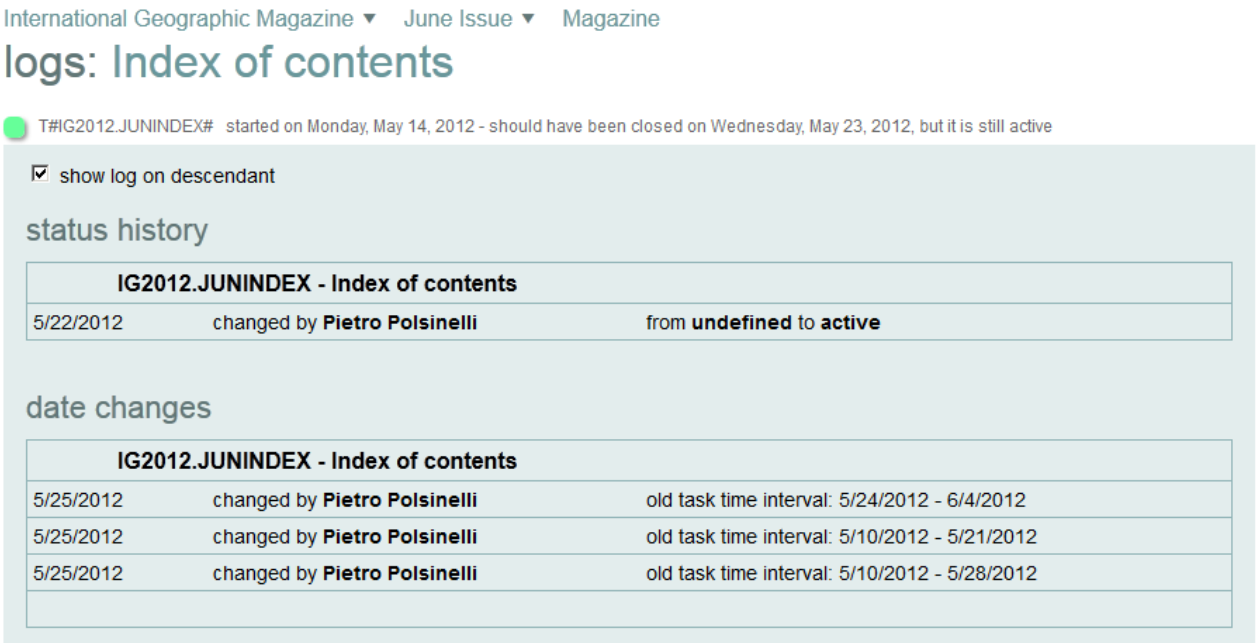
Each post can contain as much text as you want, so you can use it for example to record past notes, ask for comments, etc. The post editor allows inserting also html formatted text:



The discussions page will contain also links to meetings related to this project: see “8.4 Meetings”.

3.10.2 Logs

As stated previously, every status and schedule change on the task is recorded: in the “logs” tab you will find the history of all changes.



3.11 Business processes

Teamwork has an integrated business process management tool, which greatly widens the modeling possibilities w.r.t. project trees, and improves usability even for quite complex use-cases.

It is now possible to model intrinsically complex processes, while maintaining the basic project based organization. The underlying technology is our implementation of Hibernate + JBPM, which gives the full generality and power of a proven business process framework, integrated with Teamwork's model.

In our meetings with customers we often presented two way of modeling their business processes: with projects, aimed at giving a minimal structure to work and collecting a maximal amount of feedback, work logs etc., or using business process models, which are workflows. Workflows are more rigid but more accurate. They are more complex to plan but often easier for the final user, who has just to say "proceed" on her/his tasks when it is the case.

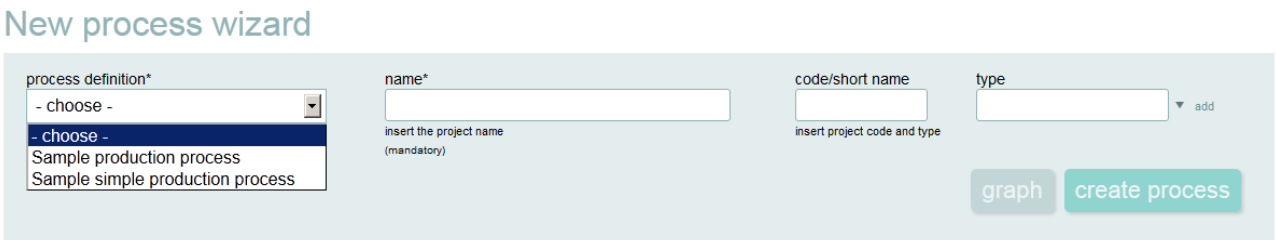
Process steps are actually tasks, so that say search in projects would find also steps contents.

Steps to be done (read: tasks to be closed) will be presented to users in the same locations where she usually finds her assignments and tasks.

So basically we have a wizard which given a process definition, lets you pick the assignees for the process' nodes, and will generate a process instance which will guide project advancement, notifying and recording step progress. This is more flexible than the classical swimlane based business process assignment, because to the same swimlane there may . Processes are defined in JPDL, a powerful business process definition language which covers all the usual fork/join/milestone etc. needed in process management.

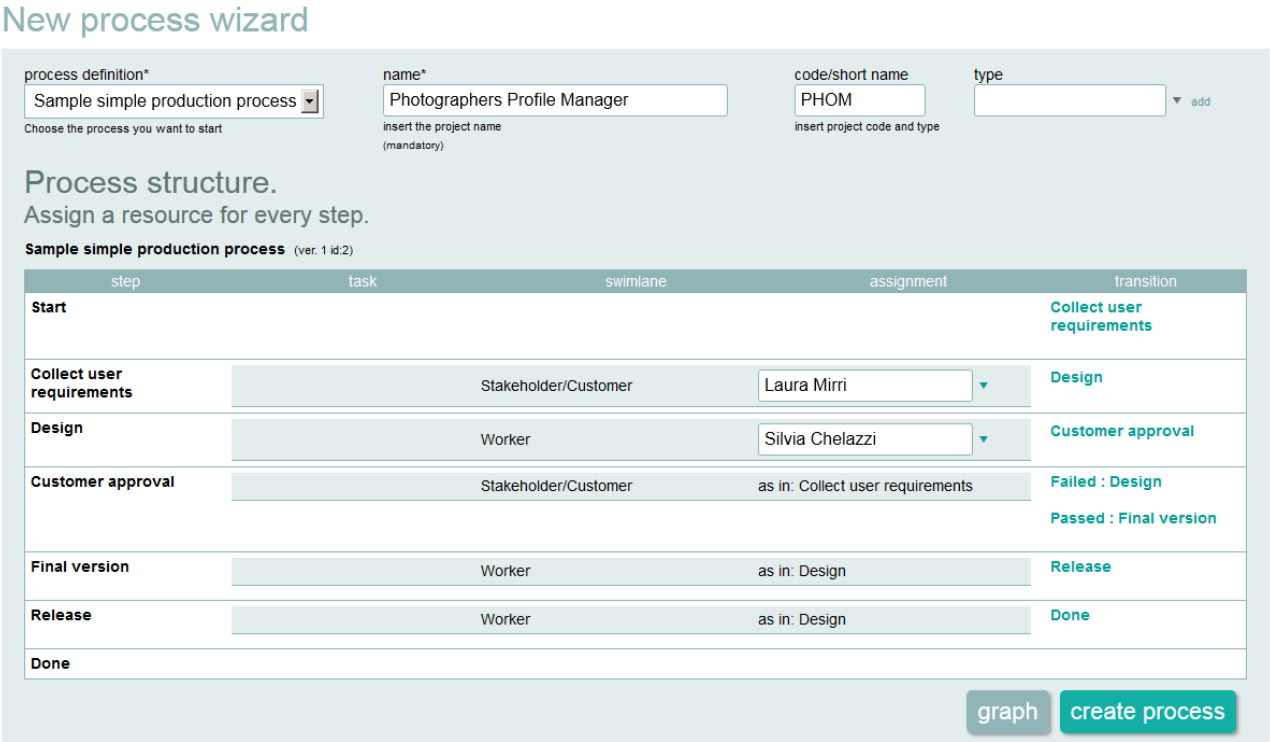
There are three aspects of workflows in Teamwork: usage, administration and customization. In this section we will examine the usage only. See chapter "15 Advanced business process" for details on the other two points.

In order to start a new project as process go to the projects page and click on "create process" button:



Fill the form with process definition, name, code, type. Some test process definitions are usually loaded by the installer.

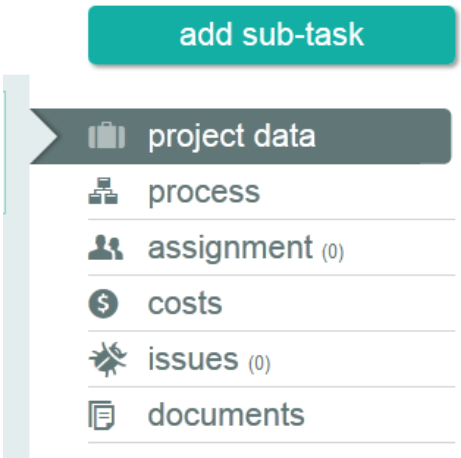
The page will show you the process structure in tabular form, moving on transitions you will see the next step.



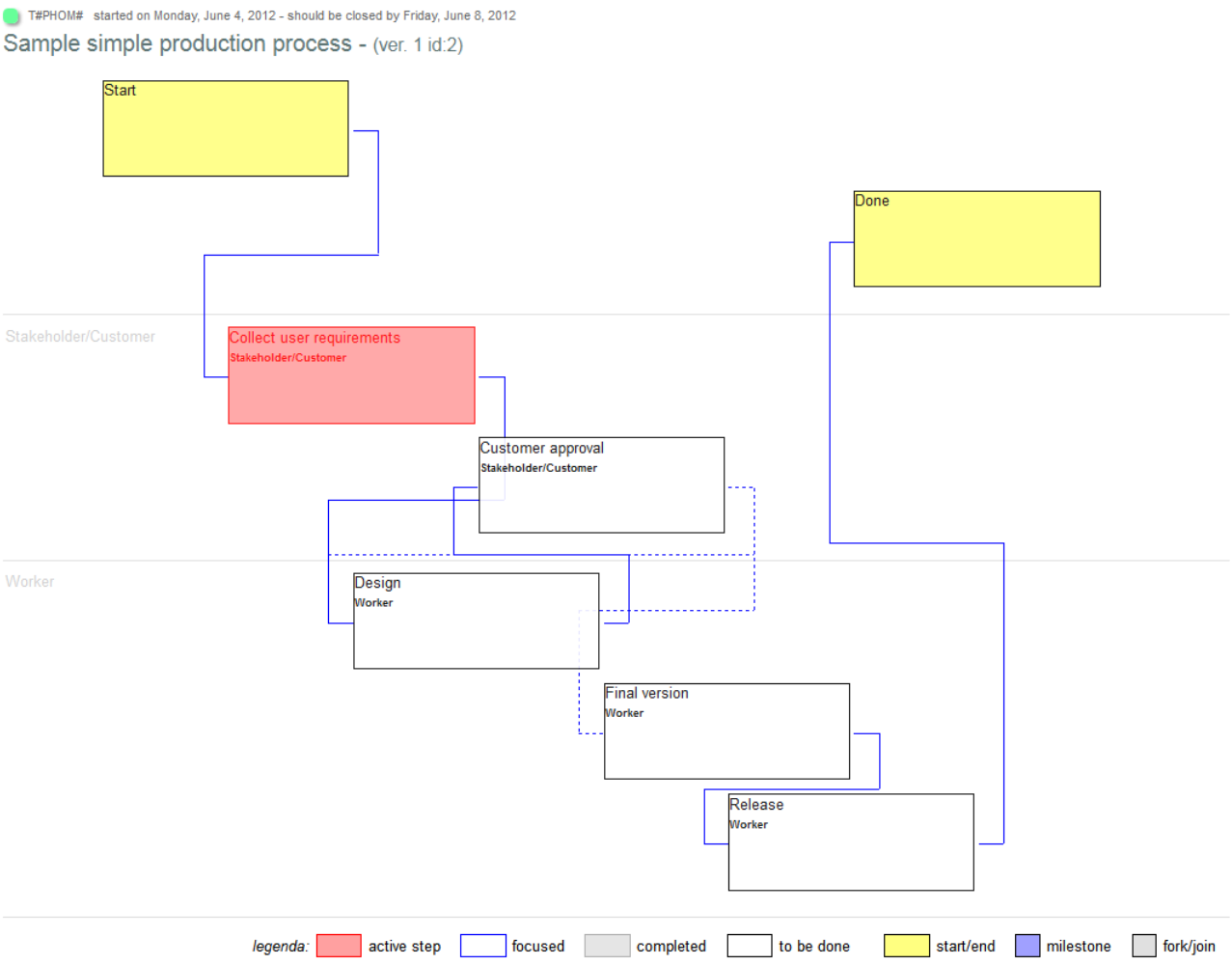
For every step a task will be created, so here you have to assign a resource. The role is pre-filled in the process.

Fill every assignment and click “create the process”; you will redirected to the task editor page:

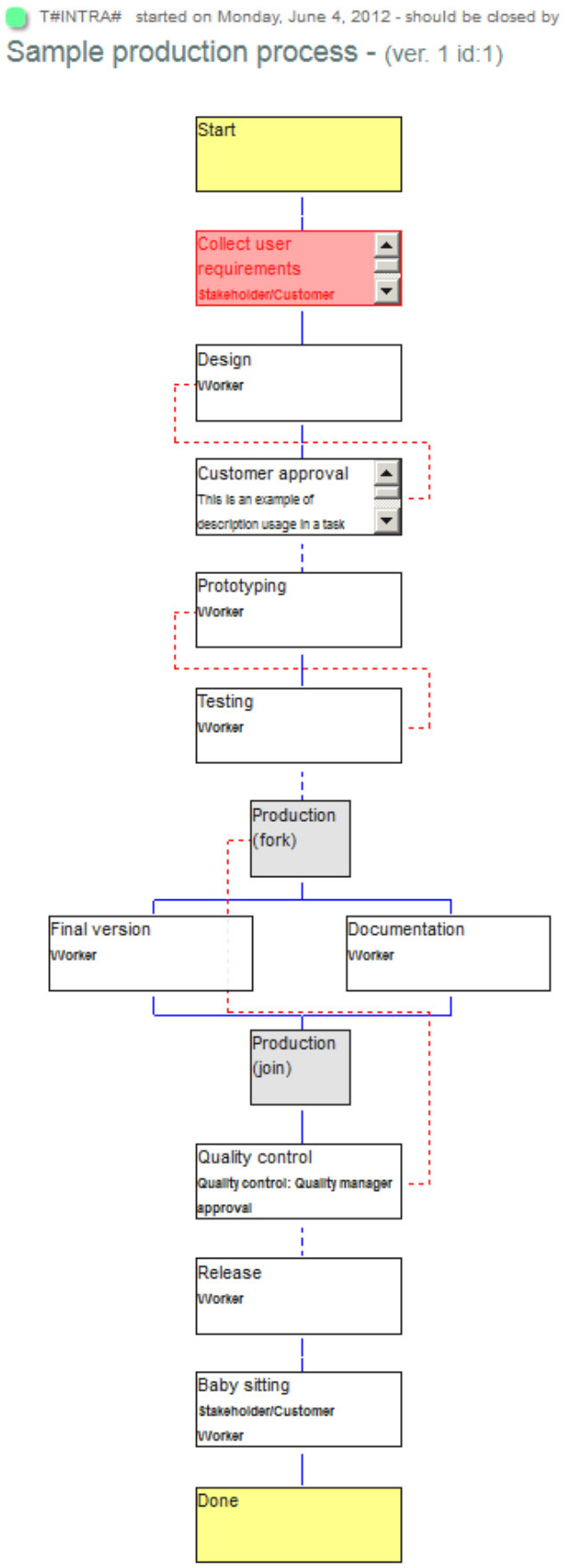
As you can see process driven tasks have a slightly different editor and have a “process” button instead of the Gantt / tree link. This because processes are managed by a separately defined process structure, and not by editing a tree.



The swimlane view:

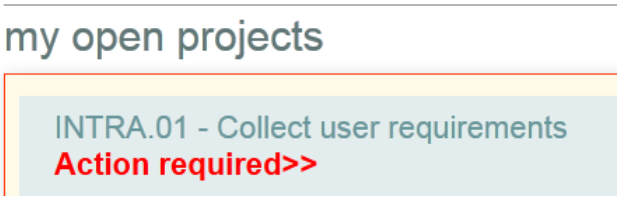


And you can see it as a graph:



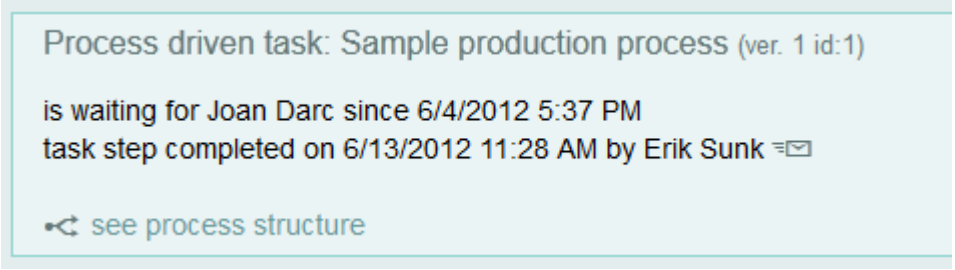
Pink task are currently active, gray ones are completed.

When the current task is waiting for you action, you'll have it on your home and a button will allow

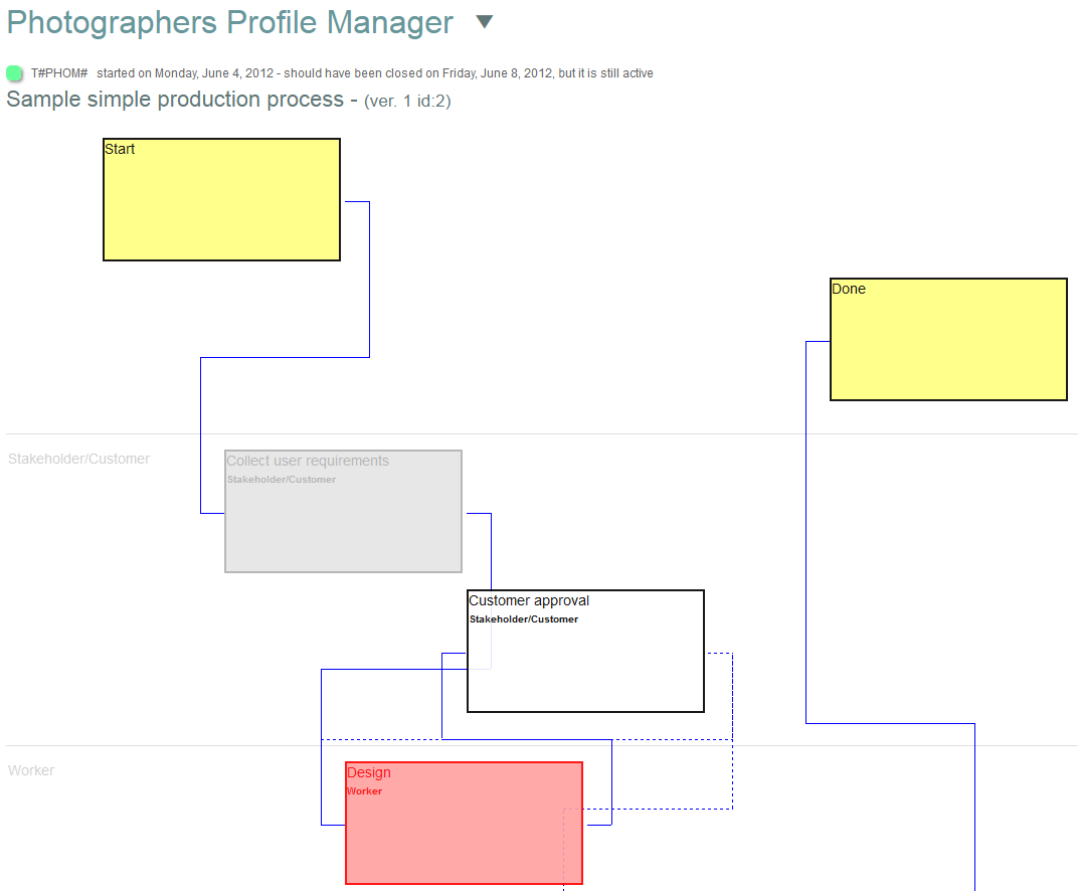


you to complete the task:

by clicking it a message will inform you about execution result.



Executing a step will automatically change status of tasks in the process flux.



Seems simple? Ok, now follow this hyperbole….

Reducing trees to business processes

Converting a business process, that by default can be a complex graph (not just a tree), in a tree structure (that is an "oriented" non circular graph[5]) is not only not trivial but also impossible, so we adopted some reduction rules.

[5] Technically: "A tree is an oriented graph with a distinguished node (its root) from which to each node there is a unique oriented path."

First of all every process' node that requires user action (called task-node in process idiom) is converted to a Teamwork task. For every process-"task" (in processes a task is an action required from a swimlane in a task-node) we create assignment with the role of the swimlane.

By doing this you can have task that require double confirmation (so you can have for instance processes with joint signing).

Example: a task-node of the process requires one action from swimlane A and an action for swimlane B. This is converted in a single task with two assignments.

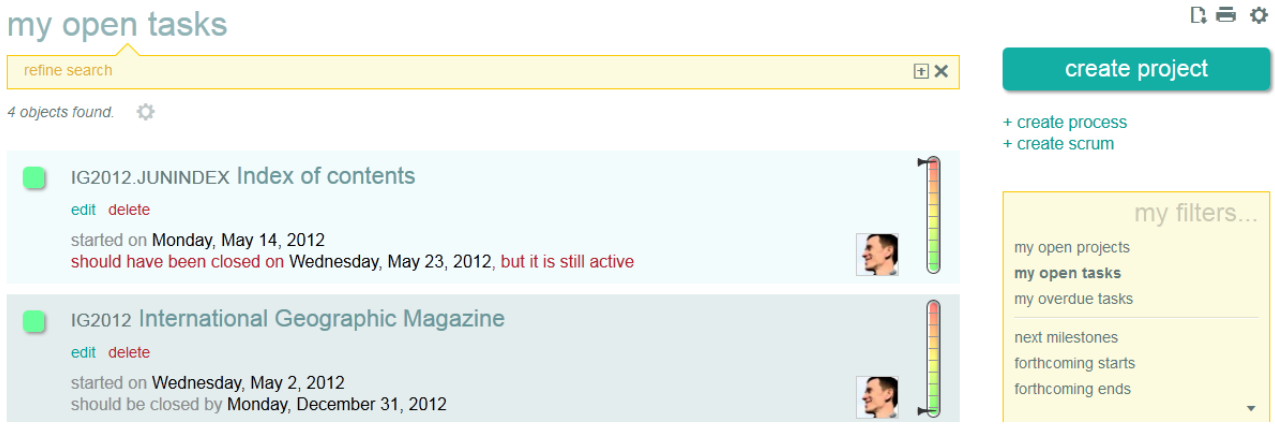
Then there is the graph reduction problem…. We solved this by applying a "running reduction"; this means that every loop is cut and straightened during the instantiation phase. During the running phase when the process loops back, the statuses of tasks (Teamwork tasks) are changed accordingly, a sort of rewinding of time, but the path is always straight.

For a complete overview about Teamwork business process, administration, maintenance, customization and JBPM technicalities see "15 Advanced business process".

3.12 Search / reporting

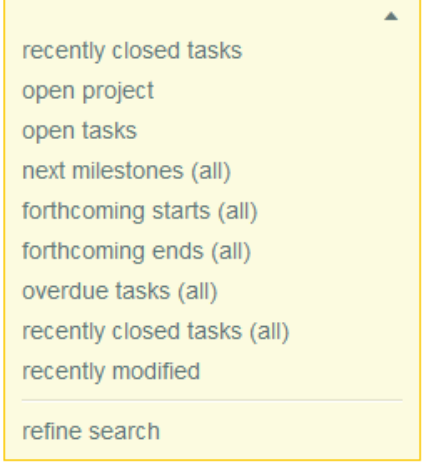
Teamwork does everything possible to let you find your projects at one click distance but also includes a powerful specific search/filtering function for tasks/projects.

Click on the project button:



This page contains some useful prefilled filters:

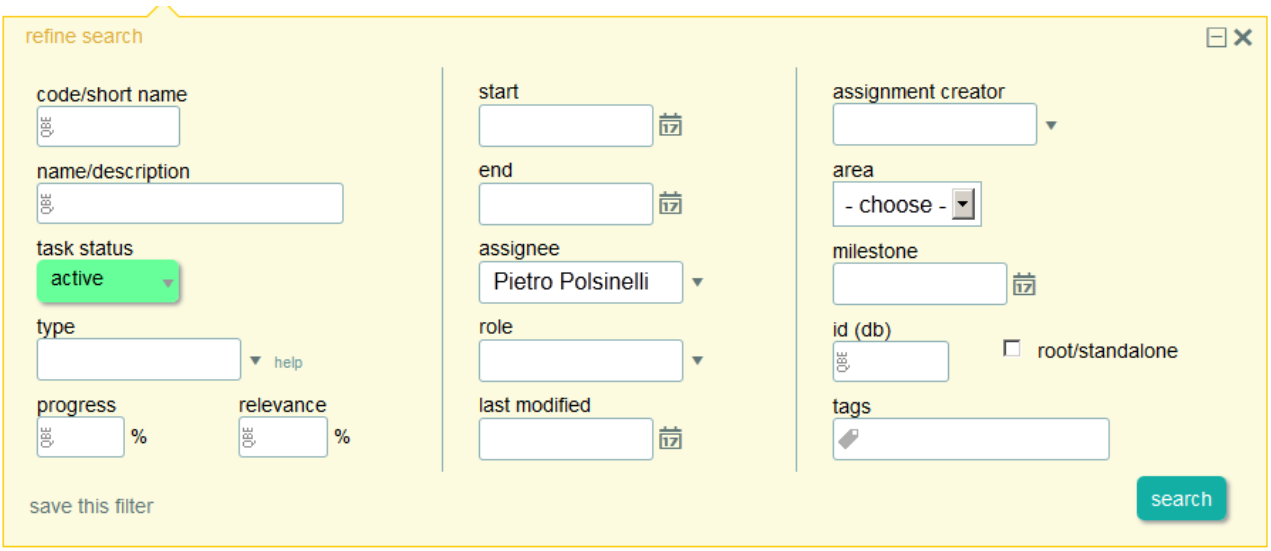
1) **my open projects**: all "root" open tasks where you are assigned
2) **my open task**: all open tasks (root or not) where you are assigned (this is the default one)
3) **my overdue task**: task where you are assigned, that are still open even if end date is overdue
4) **next milestones**: list of forthcoming milestones (in two weeks) where you are assigned
5) **forthcoming starts/ends**: forthcoming start/end (in one weeks) where you are assigned
6) **recently closed tasks**: task where you are assigned closed in last two weeks



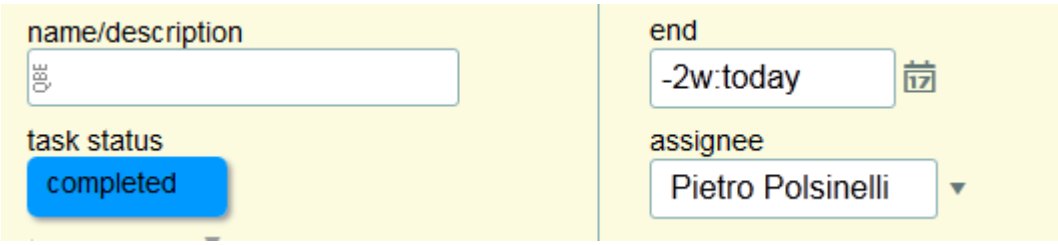
There are more filters: click on ▼ to see some other prefilled filters.

These filters are similar to the ones above but are not limited to your assignment.

If these filters are still not enough you can open the “custom filter” area by clicking on “refine search”:



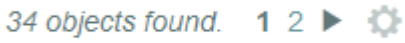
In this example you can see how for example the filter “recently closed tasks” is composed:



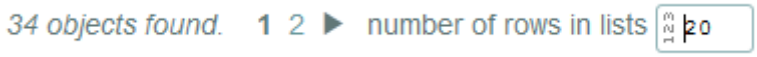
1) task status is set to “complete”
2) assignee is set to yourself (in this case “Pietro Polsinelli”)
3) end is set to “-2w:today”: wondering what the heck is that? This is a Query By Example (QBE) syntax meaning that “end” should be in an interval (semicolon means interval, [from]:[to]). The interval starts two weeks ago (-2w) and ends today ☺. Friendly, no? Also “functional”, meaning that it has no hard-coded dates (for a complete example of the power of QBE see “13.2 Custom filters and QBE”).

Every condition is used in logic AND with the others, so you can compose complex and powerful filters that can be stored to be reused on need.

Results are usually paged, and you can flip pages by using the paging bar.



Page size can be changed, just click inside the small input, change size and click the reload symbol. The page size of this list will be remembered now onwards.



This page, like most of Teamwork's list pages, allows “print” and export to Excel .

3.13 Copy/move projects

Teamwork allows coping/moving tasks. While moving is a task tree branch may sound strange, copying is a really smart feature, frequently used.

Often different projects share a common structure, and sometimes are almost identical. In these cases the possibility project copying (or “cloning”) is a real advantage and saves you a lot of time.

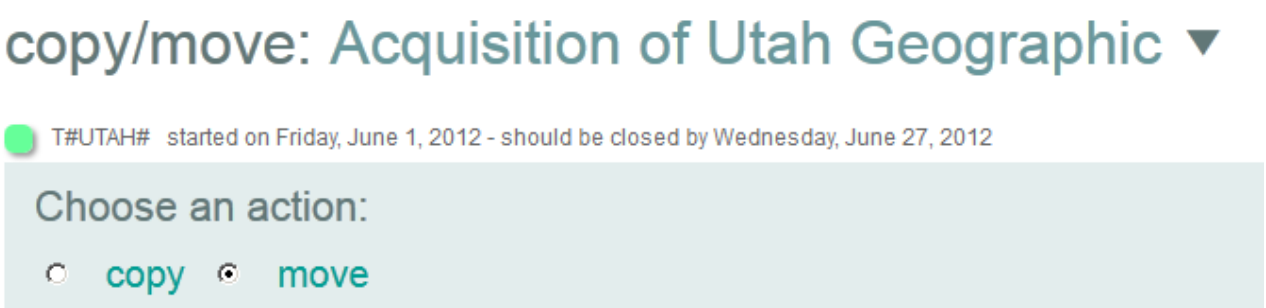
In addition you can prepare some task skeletons (called “templates”) prepared ad hoc for this purpose; this is a smart way but it is not strictly necessary: Teamwork allows you to copy every task.

Copy

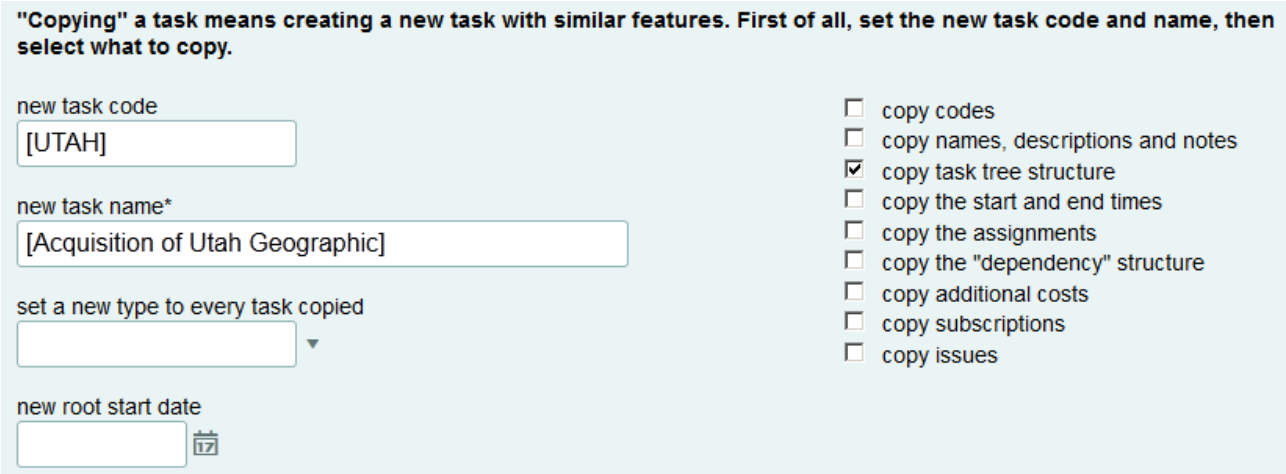
copy/move

Let’s examine the copy functionality. First of all go to the task you want to copy, then click on copy/move:

> Prepare a custom filter for task’ type=template and call it “templates”, so you can in few click have the list of “copy-able” tasks. But notice that Teamwork always lets you use *any* task as template.



Then select the “copy” action:



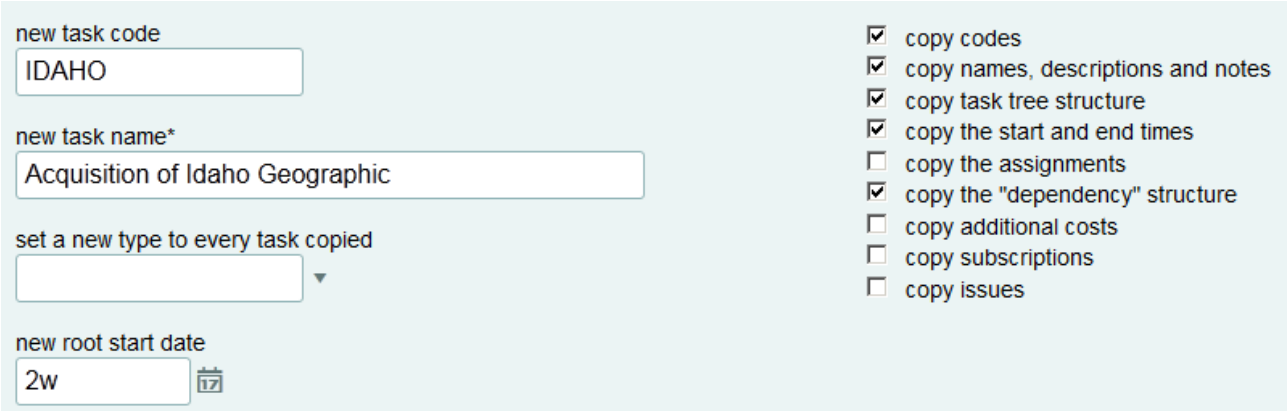
As you can see the copy function is quite flexible:

1) **code and name**: first of all the original task’ code and name are bracketed, usually here you will assign a new code and name (without brackets ☺) .
2) **type**: you can assign a new type to your new project; this because often “templates” are of type “template” and you may want to reset it to, for instance, “production”. If you leave the field empty the original type will be used.
3) **copy codes**: check it if you want to reuse codes from the template, otherwise the code will be generated from the root code and suffixed by .1, .1.1, etc. .
4) **copy names, etc**.: check it if you want to reuse names from the template, otherwise the name will be generated from the name above specified by adding the suffix .1, .1.1 etc. .
5) **copy dates**: check it if you want to use dates from template, otherwise dates will be reset.
6) **new root start date**: if you specify a new starting date the whole tree will be moved

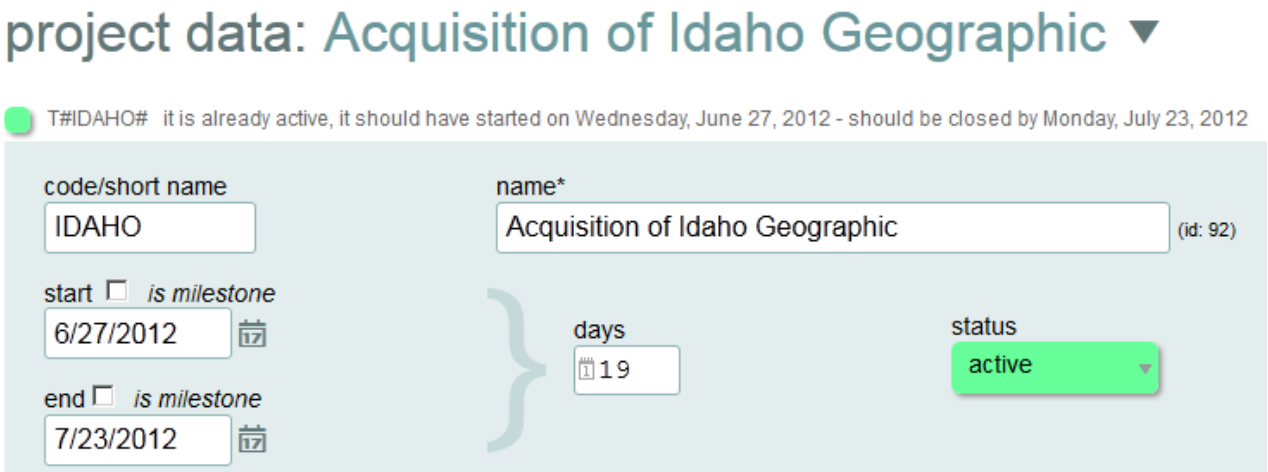
accordingly

7) **copy assignments**: check it if you want to use the same assignments of the template.
8) **copy dependencies**: check it to copy dependencies.
9) **copy additional costs**: check it if you want to copy additional costs. This means that you have inserted also fixed cost in you template.
10) **copy subscriptions**: check it to copy also subscriptions.
11) **copy issues**: issues also could be use as templates, by checking this flag Teamwork will copy issues too.

This could be a common set of checked options:

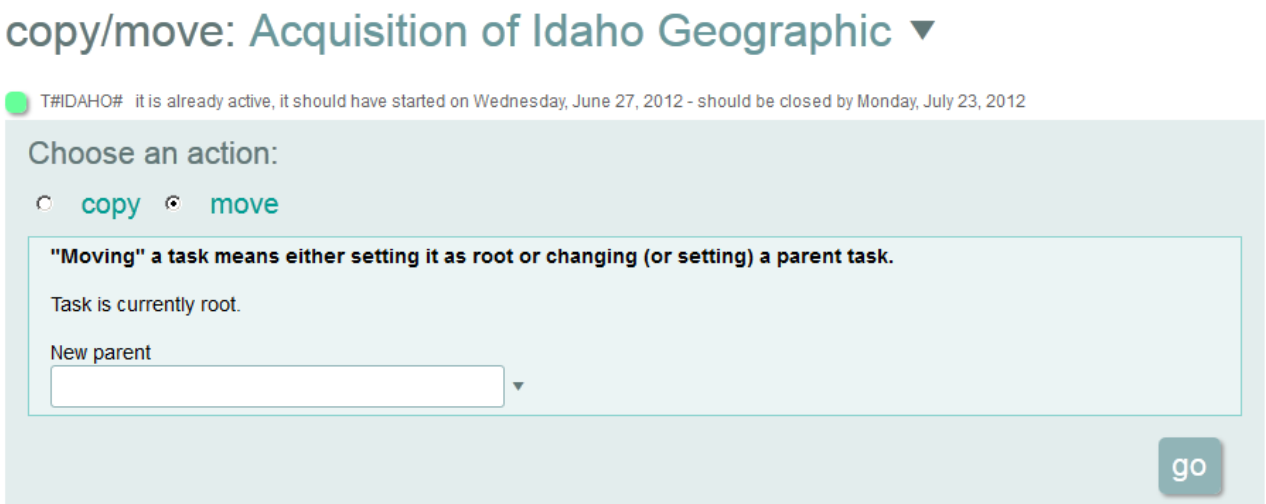


Press “go” and you will be redirected to the just created task, in the task tree editor:



Move

A second option in the copy/move page is move: if you want to move a task somewhere else, in the previous page, just click “move”:



Then specify a new parent, or if you are moving a child, just leave “new parent” empty to “promote” a sub-project (a child) as project (a root). Again you’ll end up in the task editor.

> All data recorded for a task is local to the task. So when moving a task, worklogs, costs, issues will be moved with it.

4. Issues

4.1 Introduction

What is an issue? It's something smaller / less relevant than a task / project. Issues can be to-dos, remember to, bugs, notes, suggestion, tickets etc. .

Usually issues are relative to projects / tasks, in this case a task is also a collector of issues.

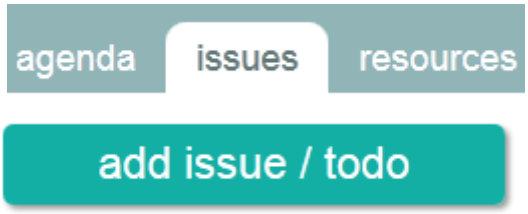
We have introduced issues in order to meet managing requirements for something lighter than a task.

"Issue" is common term in the programmer's world, but issues usage is wider.

> If you are used to bug-tracking, an issue can be seen as a wider notion than a bug

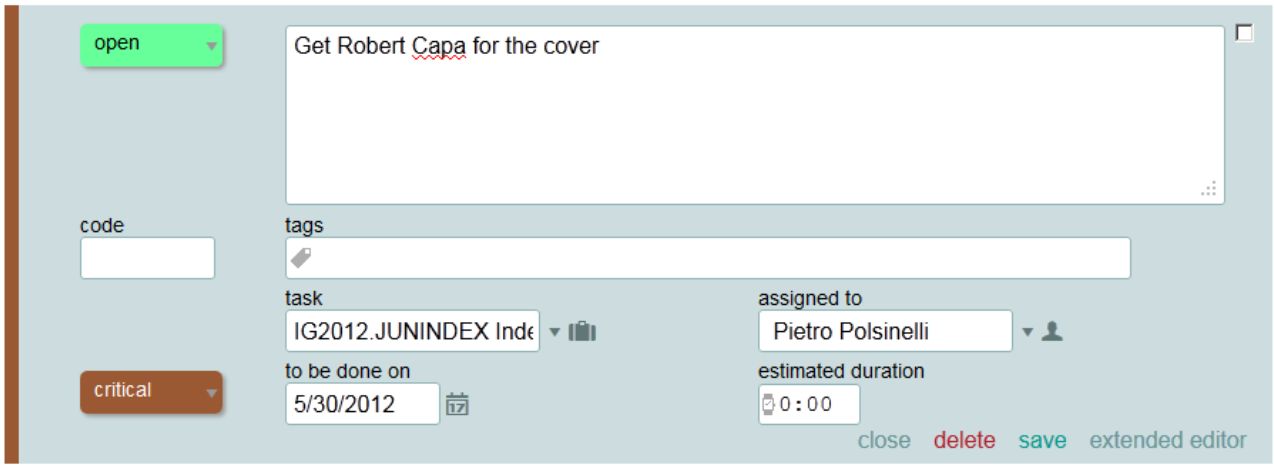
4.2 First issues

The easiest way to understand issues is by creating one; click on "issues" in the menu. Here you will find an "add button" which opens an editor in this same page.



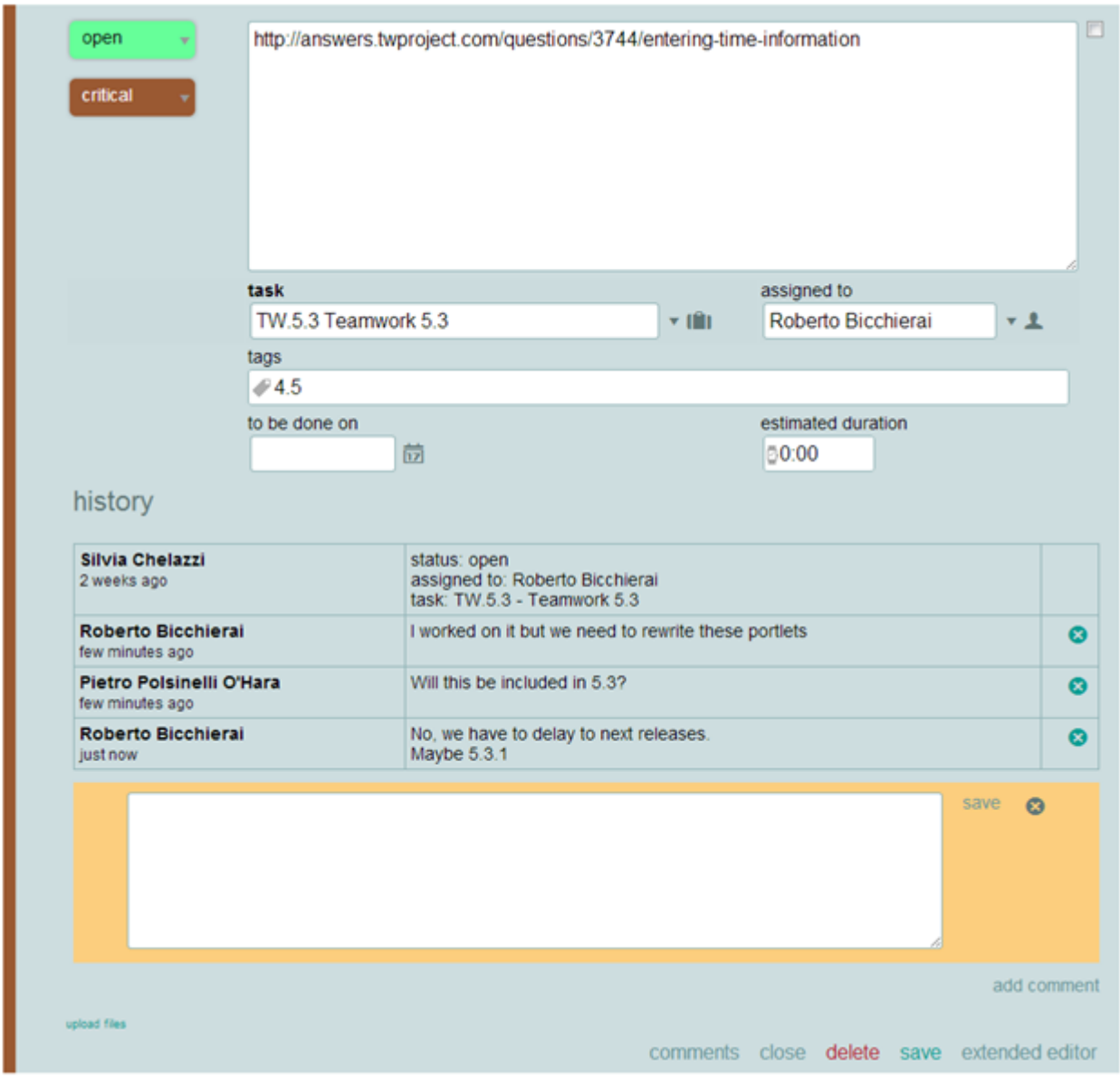
4.3 Editing

Issue state and order can be changed without entering the edit issue forms (see the part on lists below). Issues can be edited in place from the issue list – just click on edit:



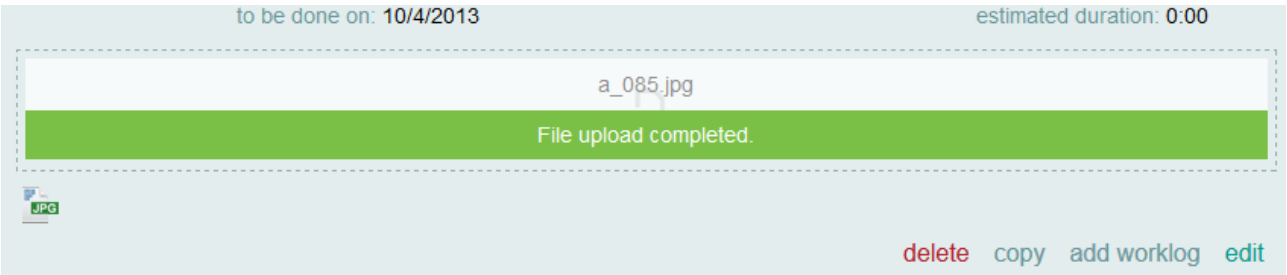
This view already gives you a quite extensive editing functionality: the issue text supports links and smiley and here you can associate people, set a task, pick a date for the issue to be done and a duration.

But issues are extremely flexible, and can be used for different purposes; to access the full spectrum of functionality click “extended editor”:

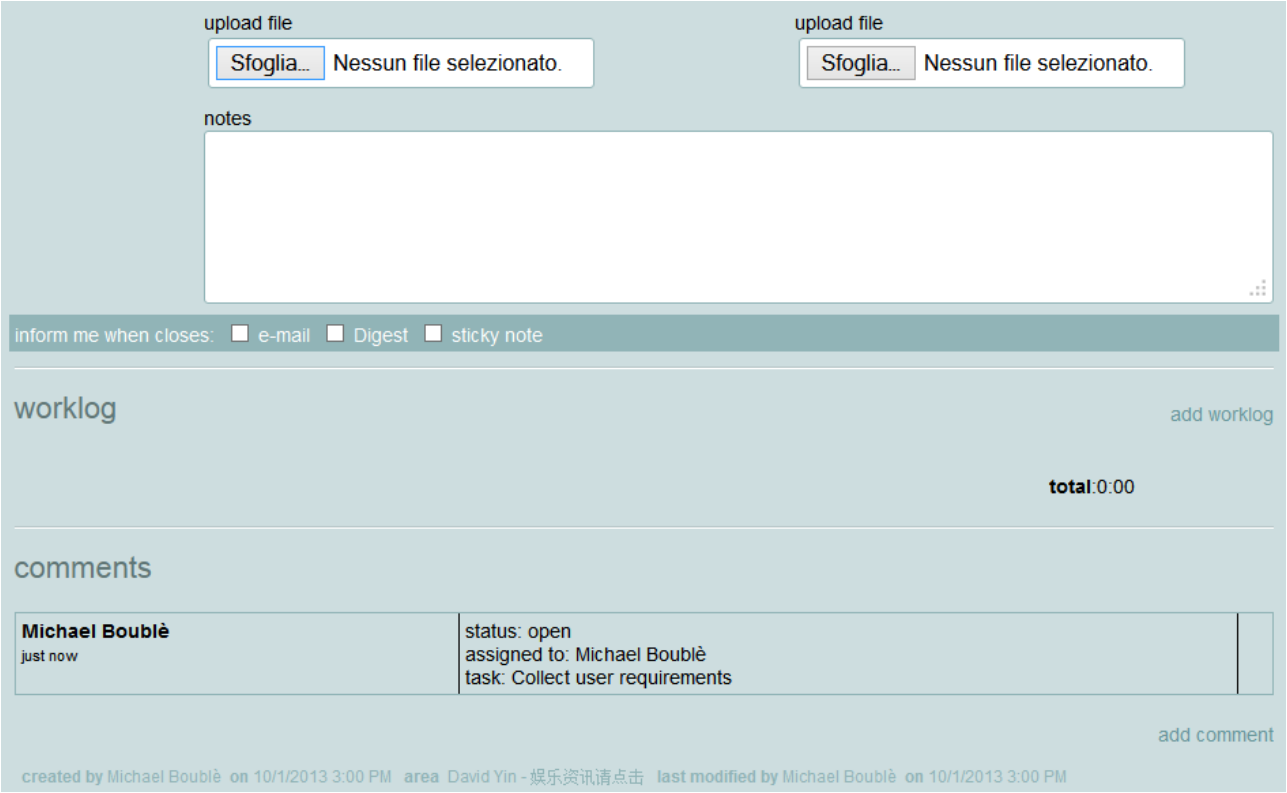


Each issue has status, description, eventually task (if not we call it to-do), type, severity, impact, requester, signaled date, work time estimation, assignee, date to be closed. And since version 5.3, there are also “comments”, that include the issue’ history.

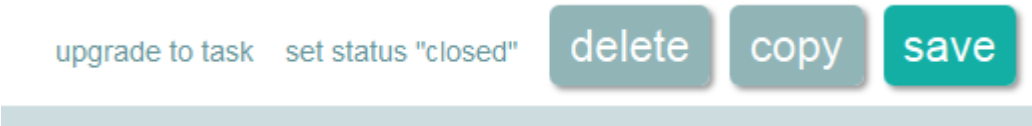
Also once an issue is saved, you can attach files by simply dragging them on the issue, finding the green bordered area:



And much more such as notes and work logs:

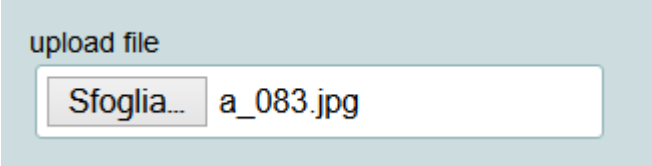


This rich set of feature makes an issue a versatile object that can meet different requirements.



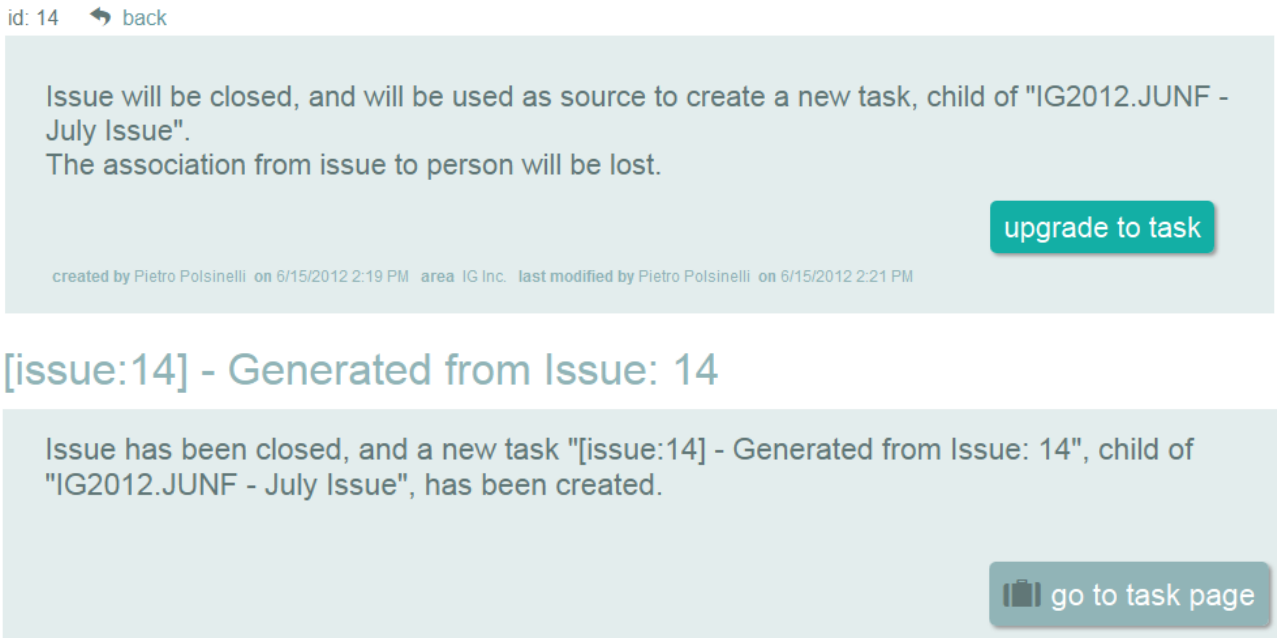
Lets check the upper button bar: by pressing the "copy" button you will obtain a new issue, copy of the current one except for the description that is left blank. In this way you can insert a lot of issues very quickly. The "set status closed" button will set the status to closed, save the issue and allow you to insert work log (if it is the case).

Another aspect of issues that increases usability is related to security. Permissions required to insert issues are distinct from permissions on task, so for instance you can create security profiles with read-only permission on tasks and creation permission on issues: we can call this set of permissions "customer" and create an ad-hoc role. Assigning customers with this role will allow them to insert issues right at developer's hand; it also facilitates creation of a "backlog".

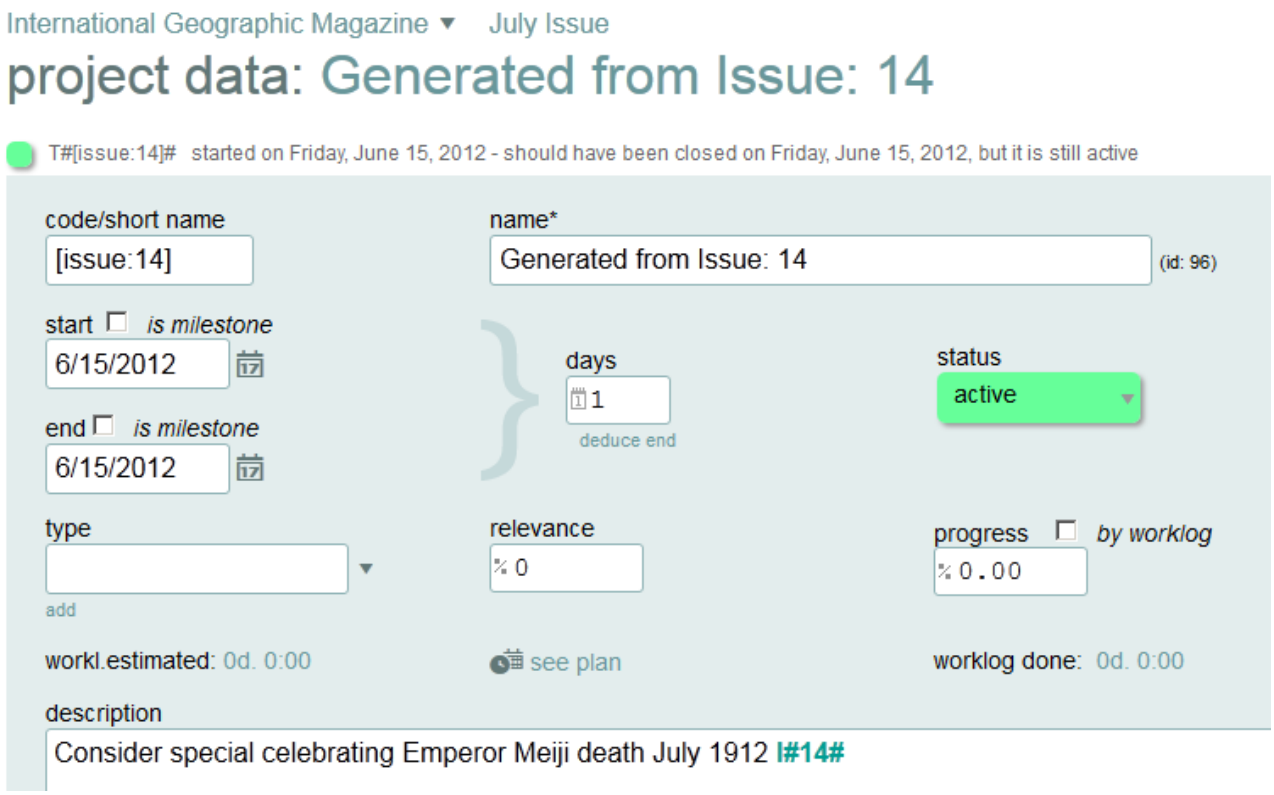


You can upload files, typically screenshots or similar, or link files on Teamwork' file storages (see chapter "9 Document"). The files are shown linked from the issue list.

If you find out that the issue is more complex than initially estimated, you can promote it to "task" by clicking on "upgrade to task" button. In this case the issue will be closed and a new task under the current one will be created:



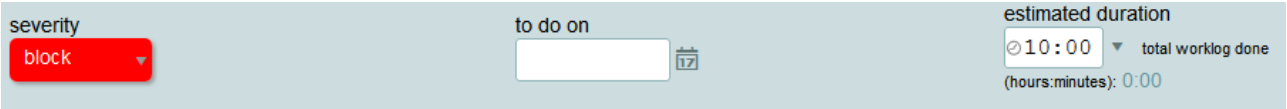
The result being:



Where you clearly have to edit the main fields.

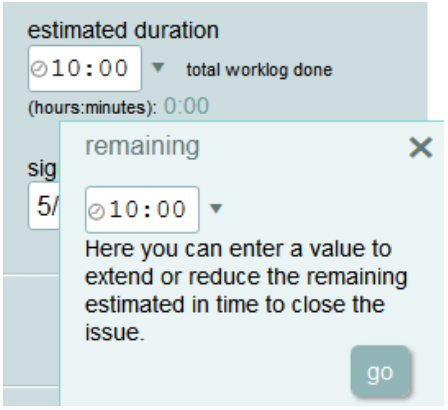
Moreover you can enable use of external codes on issues (admin -> default for projects).

Issue re-scheduling



A smart Scrum team leader that is using Teamwork remarked the following: suppose that you are a developer and are assigned on a set of issues, on which you do your development and record development time spent. You did with the team the initial evaluation of needed development time, and suppose for a particular issue you decided to put 10 hours.

You recorded time elapsed, but as happens in life all the time, you have to reschedule some of the issues. Now the users remarked that it is quite cumbersome to reason on the base of estimated duration - worklog done, because all you are actually focused on is time remaining.



So, here is the change: by clicking on worklog done, a time remaining panel appears, and its editable right there.

This little practical change can make a difference; think when you planned 34 hours, and have done 27:30, how many hours to go, will it suffice.. I don't want spend time on that: just let me reschedule that.

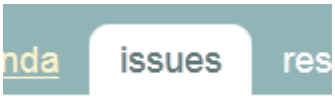
If you want to link directly to an issue via a URL, the URL will be:

http://[your root]/i:X

where X is the ID of the issue.

4.4 Search / filter / close / order

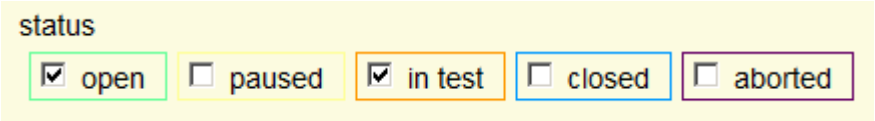
If you want to manage your issues in a tabular form, just click on the “issues” button on the top menu.



You will get the list of you open issues:

First on this page are some prefilled filters:

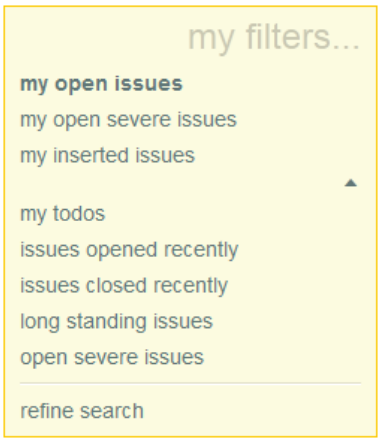
1) **my open issues**: issues where you are the assignee and are either open or in test.



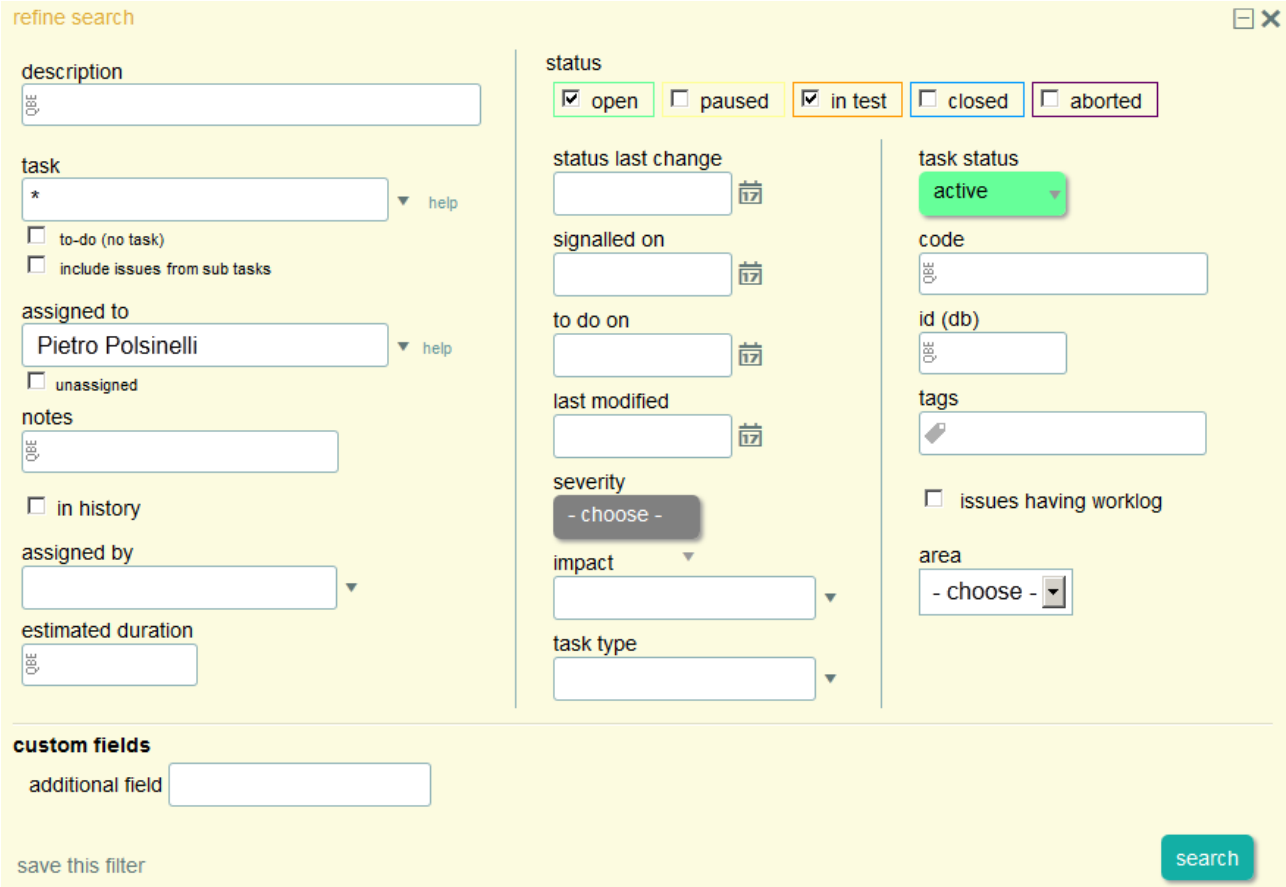
2) **my todos**: is the list of your to-dos
3) **my open severe**: if you have lots of issues you should start considering severe ones first ☺
4) **my inserted issues**: issues that you have inserted that are still open

and more:

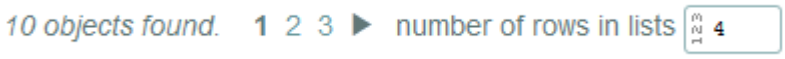
5) **issues opened recently**: two weeks
6) **issues closed recently**: two weeks
7) **long standing issues**: one month
8) **open severe issues**: not only yours (filtered by security)



As usual by expanding the custom filter you can see how pre-filled filters work and also compose your preferred filters.

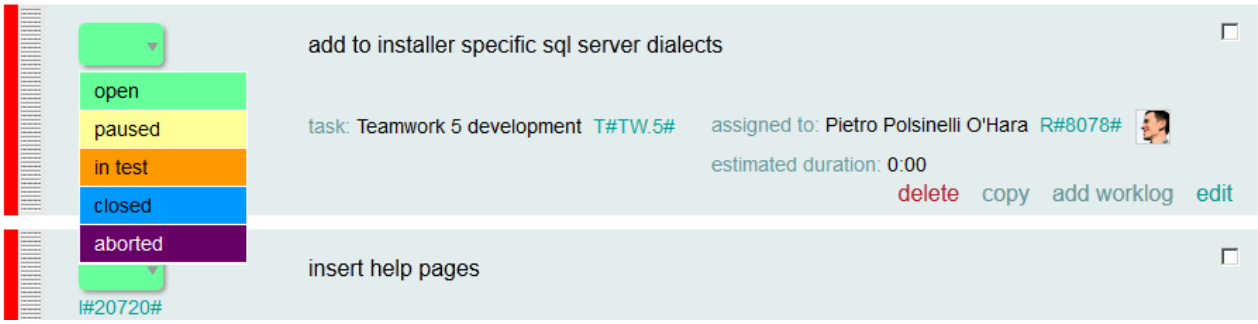


Results are usually paged, and you can flip pages by using the paging bar. Page size can be changed:

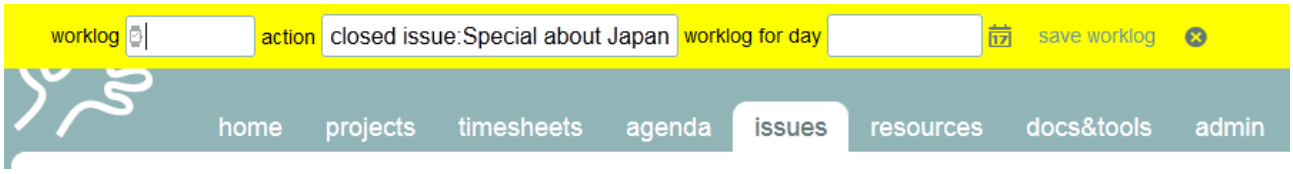


But the list view allows also some operations on issues without entering editors.

First you can change issues' status, typically closing them:



When you are closing your issues that are related to a task where you have an assignment you can insert worklog right there:



But this page has even more features. First: notice that on the right there is a drag handle: you can sort issues as you like. One constraint only: severity wins over your whims.

Hey I don't understand sometimes the handle is there and sometimes it isn't!

Notice that you can order issues **only** when we are talking about *your* issues or *issues from a single task*. If you compose a general filter the drag handle will be hidden.

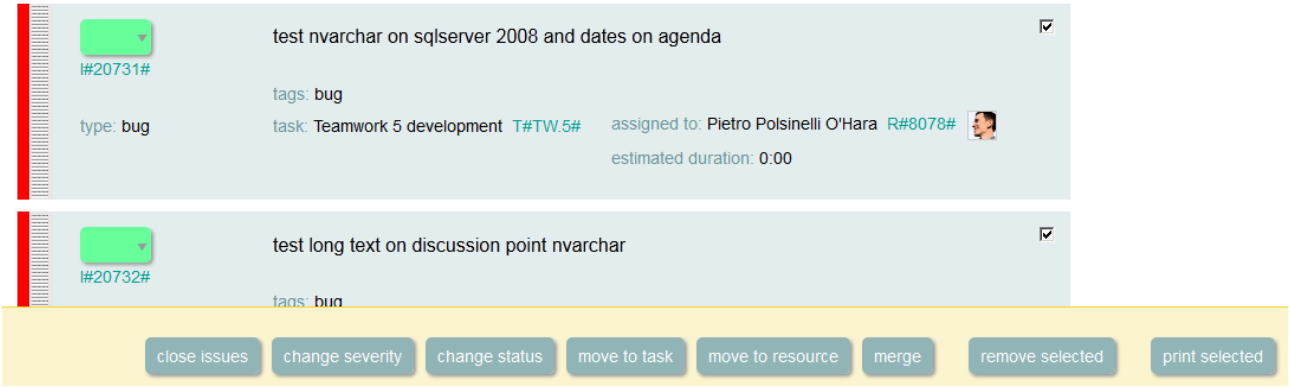
If you get to the issues page from a task by clicking on the "issues" button, you can sort issues as well, but the sorting will be strictly relative to that task. Summing up, issues have two sorting factors: one by assignee and another by task.

This page, like most of Teamwork's list pages, allows "print" and export to Excel .

4.5 Bulk management

There could be little worse in work management than discovering that the set of issues that you've carefully inserted on a task should be somewhere else and… should be re-inserted; this will never happen in Teamwork, thanks to "bulk operations".

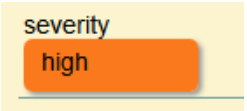
From the issue list page you can perform some simple bulk operations.



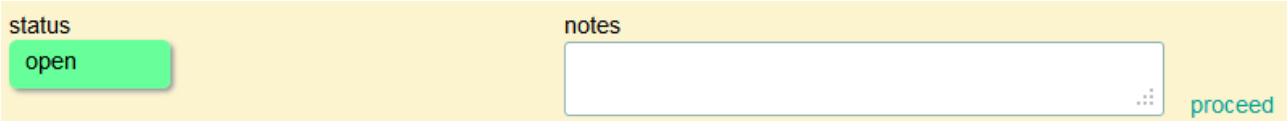
By selecting one or more issues, a button bar will fade-in at the end of the page. You can select every issue by clicking on the checkbox on top of the first column. Here are the possible operations – all will ask for confirmation before being applied:

1) **close issues**: you will set status of issues to "closed"

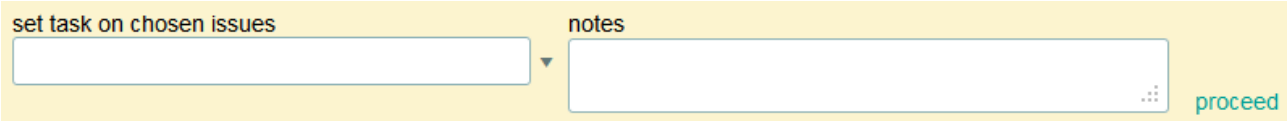
2) **change severity:**



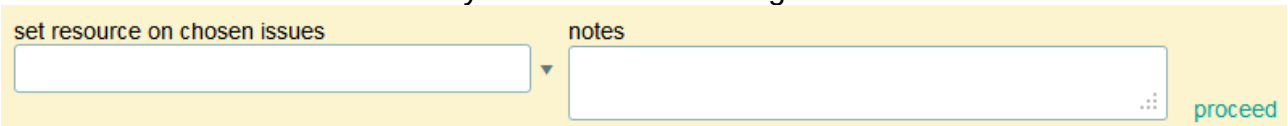
3) **change status**: a bar will let you select a new status with notes, by clicking proceed you will apply the status to each issue.



4) **move to task**: a bar will let you select a task where to put you issues.



5) **move to resource**: a bar will let you select a new assignee for issues.



6) **merge**: descriptions of selected issues will be "merged" in a single one. Issues must be "compatible" in the sense that task and assignee must be the same.

7) **remove selected**: remove selected issues.

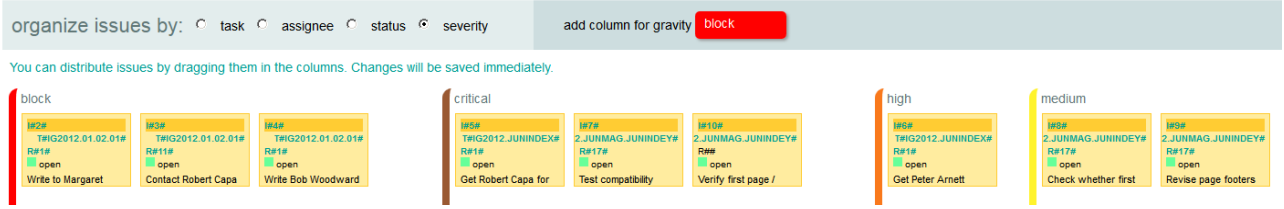
8) **print selected**: print selected issues.

4.6 Organize – Kanban - Planner

Agile methods, Scrum and Kanban have already been discussed in *ix Agile methods, SCRUM and Kanban*.

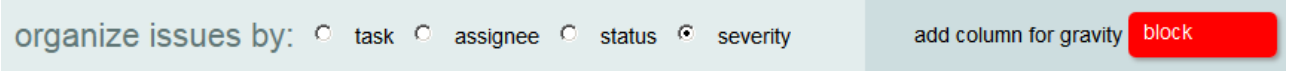
Issues can be now be organized in a completely visual way by dragging and dropping them: filter the issues in which you are interested in, and then select the “organizer” button, getting:

organize (Kanban)

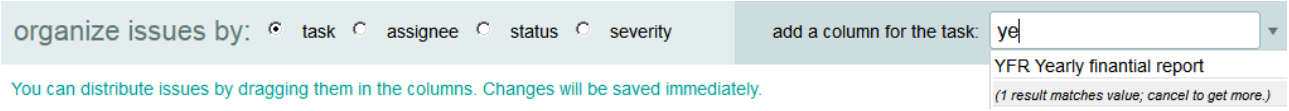


You can see the full text of the issue by simply hovering on it.

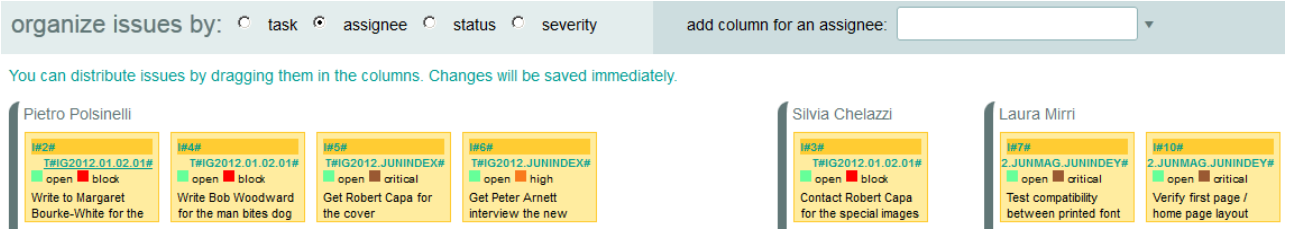
Issues will be presented in distinct columns, e.g. by severity level: now you can change their severity by just dragging them around. And you have several grouping/editing criteria you may choose:



For most types, you can add columns if you need to. For example, if you group the issues by task, and say you just created a subtask (a sprint, maybe) to which you want to move some of the issues, just select add column:



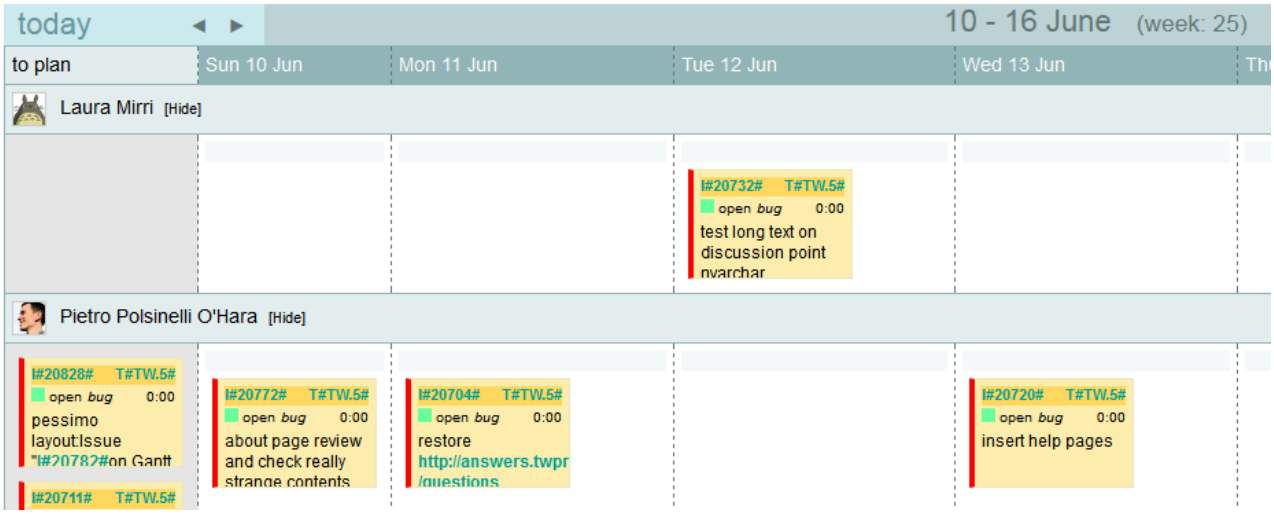
Once done, and moved the issues to that task, you may distribute them to different resources by just selecting “assignee” as grouping and adding columns:



It is a form of multi dimensional management – extremely powerful. And for those into agility, it is a great tool for management.

A new section available since version 4.8 is the Issue Planner:

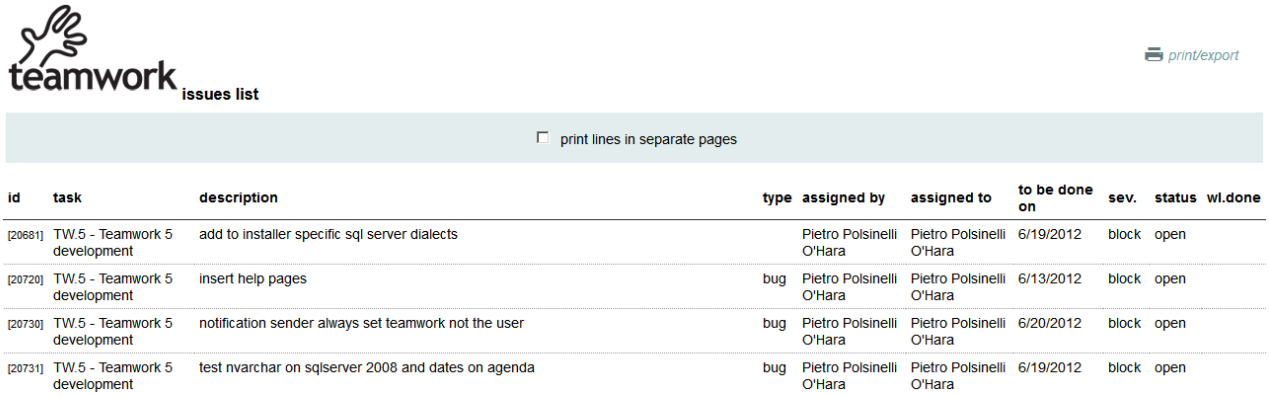
planner



For the given filter, the issues will be grouped as “to be (re)scheduled” and scheduled, and all the users involved will be displayed. You schedule the issues day by day by simply dragging them around. You can of course also change assignee and add users rows.

4.7 Print

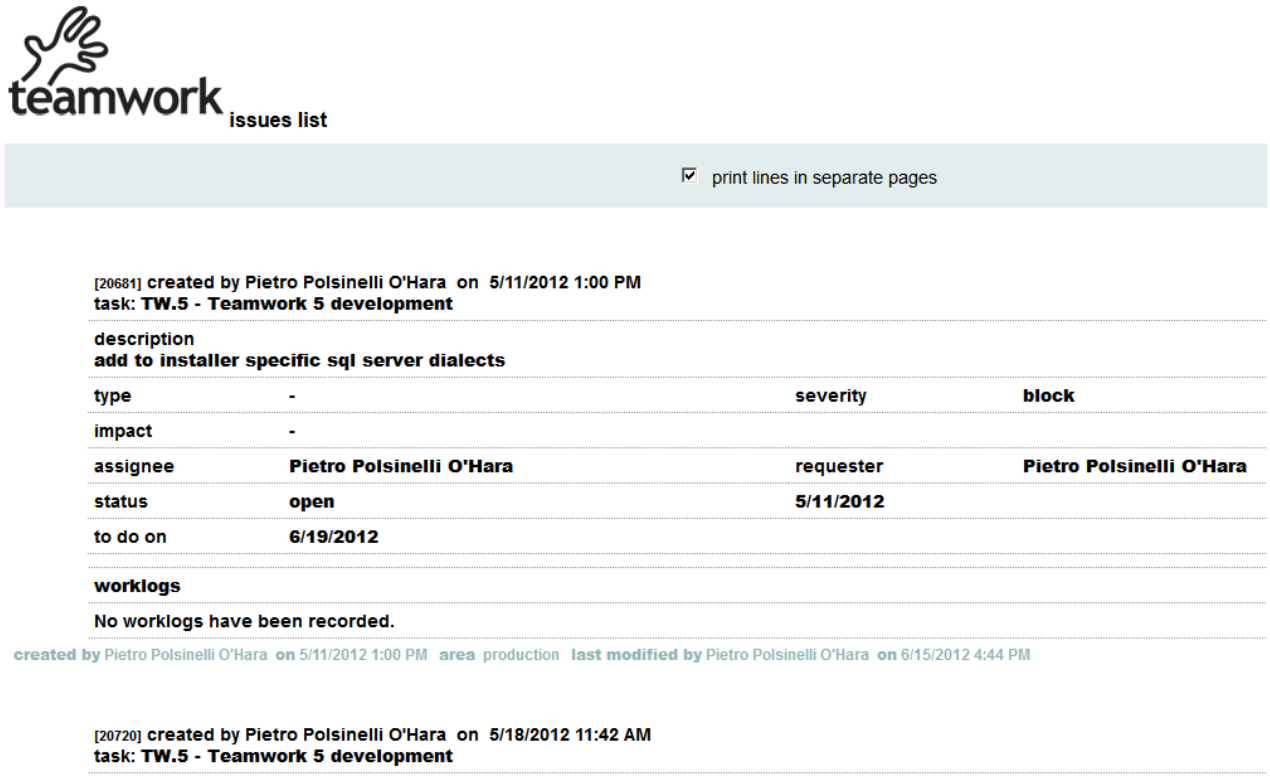
When you print issues you can have two different layout:



id	task	description	type	assigned by	assigned to	to be done on	sev.	status	wl.done
[20681]	TW.5 - Teamwork 5 development	add to installer specific sql server dialects		Pietro Polsinelli O'Hara	Pietro Polsinelli O'Hara	6/19/2012	block	open	
[20720]	TW.5 - Teamwork 5 development	insert help pages	bug	Pietro Polsinelli O'Hara	Pietro Polsinelli O'Hara	6/13/2012	block	open	
[20730]	TW.5 - Teamwork 5 development	notification sender always set teamwork not the user	bug	Pietro Polsinelli O'Hara	Pietro Polsinelli O'Hara	6/20/2012	block	open	
[20731]	TW.5 - Teamwork 5 development	test nvarchar on sqlserver 2008 and dates on agenda	bug	Pietro Polsinelli O'Hara	Pietro Polsinelli O'Hara	6/19/2012	block	open	

Tabular or (by checking

) one issue per page:



4.8 Importing and exporting issues

You can import issues from a CSV file, in particular it is the default format in which Bugzilla exports in CSV its entries. The page of the export also documents the CSV format:

issues import

You can import issues from a CSV file, in particular it is the default format in which Bugzilla exports in CSV its entries. The format is the one listed here.

bug id, bug severity, priority, op sys, assigned to, bug status, resolution, short desc

file to import

Browse...

Bugzilla	Teamwork
bug_severity	Not mapped, as severity in Teamwork is a fixed set (as it is used from some refined business logic, differently from apps where everything is a lookup and the application does nothing ☺). Always set to "high".
priority	Mapped to "impact".
op_sys	Added to the end of the issue description.
assigned_to	Mapped to "assignee".
bug_status	Mapped to "open" or "closed" depending on wehether the issue is marked as "resolved" or not in Bugzilla.
resolution	Mapped to "notes".
short_desc	Mapped to "description".

read issues from file

The export can be launched both from the root, or from a specific task, in which case the issues are obviously set on that task, in the former you will have to pick one.

4.9 Custom fields on issues

In order to supply a quick solution for adding a field to the issue form, Teamwork supports custom fields.

In order to activate them you just need insert a value for specific labels. To activate them, go to admin -> Customization -> on issues, you'll get to a dedicated page:

Issue custom field

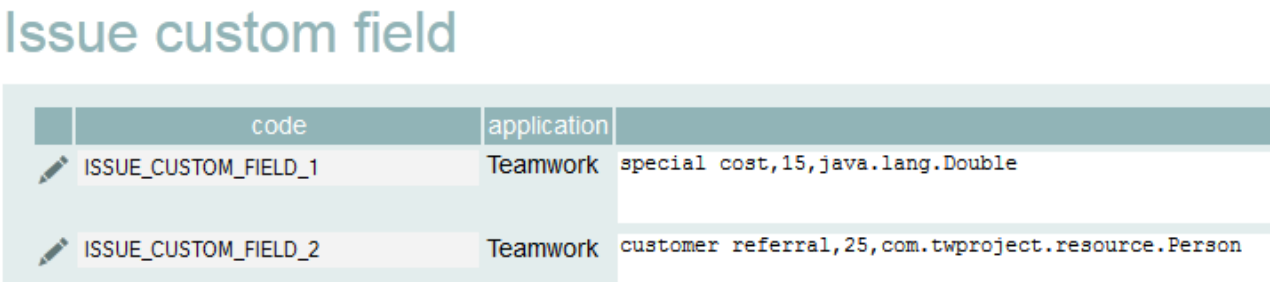
	code	application	
	ISSUE_CUSTOM_FIELD_1	Teamwork	no
	ISSUE_CUSTOM_FIELD_2	Teamwork	no

On such page you also have detailed examples and a technical reference, which we do not duplicate here.

Custom Field examples:

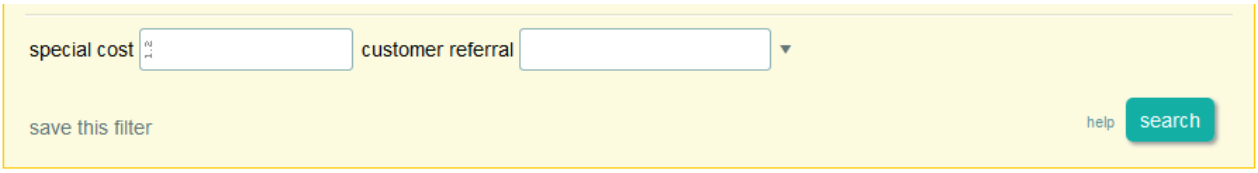
- Unused field: `no` -> the field is unactive
- Simplest case: `my code` -> a string
- Lenght defined case: `my code,20` -> a string 20 chars lenght
- Double case: `cost,15,java.lang.Double` (also Integer or Long)
- Date case: `when,10,java.util.Date`
- Teamwork object case: `customer referral,25,com.twproject.resource.Person` -> a Person
- `see this task,25,com.twproject.task.Task` -> a Task
- `attachment,15,org.jblooming.ontology.PersistentFile` -> an uploaded file

So if for example you add two such custom fields:



And you get on the issue form:

You can also search / filter by these fields in the issue search page:



You can any time add and remove such fields, the connected data will not be lost. But be careful **not to change data type** for the same field, name: that indeed will crush data integrity.

4.10 Mobile support

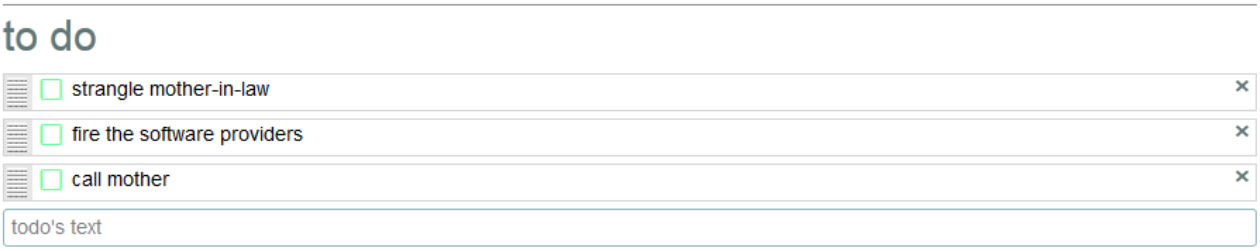
You don't need to install anything in order to use Teamwork from a mobile device: you just need a browser and to be online, your Teamwork has to be online too. You will find the mobile application at this address

http://[yourdomain]/mobile

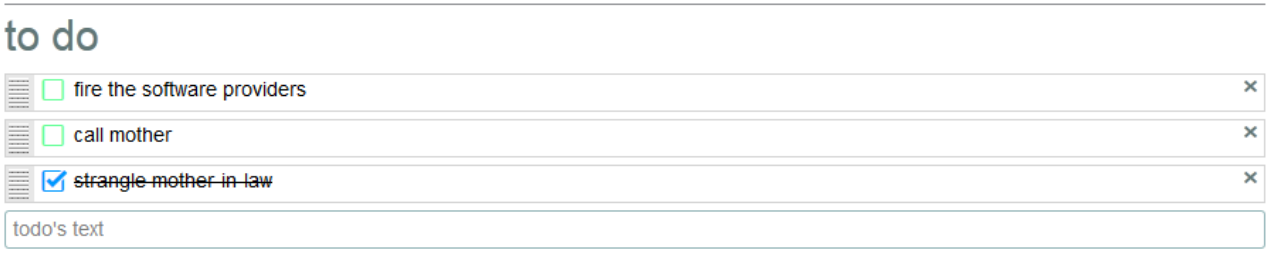
See chapter *20 Teamwork Mobile*.

4.11 To-dos

Teamwork operative homes always have the to-do list portlet.



Just type text in the “to-do’s text” field and press enter, a new to-do will be created. Just click the “done” check when you have completed the activity (will be saved automatically).



You can, of course also sort or delete to-do's.

5. Groupware

5.1 Introduction

In successful workgroups one of the secrets of success is how easily information is collectively updated and flows through. Teamwork messaging system, with subscriptions, alerts, remainders, can give you a hand.

Teamwork helps collect and distribute information about tasks, issues, appointments, and in general on everything concerning the working group.

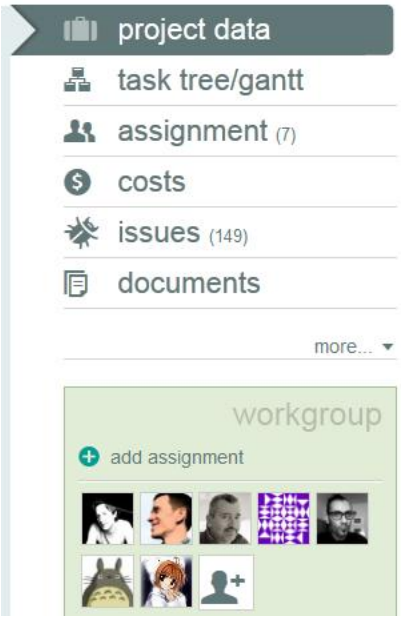
First of all, what are workgroups?

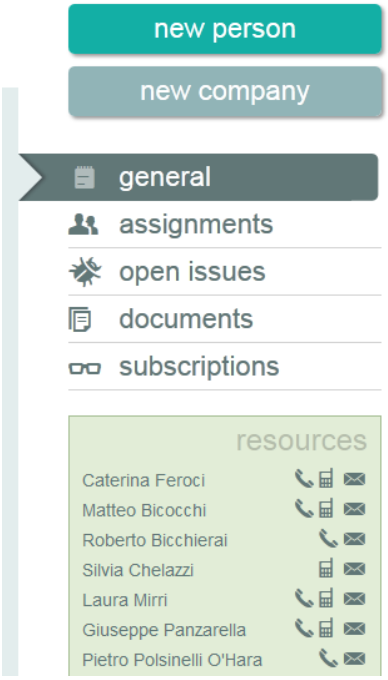
5.2 Workgroups

Teamwork implicitly defines workgroups on every project, company or department. For example, when you are assigning resources to projects/tasks, you are with the same effort planning the task, defining security **and** composing a workgroup. No repetitions required!

The fact that a company/department implicitly defines a workgroup is probably quite intuitive; but if you think about it for a moment, the same can be said for any project/task.

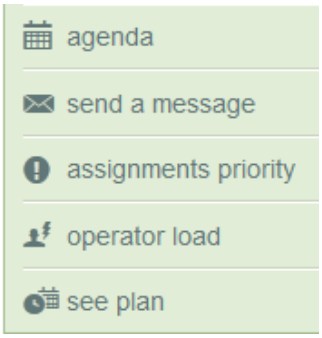
So you can access a workgroup from both the resource editor and the task editor:



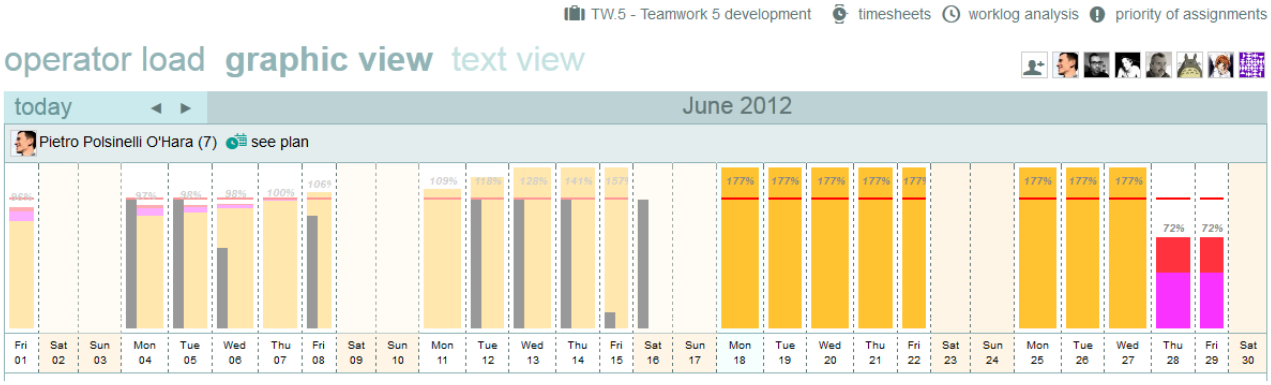


What can you do with a workgroup? For example you can access the group agenda, say to plan a meeting for the whole group; or you can send a message to the group.

So if you have a group of people as workgroup on a task, just click on agenda and you'll get the agenda filtered by such workgroup:



Same for "operator load" and the others:



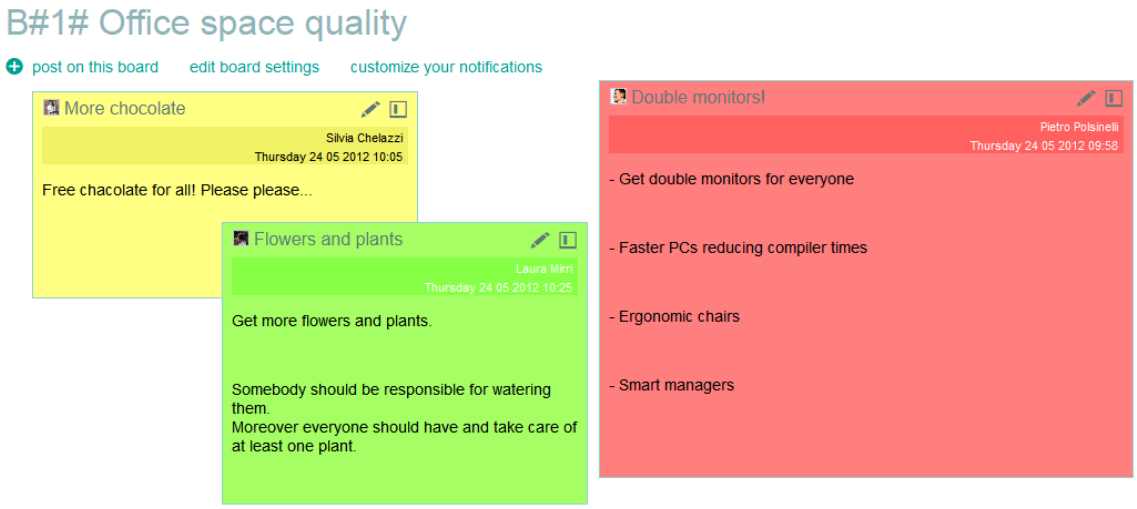
Two functions for the work group related to resource management will be discussed in detail later, load and plan (see section "6 Work Planning").

5.3 Boards

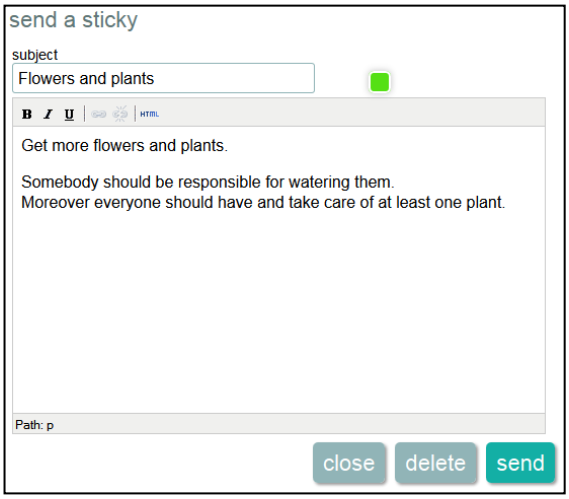
Physical boards are a common way to share information, say for collecting topics for a meeting. In Teamwork there are “digital analogues” of physical boards.

A board is an “open space” where everyone (almost everyone, security is always on background) can stick a message.

Once a board has been set-up and used for a while, it will probably look like:



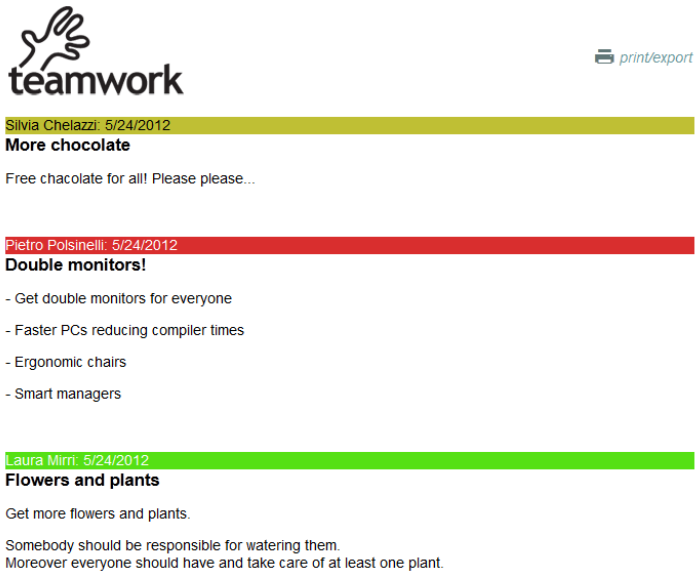
You can edit every message by clicking on button on each message.



You can edit the board data like name, description, status or subscriptions by clicking the links:

edit board settings customize your notifications

You can print the list of message:



5.4 Manage by e-mail

Not everybody can (or want) to use Teamwork all the time. Users (bad ones ☺) may prefer to perform some operations using their preferred e-mail client.

First of all notice that the messaging system can send notifications via e-mail. This is a basic feature, when you subscribe an event just check the e-mail channel.

This kind of communication is from Teamwork to the user, but also messages from the user to Teamwork are supported: Teamwork has a rich set of features to facilitate e-mail interaction with it, which go beyond the (quite powerful by itself) subscription/notification engine of tasks and issues; here are the possible actions:

1) Adding document to task
2) Adding issues to task
3) Adding to-dos
4) Sending messages
5) Synchronizing agenda (for detail see chapter “8.3.1 iCalendar (Outlook/iCal)”

There are software packages that do just this; Teamwork does this in context, automatically integrating security checks and project links.

There are some simple actions that can be done in Teamwork simply by sending e-mails to Teamwork. This can be useful for example in cases where the web interface is not accessible, or when sending e-mail to someone and also in copy to Teamwork.

In order for this functionality to work, configuration of e-mail from users to Teamwork is necessary (see “16.3.2 Configuration of e-mail from users to Teamwork”). For the moment it is sufficient to be aware that there will be an e-mail address whose account will be checked by Teamwork; writing to this address is for us “writing to Teamwork”.

Teamwork in receiving e-mails will check that the sender has the right to do the action intended: there must be one and only one resource having as e-mail the one which you are using to send the message. This way Teamwork will recover the resource and check security rights.

The possible actions are now detailed.

5.4.1 Adding documents to tasks

You can send an e-mail to Teamwork with documents attached, and all the attachments will be added as documents on the task intended. In this the task on which you intend to operate via e-mail is identified in the following way:

the subject of the e-mail you are sending is parsed, and if it starts with TASK, what is between # # is considered, say it is A342; first a task with code A342 is searched; then, if the value is numeric, say 342, a task with database id 342 is searched; lastly, a task named A342 is searched.

Some examples:

TASK #IG-JUL09# ATTACHMENT
task #2646# ATTACHMENT
TASK #International Geographic - July 2009# attachment

are all valid e-mail subjects by using code, id and name.

Task’s code and name are not mandatorily unique when saved, so if you have homonymy

you will get back an error message by e-mail. Using the task's id is the only always safe way, not necessary the most comfortable.

If at least in one of the cases above the task is found, the documents are created (this is the action of this section).

5.4.2 Adding issues to tasks

Using a single e-mail address

You can send an e-mail to Teamwork where the text of the e-mail will be used as description of a newly created issue on the task intended. If there are up to two documents attached, these will be attached to the issue. E-mail priority is taken into account to set the priority of the issue.

Some examples:

TASK #IG-JUL09# issue
task #2646# ISSUE
TASK #International Geographic - July 2009# issue

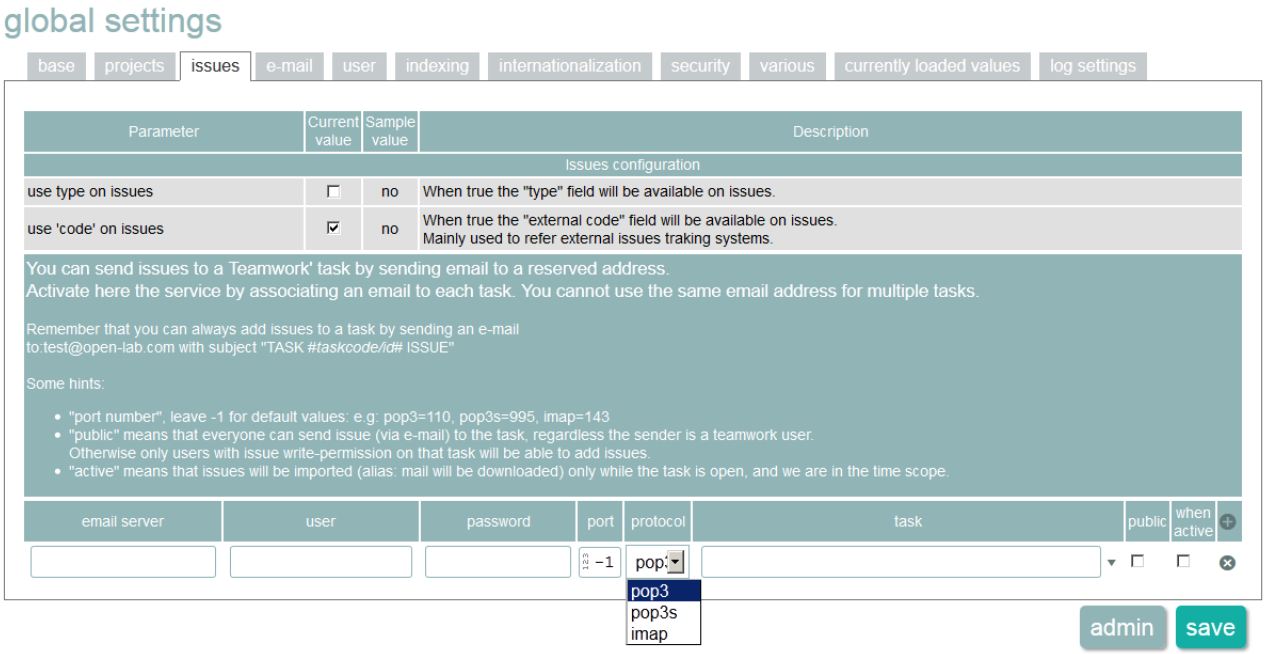
Teamwork will always check security, so from e-mail sender it will infer the user, then check on the task if such user has the permission to perform the action.

This could be a nice way to allow your customers to send issues/feedback on your project.

Using multiple incoming e-mail address

This feature considerably expands the management possibilities for anyone handling helpdesk like situations. You can now have multiple "background jobs" running in Teamwork that check several e-mail accounts (say, one per main project) and create issues for incoming e-mails.

To configure this you must access as administrator and go to *admin -> default for issues* (issues tab).



Activate here the service by associating an email to each task, and then adding a configuration line here. You cannot use the same email address for multiple tasks. The

priority of the issue created will be taken from the e-mail.

Some hints:

"port number", leave -1 for default values: e.g.: pop3=110, pop3s=995, imap=143

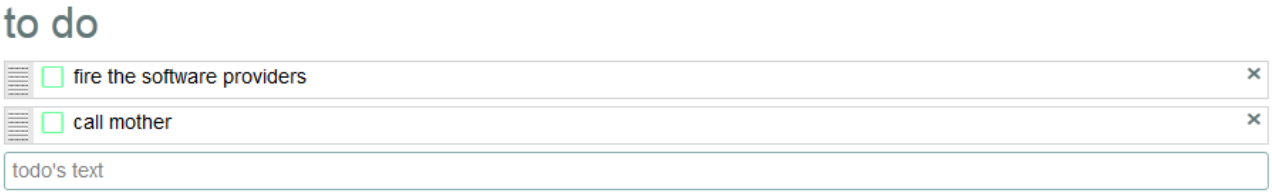
"public" means that everyone can send issue (via e-mail) to the task, regardless the sender is a teamwork user.

Otherwise only users with issue write-permission on that task will be able to add issues.

"active" means that issues will be imported (alias: mail will be downloaded) only while the task is open, and we are in the time scope of the task.

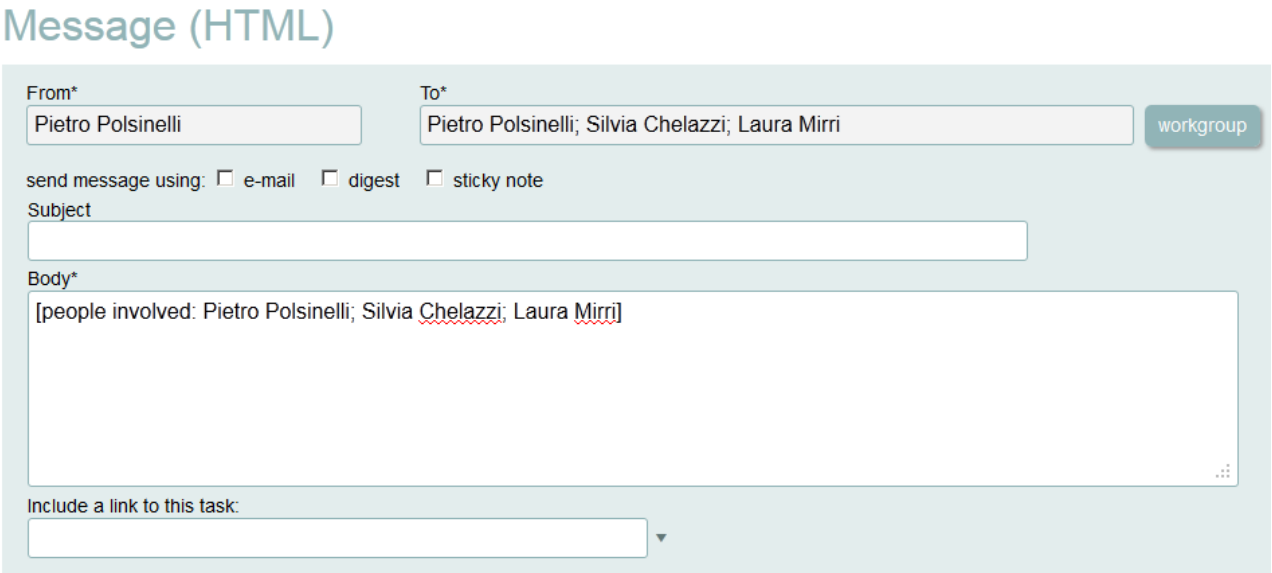
5.4.3 Creating TO-DOs

This supposes that you have added the “to do” web part to your home page (it is there by default after setup), in order to see the effects of your actions: to create to-dos by e-mail, just use a s subject “to-do”.



5.4.4 Send messages

In order for this functionality to work through e-mail, configuration of e-mail from Teamwork to users is necessary (see “16.3.1 Configuration of e-mail from Teamwork to users”). You access this section through *docs&tools -> boards & messages -> send message*.



This is a page quite self-explanatory.

Multiple e-mail, e.g. SMS forwarding

In order for this functionality to work, configuration of e-mail from Teamwork to users is necessary.

Multiple entries can be set in the resources editor; so for example, a secondary SMS-forwarded notification e-mail can be set on the personal details data e-mail field by adding more personal details.

6. Work Planning

6.1 Introduction

Teamwork comes with a broad set of planning tools. You can choose to go more or less in deep in planning depending on your habits, you organization, your team.

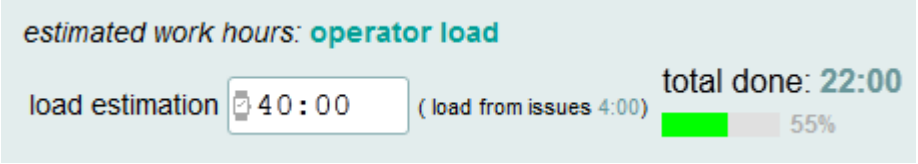
The first level of operator activity and load is the flat list of assignments for each resource. From the resource editor you get a list of assignment for open tasks:

Of course this may be too simple for your needs; let's examine some useful tools.

6.2 Work load

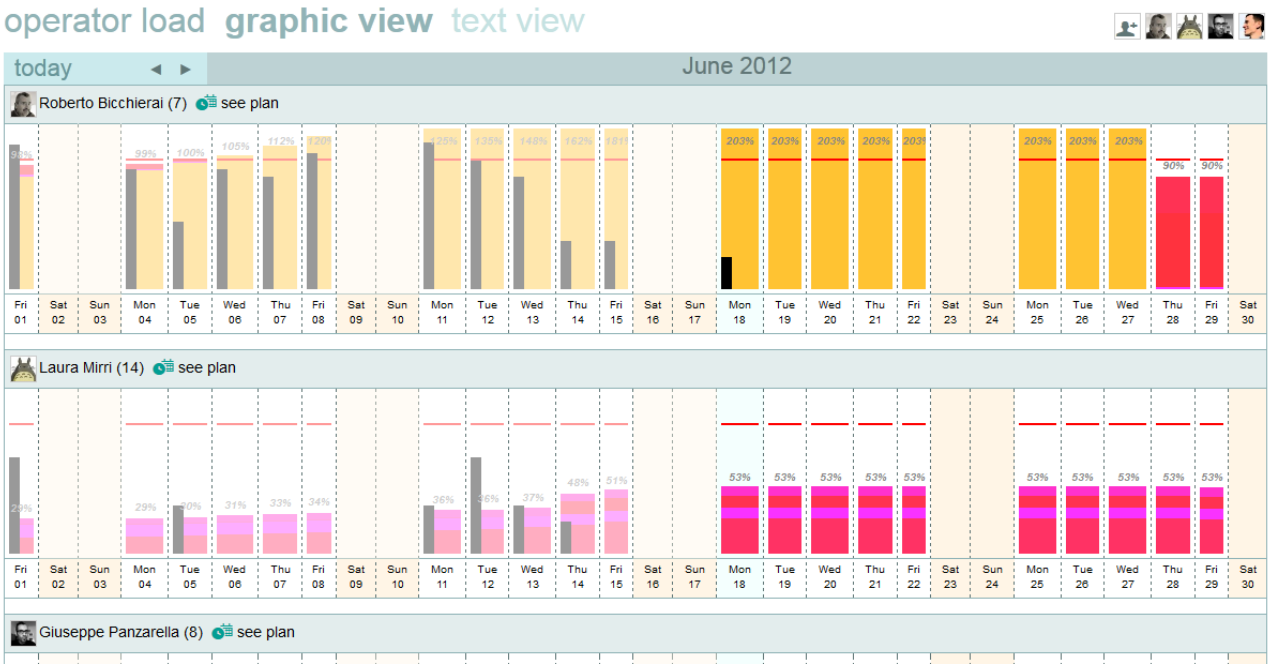
How to know what is the load of a resource? First, let's ask *how did you set the load*?

You can set it on assignments – that's already quite fine grained. If you are breaking down the task with issues, estimating by quantifying each issue is a very good idea. You're likelihood of getting close to reality is higher, and it will also be easier to manage the real-world evolving situation. This because people tend to be late – and every single issue too, so you will be able to fine tune during production – and not getting a bad surprise at the end: little delays sum up, and can generate a big problem.



A schedule, if set as an agreed result, is a way to get people involved and committed: it is a public commitment. So it is useful even if it turns out to be inaccurate.

Anyway, if you've set the load, from the timesheet or workgroup submenu click on "operator load", you'll get:



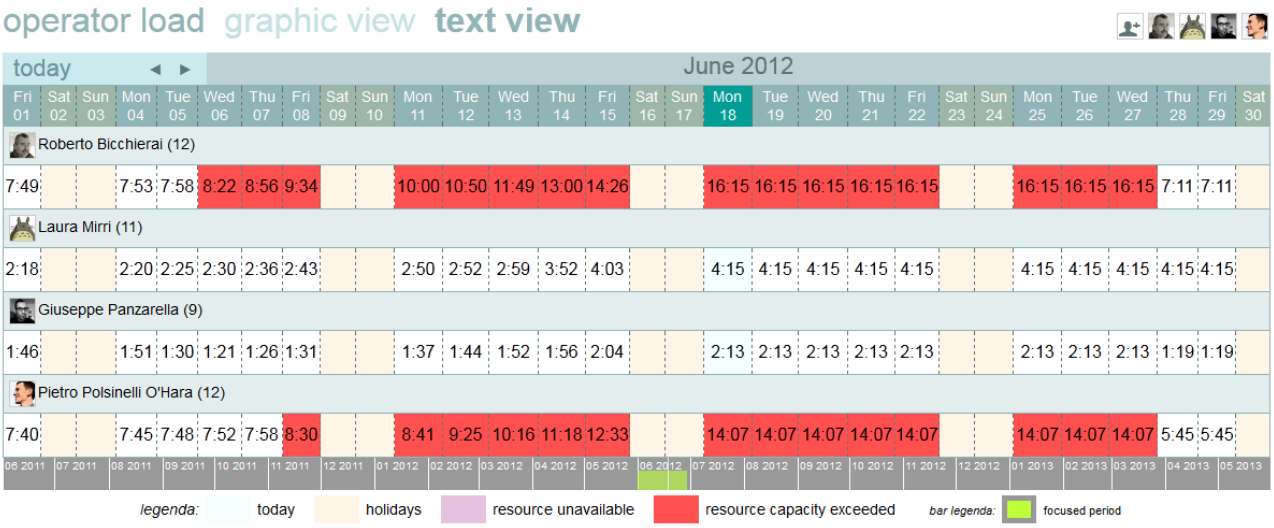
In this page for every resource in the selected workgroup you will get a graphical representation of the load total by day. By clicking on a column you will get a detailed explanation of the components of the load.

load detail

Assignments of Roberto Bicchierai on Friday, June 15, 2012

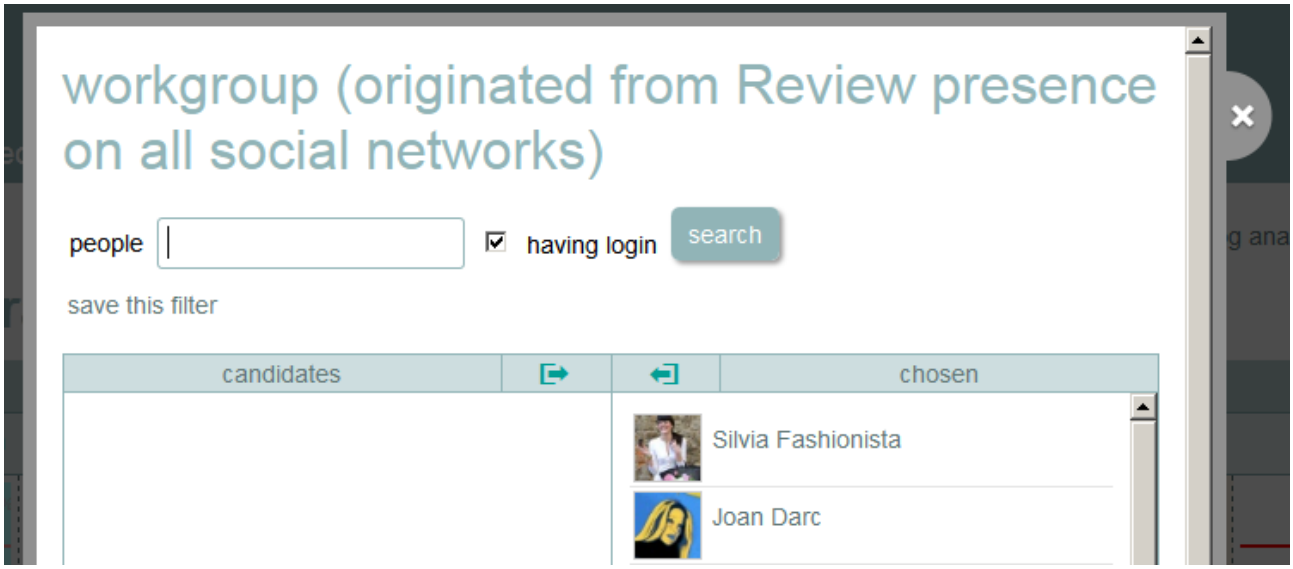
AVA - Avagliano API first part	14:26
TW.5 - Teamwork 5 development	0:00
TW.4.7 - Teamwork 4.7	0:00
TW.5.01 - MagicDiary	0:00
LAMMA - Lamma meteo mobile	0:00
total	**14:26**

If you prefer a text-based representation, click on “text view”:



This is a monthly view, by clicking on a day you will get the detail for the day.

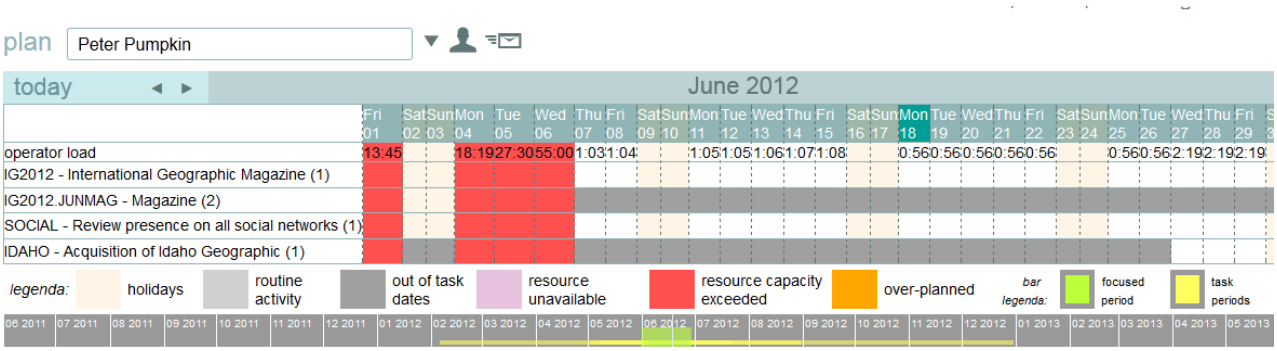
Go back to the graphical view, and notice that you can move in time by clicking the bottom bar or add someone to you workgroup by clicking workgroup.



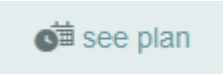
6.3 Resource work plan

We have already seen in previous sections that you can estimate how much work is necessary to complete a task or close an issue. These estimations are taken care of in evaluating the load.

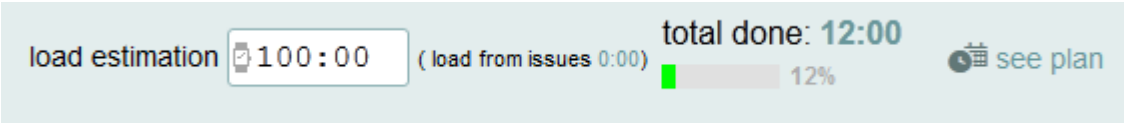
Teamwork supports a third way to track load: the plan, by task and by resource.

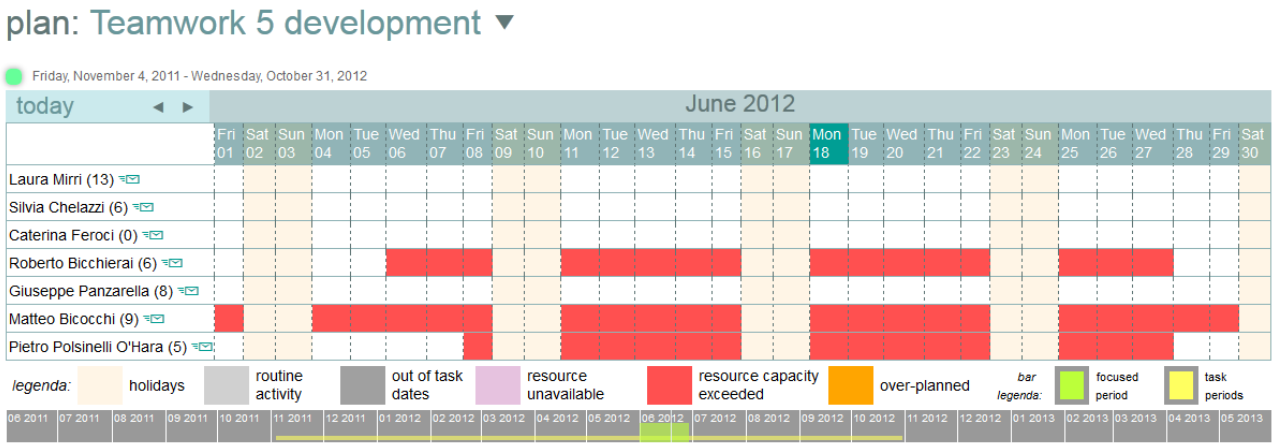


Every assignment can have a plan of activity, in terms of working hours per day. This is the finest level of detail with which to plan the activities of resources.

You can access the plan for a task from the "general" view" 

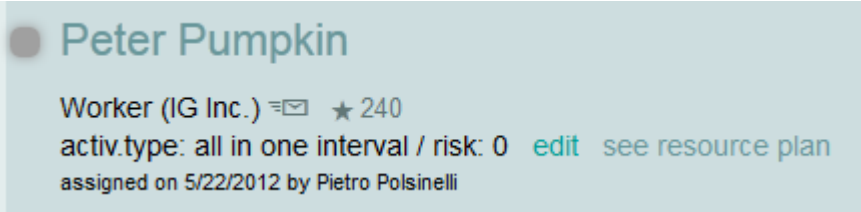
, from the assignment editor



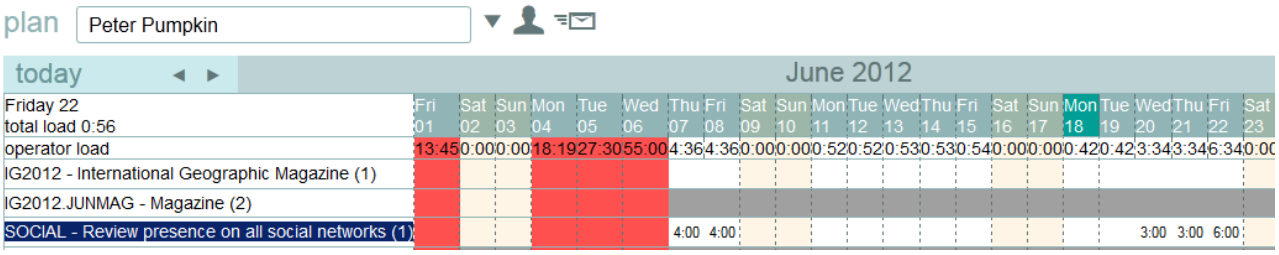


Here you can insert planned hours per day for each assignment.

The plan we've seen up to now is by project / task. Planned activities can be seen by resource, you access it both from the assignment and from the resource editor:



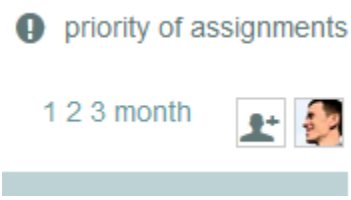
Getting



Red cells mean that there is an overflow for that day: there is a legenda on the bottom. Pink cells represent days of unavailability for that resource.

Every resource has a default working time per day (see “2.3 Teamwork’ resources with login”).

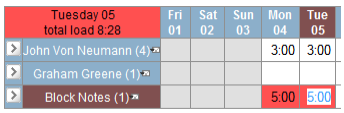
You can move in time by using the bottom bar o through the tree structure from the navigation bar.



Since 5.2 you can also see the plan for several hand-picked resources, through the usual resource picker .

Example.

Only 5.00 hours of planned work to go overflow, why that as I have set 8 working hours per day? Well, 5.00 hours for *this task*, but probably the involved resources have some other load for some other task (you can explore that by simply clicking on the resource name on the left, see below). Click on a cell to see the total work amount:



in this case the total amount is 8:28.

Work estimation by assignment and/or issue is spread uniformly on the period of activity of the task, by counting working days only.

Example.

If you have assigned a resource for 60 hours on a task that opens Jan 1st and ends Jan 25th, for 15 working days (Saturdays, Sundays, new year eve, and eventually your company holidays are considered). Then on this task there are four issues for a total estimation of 20 days.

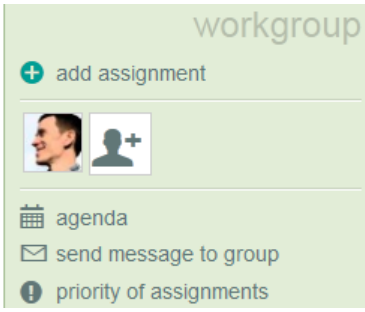
Question: how much is the estimation for each day? Upside-down solution: **(60+20)/15=5.33**

6.4 Priority

Another aspect of planning is to set the right priority to every assignment.

Teamwork stores a plan of priority changes. When you set a new priority on a assignment, priority is valid from that point in time onwards, until it is changed again.

You can change priority from the task editor on the assignments tab:



This sets a change priority point to today. If you want to planning on a wider period select from workgroup menu:

Here for each resource involved in the workgroup you will see the assignments active for the week. You can change priority, or remove change points.

Teamwork shows assignment priority wherever possible. In your assignment part in the dashboards for instance:

6.5 Operator load calculation

To describe clearly how Teamwork calculates the work load for a resource let's see a practical example:

Say we have two tasks:

- Project A lasting 5 days with a resource assigned (Resource R1) with an estimated worklog of 25 hours.

- Project B lasting 2 days with a resource assigned (Resource R1) with an estimated worklog of 12 hours.

Task in the future:

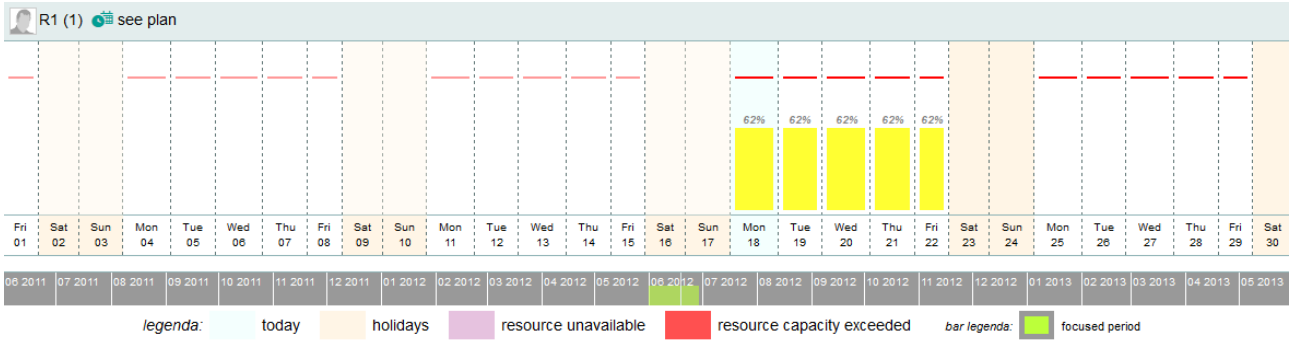
This is the simplest case:

- one single task (Project A)
- no unavailability events in the task period

According to this data the work load will be distributed equally in the five days (5 hours per day).

The percentage represents the estimated worklog for that day according to the working hour per day of the resource (represented by the red line).

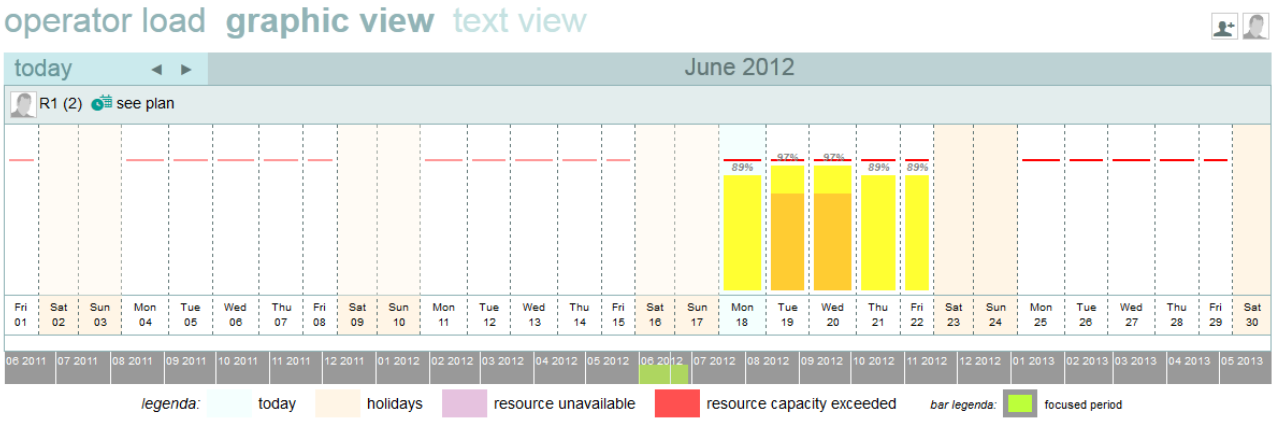
Here is the load for Project A:



Now we can consider a more complex case:

- 2 different tasks (Project A and Project B) having intersected period assigned to our R1

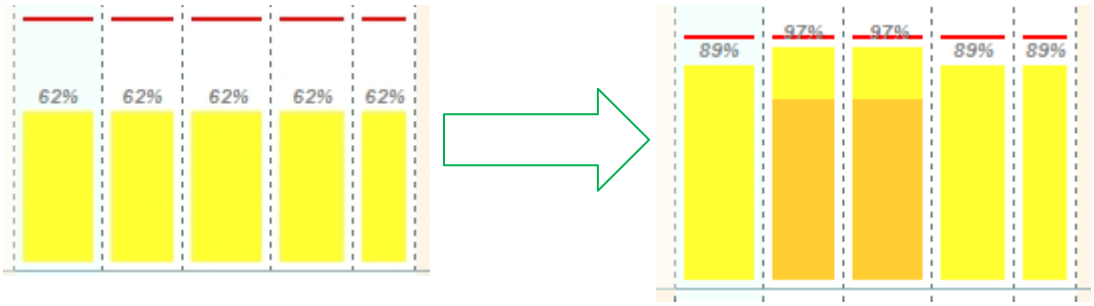
You see the load of these two tasks below.



As said before:

Project B lasts 2 days with an estimated worklog of 12 hours which should generate 6 hours per day of work.

Project A lasts 5 days with an estimated worklog of 25 hours which should generate 5 hours per day of work.



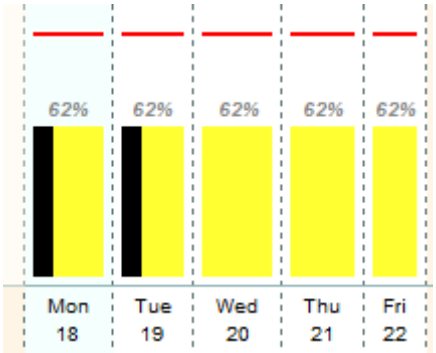
As you can see from the picture Teamwork does not simply sum the two estimations for each day but tries to distribute the work load so that the resource does not exceed his/her working hours per day, in this case by increasing load in the less booked days.

Task in the past:

The main difference looking at the operator load including task in the past is that in this case we have to consider also the work done and not only the estimated one.

The picture below shows again the load of Project A (moved to intersect today).

If we consider again a simple case (a single project without unavailability events) and we suppose that the resource has correctly inserted the worklog here is the result:

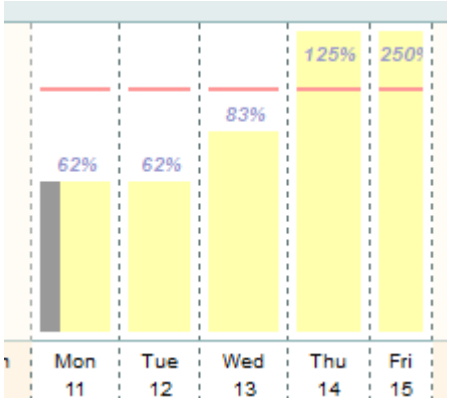


The dark bar represents the inserted worklog.

The total estimated worklog was 25 hours, we already did 10 hours (5 hours on Monday 13 and 5 hours on Tuesday 14) so we still have 15 remaining hours which have to be done in 3 days.

In this case the load still shows you five hours per day also for the future days.

Now we consider what would have happened if the resource had not recorded worklog for Tuesday.



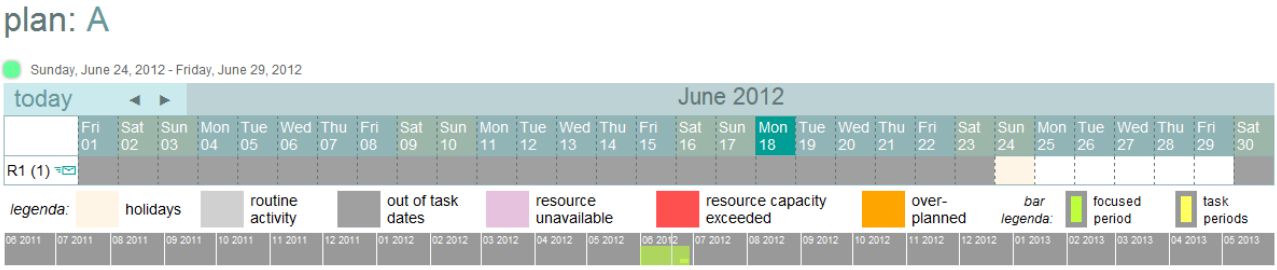
As the estimated worklog for Tuesday has not been done the load for the next days increased.

The total estimated work was 25 hours, we did 5 hours (instead of 10, there is not the grey bar on Tuesday) so we have 20 remaining hours of work which have to be done in 3 days, this is the reason way the load for the future day is increased to 83%.

If you use the plan to organize the estimated worklog we have to introduce an important consideration:

Let's use again the simple example of Project A (again it has been moved in the future):

Here is what you see when you enter in the plan for the resource R1.

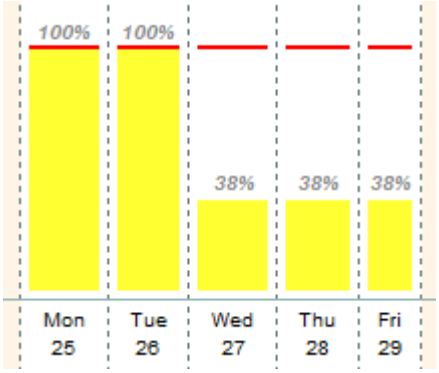


The first row represents the operator load that is there to help the user to correctly insert the plan.

In this case we are looking at a future task so the estimated work is distributed equally through the task days.

Now we supposed that we want to work on project A for 8 hours on Monday and Tuesday:

Here is the result:



The pictures show that according to the plan the operator load is changed; in fact we have a load of 8 hours for the first two days and the remaining hours are distributed on the 3 days left.

What we hope to have shown is that plan and load interact and get filled up from every possible source in real time – which is Teamwork's focus.

6.6 Handling future projects by resources as skills

Consider now this use case:

- handling projects open now with real people

- modeling future projects with assignments by skills

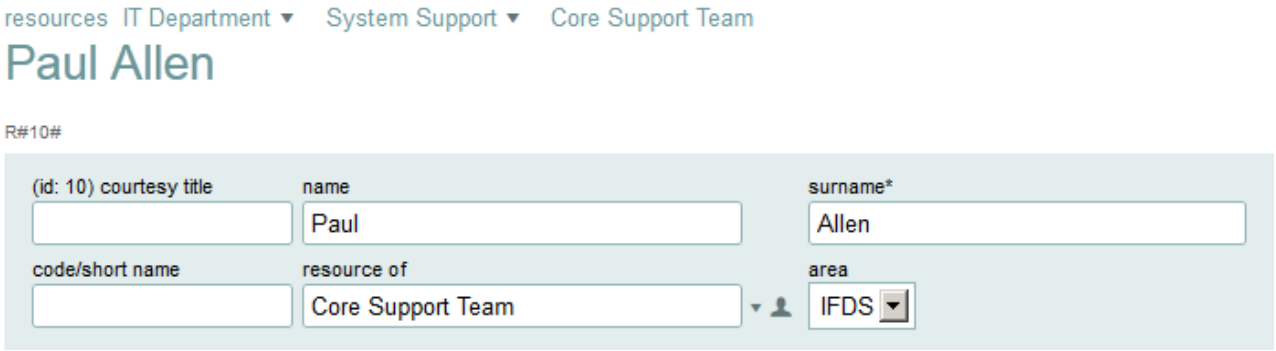
Having both in the same system makes it possible for example to share project tree creation and maintenance between planning project managers and production project managers.

We assume below that we start from a future project.

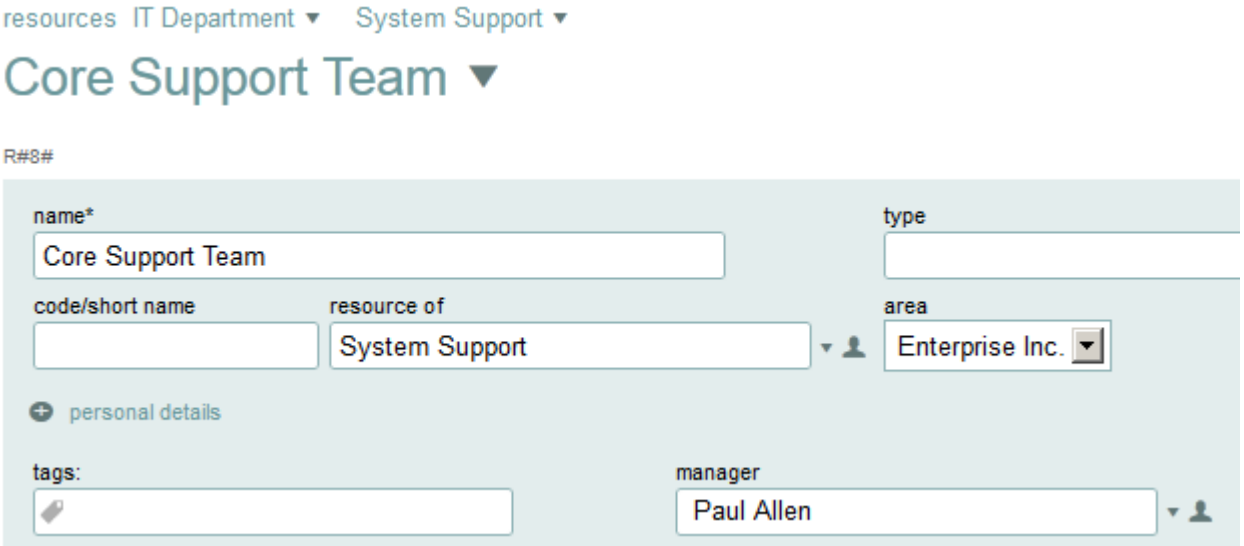
6.6.1 Resource hierarchy

We created departments and teams, with people in teams. Each department team has a manager.

Suppose that Paul Allen is the manager of the Core Support Team; after creating company, department, teams, we create Paul Allen:

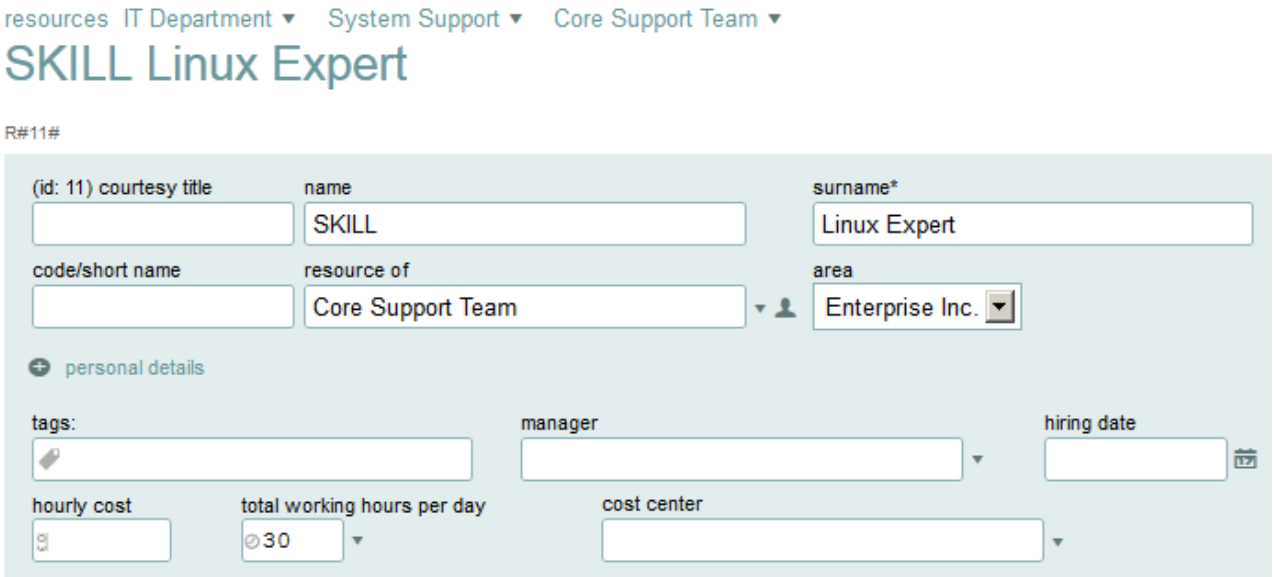


Now we have to set Paul Allen also as manager of the Core Support Team:



Then for each team we need the members of the team (obviously) but also a (or also several) placeholder resource which will play the role of team skill(s).

Example:

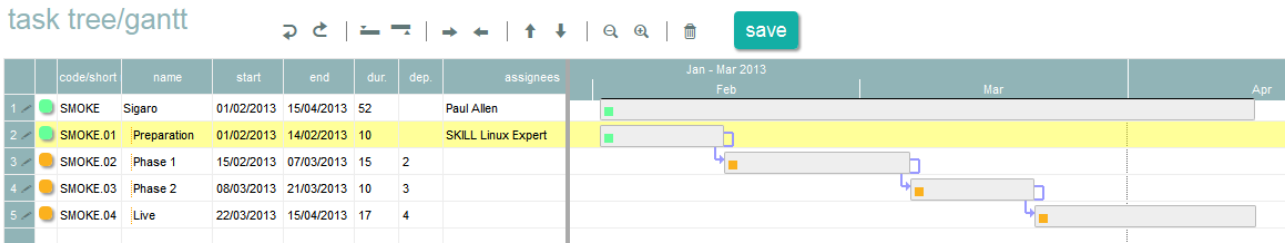


This resource should not have a login, and should be put in the team where it belongs to.

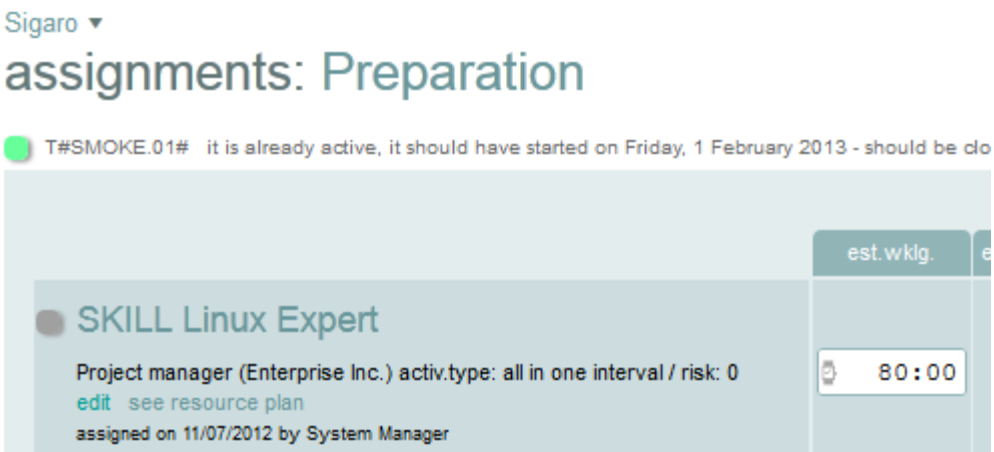
The crucial point is the resource capacity – working hours per day, which actually is the potential output of the team for such skill. In this case we have 6 Linux experts in the team, who can dedicate 5 hours out of 8 (3 are taken by say "Business As Usual"), so the potential output is 6*5 hours of "Linux Expertise" per day. Notice that this holds only for planning tasks, as in the day to day operations load is not given by such simple availability but you have to use the operator load where are taken into account open assignments but also worklog inserted which progressively frees the team members from the assigned hours.

6.6.2 Future project

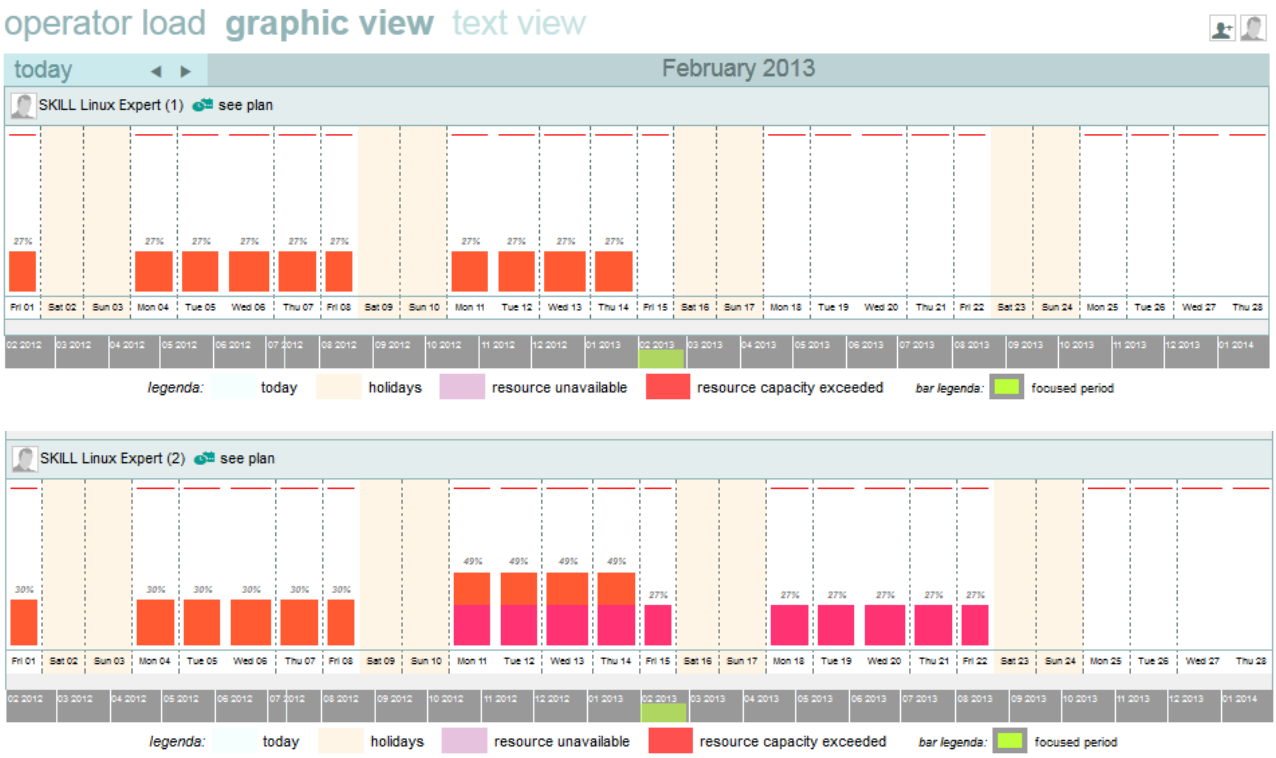
So we suppose to have a future project called "Sigaro", the idea being that at this moment we know skills required and can only predict effort. We surely now know the core structure, and we even have some milestones. All this can be set in Teamwork.



Created the Gantt/tree structure, we assign the "Linux Expert" skill with an estimated effort of 80 hours:



The operator load function is now already effective and useful, and can handle a skill “booked” in several future projects, giving you a report of the state of commitment:



More future projects you will have, more useful it will become.

6.6.3 Running project

What happens when the time comes that the project has to be started? First generate a “snapshot” of the project, which will be the project as planned.

You will add assignments for real people: you will assign a project manager, resources with the skill’ skills (sorry ☺), and the skill’ assignment can remain, as it does not record workload and worklog accumulation is not done for resources without a login.

Notice that the worklogs you will be inserting will subtract from the operator load.

The last point is important because this way you can check in the same operator load page the load of both “real” operators and of skills. As you have in the team both the skills and the real operators, by clicking “operator load” on the team you will see a load / availability overview.

That’s it.

7. Work time tracking

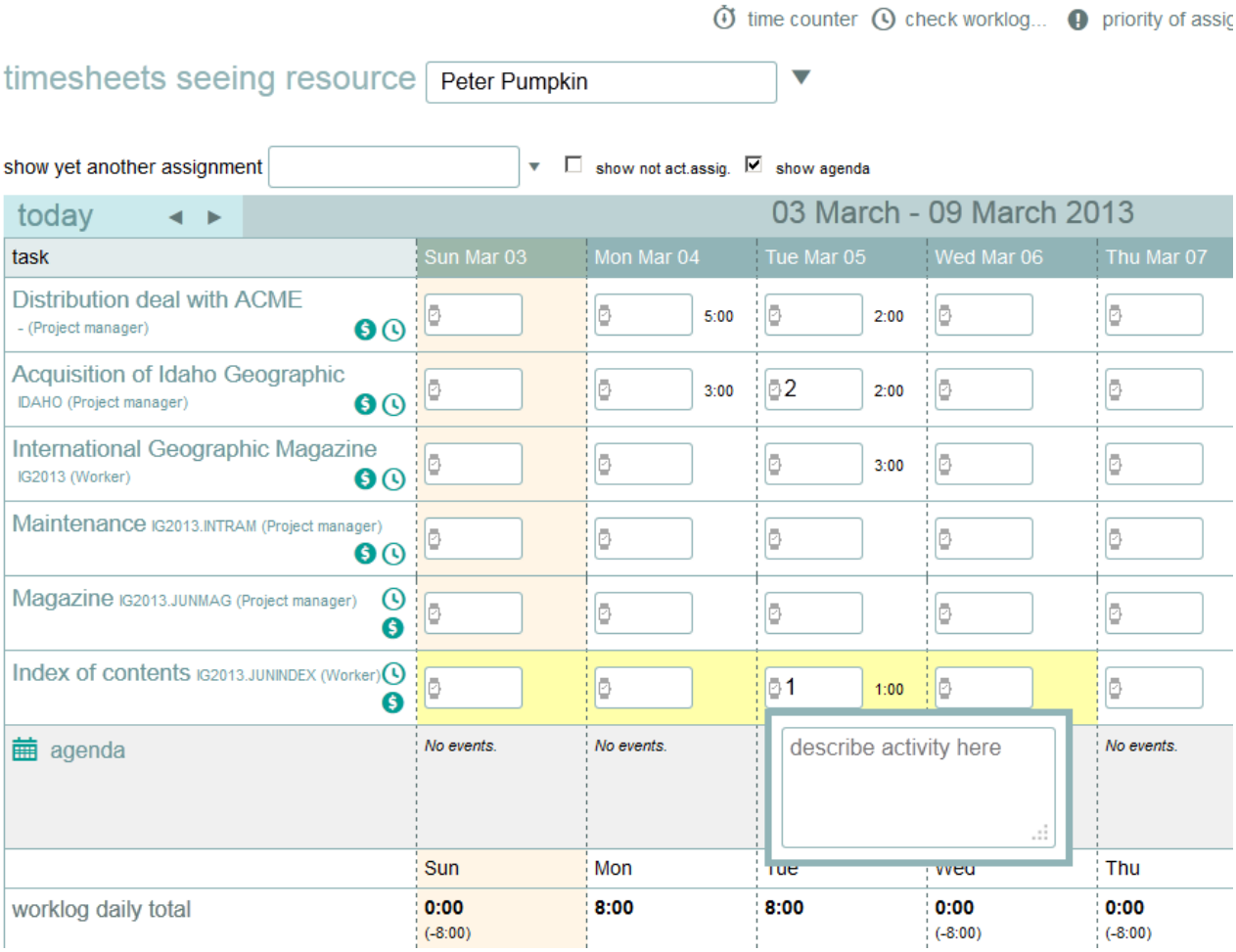
7.1 Introduction

Working time tracking is a critical aspect of work management. Only when PMs have complete and accurate time reports, they can evaluate and manage projects. As it can be quite boring for users to insert work logs, we did a lot of research in trying to make worklog insertion as fast and friendly as possible; we are aware that when users feel that an activity is “heavy” they will simply skip it or, even worst, insert random data. So Teamwork provides many different ways to collect work logs.

7.2 Editing

7.2.1 Weekly time sheet

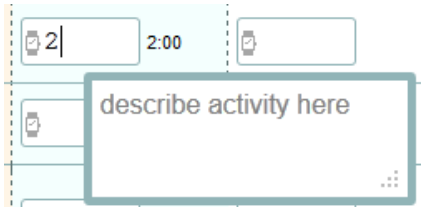
From the menu bar click on timesheets:



This page sums up all you work log recordings.

In order to record work just insert the duration (in hours:minutes) in a cell.

If you want to insert a description after inserting duration, press tab (a field will be opened) and type the description.



Sums are computed on-the-fly by row and by column. Yellow cells means that such column’s data is out of the task time-scope.

Filters: you can see that you may show cells for not active assignments and visualize a row for agenda events.

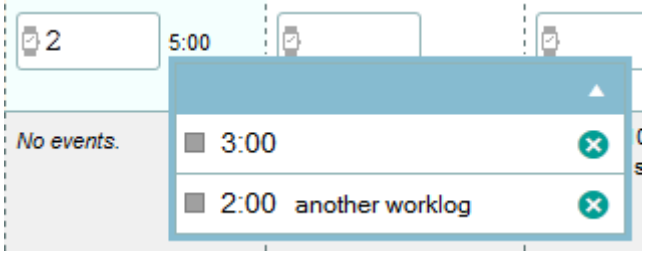
With “show yet another assignment” you can add rows for assignments whose task state is not active and/or date interval does not intersect with the week visualized.

Is it correct to insert worklog on these invalid situations? Well we think that is better to allow users to insert worklog wherever they think it is correct. Probably they really work on these tasks out of scope; what will happen if we block them (as instead happens in all traditional PM software)? Probably users will insert worklog somewhere else just for the sake of inserting 8 hours per day.

Actually any “non-standard” worklog insertion will generate an event that can be subscribed.

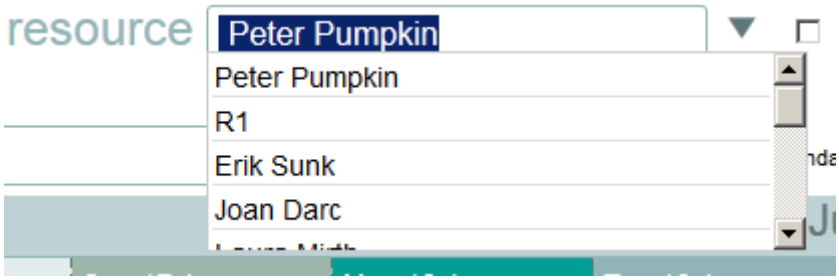
The lower part of the shows you agenda appointment. This could be very useful when you are inserting work log after some weeks, filling “holes” (ok, this never happened to me, but….).

You can insert more than one record per day per task, say if you return here several times a day, maybe for different reasons.



By clicking on the total you will open the detail. You can remove a work record or modify it by clicking on a row.

If you have permission to do it, you can also watch/insert work logs for somebody else: just select it from the combo in the upper part:

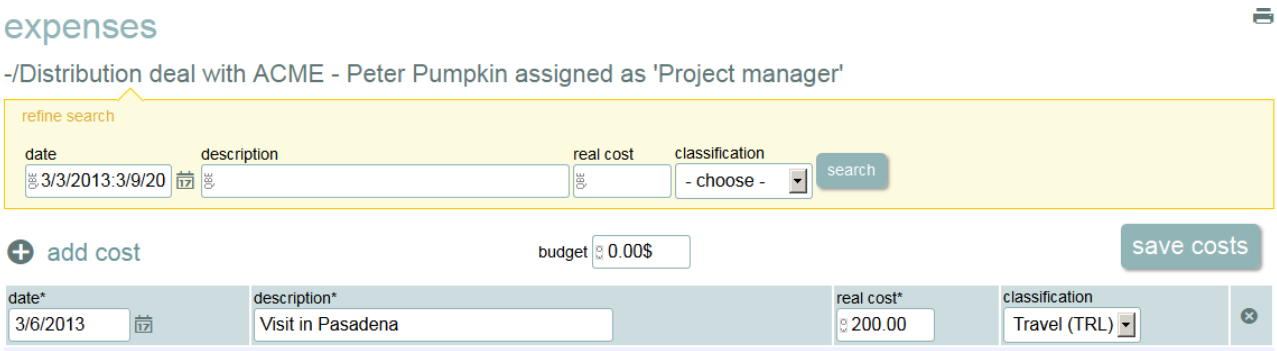


You can move through time using the bar at the bottom.

By clicking on in the first column you will get the list of work logs for that assignment:

You can change/remove/modify each record.

On the default configuration since version 5.2, you get also a further useful link , to assignment-specific expenses:



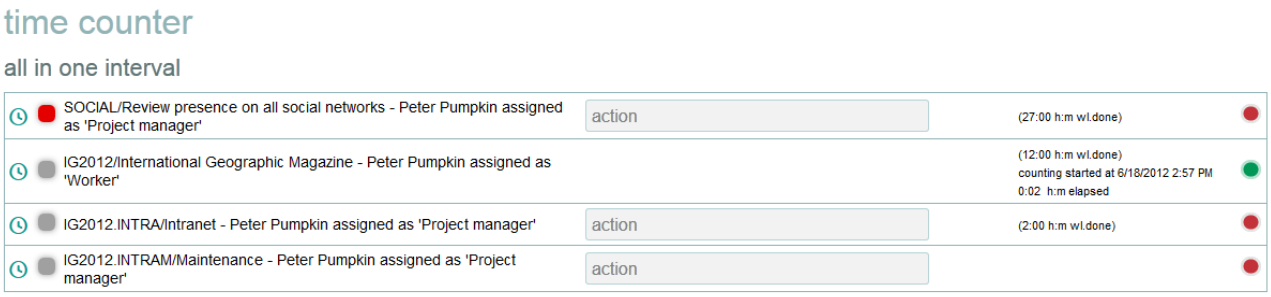
Do note that here by default expenses are filtered by the week in focus (differently from the assignment editor), which is very handy.

In order to enable / disable such link, go to Admin -> Extra features -> CUSTOM_FEATURE_SHOW_EXP_ON_TIMESHEET, set the value to yes or no.

7.2.2 Time counters

Counters are intended to be used by those that usually work most part of their time on a project.

There are two tools for using counters: a page and a portlet.

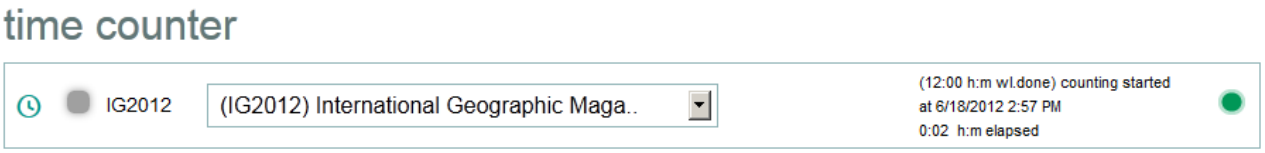


Both work similarly. Insert in the "action" the description of what you are about to begin, then start the counter by pressing the red dot.

When you stop recording or you start another counter a work log is saved.

If you forget to turn off a counter Teamwork will do it for you after recording more than the daily work hours from your options.

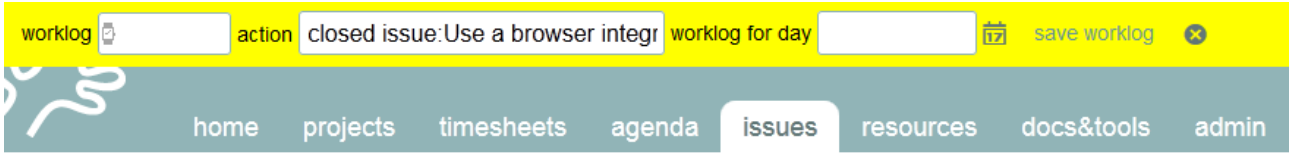
The slim portlet looks like:



And you can add it to your home page.

7.2.3 Issue driven

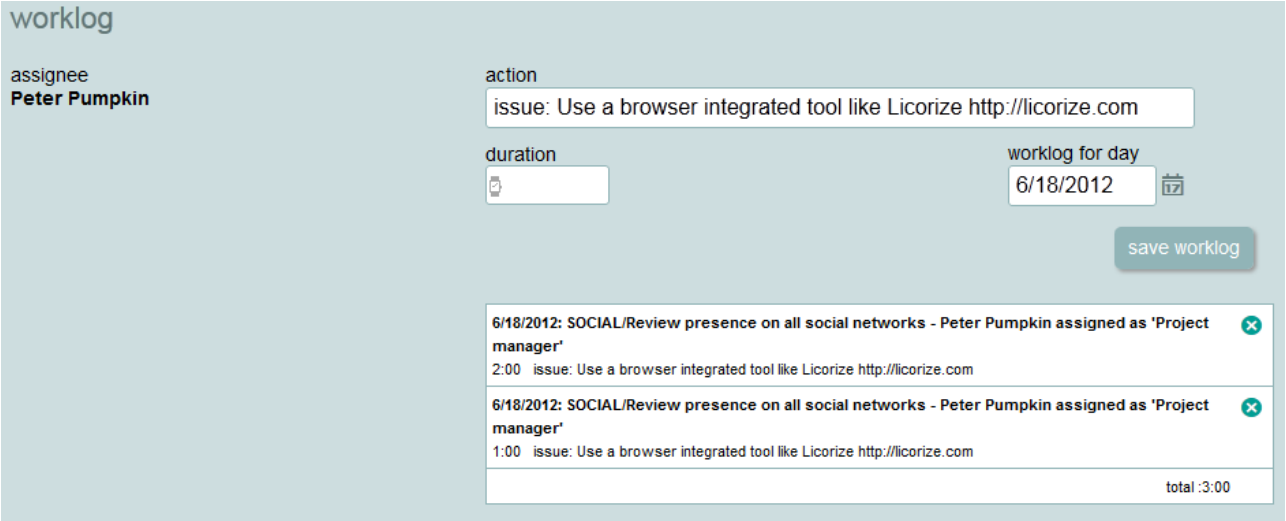
If you mainly work on issues probably the most friendly way to insert worklog is when you are closing an issue.



When in the “issues” page you change an issue status to “closed”, a yellow bar on top will appear and allow inserting worklog; actually, you can do that even for issues that are not closed, just click on the watch icon. Of course this happens only when the issue is yours, and is related to a task where you are assigned.

7.2.4 From the issue editor

You can insert worklog also in the full issue editor:



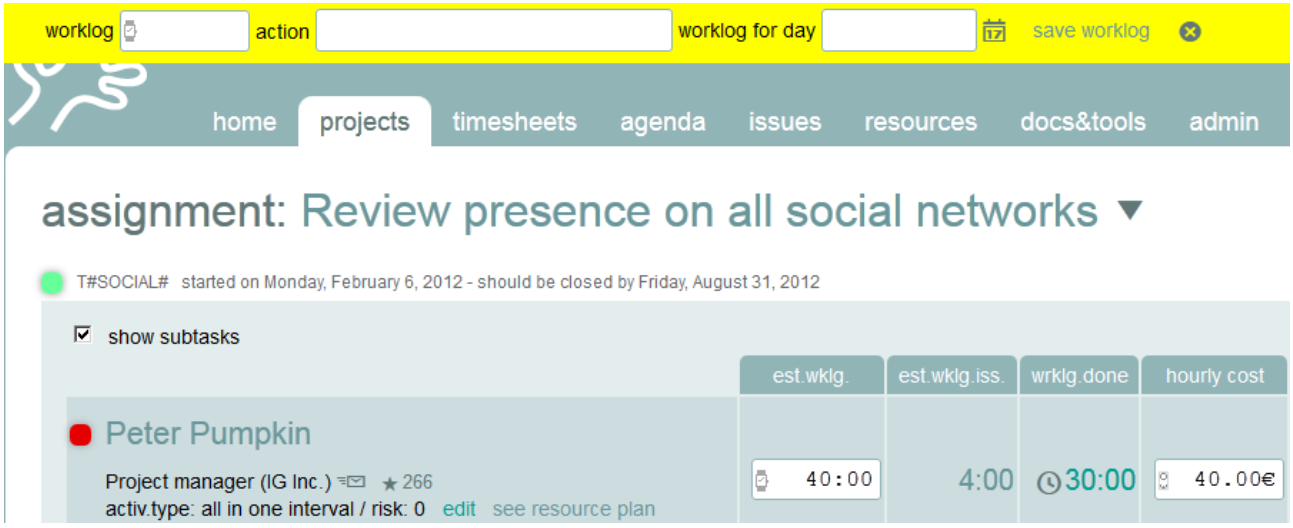
By clicking the “close issue” button you will be asked to insert worklog.

The action description is prefilled, and you have just to insert the duration.

If an issue requires more than a work log, you can add here as many you need.

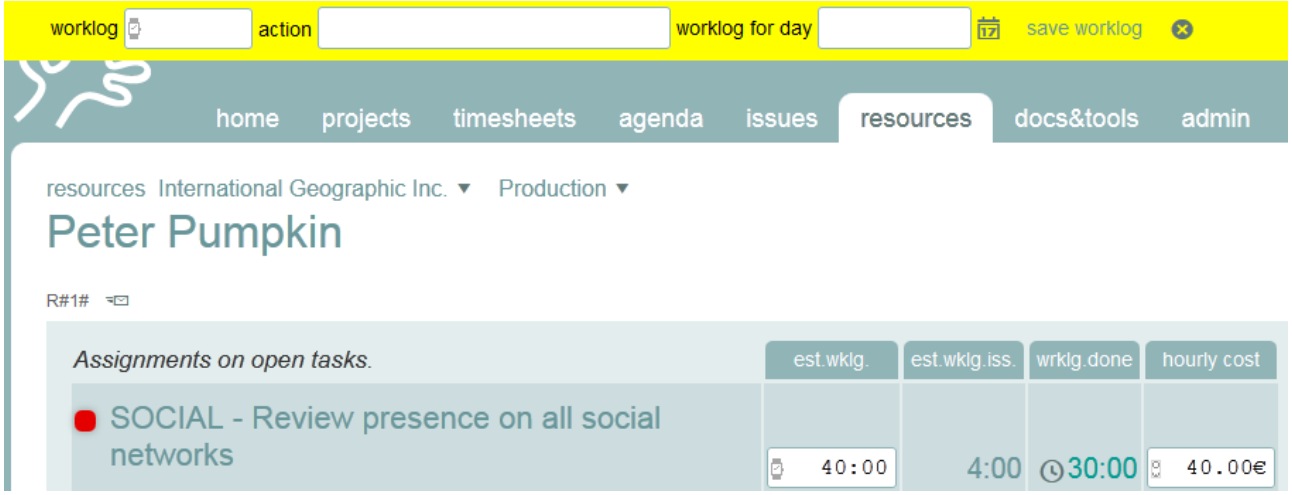
7.2.5 From the task editor

If you are assigned on a task, from the assignment list just click on the watch:



7.2.6 From the resource editor

From your own editor, go to the assignments, there you’ll be able to insert worklogs:

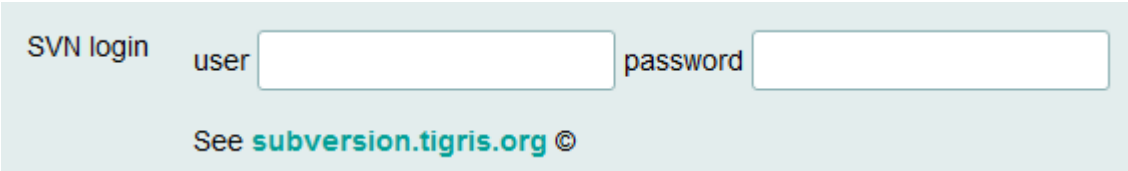


7.2.7 Subversion

Teamwork can import worklogs from Subversion (SVN, http://subversion.tigris.org) commits.

This functionality could easily be extended; Teamwork connects to various servers and recovers traces of your work. Then you can choose those that are significant, specify the time you spent on doing them, choose the project linked to the activities and start importing.

In order to use these features you must configure the relative accounts on the user's option page:



In order to start importing logs from SVN press "import SVN logs"

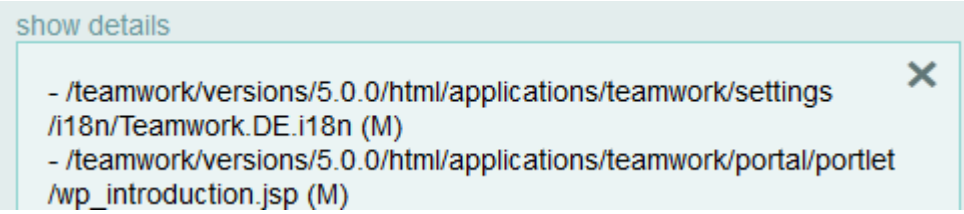
from timesheets.

Fill the form with SVN account data and specify a date to narrow the search scope.

Then press "find logs" button (it may take a while):

Then you can import every single log by specifying time, choosing the task related to the log and press the “go” button, or faster, by checking multiple lines and importing logs to a single task.

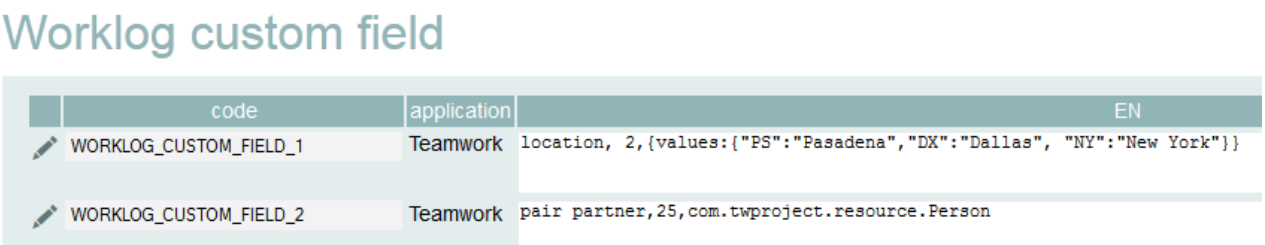
You can see details of changes by clicking on “show details” link



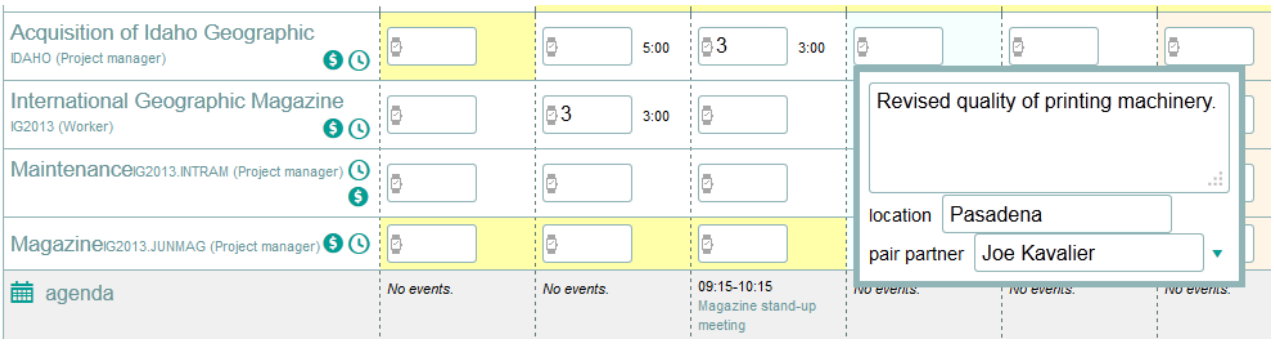
7.2.8 Worklog custom fields

One of the coolest features of version 5.2 is adding custom fields on worklogs: say you want to make it possible at worklog insertion to insert additional info, e.g. location and “pair partner” (say that your quality policy requires work in pairs). Just define the two custom fields in the admin section, and you’ll get an expanded worklog insertion form in the worklog week page. Sample values inserted are:

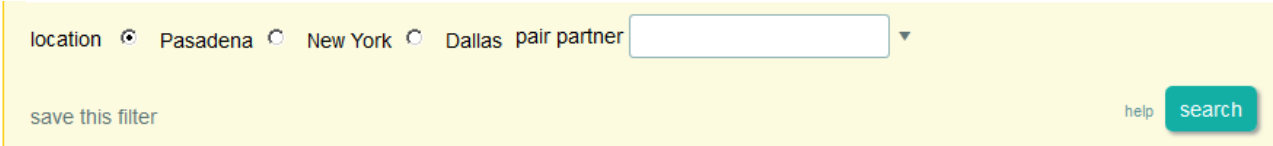
```
location,2,{values:{"PS":"Pasadena","NY":"New York","DX":"Dallas"}}
pair partner,25,com.twproject.resource.Person
```



This means that in worklog week when inserting worklog you will contextually get additional fields:



It will be possible to filter worklogs using such fields:



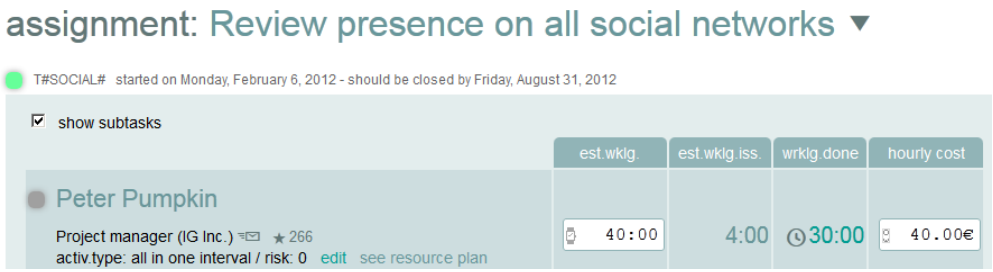
And you get them also in print:

id	date	worklog status	assignee	action
IG2013 - International Geographic Magazine()				
63	Thursday, February 28, 2013		Peter Pumpkin	Carefully inspected local wine bars. location **Pasadena** pair partner **Joe Kavalier**

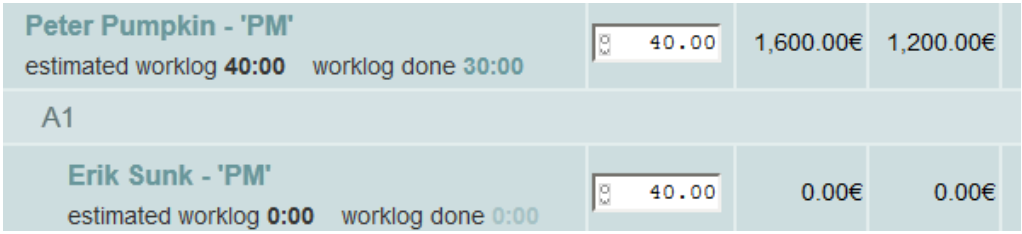
7.3 Monitoring worklog

Inserted work logs are really relevant for monitoring projects, so they are visible from several parts of Teamwork:

1) In the task editor's assignments view:



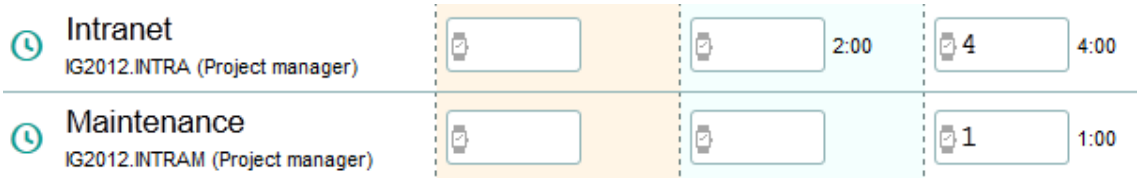
2) In task cost analysis:



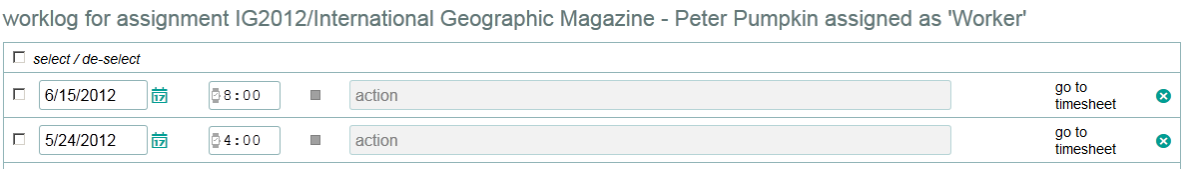
3) In "my assignment" home portlet:

4) In resource editor "assignments"

5) In timesheet week:

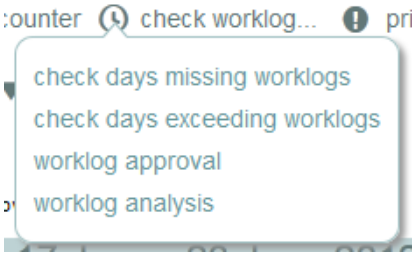


6) In “worklog for assignment”, in detail:



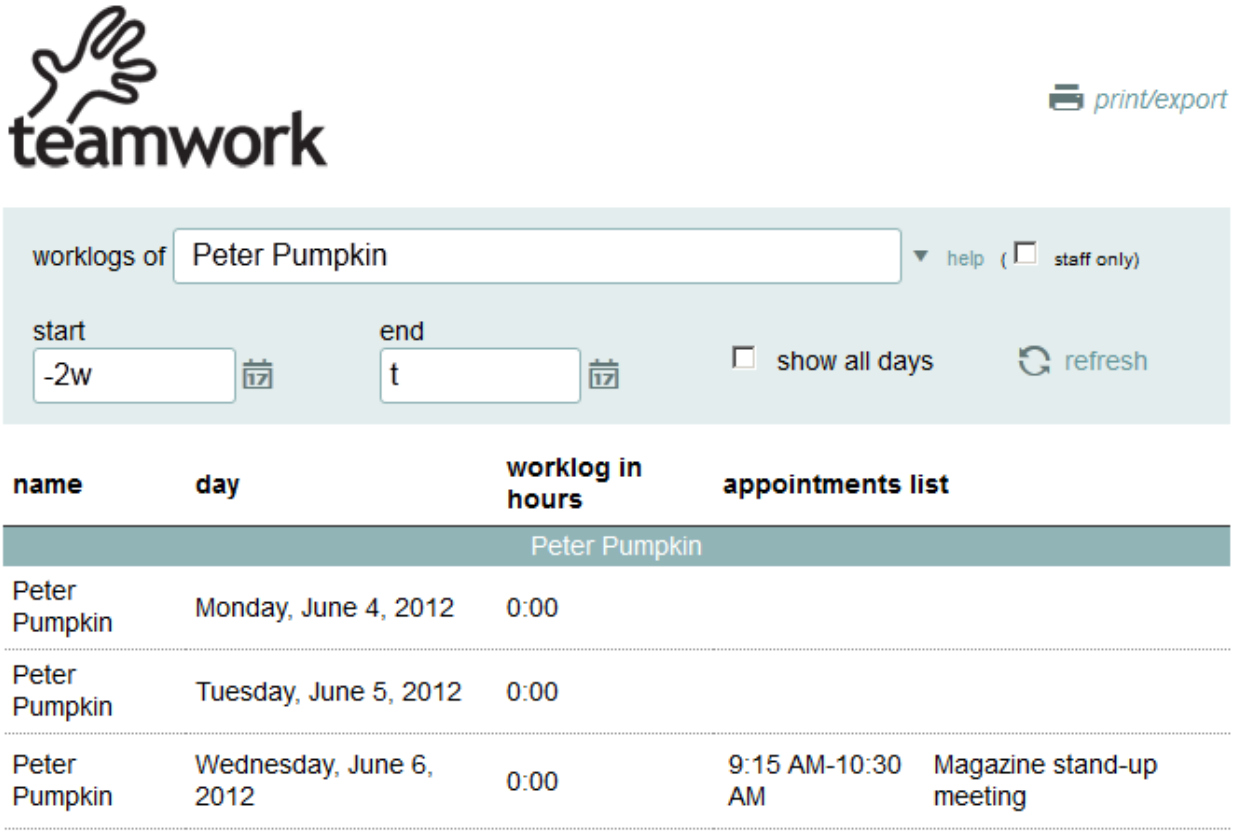
There are two other useful tools for checking work logs next sections.

7.3.1 Check missing worklog



This tool is designed to find days where work log is absent or below the total per day.

From timesheet week page press “check days missing worklog”:



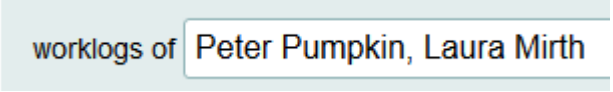
Fill the start and the end fields and press “refresh”.

The tool lists days with insufficient or missing work logs and, on the side, the list of appointments for each day; this may be useful for “remember” why there is no work log on that day.

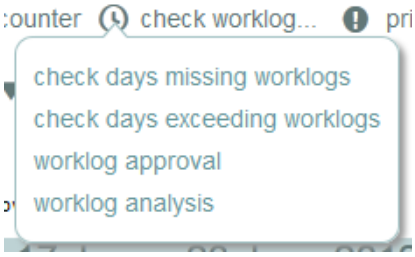
- Use date shortcuts as “lm” (for last month) or “t” (for today). See the complete list on section “13.2.2 Date shortcuts”)
- The tool shows working days only. You can configure “company’s holidays calendar” on the administration page, holidays link.

You can print/export the list, and go fill the missing data.

You can also check the worklogs in a group:

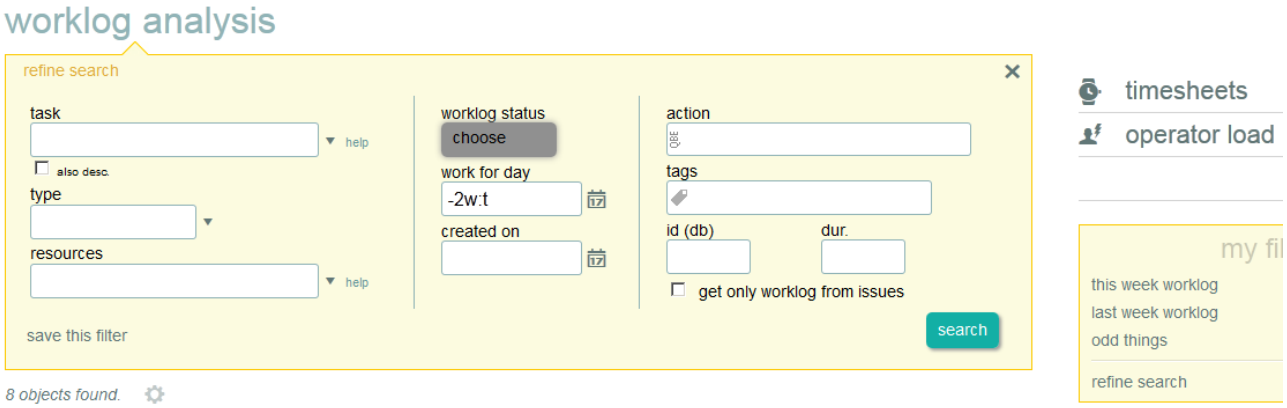


7.3.2 Worklog analysis

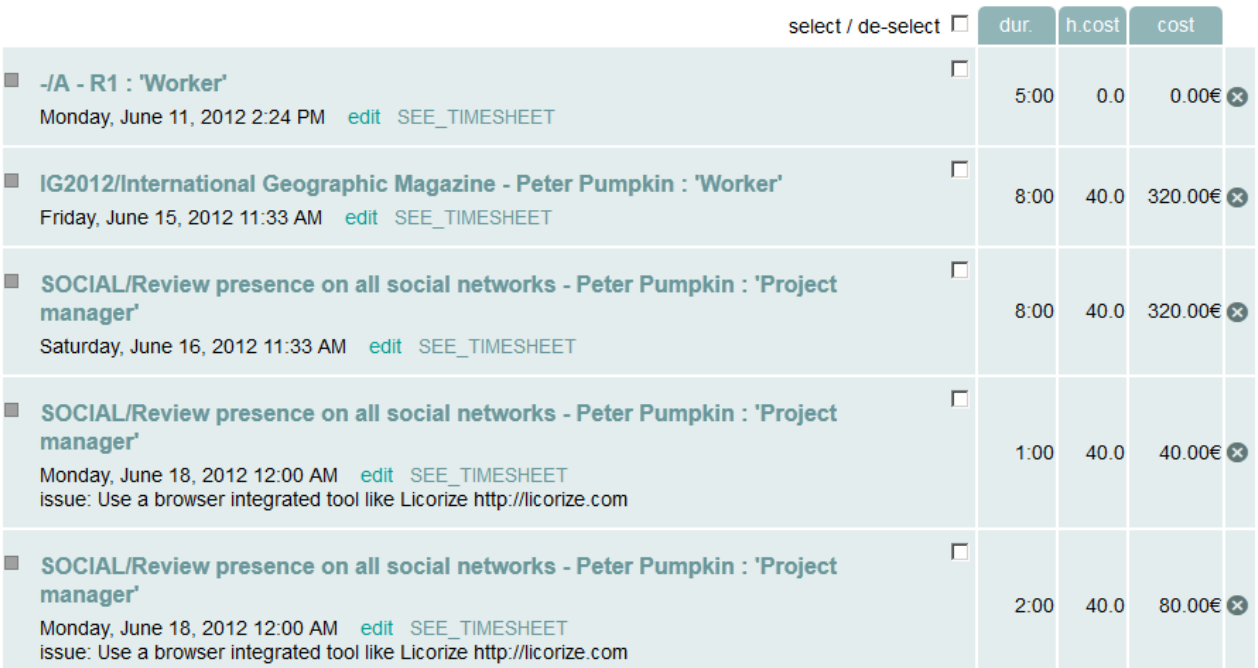


This is a tool designed for both analyzing and managing worklog.

From timesheet week page click on “worklog analysis”:



Compose a filter and press “search” button.



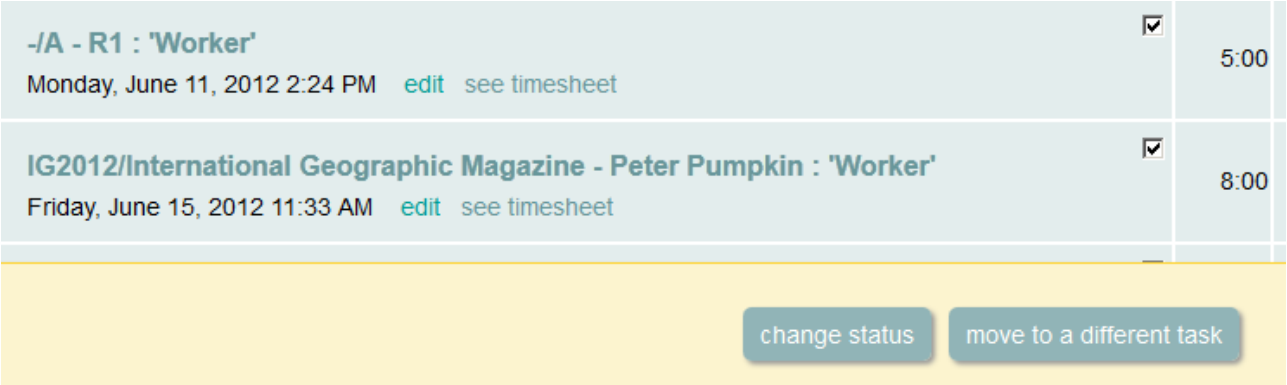
You will get the list of work log matching your filter. As usual you can store you preferred filters for future uses.

At the end of the page, totals are reported both in hours and as cost.

partial worklog (in this page) (hh:mm):	**31:00**	1,040.00€
totals from worklog (hh:mm):	**31:00**	**1,040.00€**
totals from worklog (dd-hh:mm):	**4d. 1:00**	

You can edit every line in order to fix timing, insertion dates, or descriptions.

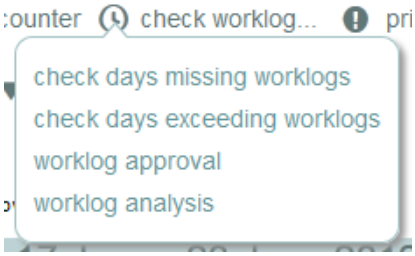
Filtering on this page can have two aims: getting worklog totals for a period, or moving worklog to another assignment/task, selecting it at the bottom of the page. As soon as you check a row:

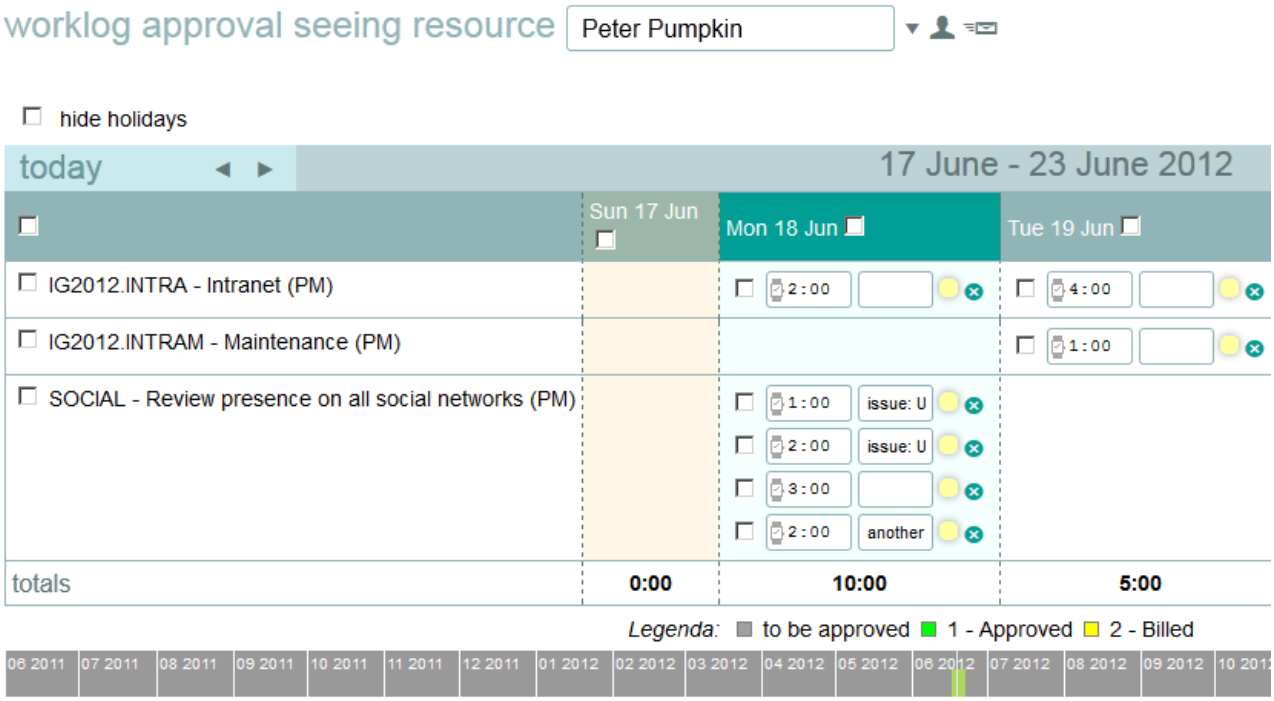


Actions appear that allow to bulk change worklogs. Be careful with this "potentially destructive" feature ☺!

7.4 Validating and billing worklog

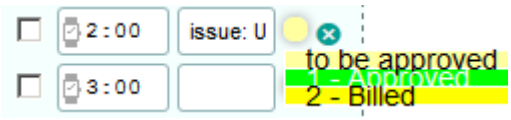
The "worklog approval" functionality covers needs of classifying work logs. Classification can mean in function of your needs, billed/unbilled, approved/not approved, and you can create your own classification categories. You reach this function from "worklog approval":



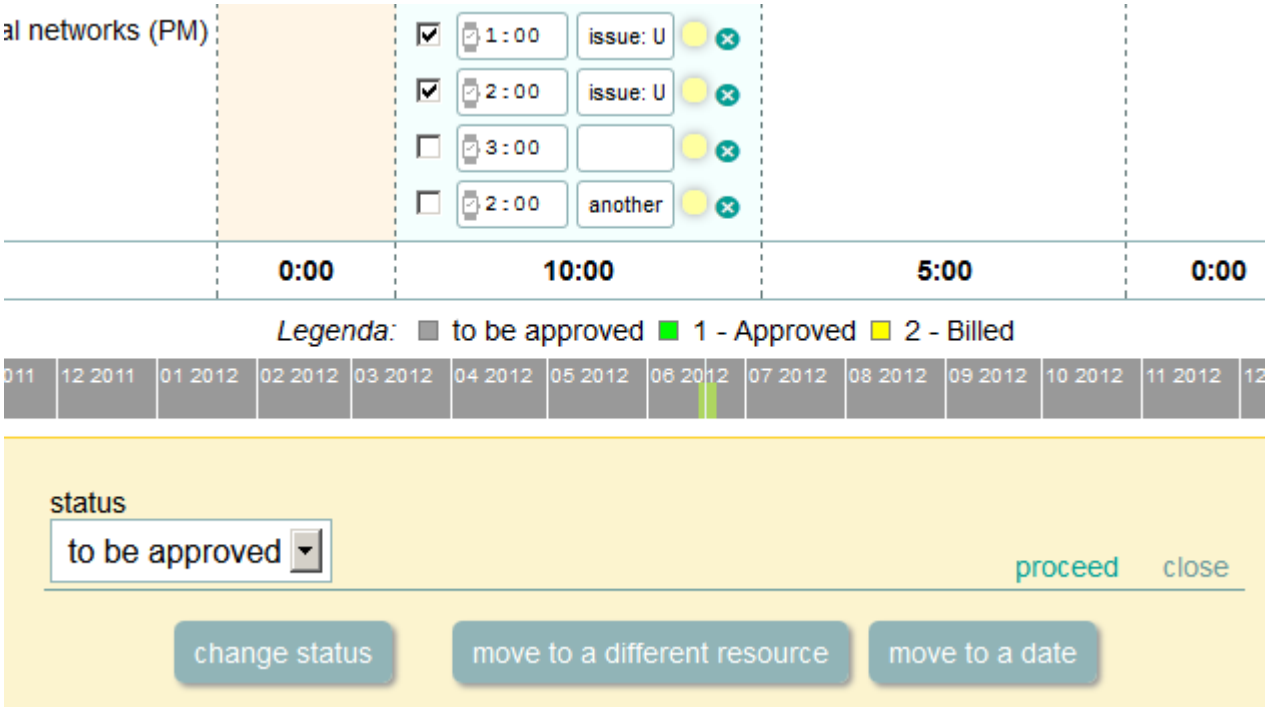


And also through the resource editor, if you have the rights to do so.

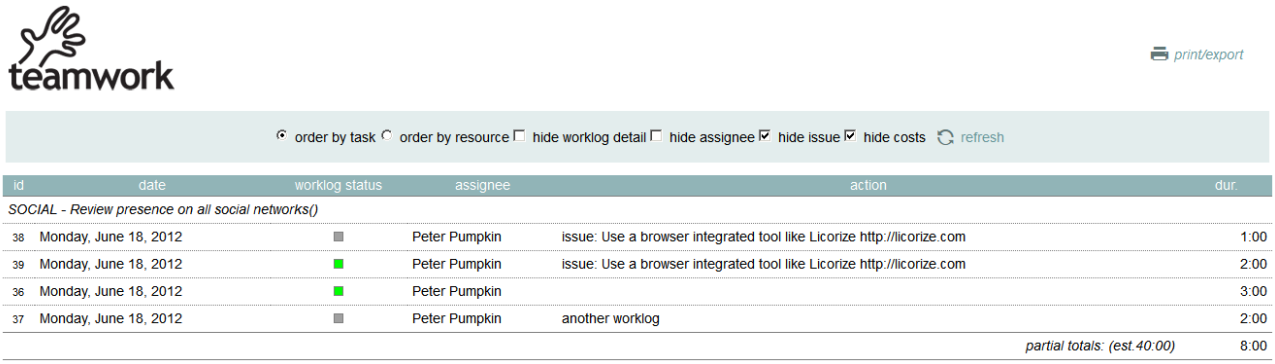
Click on the colored status little box, and pick the status you want to bring the worklog to.



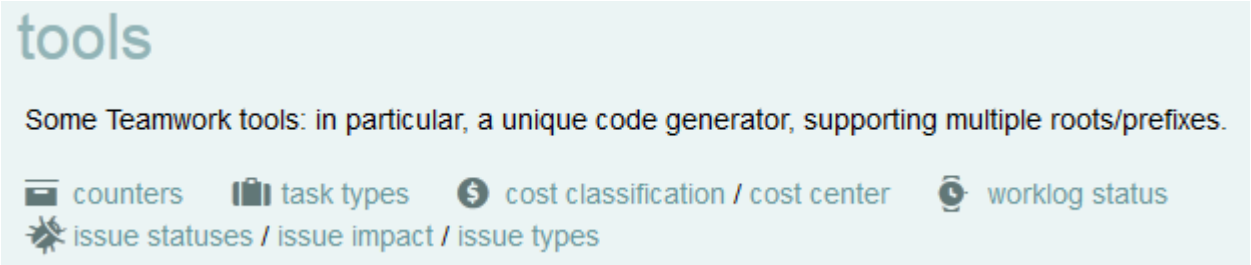
The web interface allows you bulk operate when you multi-select, as in several other screens:



So here you can do not only worklog classifications, but even bulk moving to task and date operations. The resulting page can of course be printed:

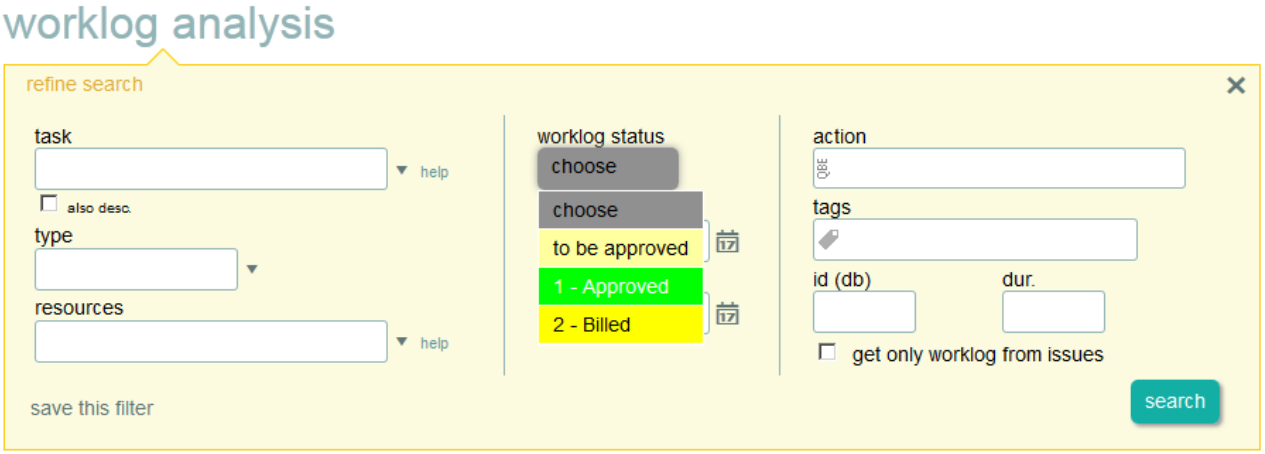


To create new work types, go to doc&tools and if you have the rights, you'll find



Click "worklog status" and there you can create them.

Of course you can always filter worklogs by type in your searches, say in worklog analysis:



And the print function has been expanded accordingly:

id	date	worklog status	assignee	action
SOCIAL - Review presence on all social networks()				
23	Saturday, June 16, 2012		Peter Pumpkin	
38	Monday, June 18, 2012		Peter Pumpkin	issue: Use a browser integrated tool like Licorize http://licorize.com
39	Monday, June 18, 2012		Peter Pumpkin	issue: Use a browser integrated tool like Licorize http://licorize.com
36	Monday, June 18, 2012		Peter Pumpkin	
37	Monday, June 18, 2012		Peter Pumpkin	another worklog

Who can approve worklog? There is a new permission “worklog management” local to projects, which lets “manage worklog, approve it, bill it”: you probably should enable it on project manager roles.

8. Agenda

8.1 Intro

Teamwork includes a complete agenda, capable of synchronization with several e-mail clients (Outlook, Entourage, Mail, iCal, Google calendar etc.).

> This because the agenda is compatible with the iCalendar standard.

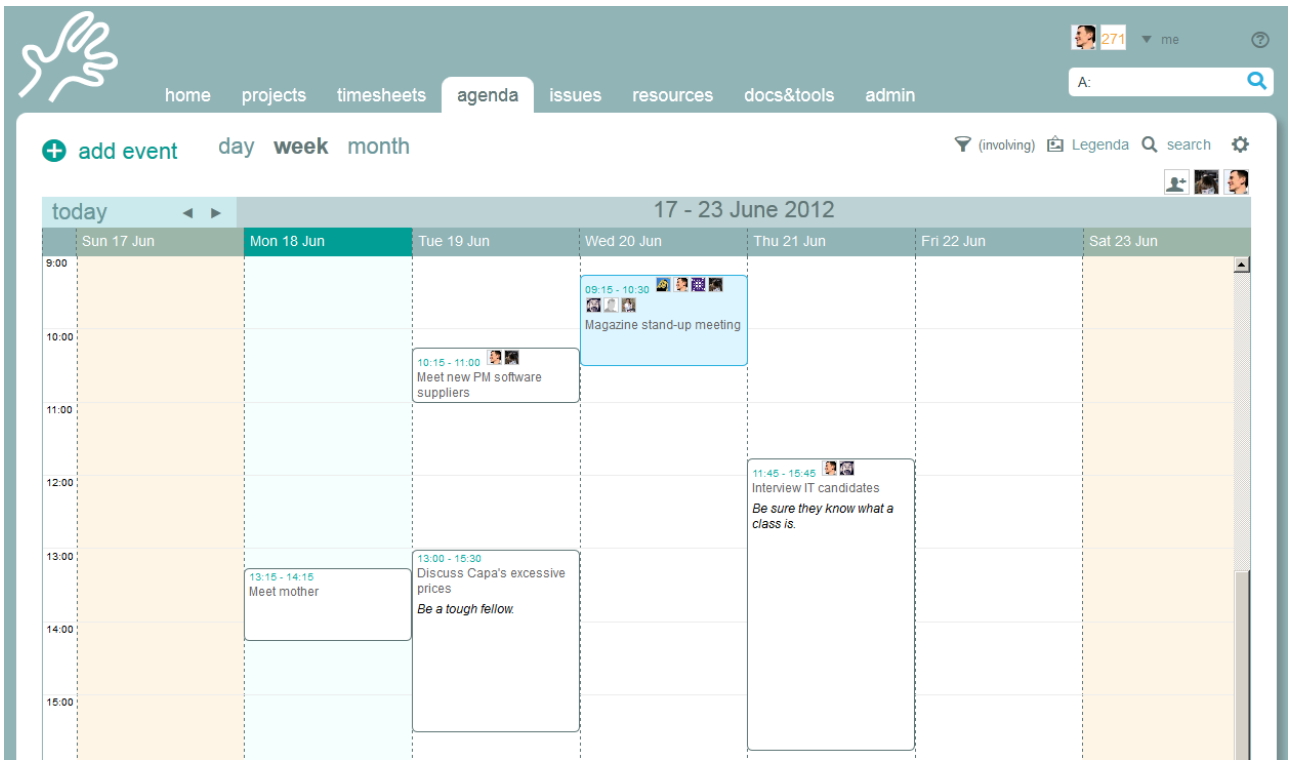
Teamwork smoothly integrates the workgroup agenda, meeting management, and external calendars visualization.

Let's start with something commonly used and easy.

8.2 Personal agendas

8.2.1 Weekly view

On menu bar click on the "agenda" button:



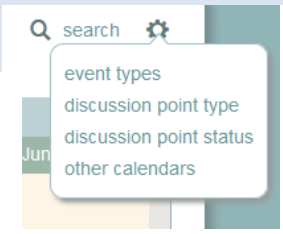
This is your week plan.

At the top you have a filter bar with prefilled filters:

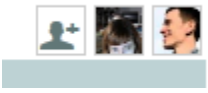
1) **involving**: every events your are invited in
2) **works event**: works events only
3) **personal**: your personal events only
4) **only me in it**: events where you are alone
5) **that I created**: events that you created. You are not necessary involved
6) **unavailability**: when you are on holyday, or at least not at work

Then there are filters by appointment type.

You can add as many new types you need by clicking “event types”.

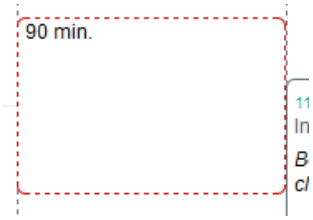


The usual workgroup buttons will allow you the see the agenda of other resources, we’ll get back to it later.



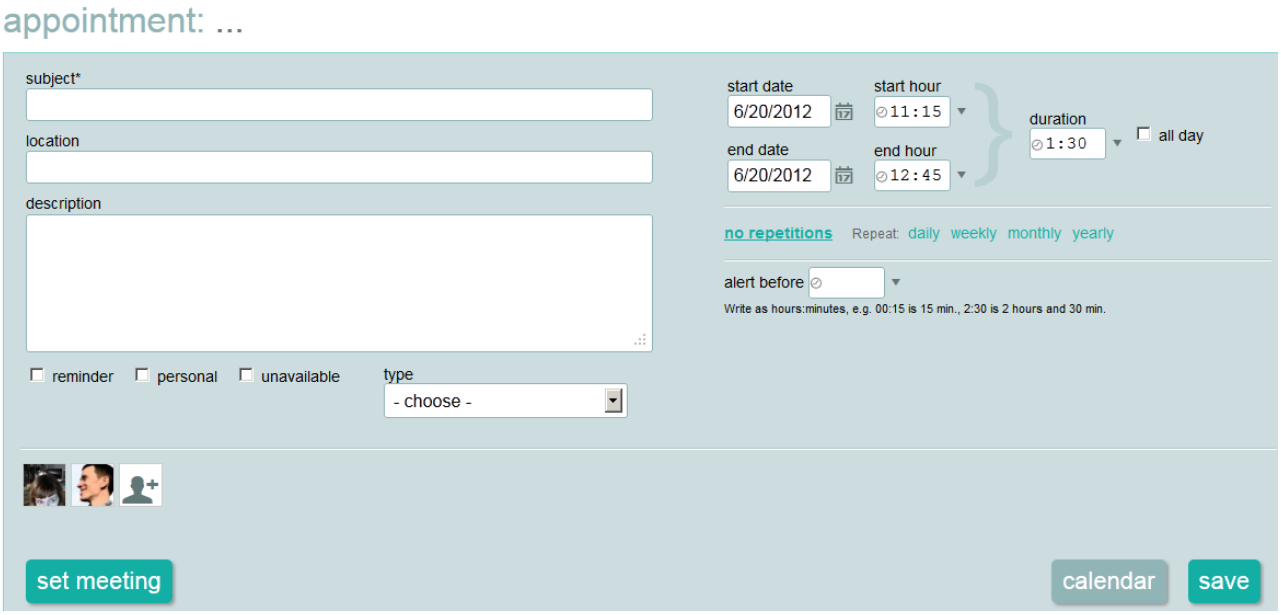
The dotted blue line on the plan is the current time.

If you hover the mouse on the plan and drag you will add new appointments, at the time where you were hovering.



8.2.2 Editing

Every event has an author, a subject, description, location and type (at least), some flags, plus its schedule.



The event type field is used for filtering events, but it is not mandatory.

Usually we check "reminder" when you want to remember something seeing it on the agenda, without “covering” a time interval.

Marking an appointment with the “personal” flag will hide its description to everyone (except you of course). Your colleagues will see that you are not available in that time interval, but they do not know why.

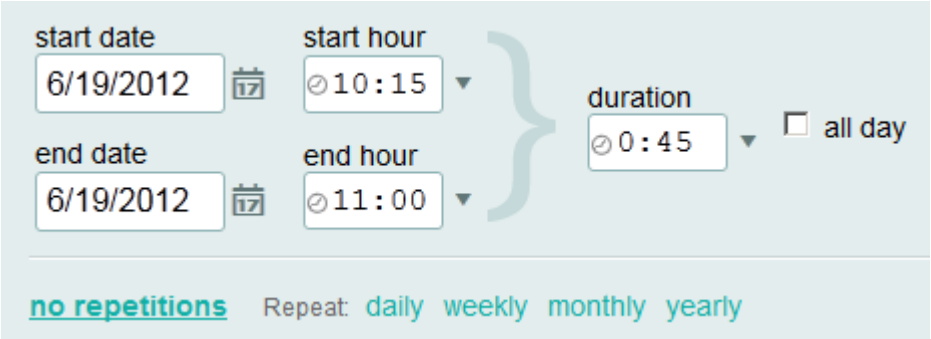
“Unavailable” means that you are not available for working. Unavailable events are reported on operator load and on plan on pink color. It is usually used for personal vacations (not company’s holidays).

Enabling the “meeting” checkbox will open the meeting management section after saving; we will see it in detail later.

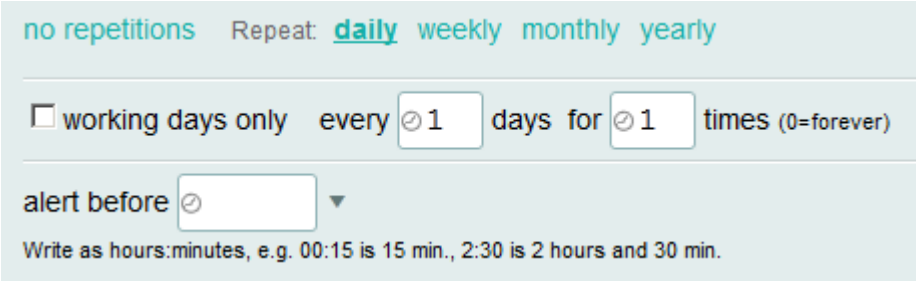
Teamwork supports recurrent events, in the schedule section you can pick many types of

schedule:

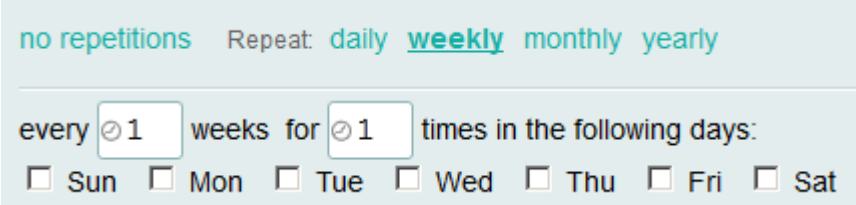
1) Single:



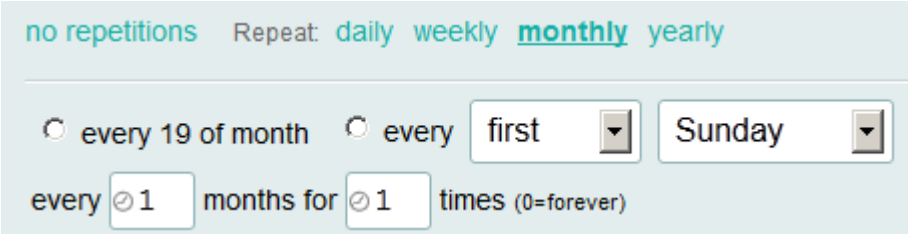
2) Daily recurrent:



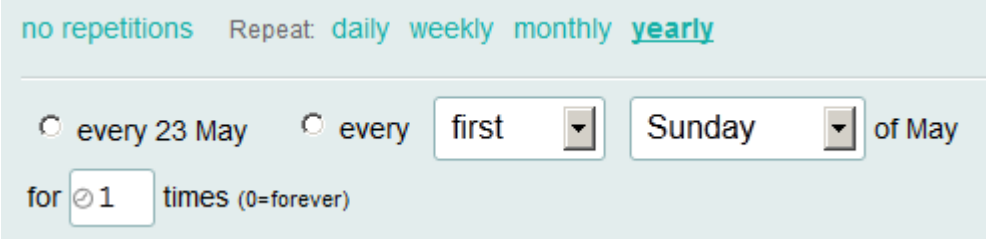
3) Weekly recurrent:



4) Monthly recurrent:

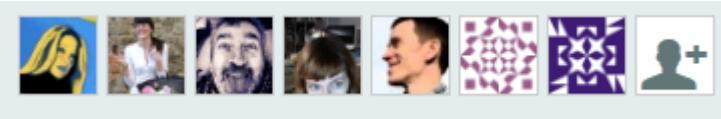


5) Yearly recurrent:

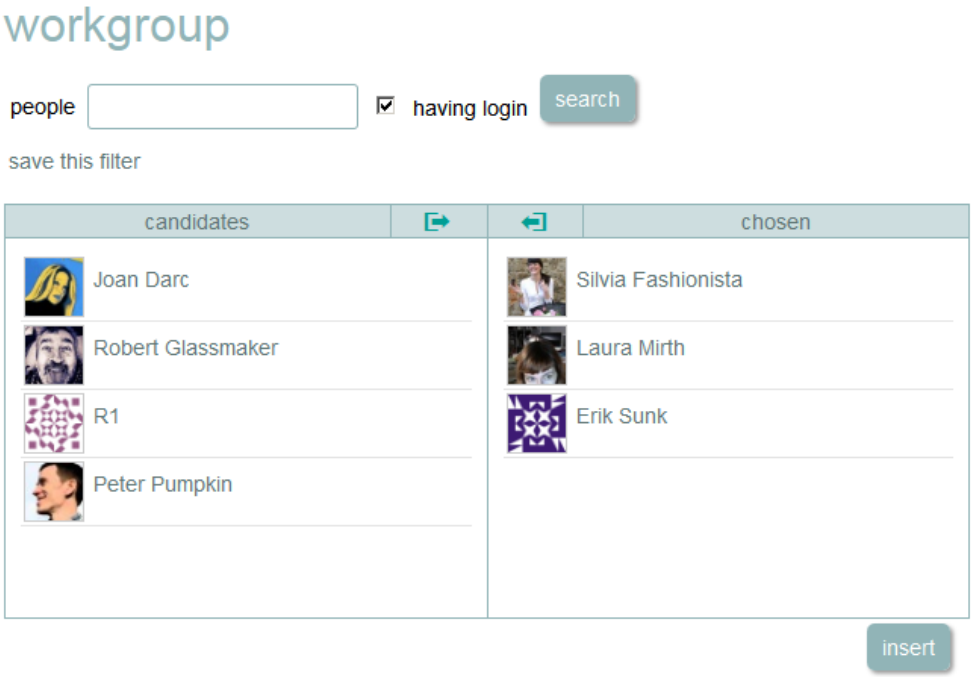


If you are configured to synch with external iCalendar clients (Outlook, iCal, Google calendar etc.) in your options, you can set even the alert time (e.g. Outlook's remainder).

You can add someone else to an event by clicking on the button on the “attendees” section.



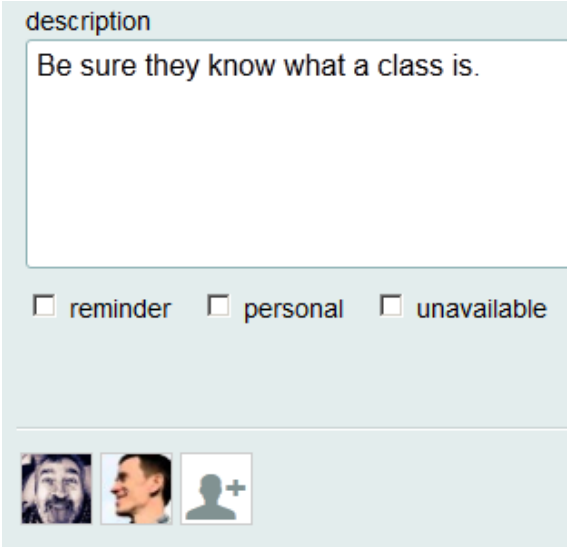
Similarly to clicking the workgroup button on the weekly view, it will open the resource selector:



Insert a filter or just press search to find candidates, then select one or more candidates and move them to selected. Additional searches will refresh the candidate part only, so you will not loose the current selection.

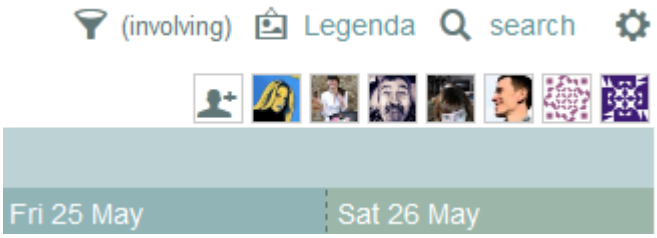
You can save prefilled selections using “save this filter”

When you are ready with your selection press “insert”. Attendees will be displayed on the editor:



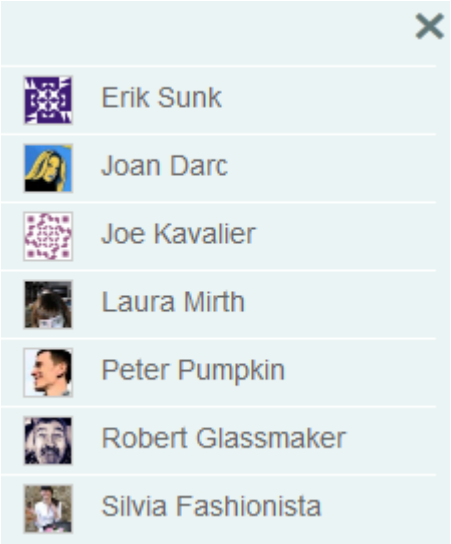
Then save the event.

If you have selected a workgroup, the weekly view will show the workgroup:



And the participants are shown on each event:

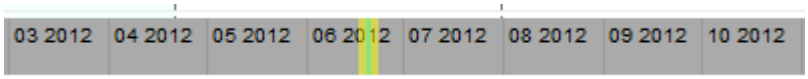
And there is a legenda:



When the legenda is on, you are working on a “workgroup” agenda, so for instance if you add a new event, by default the attendees are pre-filled with the current ones, and you can change them.

When one or more event overlaps the column width is split.

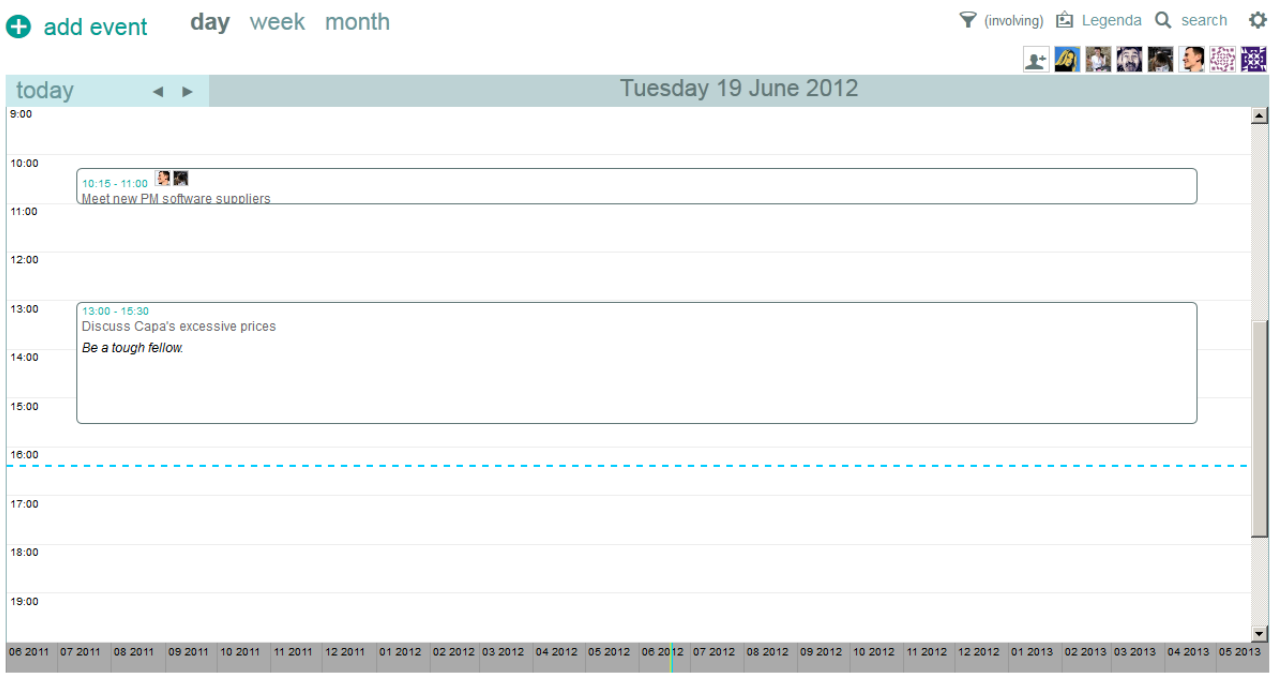
You can move quickly along time by clicking on the bottom time bar:



The green box is the period currently displayed on the screen, the blue bar means “today”.

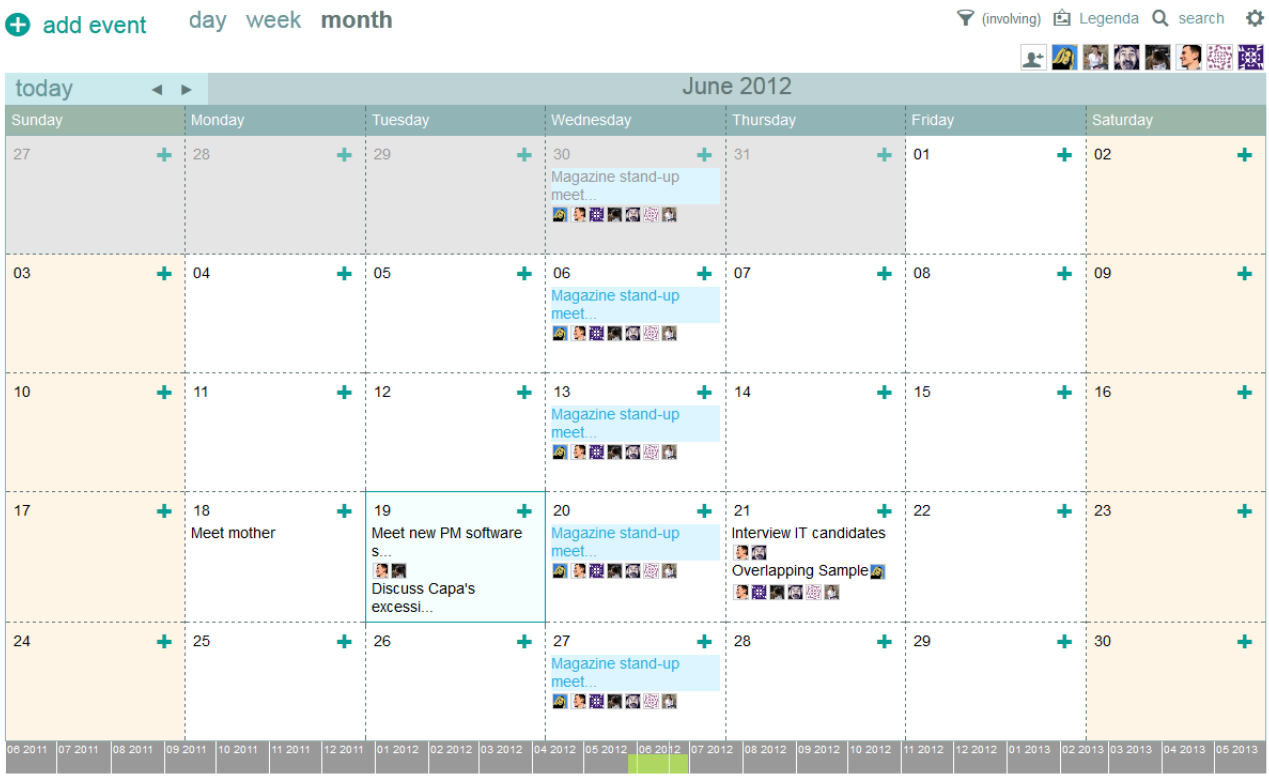
You may use daily or monthly views as well – following.

8.2.3 Daily view



This view could be more readable than the weekly one when there are many events.

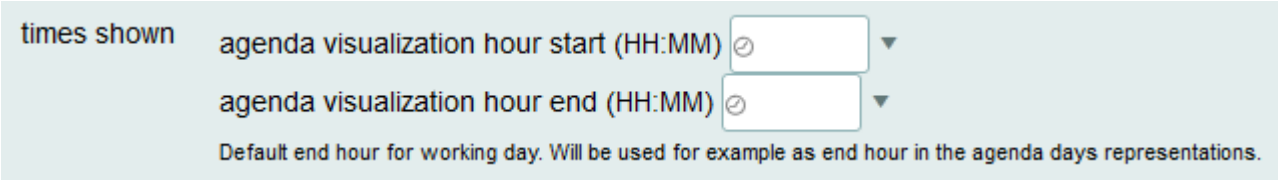
8.2.4 Monthly view



This is mainly a textual view.

8.2.5 Personal agenda configuration

You can change start and end time on your agenda from your options:



8.3 External calendars

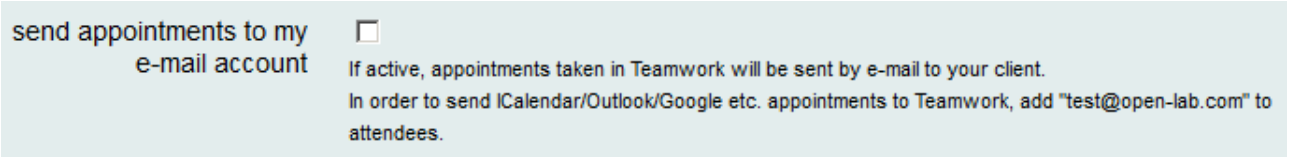
Teamwork can communicate both ways with external agendas.

8.3.1 iCalendar (Outlook/iCal/Google calendar/iPhone calendar…)

Lots of users have their calendar integrated in the e-mail client or in similar client solutions (Outlook or iCal are this kind of clients).

Teamwork may send an iCalendar messages to your client whenever an event is created/modified/removed on Teamwork's calendar.

In order to make this possible you must check in your options:

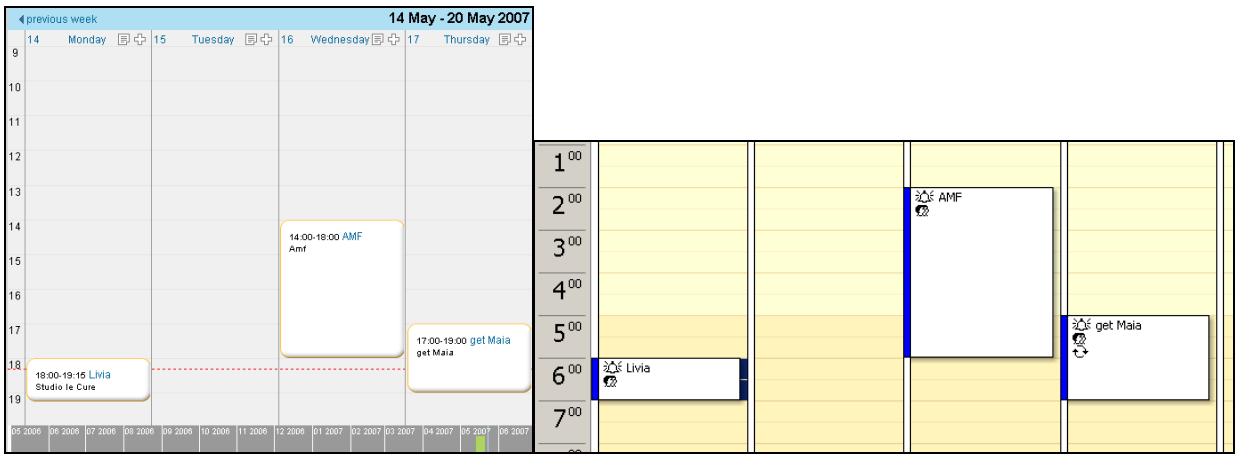


Following screens are from Outlook, but strange as it may seem, it works on every client.

When the client receives an iCalendar message, it recognizes it, presenting the mail received as in the picture.

The appointment will be put in Outlook's calendar when received, and it will be possible to accept it or not.

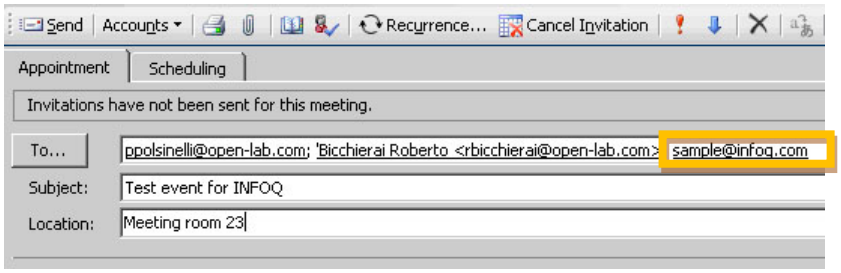
We had to do a special treatment for Outlook 2007, as there the compatibility with ICalendar has been extended, and so the handling of the Organizer is different from previous versions; in particular if you want to receive your events as subscribe-able ones in a 2007 client, ironically you must not be set as organizer of those events, otherwise the client will (rightly, from its point of view) not allow you to add them to your local calendar, assuming that the event already exists.



Notice that if you update an event in Teamwork, and resend it, Outlook will recognize it as an instance of the preceding event, and automatically update it - very nice!

How to create an appointment on your client and send it to Teamwork?

We did this simply by making Teamwork capable of downloading e-mail. Hence when you create your appointment in Outlook, you just "send" it to Teamwork, by having among the attendees the Teamwork's e-mail. If for example Teamwork downloads e-mails at the **sample@intergeo.sample** address, put this e-mail among the attendees:



A suggestion: manage the events where they are created. If you create an event on Teamwork if you have to change it the best way is to change it on Teamwork. Similarly if you create an event on your client, modify it there, not in Teamwork.

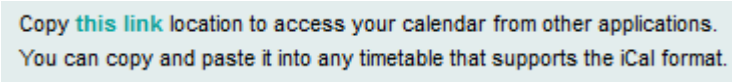
This Is not e Teamwork's bug, but e feature of some iCalendar clients that recognize the event organizer from the e-mail sender, not from the vEvent tags. See for detail http://www.twproject.com/icalendar.page

All e-mail downloaded gets logged in Teamwork's e-mail log: see "17.3 Logging".

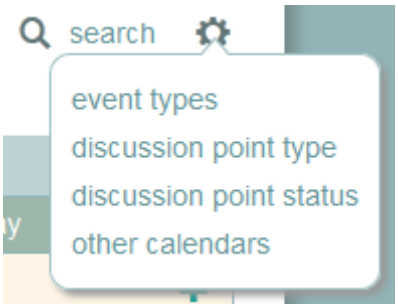
8.3.2 Google calendar

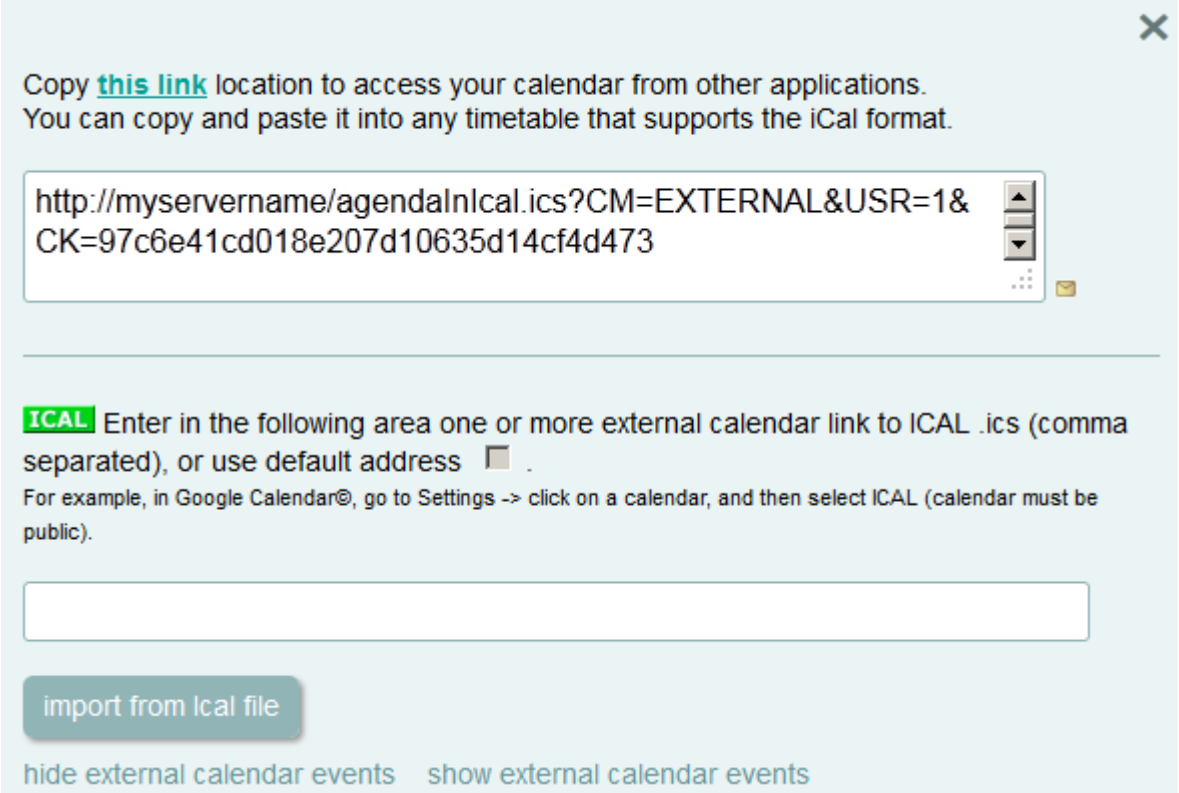
There are other options regarding calendar. You can publish your whole Teamwork calendar on a web-based calendar service like Google calendar.

To do this go to your options and copy the proposed address into your on-line service:



You can reach the same link from the agenda. Click on "other calendars":





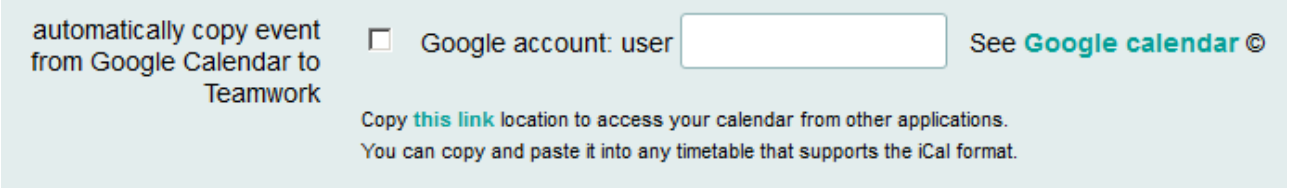
As this box shows, you can also include an external calendar inside Teamwork's one. Just paste the external calendar URL in the field above.

In both above cases, events remains in their respective places (Teamwork' calendar on

your server, Google calendar on Google).

There is a third way that allows you to copy appointments from Google calendar to Teamwork's one.

On your options set the Google account:

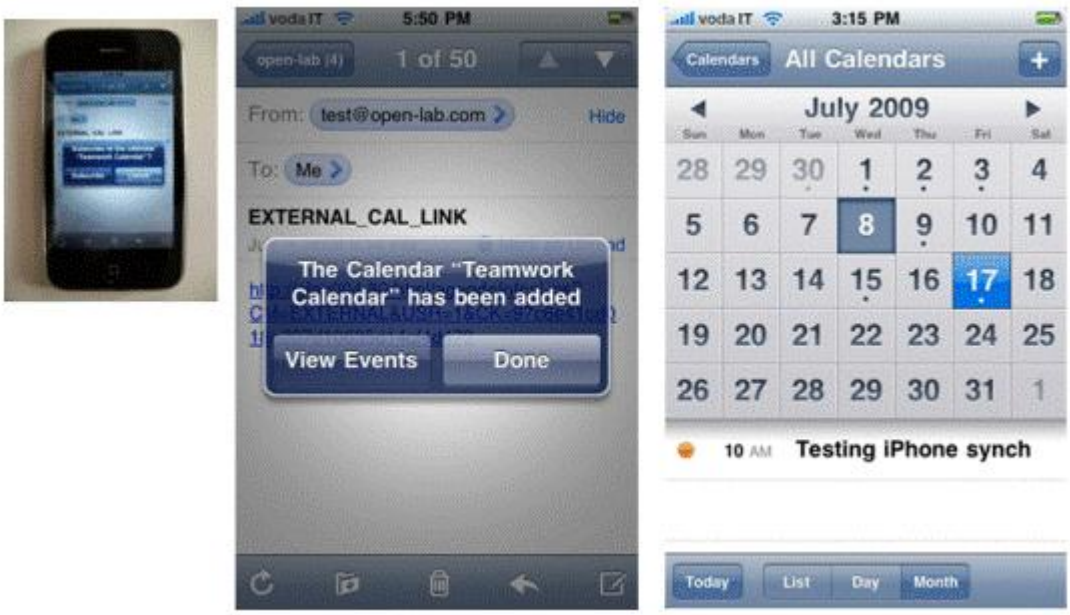


A scheduled action will check Google calendar changes.

8.3.3 iPhone calendar

This works similarly as above: the URL seen when selecting "other calendars" can be sent by e-mail to your iPhone (click on ; if it does not appear, it's because you haven't set an e-mail on your Teamwork user).

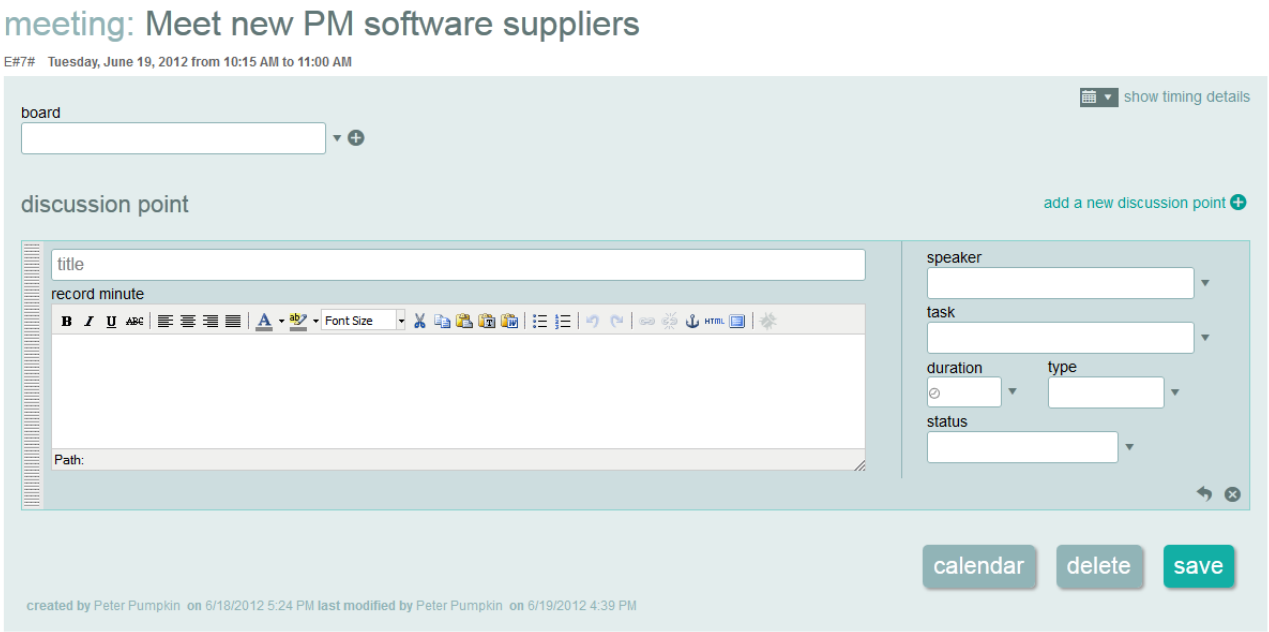
Then when the e-mail is received on the iPhone, the link will be recognized as a "calendar" link:



8.4 Meetings

A meeting in Teamwork is an event linked to a section that manages discussion points and meeting minutes.

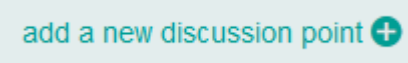
You can create a meeting simply by click “set meeting”: the agenda part will be collapsed and a meeting part will magically appear:



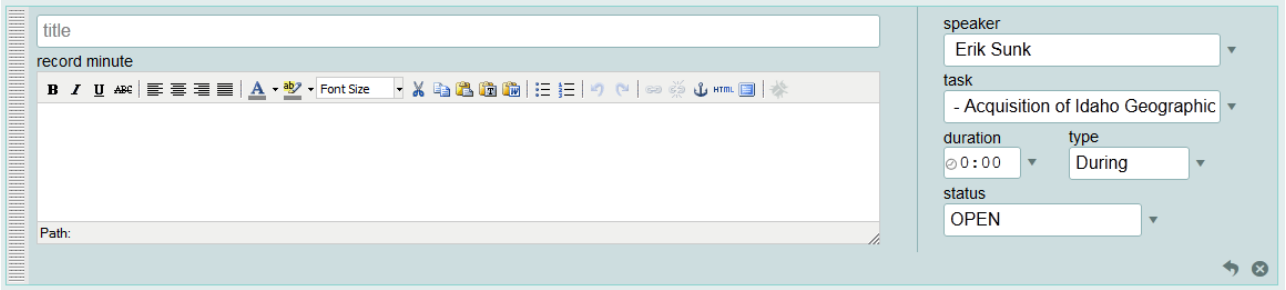
You can edit meetings title and descriptions, and evenutually link a board (see “5.3

Boards”).

You can always edit also the timing details by clicking on:

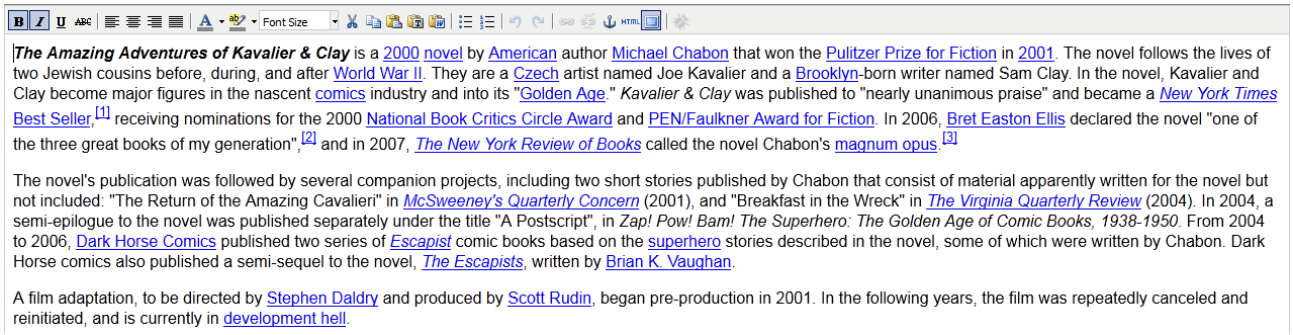
Add discussion points by clicking on 

:



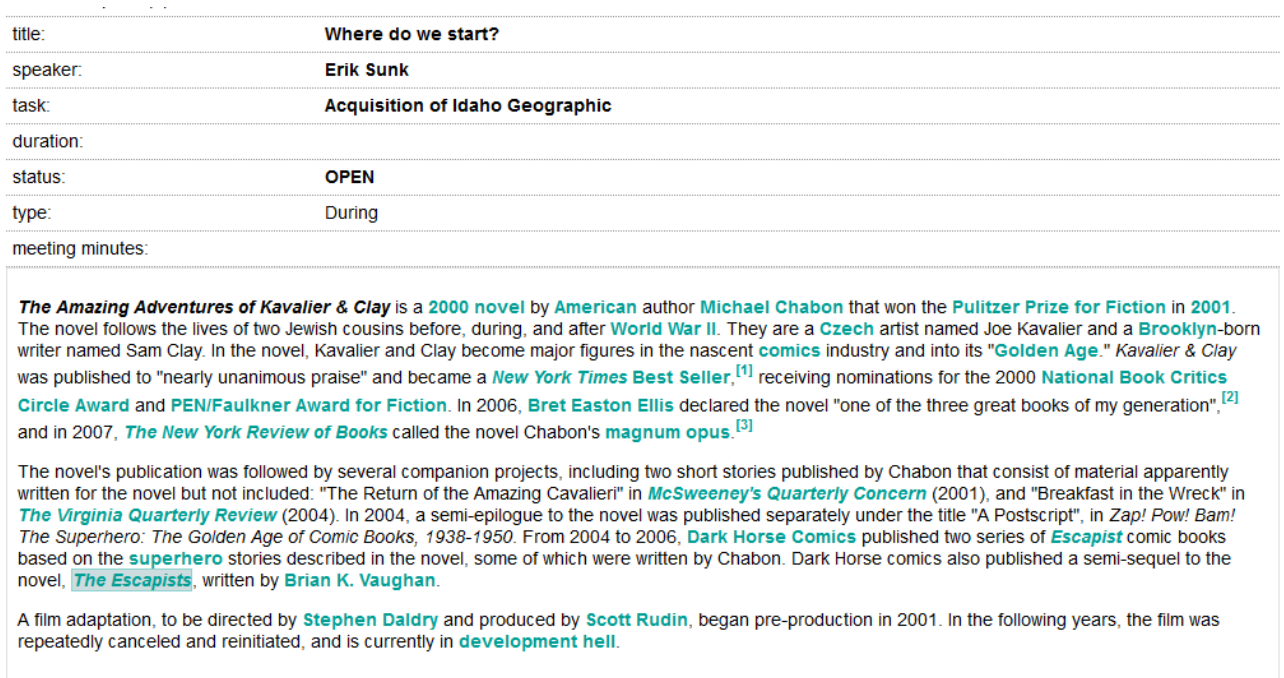
Each discussion point is constituted by a type, a speaker (a resource), a title, eventually a task subject of the discussion, and a minute, which is a part of the total one.

When the meeting starts, you can use Teamwork to record the minute – you can set the editor to full screen:



You can write an html text, and paste images as well. This is a long text field so you have no limitation on text size.

You can print the meeting report, export it to pdf or send it by e-mail, as usual.



9. Documents

9.1 Introduction

Document management is by itself wide enough to have dozens of specific applications (called DMS). Teamwork does not want to compete with specific tools, and we intentionally kept document management to the essentials, with some powerful and simple techniques. We believe that it can as it is satisfy a wide spectrum of companies.

Where do project and document management meet most frequently? Probably one of the most common requirements is to find documents related to a task. There are many solutions to this requirement.

Supposing you have a local network in your office with at least a shared file repository (a server, NAS etc.) a common solution for keeping documents classified by project is to have a shared folder called “projects” and a sub-folder for each project. Probably the sub-folder name is the project code or the project name. This solution works really fine until you are in your office but when you need to access the document from the web, you may end up in troubles.

Usually the counter-proposal is to provide a web-accessible repository where the user uploads documents and tags them with project codes in order retrieve them. This solution fulfills the needs of remote access but puts a big burden on users. Uploading documents on the browser is *much* less comfortable than using the file system

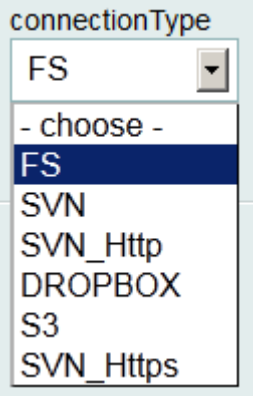
For instance if you need to link your project to “working” files like sources, cad files, .psd, audio or even worst video sources, upload is not an alternative.

Teamwork meets these needs with a simple and pragmatic approach.

9.2 File Storages

If you have a file server accessible from the server where Teamwork is running, you can create an entry point in Teamwork to access the server contents through the web.

What is meant by “accessible” above? Teamwork serve can access the **file system** and can contact a **Subversion server**, but proprietary protocols could easily be added. We recently added **Amazon S3** and will soon add **Dropbox**.



9.2.1 Configuration: file system

In order to configure a file storage in the case of a network file system access you have first to set-up the file storage root. File storage root is the lowest level access to you file server. For instance if your server folder structure is something like:

[server]/userdata/documents/projects

and if you want to limit the access to the project folder just set-up a “file storage” there, with content path [server]/userdata/documents/projects.

First of all you need access as administrator and set the allowed file storage paths:

file storage path allowed	\\olfs03\works\open,\\oldb02\bk	c:\demo\projects,d:\shared \files	Use comma separated paths.

You can insert several roots comma separated. The content path of any file storage of file system type must be a path visible from the server where Teamwork runs. Both local or network paths are allowed

c:\documents\projects, /usr/docs/prj,

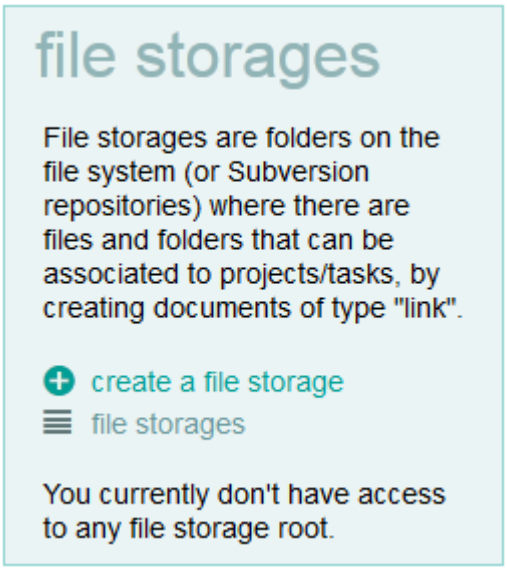
\\serv1\share1\docs

are all valid paths.

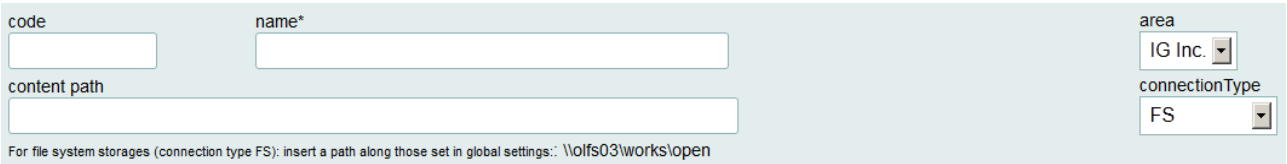
> Usually files servers contain relevant data, so you should be careful in creating a file storage root. Teamwork in order to improve security in case of file system connections requires that the administrator defines some file storage seeds in global setting, which restrict available paths. So a standard user cannot create a storage outside pre-authorized paths. See “16.2 Paths, network and security”.

9.2.2 File system example

Click on “docs&tools” on the menu and then on “create a file storage”:



The default file storage type is FS which is “file system”:



Saving it:

file storage

Teamwork user guides

code TUG (id: 5) name* Teamwork user guides area IG Inc.

content path \\olfs03\works\open\teamwork\documentation\userGuide connectionType FS

For file system storages (connection type FS): insert a path along those set in global settings:: \\olfs03\works\open

explore delete save

created by Peter Pumpkin on 6/19/2012 5:20 PM area IG Inc. last modified by Peter Pumpkin on 6/19/2012 5:20 PM New "FILESTORAGE" inserted in database.

Fill code and description, then the content path; choose also the connection type:

FS: file system

SVN: Subversion (Tigris.org)

9.2.3 Using Subversion SVN

If you want to use a SVN connection, first specify connection type; then specify host, username and password as well. The SVN server must be visible from Teamwork server.

connection data

host user password

Currently the SVN implementation supports the “svn:”, http and https protocols.

Example for Subversion only: for a repository url of svn://olfs03/platform, the host is olfs03, the content path above is /platform

9.2.4 Amazon S3

connection data

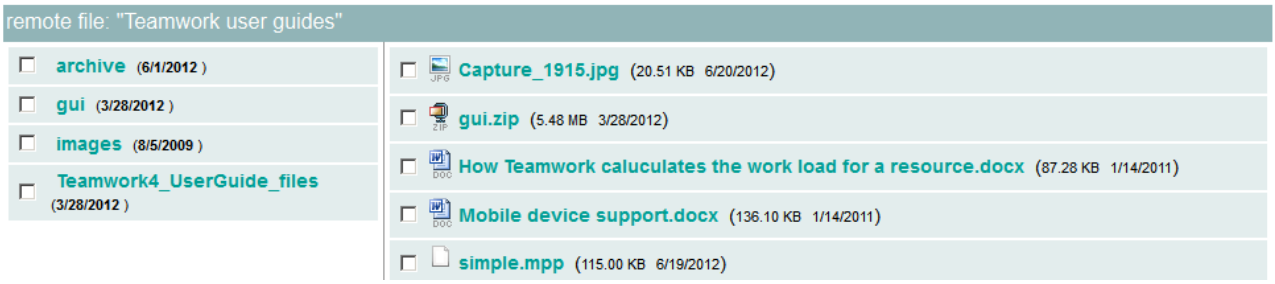
bucket access key secret key

These are the standard Amazon S3 authentication parameters.

9.2.5 Usage

If you wrote the correct parameters, after saving the “explore” button should appear.

explore



The left part lists directories, the right part lists files. This is a “browser window” on the files on the server. By clicking on a directory you will explore it and the page content will be refreshed. By clicking on a file you will download it.

You can create or remove directories or files and can perform multiple actions by using selection checkboxes.

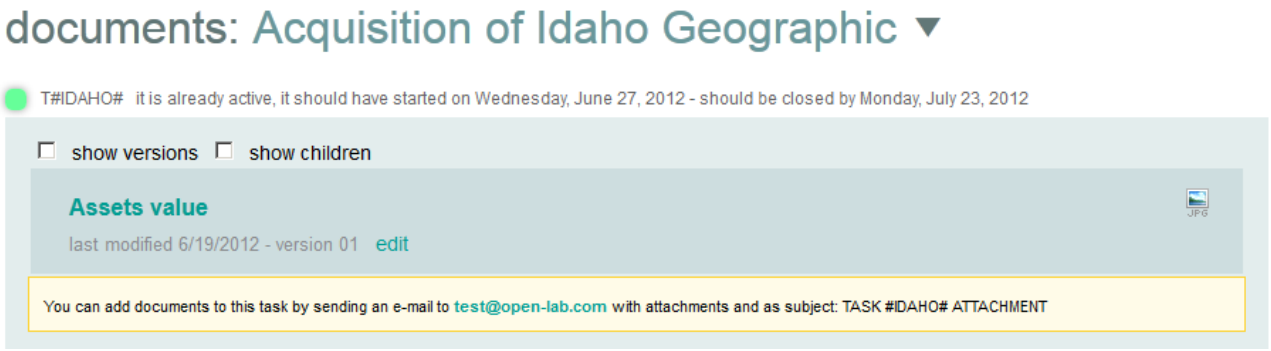
Download more than one file at once by selecting several and using zip selected. This makes download faster.

You can upload a file in the current folder by selecting one on your local file system and by pressing “upload” . You have hundred of files and folder to upload at once? Just zip them and Teamwork will ask to unzip the file once loaded. Check “overwrite” to overwrite existing files.

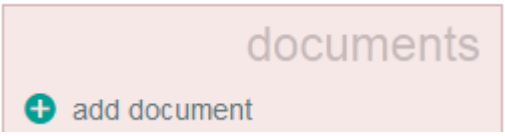
Now you know how to make your files accessible from the browser, but how to link documents or folders to a task?

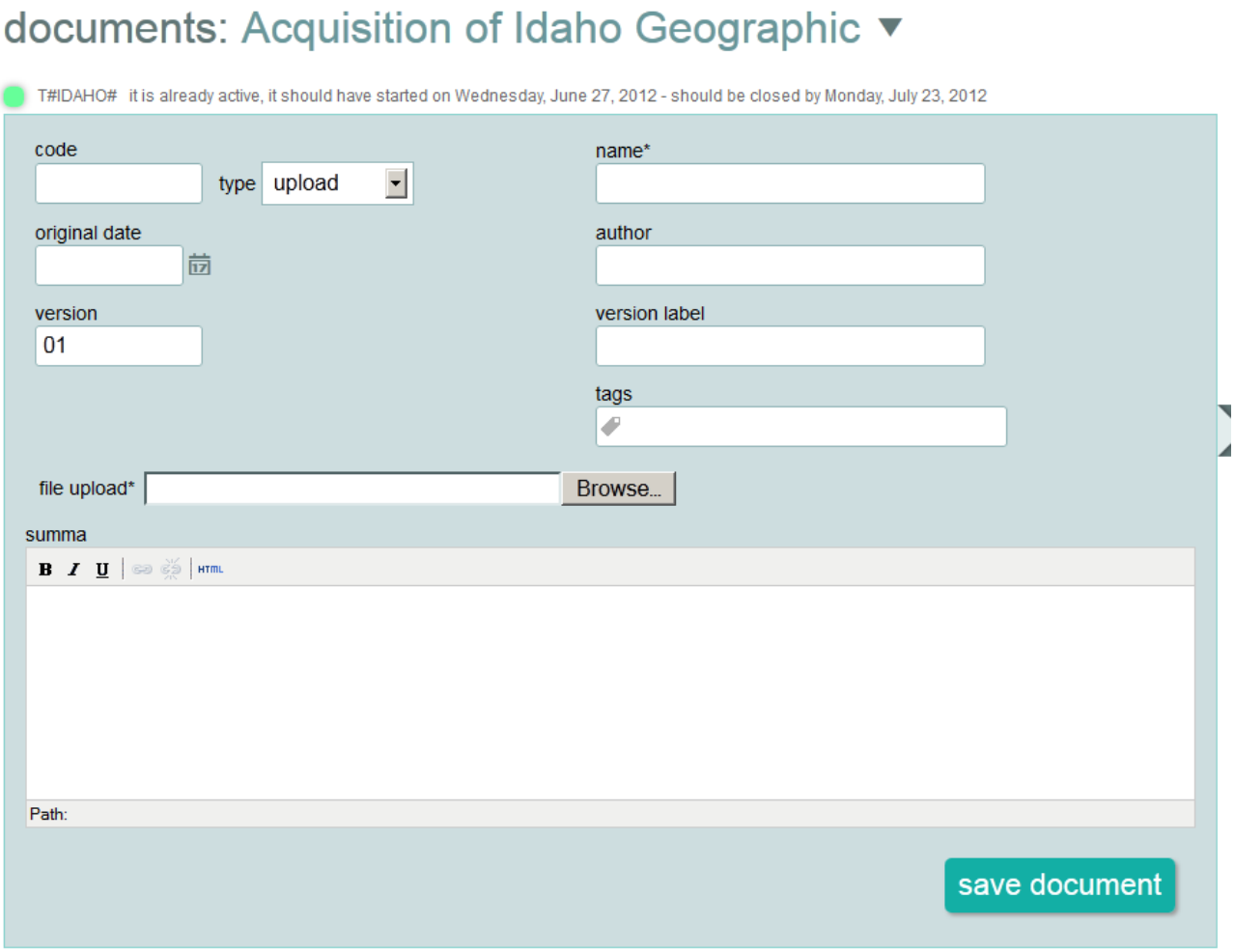
9.3 Linking a document

Teamwork allows document creation in both tasks and resources; in both editors you have a “document” view with similar functions:

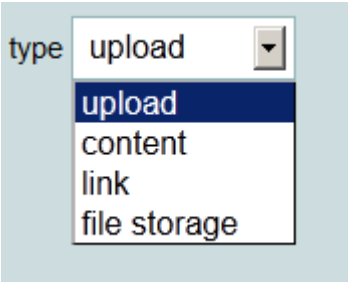


In order to add a document just click “add document”:



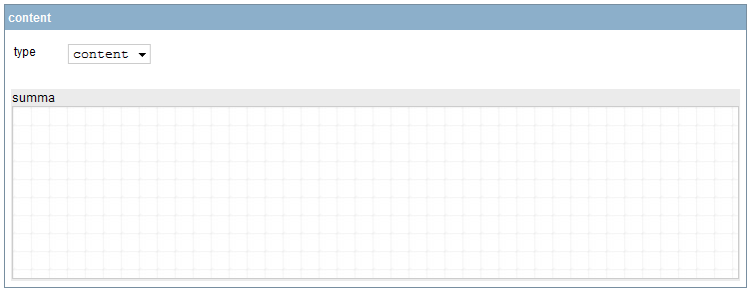


"name" is the only mandatory field for all document types.

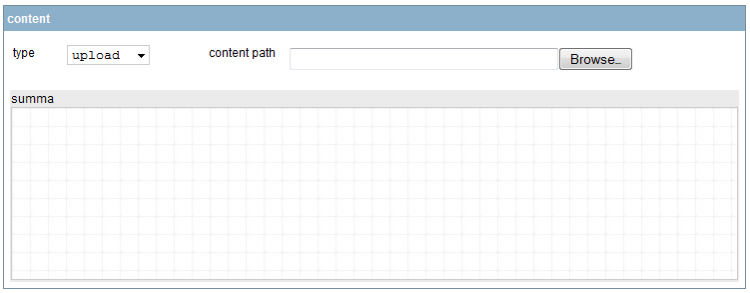


Then you have to define the document's content type. There are four different types of content:

1) **content**: is a sort of note. You can write a plain text:

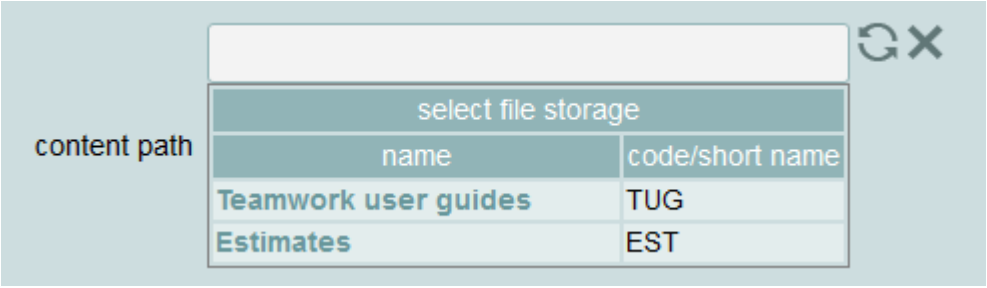


2) **upload**: you can chose a file to be uploaded to your repository (see next section)



3) **link**: you can specify a link to a “document”; it is simply a URL.

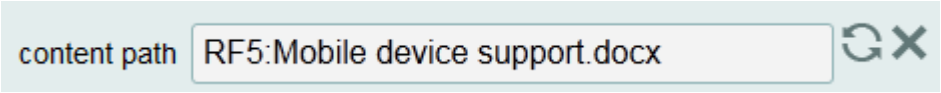
4) link to a folder or file in a **storage**. In this case click on then select a file storage:



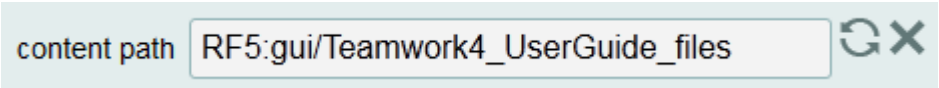
by clicking on a storage a popup will appear:

Select a folder or a file to link it, then save the document.

A file will look like this:



A folder:



It’s a relative path to the files; the fact that it is relative is the way to give only partial visibility to the file system, ensuring that permissions are preserved.

Once a link is created you can access it from task (or resource) editor:

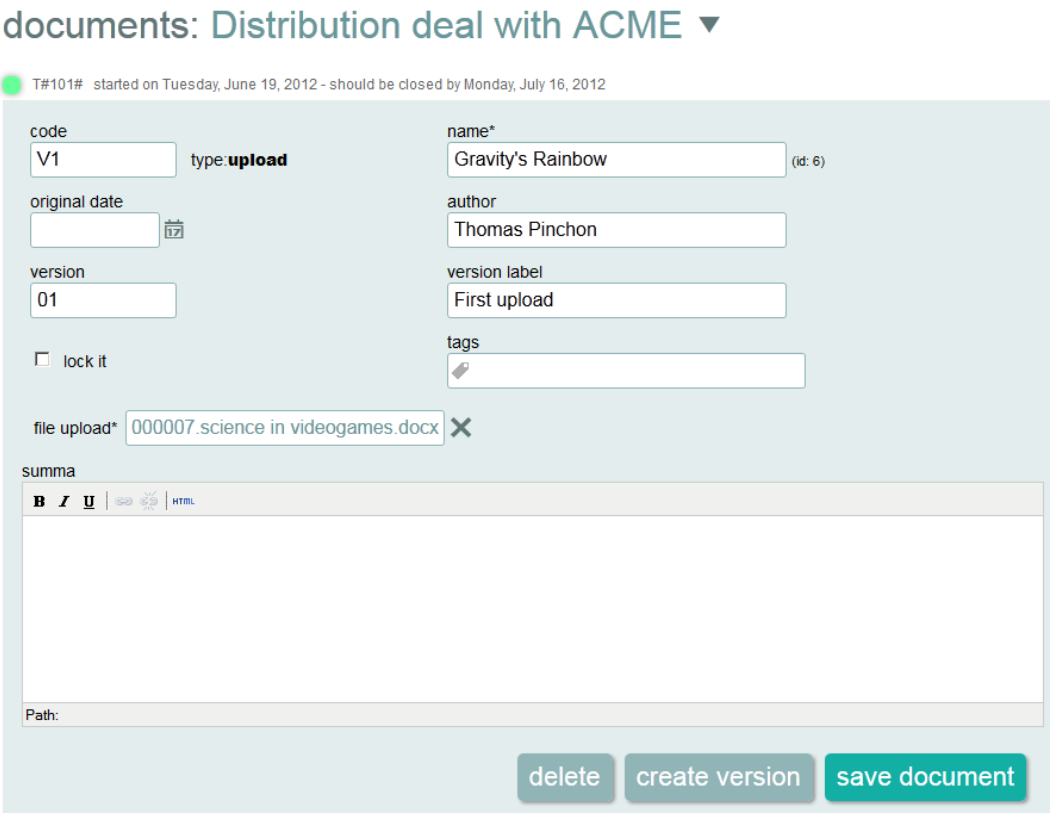
A common solution is to upload you files in a special folder managed by Teamwork, the repository:

9.4 Repository

You can specify a folder on Teamwork's server where to upload documents (see "16.2 Paths, network and security").

In order to upload files, just select a content of type "upload", select a file on your computer and save the document.

Documents managed by Teamwork have some interesting additional feature. First of all you can create versions; if you have updated a document and you want to upload the new version, go to the editor:



and press "create version".

Then select your new file and save the document. By default only the latest version is shown, but you can enable "show versions":

You can "lock" the document in order to avoid that someone uploads a new version while you are using it. You should lock the document while you are editing it.

Documents will be full-text indexed; these are the extensions supported:

".txt", “.rtf”, ”.log”

“.pdf”.

“.htm”, “.html”,

“.zip”, “.war”, “.jar”,

".xls", ".xlsx", ".xltx", ".xlsEmb",

".doc", ".docx", ".dotx", ".docEmb",

".ppt", ".pptx",

“mpp”, “mpx”,

".msg", ".msgEmb",

".vsd",

".pub".

9.5 How do I make someone access a folder through a project without giving access to the entire repository?

Suppose you are the administrator and you want the user Andy Whirl to access and operate on a folder of the file system. You should assign Andy on the project with a role that contains permissions relative to file storage (admin -> roles management):

file storage explorer: read	☑	Allows to read files within a file storage
file storage explorer: write	☑	Allows to write files within a file storage
file storage explorer: create folder	☑	Allows to create folder within a file storage

Then create on the project a file storage document pointing to a folder. When Andy accesses the documents folder on the project, and goes to a file storage document editor, he gets this:

content path RF5:archive

Andy can click on the link, but can’t change the path; and the popup opens with file system operations enabled:

create directory

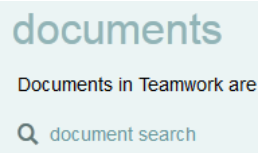
zip selected delete selected

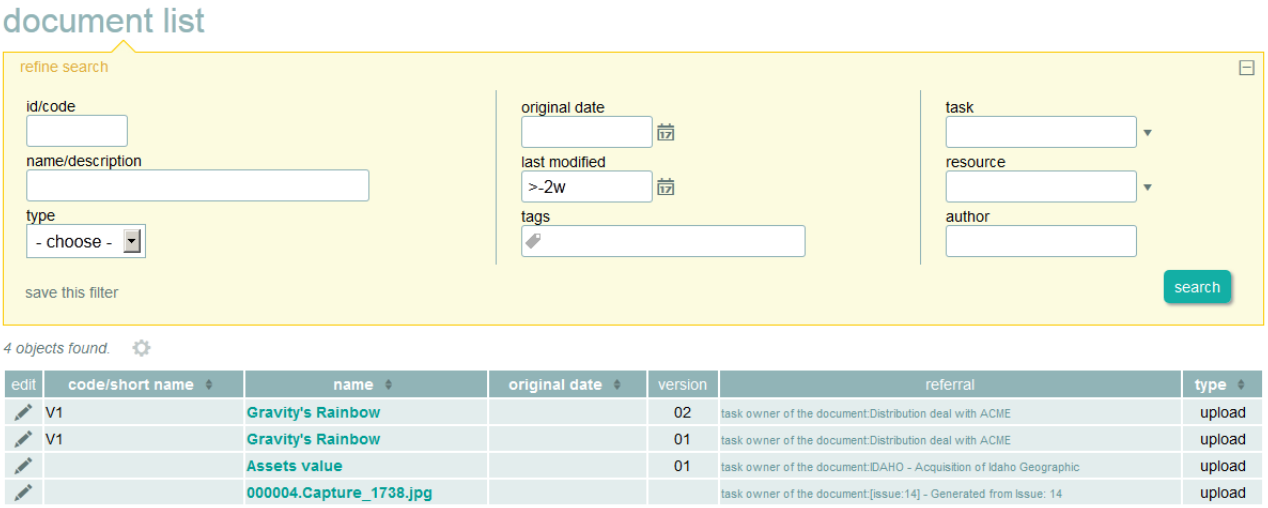
Choose File No file chosen upload

So you have all you needed.

9.6 Document search

Teamwork has a specific document search page (of course the generic search field searches in documents). Go to docs&tools -> document search:





From every document you can access the “container object”, being it a task or a resource.

10. Portal/Dashboards

10.1 Introduction

Teamwork has a quite flexible home page configuration functionality so that every user can customize it to fit her/his needs.

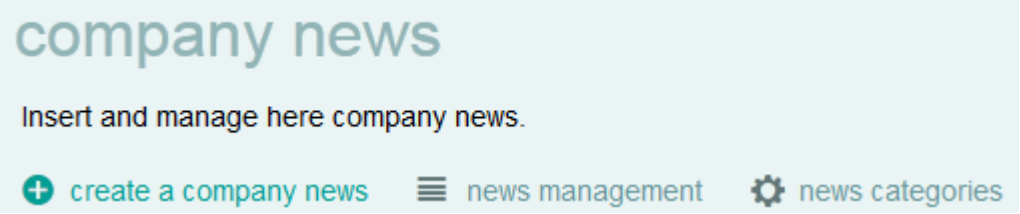
Management software can be felt as a burden; a custom home page can help in developing a better relationship with the software.

In particular Teamwork supplies tools for managing “news”, and a complete dashboard customization system.

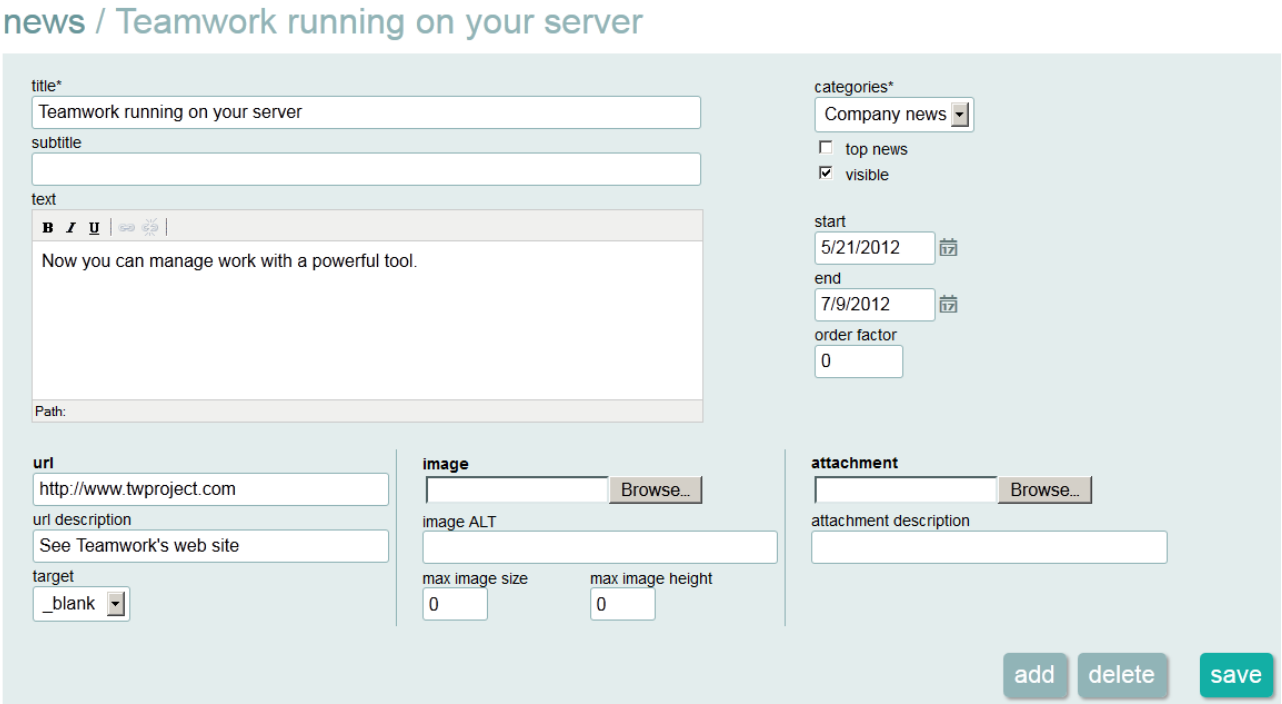
10.2 Company news

Default Teamwork’s pages (default? Can I create others? Yes, keep reading) include a box (a portlet) called “company news” that allows to display news on user’s pages.

In order to publish news, select the menu button docs&tools, then



Click on “create a company news”:



a news is rich object with several properties. Most relevant are:

title, subtitle, visibility, start end dates, and text.

You can also insert a link, an image, an attachment and eventually publish the news on Teamwork’s RSS channel.

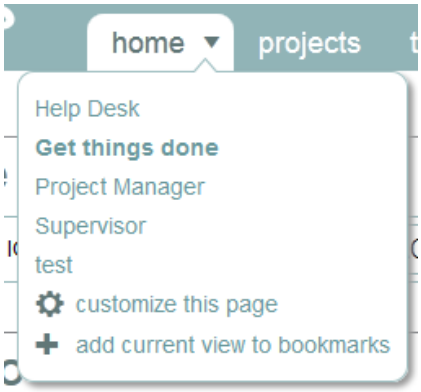
In order to create/manage RSS feeds just go to “RSS feed management”.

10.3 Dashboard Customization

Every Teamwork' user by default can customize its home page/dashboard.

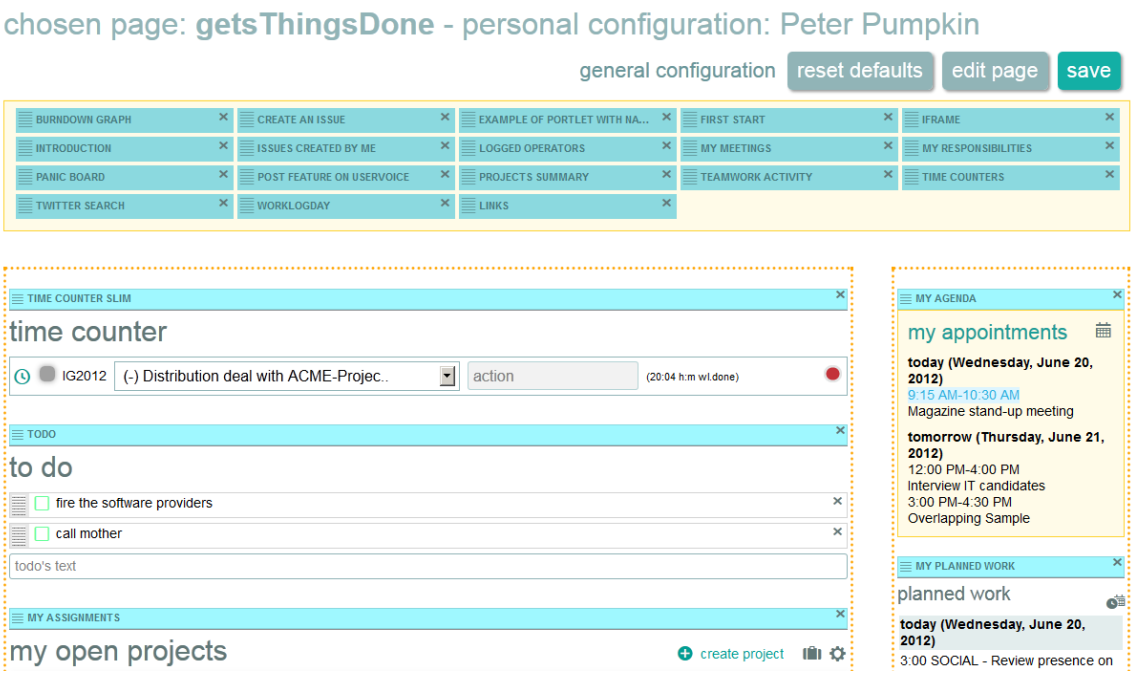
> You can block parts or even the entire page if you want to – see "10.5 Templates".

A page is composed by a template (a sort of grid) where portlets can be inserted.



> Only page with .page extension are customizable, .jsp are not.

In order to customize a page, go to say your home page, click on "customize this page" and you will be redirected to the page editor:



The top part of every portlet can be used to drag it around. Drag to move, click "x" to remove.

In order to add new portlet, drag from the portlet list on top in the desired slot.

Administrators will have a double option: they can change their own page or the "default view" for other operators, so pay attention to which radio is selected.

The two checkboxes let you set the page as your default or (for administrators only) for every user.

Dashboards are composed of portlets, pages and templates; let's first see portlets.

10.4 Portlets

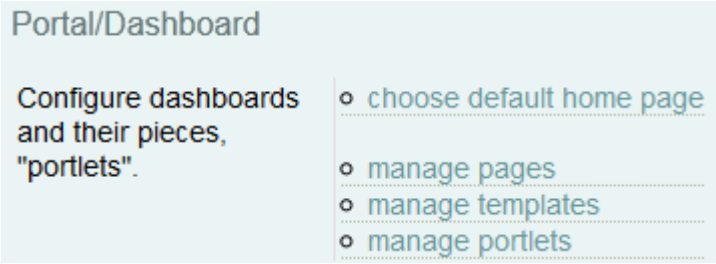
Portlets are the small parts that compose a page; they a practical way to extend Teamwork functionality, adding say a new way to represent data.

We provide a quite wide range of built-in portlets, but brave users (Java programmers) may try to create their own.

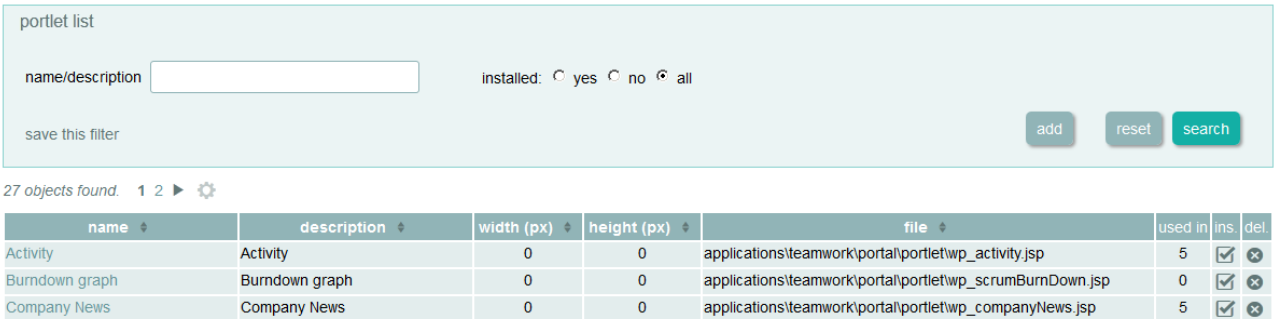
You can always reset the page to its default structure:

10.4.1 Editing portlets

From the administration page (admin on top menu):



Click on “manage portlets”:



Here you will see the status of installed portlets. A portlet is a simple self standing .jsp page that is in the configured directory (by default

[root]/applications/teamwork/portal/portlet

).

In the portlet editor you can change name and description, remove the portlet from where it is used and modify it.

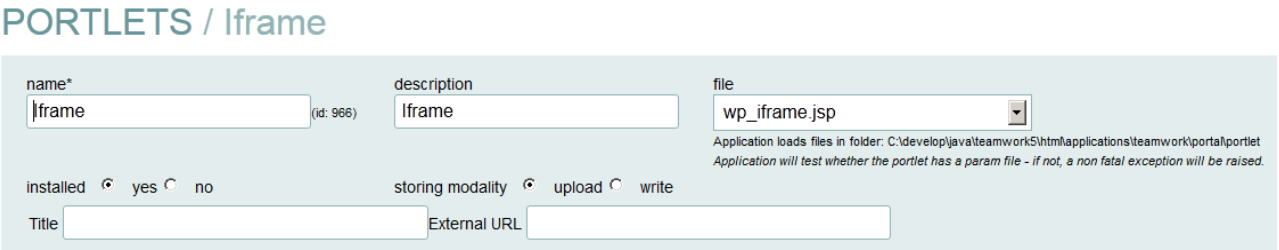
You can select a portlet file from the combo that lists the .jsp files in the folder.

> By selecting “storing modality” to “write” you will be able to change the portlet source code. This requires Java skills, do not do it at home ☺

The “security” tab will allow restricting portlet usage to users that have particular permissions.

10.4.2 Show some external content in your dashboard

If there is any web site which you’d like to be available on your dashboard, you can configure an “iframe” (i.e. a viewer of an external website) portlet to point there: open the “iframe” portlet in the portlet list, configure it:



setting title and url, and then add it through the customization functionality.

Of course this is a *rough and ready* solution; if the web site content is available through a web service, generating say a widget view, you just have to paste the content in the portlet body.

10.4.3 Create your own portlets

You should be fluent in Java in order to create your own portlets.

If you are still reading this section probably you didn't see the previous disclaimer, anyway…

Let's start with something really easy: we want to create a portlet that shows a satellite photo of weather on NE USA.

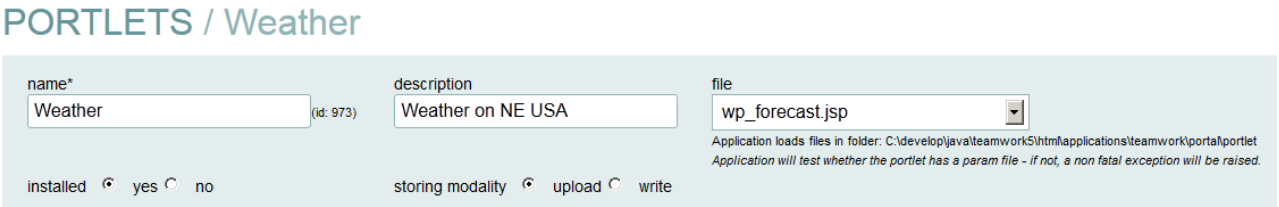
First create a new file on [root]/applications/teamwork/portal/portlet folder: we'll call it "wp_forecast.jsp".

Here is the content:

```
<%@ page import="org.jblooming.waf.html.container.Container" %>
<%
  Container cont = new Container("MYAPPS");
  cont.title = "Today's forecast";
  cont.height = "100%";
  cont.start(pageContext);

  %><img src="http://weather.yahoo.com/images/northeast sat 440x297.jpg" alt="satellite forecast"><%

  cont.end(pageContext);

%>
```

Then from the portlet editor create a new portlet: call it "Weather on NE USA", select "wp_forecast.jsp" as file, save it.

Probably you will get a warning about the absence of a parameter configuration portlet. This is a more sophisticated feature that allows you to customize portlet with parameters through the web interface. Study the supplied example wp_RSSreader.jsp and wp_RSSreader_param.jsp



You can change the code by editing it "on-line"; just select the "write" radio.

yes ○ no storing modality ○ upload ◉ write

write

```
<%@ page import="org.jblooming.waf.html.container.Container" %>
<%
  Container cont = new Container("MYAPPS");
  cont.title = "Today's forecast";
  cont.height = "100%";
  cont.start(pageContext);

  %><img src="http://weather.yahoo.com/images/northeast_sat_440x297.jpg" alt="satellite forecast"><%

  cont.end(pageContext);

%>
```

Your new portlet is now ready to use: from “customize page”, drag and drop the portlet in your page, save.

10.4.4 Additional portlets in distribution

With respect to the portlets that you can see in the predefined home pages, there are many more, and more get added with new distributions.

When you do the initial setup, all those of the setupped release get recorded as available; but if you do an update, you may have to create the portlet by yourself n the backoffice: but that is easy, as you will already have the file from the update, and the rest has been explained above.

Panic board

Since version 4.7 there is a new web part: panic board, to follow closely the progress of a single project:

T#IG2012.JUNE# June Issue

expired since: days:8 hours:17 minutes:47 seconds:23

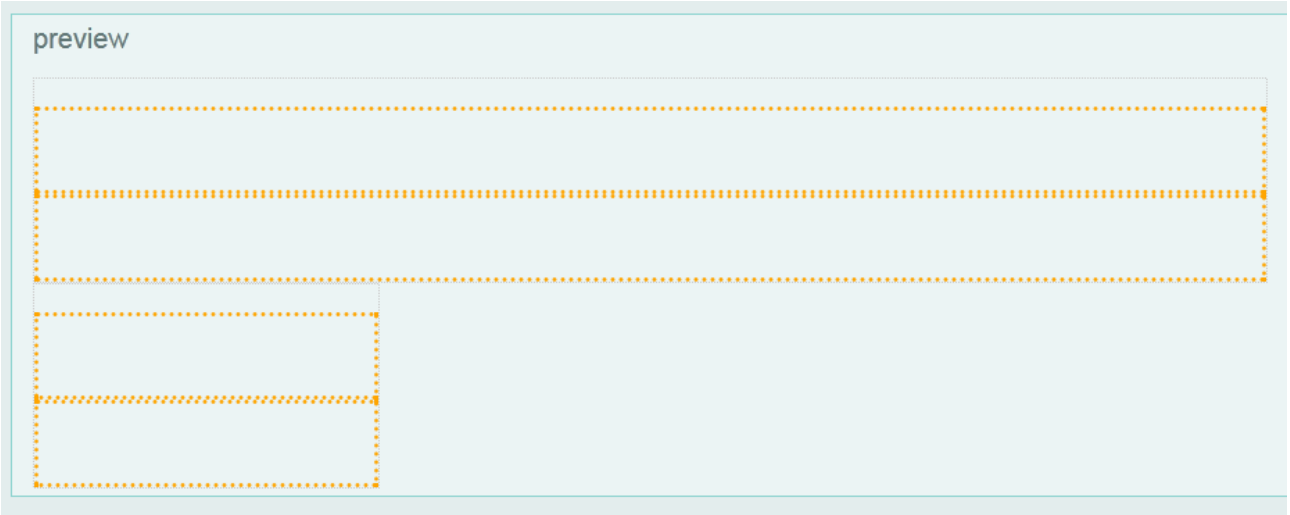
work done/estimated	time elapsed	descendant task closed/total
0%	100%	0%
team/all users	issues closed	block issues closed
43%	0%	0%

10.5 Templates

A template is mainly a .html grid where portlets can be inserted on.

Teamwork comes with some default templates pre-installed.

The template editor is similar to the portlets one; a template preview is available:



Sample code of a template:

```
<script>$("#HOME MENU").addClass('selected');</script>

<div class="mainColumnHome">
    <div areaname="LEFT" custom="yes"></div>
    <div areaname="LEFT_BOTTOM" custom="no"></div>
</div>
<div class="rightColumnHome" >
    <div areaname="RIGHT" custom="yes"></div>
    <div areaname="RIGHT_BOTTOM" custom="no"></div>
</div>
```

You can use any kind of container for dropping your portlets (TD and DIV in the example). Each area is identified by “areaname”, use it to assign a name, and “custom”; use “yes” if the standard user can customize this part, use “no” if only administrator can; this is the way to have mandatory-non-removable parts (e.g.: users cannot remove company news).

A requirement for template’s areas is not to be nested. E.g.:

```
<div areaname="VALID" custom="yes"><div areaname="INVALID" custom="yes"></div></div>
```

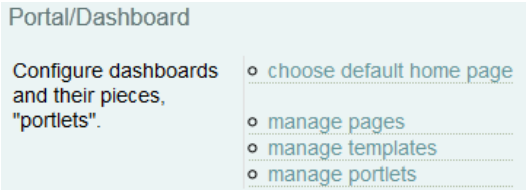
is not a valid template.

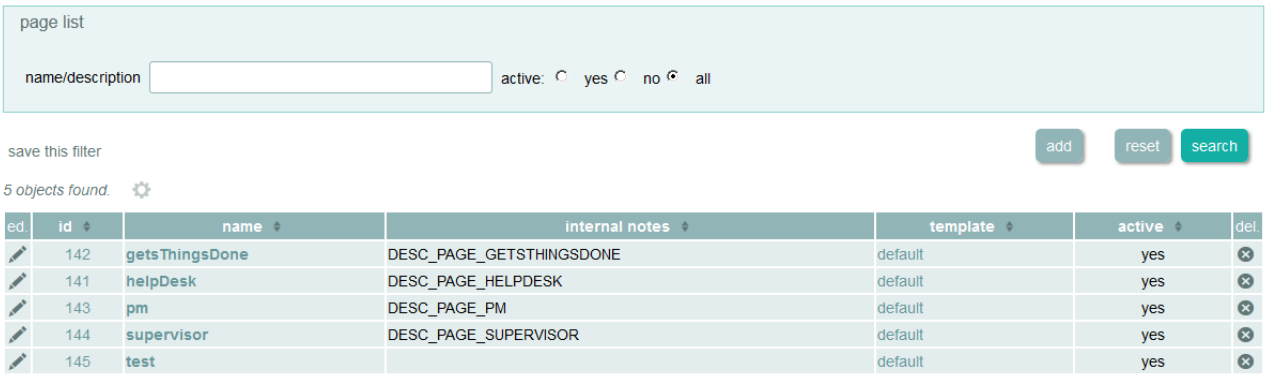
Every HTML put outside an area will be left untouched, and reported as-is in each page built on the template.

10.6 Pages

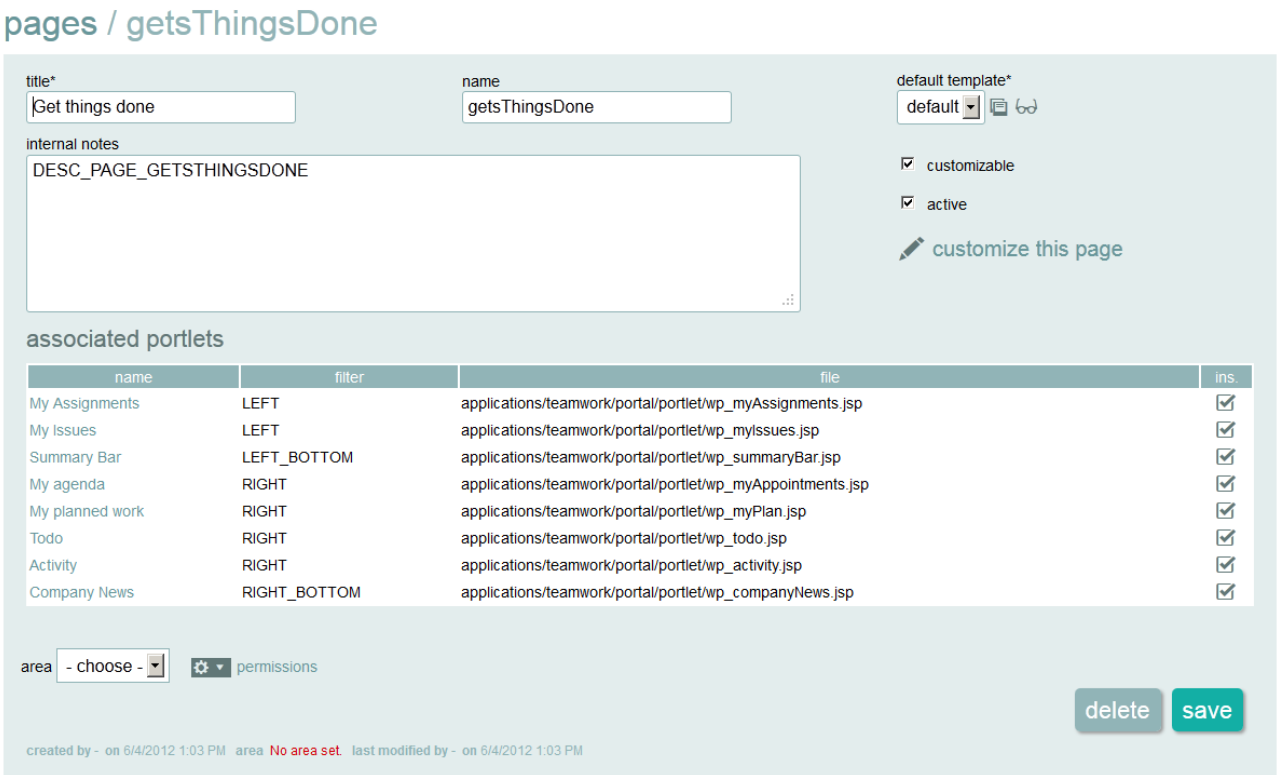
A page is defined as a template with some contents (portlets).

From the administration page click on manage pages:





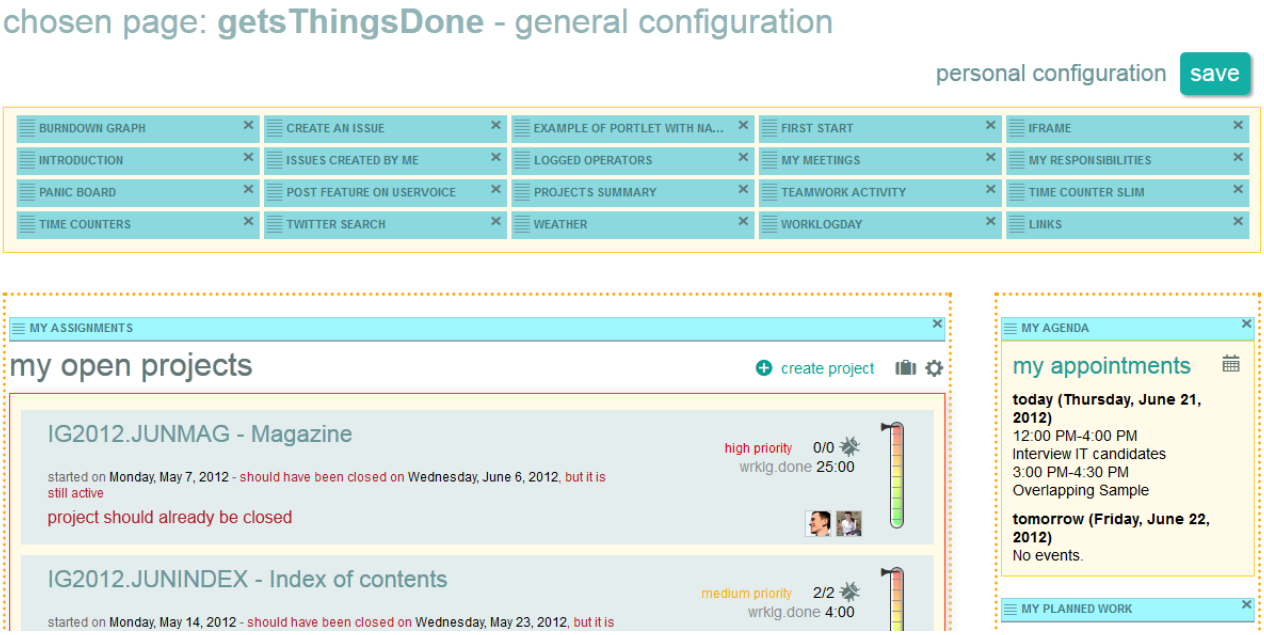
Then edit a page:



Define a title, a name, the template.

> The name is used for composing the URL, the title is what will be shown in user interface.

You can mark a page as customizable or not, active. You can set permissions on the page, and then insert portlets as shown above (see “10.3 Dashboard Customization”). The of course the fun starts by selecting “customize this page”, with drag’n’drop configuration:



General configuration

This is the base configuration that all users will see before customizing their own view (eventually).

Personal configuration

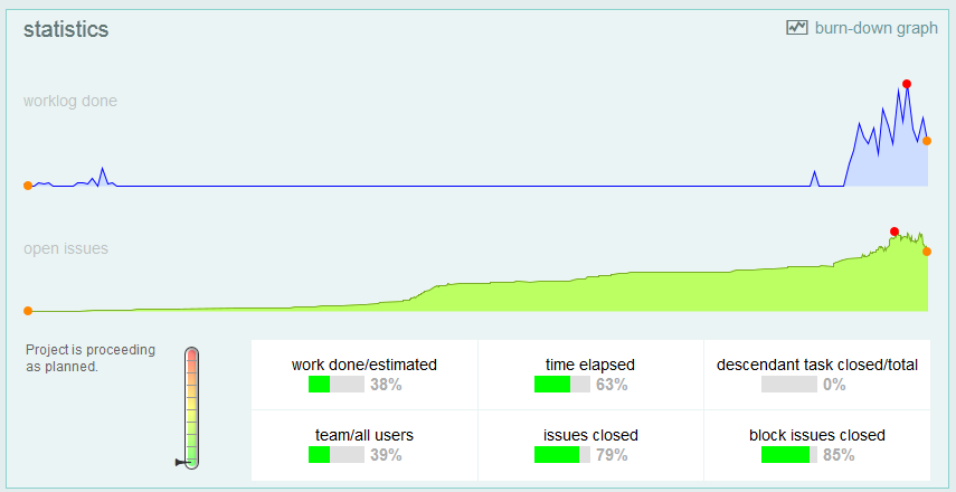
If you click on “personal configuration” you will edit the page only for yourself, and you get more options:

Which are just ways to get back to other options; “reset defaults” remove the customizations you did on top of the general configuration.

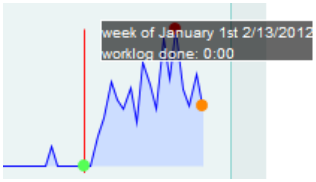
11. Agile / SCRUM

This theme is discussed in *ix Agile methods, SCRUM and Kanban*. Here we see how to get some graphical representations of the task progress in time.

Since version 5 we have statistics directly in the task editor



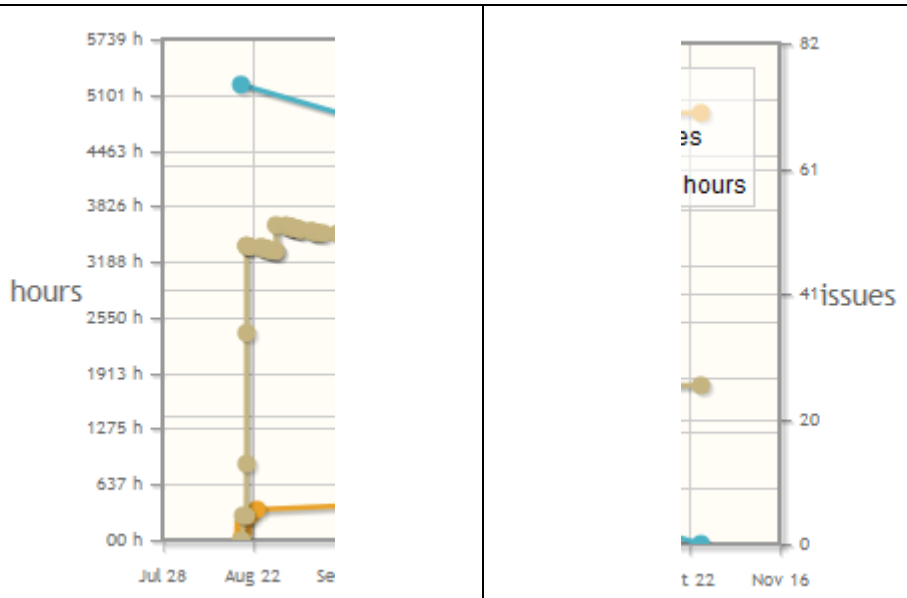
Mouse over the stats to get more info.



You access the burn down graph by selecting “burn-down graph” here:

assignee	done/assigned	% on assigned	% on total done
Pietro Polsinelli O'Hara	228:00/500:00	45,6%	22%
Giuseppe Panzarella	25:00/40:00	62,5%	2,41%
Laura Mirri	215:30/300:00	71,83%	20,8%
Matteo Bicocchi	6:00/500:00	1,2%	0,58%
Roberto Bicchierai	514:48/501:00	102,75%	49,68%
Silvia Chelazzi	47:00/107:00	43,93%	4,54%

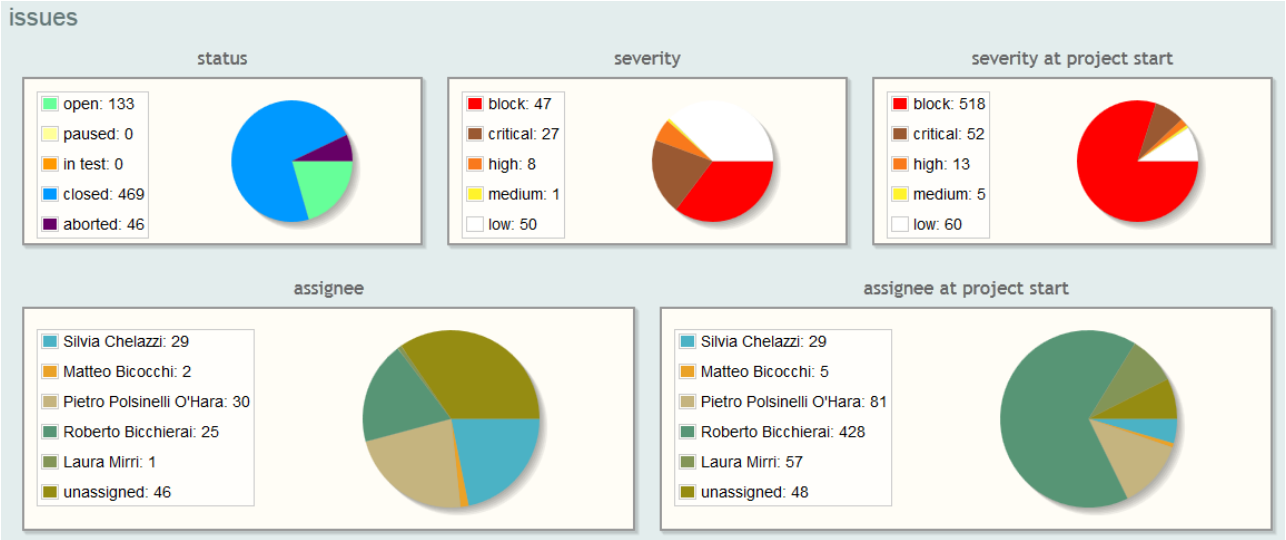
The graph generated is useful if estimation and worklogs are being inserted during the task life.



Notice that there are two scales, on the left and the right of the graph, one for the work hours, one for the open issues.

> Burn down graph is not supported in Internet Explorer – for this functionality use another browser.

And on the lower part of the page, we get several pie charts representing the current state of issues and how it was at project start.



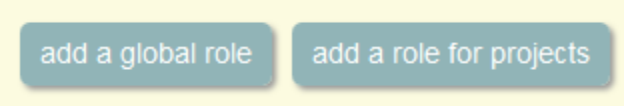
12. Security

12.1 Introduction

Teamwork integrates a really fine grained security model without bothering too much neither the user nor the administrator for setting it up.

In order to understand Teamwork's security, there are some key points that we will explain in this chapter.

First of all, Teamwork's security is *role based*; having a role means gaining permission for performing certain operations, for example crating task, inserting worklog, reading resources. We call this ability "permission"; a role is a collection of permissions.



There are two kinds of roles, "local" and "global". Local roles have the scope of a project: this means that permissions work on the project where the role is set (through an assignment). Local roles are assigned to resources during the assignment phase; so when you assign a resource on a task as, for instance, project manager (that is a *local* role), you are giving the resource the set of permissions associated to the PM role.

In this way you will create a really fine grained security structure, but with some limitations: setting local permission will not allow, for instance, a supervisor to read every data of your project without assign her/him on every task, which would be a waste of time.

In order to solve this kind of problems Teamwork supports also "global" roles. A global role is a set of permissions that is directly associated to a resource, not through the mediation of an assignment. So if a user has a global role with "task read" permission, she will read every task, bypassing assignments.

This model is really refined and works well in most cases, but Teamwork goes beyond that, and introduces a more sophisticated object called "area". An area is a sort of "sandbox", and almost all Teamwork' objects belong to one and exactly one area. Objects from different areas cannot "see" each other (with few exceptions), so for instance if you have two areas, "production" and "accounting", you may have distinct, separate projects, roles, task types, etc. .

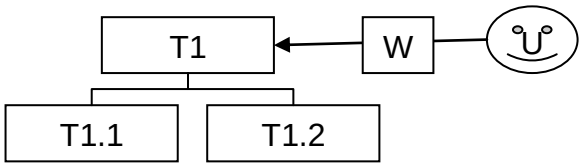
Obviously having two *completely* separated areas may also be a problem, say for a single company, where probably some users should be cross-area. Teamwork supports also this kind of solution, by allowing having on the same user global roles and assignments from different areas.

Another interesting feature is security management delegation: in each area you may have a sort of sub-administrator, the "area manager", that is responsible of new user creation and area administration.

Setting up this kind of environment is simple but not trivial, we warmly suggest to avoid multi-area management until you have really understood Teamwork' security model.

Last point is how security works for tree-structured object (like task or resources); well by default security is *propagated* so if you have a permission on a task, you have the same permission on each descendant. This is the default behavior, but this setting is local to the node, so for instance Scrum based projects may have a different configuration (on Scrum a customer can add issues on the backlog, but cannot interfere with sprints, so permissions are not to be propagated in that case).

Summing up how security works, we will examine an example of how Teamwork answers this question: can user U add an issue on subtask T1.1. Here the structure of the example:



The resource U is assigned on T1 with local role W(orker) that contains some permissions like task read, issue add/read/modify, and others. U has no global roles.

This is the flow followed by Teamwork in checking security, when a check is true the testing stops, otherwise the following clause is checked:

1) Is the user owner of the task T1.1?
2) Is the user an administrator?
3) Has the user a global role in the same area of the task T1.1 containing the “add issue” permission?
4) Is U assigned to T1.1 with a role containing “add issue” permission?
5) Finally check if parent (T1) propagates permissions and child (T1.1) inherits. The answer is “yes” by default so it will check steps 1-4 with T1.1 parent T1.

Making this kind of test faster has been a really challenging task.

Security editors are really simple with respect to the security model ☺.

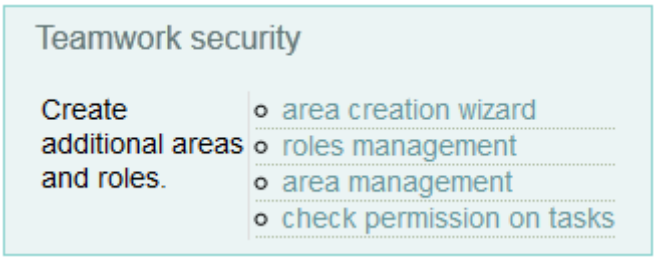
12.2 Areas

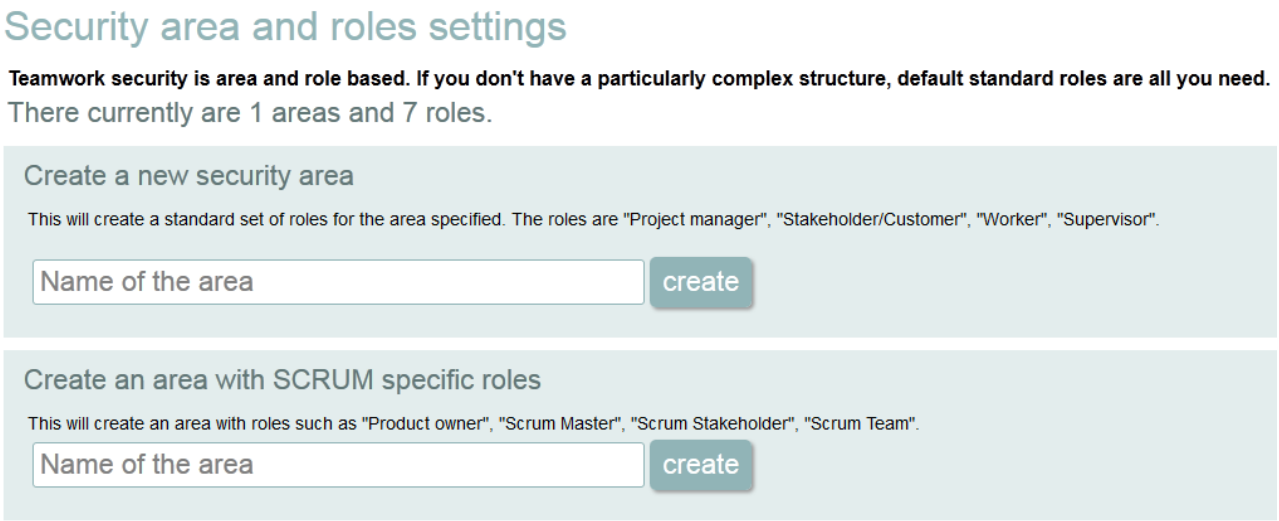
The installer creates a default area, and normally you should not need any more.

Think twice before creating a second area: things may quickly become horribly complex.

But for advanced usage, in order to create a new area go to the “admin” page, focus the security box.

Here you can use the “area creation wizard” that creates a new area and standard roles on it:





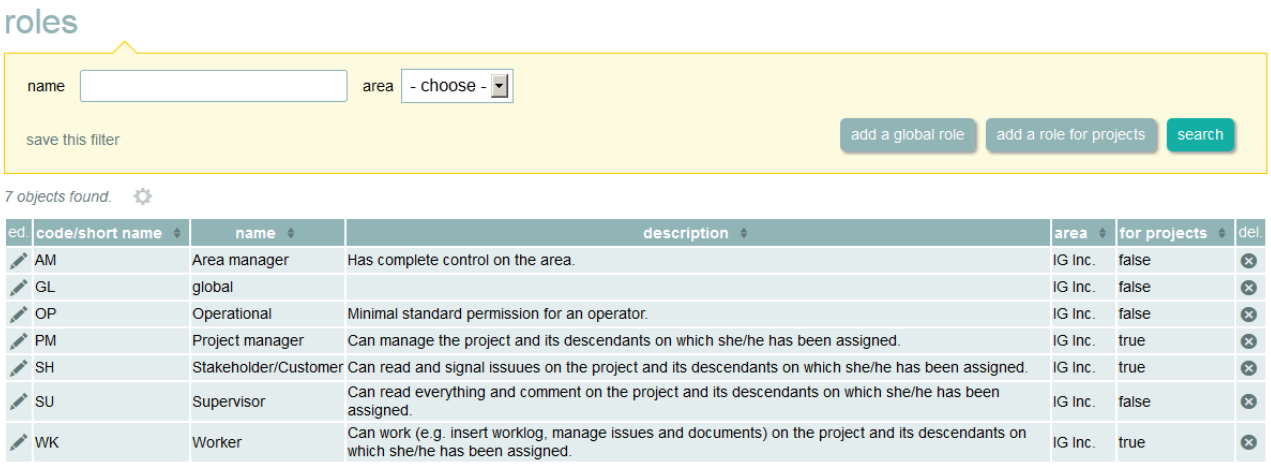
By going on “area management” you will have the standard find-and-edit pages.

The area creation wizard also supports creation of Scrum based roles.

12.3 Roles

Teamwork has built-in default roles, but you can create your own in order to model more closely your business model.

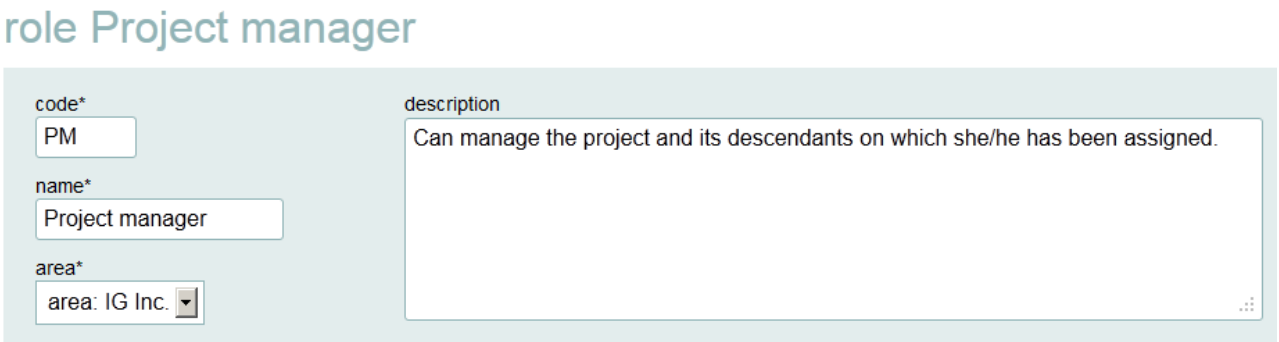
From the “admin” page go to “roles management”:



ed.	code/short name	name	description	area	for projects	del.
	AM	Area manager	Has complete control on the area.	IG Inc.	false	
	GL	global		IG Inc.	false	
	OP	Operational	Minimal standard permission for an operator.	IG Inc.	false	
	PM	Project manager	Can manage the project and its descendants on which she/he has been assigned.	IG Inc.	true	
	SH	Stakeholder/Customer	Can read and signal issuues on the project and its descendants on which she/he has been assigned.	IG Inc.	true	
	SU	Supervisor	Can read everything and comment on the project and its descendants on which she/he has been assigned.	IG Inc.	false	
	WK	Worker	Can work (e.g. insert worklog, manage issues and documents) on the project and its descendants on which she/he has been assigned.	IG Inc.	true	

As you can see there are some role “for projects” (local ones) and some not (global ones).

Go to edit a local role, say Project Manager:



The “role local” check reflects the fact that PM is a role for projects, you may be PM on

some project but not in others.

Go to the “permission” part below:

Permissions for projects

name		description
resource: read	☑	Allows to read resources.
task: read	☑	Allows to read tasks.
task: create	☑	Allows to create tasks.
task: write	☑	Allows to write tasks.
task: can assign	☑	Allows to create assignments on tasks.
worklog: manage	☐	Allows to manage worklog, approve it, bill it.
task cost: read	☑	Allows to read task costs.
task cost: create	☑	Allow to creat task costs.
task cost: write	☑	Allows to write task costs.
issue: read	☑	Allows to read issues.
issue: write	☑	Allows to write issues.
issue: create	☑	Allows to create issues.
document: read	☑	Allows to read documents.
document: write	☑	Allows to write documents.
document: create	☑	Allows to create documents.
file storage explorer: read	☑	Allows to read files within a file storage
file storage explorer: write	☑	Allows to write files within a file storage
file storage explorer: create folder	☑	Allows to create folder within a file storage

These are permissions that make sense in project, locally.

If you edit a global role, the set of permissions is wider:

name		description
area: can manage	☐	Allows to manage all object within the area.
roles: create	☐	Allows to create roles.
roles: read	☐	Allows to read roles.
roles: write	☐	Allows to write roles.
resource: read	☐	Allows to read resources.
resource: write	☐	Allows to write resources.
resource: create	☐	Allows to create resources.
resource: read costs	☐	Allows to read resource' costs
resource: assign all	☐	Allows to manage and monitor in tasks and work all resources of an area.
worklog: manage	☐	Allows to manage worklog, approve it, bill it.
task: read	☐	Allows to read tasks.
task: create	☐	Allows to create tasks.
task: write	☐	Allows to write tasks.
task cost: read	☐	Allows to read task costs.
task cost: create	☐	Allow to creat task costs.
task cost: write	☐	Allows to write task costs.
issue: read	☐	Allows to read issues.
issue: write	☐	Allows to write issues.
issue: create	☐	Allows to create issues.
document: read	☐	Allows to read documents.
document: write	☐	Allows to write documents.
document: create	☐	Allows to create documents.
boards: read	☐	Allows to read boards.
boards: write	☐	Allows to write boards.
boards: create	☐	Allows to create board.
file storage: read	☐	Allows to read file storage.
file storage: write	☐	Allows to write file storage.
file storage: create	☐	Allows to create file storage.
file storage explorer: read	☐	Allows to read files within a file storage
file storage explorer: write	☐	Allows to write files within a file storage
file storage explorer: create folder	☐	Allows to create folder within a file storage
classification tree: manage	☐	Allows to manage classification trees.

12.4 Double area example

If you want to setup the double (or n>1) area environment described in the introduction follow these instructions:

1) Supposing to have two users U1 and U2
2) Login as administrator
3) Create area A1 with the wizard
4) Create area A2 with the wizard
5) Edit U1, assign it to area A1 and give it the role Area Manager A1
6) Edit U2, assign it to area A2 and give it the role Area Manager A2

If you stop now area A1 and A2 are completely separated, U1 and U2 can create new resources in their areas respectively, so security management is almost completely delegated.

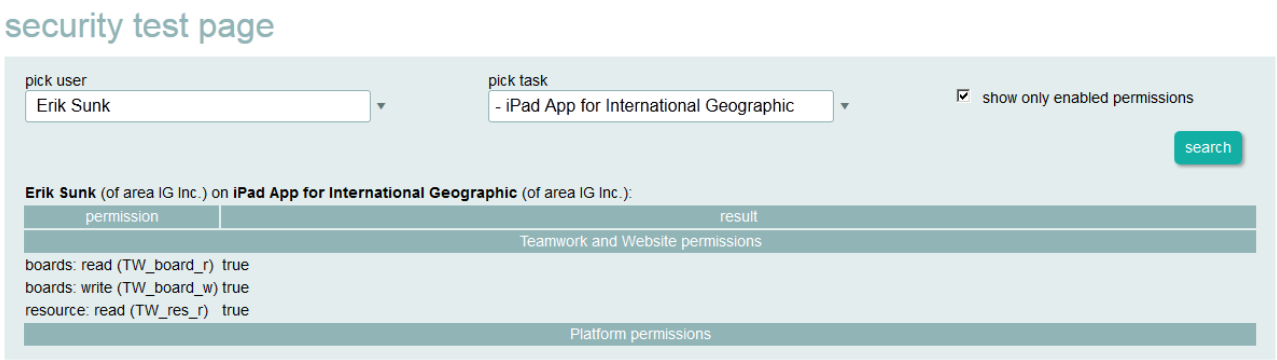
If you want to have some users with cross-area rights, you (administrator) can give them roles in both areas, or if you want to delegate you can give U1 the role of area manager even on area A2. In latter case U1 is the manager of both areas.

Note that permissions given locally go beyond area restrictions, so if you have a task T1 on area A1, you can assign a resource (U3) from area A2 with role “Project manager A2”. In order to set-up this task you must have “task create” on A1 and “resource assign all” on A2.
In this case U3 will operate on T1 without restrictions, but in general U3 doesn’t see any task in A2 except T1.

Considering that you can change roles or create new ones, Teamwork lets you have a really flexible security environment.

12.5 Check security

In order to check permission of someone on a project there is a tool (from admin page, go to “Teamwork security” section and “check permission on tasks”:



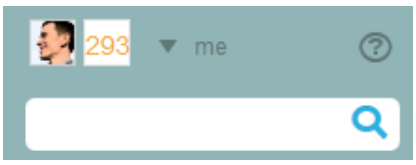
Pick a user, a task, press search and look at permissions.

13. Tips & Tricks

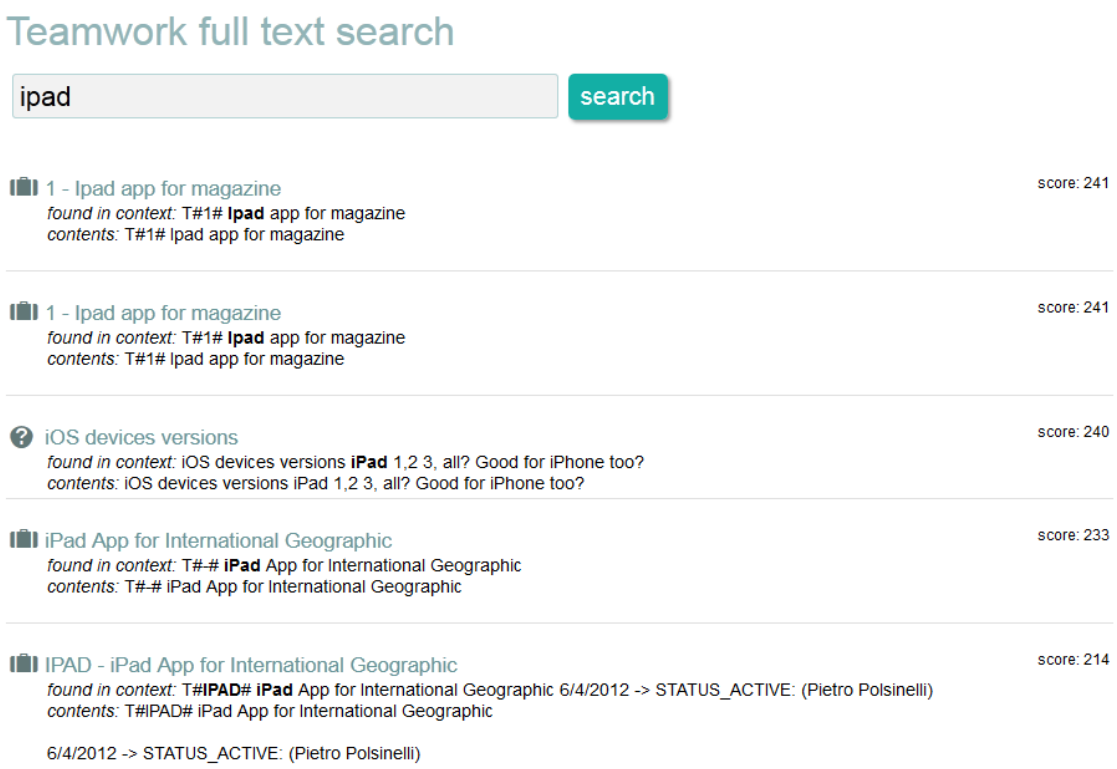
An application success is directly proportional to user's satisfaction (Metcalfe's law en.wikipedia.org). This is particularly true in the case of a work management application: if users don't log in, check information and read and insert required data, it quickly becomes useless. Teamwork has been designed to fascinate and confuse users… no we meant: Teamwork has been designed to facilitate user interaction and offer tools and tricks for the power user; this is not trivial in the case of a web application. Several examples of such techniques are listed here.

13.1 Search anywhere from everywhere

The search box in Teamwork's menu will search every text inserted on Teamwork's data. The results will be linked to Teamwork entities. Results are weighted by "group rank":



Group rank is a combination of hit ranks and text matching.



If you want to restrict search on a particular object such as task or issue or resource, you can use prefixed searches:

t: searches in tasks e.g. t:develop
r: searches in resources
i: in issues
d: in documents
b: in boards
w: in worklogs
a: in agenda

These latter searches will not use full-text search, but database search.

13.2 Custom filters and QBE

Teamwork use empowered filtering. Every list/find page uses QBE to make searches flexible and powerful.

13.2.1 QBE

The query by example (QBE) search method gives the user an easy way to compose complex queries, by using a particular syntax in the search fields. If for example in a field you write "$mixer*" and click search, you will get all results that start with "mixer", but not those that have “mixer” as a middle occurrence.

Teamwork by default is case insensitive in searches, even on case sensitive databases; this can be changed by the administrators.

If values are specified in more than one field, all these must be satisfied (fields are in "AND"). Fields where QBE is enabled are usually easily identifiable:

Queries can be composed with the following parameters:

Condition	Operator	Example
comparison	=,>,<,>=,<=	All tasks whose progress is above 5%: progress: >5 % More complex: above 3, lesser than 6 or equal to 10: unitary price: >3+<6|=10
not equal	!	All tasks whose description does not contain "jewel": name/description !jewel
exactly equal	""	All tasks whose description is "nice jewel", excluding those that have only "nice" or only "jewel": name/description "jewel"
empty field	(), []	All tasks whose description and name are "null": name/description ()
zero length field	//	All tasks whose description and name have empty content: name/description //
not empty field	!(), ![]	All tasks whose description and name are not empty: name/description !()
contains	*	All tasks whose description or name end with "spies": name/description *spies All tasks whose description or name start with "Contact": name/description Contact* All tasks whose description or name contain "an": name/description *an*
isolated word	#	All tasks whose description or name contain "jewel" as isolated word: name/description #jewel Will find "Another jewel robbery" but not "Jewellery stolen"
is between	:	All tasks whose start is between the dates below: start 10/08/2005:20/12/2006
conjunction	+	All tasks whose description or name contain "another" AND contain "jewel": name/description another+jewel
disjunction	+	All tasks whose description or name contain "another" OR contain "jewel": name/description another|jewel or name/description another,jewel

Dates are always completed with time, so when you are filtering for a date, QBE applies some conversions: in case of a single date e.g. “1/1/2009” the engine will search for dates >= 1/1/2009 00:00:00:000 and <= 1/1/2009 23:59:59:999. The same in case of an interval e.g.: 1/1/2009:2/1/2009. The engine will search for dates >= 1/1/2009 00:00:00:000 and <= 2/1/2009 23:59:59:999.

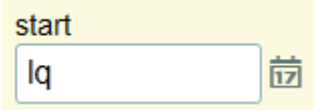
13.2.2 Date shortcuts

Date fields allow users to quickly insert dates in the correct format. You can move between months by using the arrow buttons, but more interestingly, you can use some shortcuts instead of inserting a complete date.

For instance you can insert “today” (or even “t”) instead of inserting the actual date. What is really interesting is the use of such shortcuts in custom filters. In fact using QBE in dates allows you to express something like “all those whose end date <t” intending that *end date* is in the past. There are several shortcuts:

Shortcut	“Longcut”	Description
N	NOW	Is this instant including milliseconds
T	TODAY	Is the first millisecond of today (00:00:00:001)
Y	YESTERDAY	Is the first millisecond of yesterday
TM	TOMORROW	Is the first millisecond of tomorrow
W	WEEK,THISWEEK, WEEKSTART, THISWEEKSTART	Is the first millisecond of the first day of this week
LW	LASTWEEK, LASTWEEKSTART	Is the first millisecond of the first day of last week
NW	NEXTWEEK NEXTWEEKSTART	Is the first millisecond of the first day of next week
M	MONTH,THISMONTH, MONTHSTART, THISMONTHSTART	Is the first millisecond of the first day of this month
LM	LASTMONTH, LASTMONTHSTART	Is the first millisecond of the first day of last month
NM	NEXTMONTH, NEXTMONTHSTART	Is the first millisecond of the first day of next month
Q	QUARTER THISQUARTER QUARTERSTART, THISQUARTERSTART	Is the first millisecond of the first day of this quarter
LQ	LASTQUARTER, LASTQUARTERSTART	Is the first millisecond of the first day of last quarter
NQ	NEXTQUARTER, NEXTQUARTERSTART	Is the first millisecond of the first day of next quarter
CW*n*	WEEKOFYEAR, WY	It is the calendar week number start day, e.g. CW2 is the first day of week (and this depends whether American or European format, Sunday of Monday) of the second week of the current year

Note that every shortcut is always reduced to a date, not to a period even if the name suggests a period (for instance “week”) when used in form data entry; it becomes a period when used in search pages



You can do even more: you can also use in dates fields every expression matching

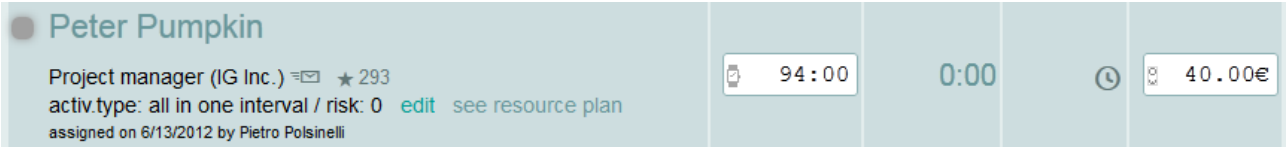
"^-?[0-9]+[DWMY]$".

This is a regular expression syntax, and to explain it a bit more clearly, some examples may help:

Example	Meaning
2D	in 2 days
-3D	3 days ago
12W	in 12 weeks
-5W	5 weeks ago
2M	In 2 months
-35Y	35 years ago

This can even be used to insert time intervals when recording work, such as in assignment list, for setting days:

Becomes:

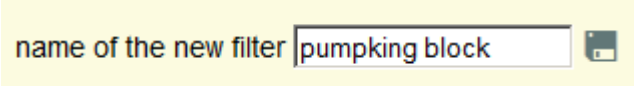


Or even in issue list inserting time (and wherever you insert time):

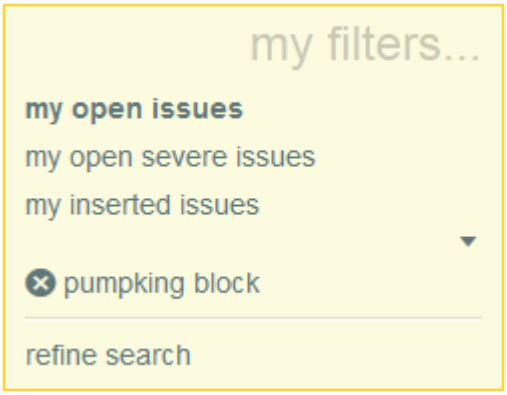
13.2.3 Custom filters

It would be horrible if you could compose such refined filters, and you'd have to retype them every time. Fortunately, you can save filters on your profile.

In every Teamwork search page you can save your filter: click save this filter and you get



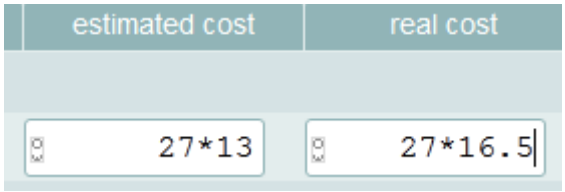
And the filter will always be available on the my filters side box.



You can decide to use a filter as your default one. In this case prefix the name with "d:".

13.3 Compute in fields

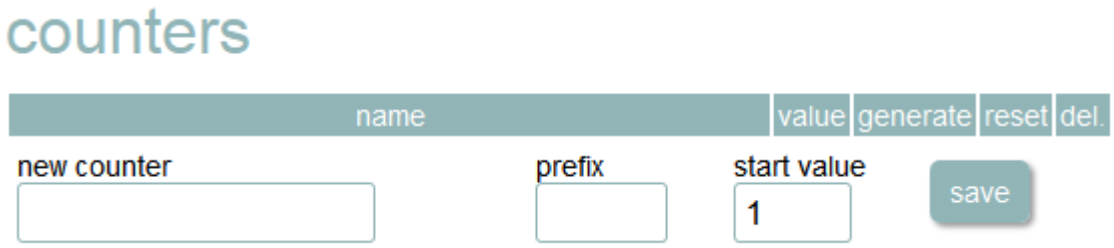
You can use arithmetic expressions in numeric fields: the result of the computation will be saved like any other numeric value:



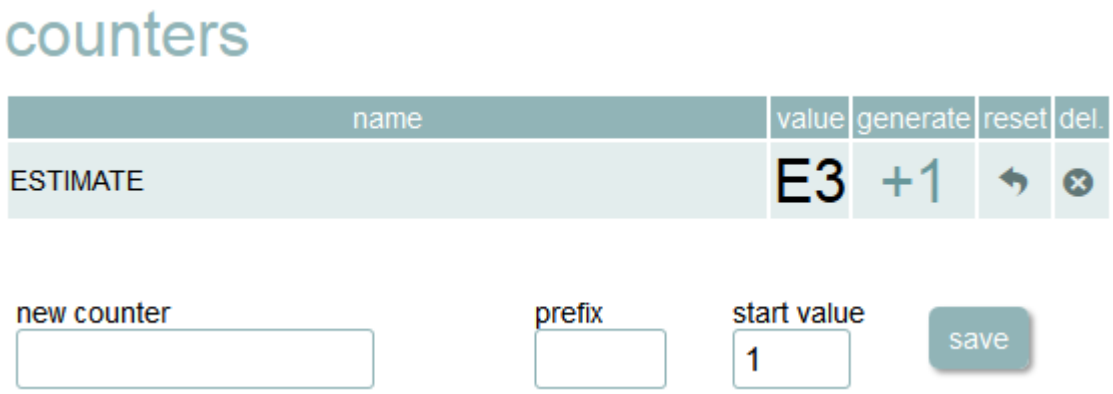
13.4 Counters

Which number should I give to the next document/task? An easy protocol numbering system is included. You don't need to keep counters in Microsoft Excel or in mind ☺

Go to docs&tools -> counters.



Then on counters:



Every time you press on +1 you will get a new unique code.

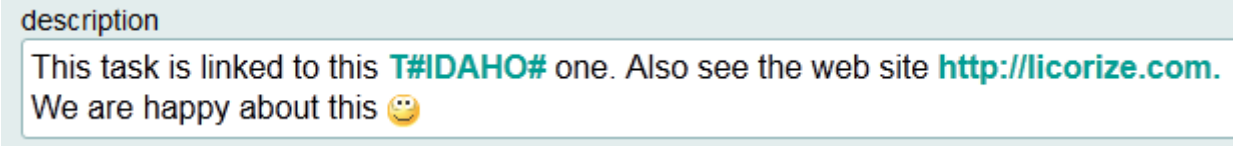
13.5 Internal links

Since Teamwork 4.5, there is a way to refer and link to any "entity" from any descriptive field. For example, you can link any task, resource or issue from any event in the agenda. You can link any other task from a project description or deliverable. Same from the issue description and so on.

Notice in this top part of the task editor on the left that there a task internal "code", which every saved task has: T#IDAHO#. This code allows you to link to this task editor from every other editor in Teamwork. For example, if on any (other) task editor in the "description" or "deliverables" fields you write such code, this will turn into a link. In this way you can cross link any task to any other, and more – see below.

Cross linking supports links between: tasks (T#CODE#), issues (I#CODE #), resources (R#CODE #), agenda events (E#CODE #), agenda meetings (M#CODE #), (B#ID#) boards, where the value of "CODE" is shown in the editors.

Actually the text in the text area supports also HTTP links, images and smileys: you can get a description like this one:



with several active links.

13.6 Remove all subscriptions plus defaults on roles

We've received complaints from users that receive "too many" notifications. It all happens because of the subscription engine, and it can be all fine-tuned, globally or on a single user basis.

By default roles have several subscriptions active: you can turn them off from the role editor.

If you have subscribed too many tasks, you can make a “clean slate” by going to your options, click subscriptions:

13.7 Smart URLs

Teamwork supports smart URLs like:

Tasks:
[yourserver]/t/[mytaskcode]
or
[yourserver]/task/[mytaskcode]

So if your server is called “teamwork”, you can get a task of code “PROD66” with the URL:

http://teamwork/t/PROD66

In the same way, this syntax has been enabled for issues, resources, events, meeting and boards:

Issues:
[yourserver]/i/[myIssueCode]
or
[yourserver]/issue/[myIssueCode]

Resources:
[yourserver]/r/[myResourceCode]
or
[yourserver]/resource/[myResourceCode]

Agenda events:
[yourserver]/e/[myEventCode]

or
[yourserver]/event/[myEventCode]

Meetings:
[yourserver]/m/[myMeetingCode]
or
[yourserver]/meeting/[myMeetingCode]

Boards:
[yourserver]/b/[myBoardCode]
or
[yourserver]/board/[myBoardCode]

14. Plugins, custom fields/forms, customizations

Parts of this section require some programming competence.

Teamwork allows the creation of complete custom fields/forms/plugins, which are automatically linked to projects, task or resources, according to the objects features.

The idea behind custom forms, also called plugins, is that you can add an entire form associated to families of tasks (or resources, or even generically present in Teamwork's menus, depending on context and user rights) just by creating a single, self standing jsp file: no new class compilation, database schema creation, or transaction handling is necessary, even if you define new fields to be saved. Of course, if you also want to create supporting classes, or add jars to the classpath, you are free to do so.

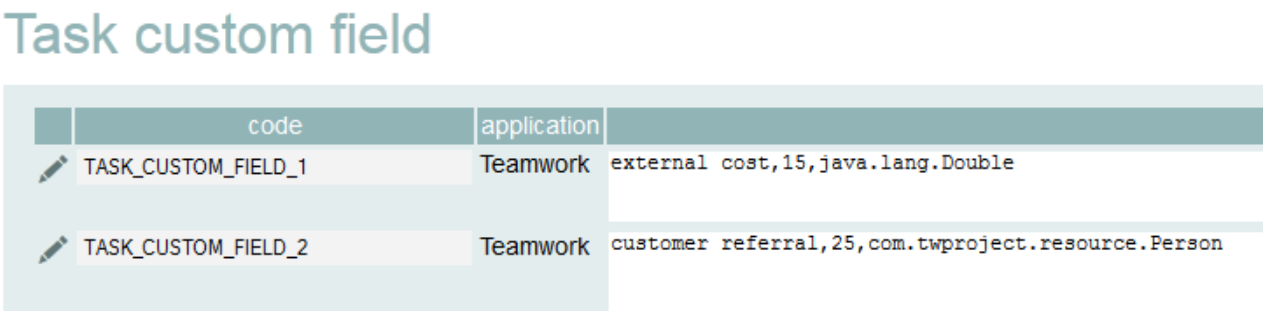
Custom forms are usually visible in the document section of tasks and resources editors: forms are used to extend properties of Teamwork's objects. Plugins are generally intended for automating actions (e.g. wizards) or for extending reporting capability.

Teamwork also provides the possibility of adding "custom wizards", which will be accessed by having an additional button on the menus and creating say a project from a set of templates. See below for more.

14.1 Task custom fields

In order to supply a quick solution for adding a field to the task editor, Teamwork supports custom fields.

In order to activate them you just need insert a value for specific labels. To activate them, go to admin -> Customization -> on tasks, you'll get to a dedicated page:

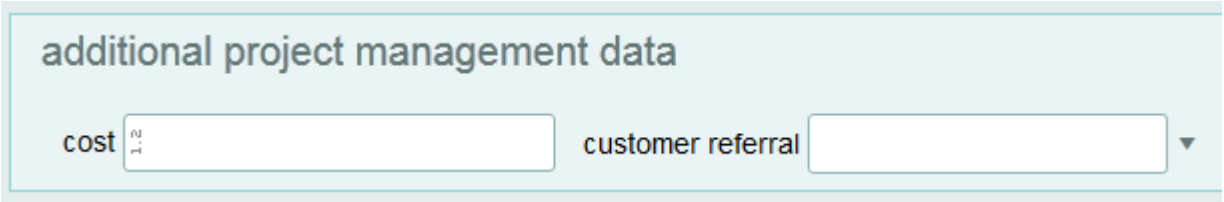


On such page you also have detailed examples and a technical reference, which we do not duplicate here.

Custom Field examples:

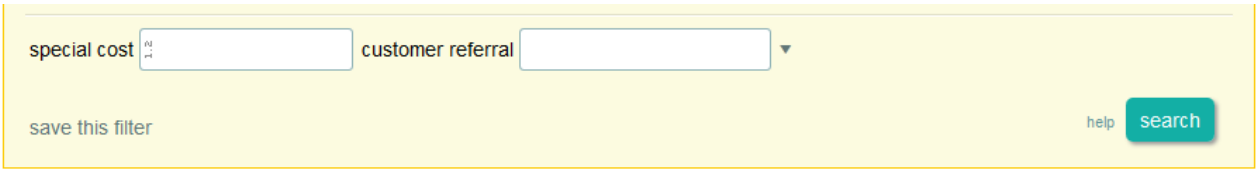
- Unused field: `no` -> the field is unactive
- Simplest case: `my code` -> a string
- Lenght defined case: `my code,20` -> a string 20 chars lenght
- Double case: `cost,15,java.lang.Double` (also Integer or Long)
- Date case: `when,10,java.util.Date`
- Teamwork object case: `customer referral,25,com.twproject.resource.Person` -> a Person
- `see this task,25,com.twproject.task.Task` -> a Task
- `attachment,15,org.jblooming.ontology.PersistentFile` -> an uploaded file

Adding the example values shown will add a section on the task



Notice that field input validation is active.

You can also search / filter by these fields in the task search page:



14.2 Custom forms usage

Actually in Teamwork's standard installation you will already have some sample custom forms on tasks and resources. These by default are not visible unless some conditions are satisfied:

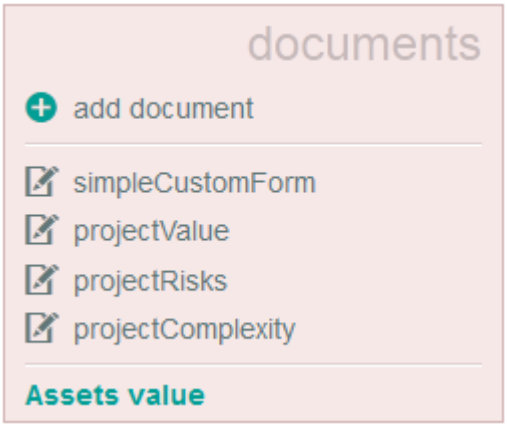
Simple Custom Form: visible only when the task name is TESTFORM.

Project Complexity: visible only when the task is root and its relevance is >= 80.

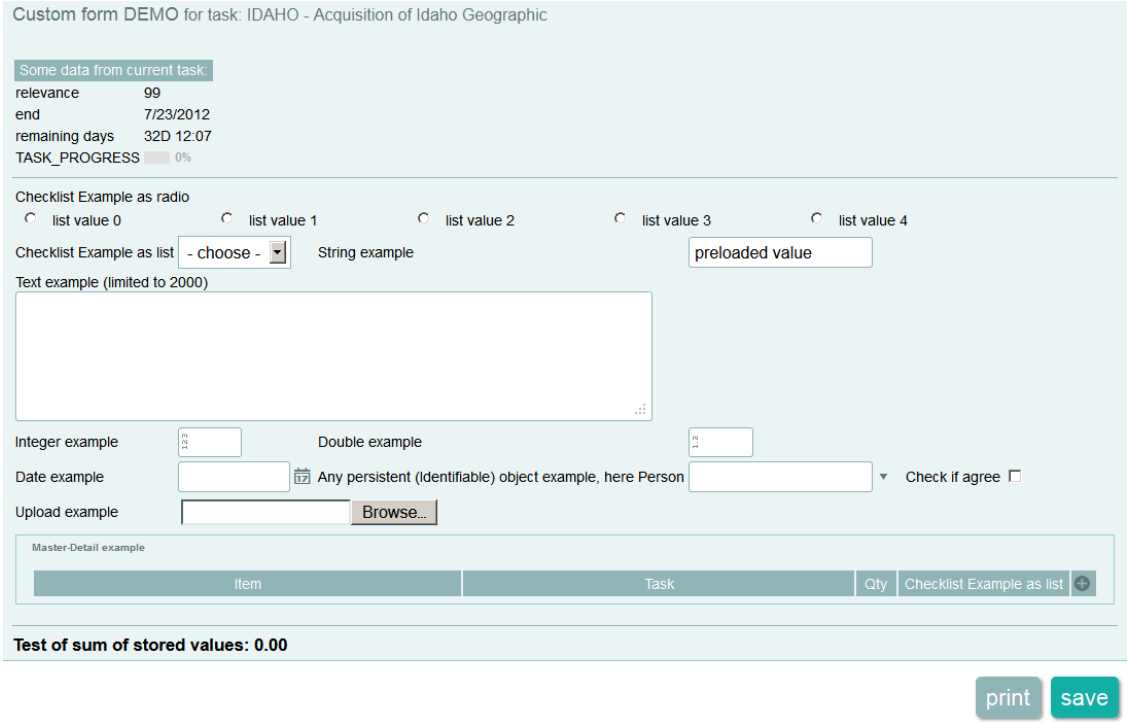
Project Value: visible only when the task is root and its relevance is >= 80.

Project Risk: visible only when the task is root and its relevance is >= 80 and the task type is PRODUCTION.

On tasks where these are enabled, just go to the "document" tab of task/resource' editor:



Click on "simple custom form":



This chaotic form is meant just as an example of the spectrum of fields that you can add on forms.

14.3 Create your own

This section is not for the faint of heart ☺: only those who know Java can benefit from this reading.

Custom forms/reports/plugins make sense only when “customized”. So in this section we will try to explain how they work and how to modify/create your own.

There are various examples forms provided; in order to start, use simpleCustomForm.jsp: it is extensively commented, and contains examples of the different fields (strings, dates, numbers, pointers to objects…) which may be used in a form. Copy it in a new file in the same folder, and start modifying it.

First of all, what makes custom forms practical is that they are “hassle free”. You can extend a task with tens of new properties without caring about saving/changing/removing data, which is done by the framework. The persistence layer is completely hidden by Teamwork.

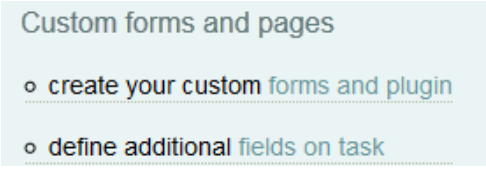
14.3.1 Where are custom forms

Custom forms are .jsp pages thatr by default are in the [root]/applications/teamwork/plugins folder.

To scan another folder for plugins, launch somewhere

```
PluginBricks.scanFolderAndInitializeQuarks(
 "[your folder path from [root]/applications/teamwork]",
 ApplicationState.platformConfiguration.getDefaultApplication(),
 pageContext);
```

In order to list active plugins go to admin page, then



press “forms and plugin”.

When Teamwork starts-up it scans that folder and initializes each plugin. You can force a new directory scanning by clicking on “reload plugins” button.

Plugin admin

name i18n	name	class	file	path
REPORTS				
RESOURCE_FORMS				
resourceTargets	resourceTargets	*delegated to impl*	resourceTargets.jsp	/applications/teamwork/plugins/resourceTargets.jsp
WIZARD				
TASK_FORMS				
simpleCustomForm	simpleCustomForm	*delegated to impl*	simpleCustomForm.jsp	/applications/teamwork/plugins/simpleCustomForm.jsp
projectValue	projectValue	*delegated to impl*	projectValue.jsp	/applications/teamwork/plugins/projectValue.jsp
projectRisks	projectRisks	*delegated to impl*	projectRisks.jsp	/applications/teamwork/plugins/projectRisks.jsp
projectComplexity	projectComplexity	*delegated to impl*	projectComplexity.jsp	/applications/teamwork/plugins/projectComplexity.jsp
WIZARDS				

reload plugins

14.3.2 How does it work

Load: At startup, Teamwork will try to call the initialize method on the jsp files found, and those that do not throw an exception are loaded in memory among the available plugins.

Visibility: A plugin can appear in the following locations: in Teamwork “tools” menu, on the task editor or on the resource editor. Whether they will appear there is entirely determined by the result of the call “isVisibleInThisContext” on the jsp page.

Persistence: Where does data get saved, and how? As a form can change any moment the type of fields present in it, its data cannot be subject to referential integrity. All data is saved in the tables olpl_des_data and olpl_des_data_value. There is nothing the developer needs to do to make data persistent: all fields present in the form will be saved, and automatically associated to the entity through which one has gone through to reach the form. So for example, if one is on a task, data written on the forms for that task will be saved in olpl_des_data_value, and linked to the task through a record in olpl_des_data: referenceId will be the id of the task, referenceClassName the task class, and designerName will be a normalized form of the jsp file name.

14.3.3 Plugin dissection

Ok, now starts the hard core….

When a plugin is initialized, it registers itself in a group, and injects an inner class extending PagePlugin, used to understand if the plugin should be visible in the current web context. Let’s have a look to the code (the example is from simpleCustomForm.jsp):

```
<%@ page import="com.twproject.resource.Person,
        … lots of import removed …
%><%@ page contentType="text/html; charset=utf-8" pageEncoding="UTF-8" %><%!
  /**
   * This inner class is used by Teamwork to know if this form applies to current context.
   * PagePlugin classes are loaded at startup (or by hand) in memory to be performant.
   *
   */
  public class PagePluginExt extends PagePlugin {
    public boolean isVisibleInThisContext(PageState pagestate) {
      boolean ret = false;
      if (pagestate.mainObject != null && pagestate.mainObject.getClass().equals(Task.class)) {
        Task task = (Task) pagestate.mainObject;
        // ----- begin test condition on task -----------------
        // this form will be visible only on root tasks
        ret = task.getParent() == null;
        // ----- end test condition on task -----------------
      }
      return ret;
    }
  }
```

```
%>
```

The jsp inner class must implement the isVisibleInThisContext() method.

This method based on data got from the PageState instance and mainly the “mainObject” field check if we are in the appropriate context.

In this case we are checking if the mainObject is a Task instance and if the task is a root one. If bothe condition are true the form will be visible in this context.

Each custom form is composed by two parts called in different application life-cycle. The first part is the initialization. This part is called at startup and injects PagePlugin instance in the system.

The PagePluginExt.isVisibleInThisContext method is called every time Teamwork is creating links for plugins for the group "TASK_FORMS".

```
<%
  /*
  */
  // ############################  BEGIN INITIALIZE
##############################################
  if (JspIncluder.INITIALIZE.equals(request.getParameter(Commands.COMMAND))) {
    PluginBricks.getPagePluginInstance("TASK FORMS", new PagePluginExt(), request);

    // ############################  END INITIALIZE
##############################################
```

Actually Teamwork uses four of groups: “REPORTS”, “RESOURCE_FORMS”, “TASK_FORMS”, “TASKLOG” that are displayed respectively in task/resource list/editor, resource documents, task documents, task log.

The second part is the definition of the form.

Definition is composed of two parts: form data definition and form html layout.

```
  } else if (Designer.DRAW_FORM.equals(request.getAttribute(JspIncluder.ACTION))) {
    // ------- recover page model and objects -----  BEGIN DO NOT MOFIFY --------------
    PageState pageState = PageState.getCurrentPageState(request);
    Task task = (Task) PersistenceHome.findByPrimaryKey(Task.class, pageState.mainObjectId);
    Designer designer = (Designer) JspIncluderSupport.getCurrentInstance(request);
    task.bricks.buildPassport(pageState);
    // ------- recover page model and objects -----  END DO NOT MOFIFY --------------
    // check security and set read_only modality
    designer.readOnly = !task.bricks.canWrite;
    // ################################  BEGIN FORM DATA DEFINITION  ##############################
    if (designer.fieldsConfig) {
```

you can have a selector as radio

```
      CodeValueList cvl = new CodeValueList();
      cvl.add("0", "list value 0");
      cvl.add("1", "list value 1");
      cvl.add("2", "list value 2");
      cvl.add("3", "list value 3");
      cvl.add("4", "list value 4");
      DesignerField dfr = new DesignerField(CodeValue.class.getName(), "RADIO",
               "Checklist Example as radio", false, false, null);
      dfr.separator = " ";
      dfr.cvl = cvl;
      dfr.displayAsCombo = false;
      designer.add(dfr);
      DesignerField dfl = new DesignerField(CodeValue.class.getName(), "COMBO",
                "Checklist Example as list", false, false, null);
      dfl.separator = "</td><td>";
      dfl.cvl = cvl;
      dfl.displayAsCombo = true;
      designer.add(dfl);
```

or as list

```
      DesignerField dfStr = new DesignerField(String.class.getName(), "STRING",
                        "String example", false, false, "preloaded value");
      dfStr.separator = "</td><td>";
      dfStr.fieldSize = 20;
      designer.add(dfStr);
```

standard text fields

```
      DesignerField dfNote = new DesignerField(String.class.getName(), "NOTES",
                       "Text example (limited to 2000)", false, false, "");
      dfNote.fieldSize = 80;
      dfNote.rowsLength = 5;
      dfNote.separator = "<br>";
      designer.add(dfNote);
```

text area

```
      DesignerField dfInt = new DesignerField(Double.class.getName(), "INTEGER",
                        "Integer example", false, false, "");
      dfInt.separator = "</td><td>";
      dfInt.fieldSize = 4;
      designer.add(dfInt);
      DesignerField dfdouble = new DesignerField(Double.class.getName(), "DOUBLE",
                        "Double example", false, false, "");
      dfdouble.separator = "</td><td>";
      dfdouble.fieldSize = 4;
      designer.add(dfdouble);
```

numeric fields

```
      DesignerField dfdate = new DesignerField(Date.class.getName(), "DATE",
                        "Date example", false, false, null);
      dfdate.separator = "</td><td>";
      designer.add(dfdate);
```

date

```
      DesignerField dffile = new DesignerField(PersistentFile.class.getName(), "FILE",
                        "Upload example", false, false, null);
      dffile.fieldSize = 40;
      dffile.separator = "</td><td colspan=3>";
      designer.add(dffile);
```

uploaded files

```
      DesignerField dfperson = new DesignerField(Person.class.getName(), "PERSON",
                     "Any persistent (Identifiable) object example, here Person", false, false,
null);
      dfperson.separator = "</td><td>";
      dfperson.fieldSize = 40;
      designer.add(dfperson);
```

lookup on other Teamwork's entities

```
      DesignerField dfbool = new DesignerField(Boolean.class.getName(), "BOOLEAN",
                     "Check if agree", false, false, "");
      designer.add(dfbool);
```

boolean

```
      // Master Detail example. You can add a detail to the form and then add field to detail.
      Detail detail = designer.addDetail("DETAIL");
      detail.label = "Master-Detail example";
      DesignerField dfitem = new DesignerField(String.class.getName(), "ITEM",
                          "Item", false, false, "");
      dfitem.fieldSize=55;
      detail.add(dfitem);

      DesignerField dfqty = new DesignerField(Integer.class.getName(), "QTY",
                          "Qty", false, false, "");
      dfqty.fieldSize = 4;
      detail.add(dfqty);
```

even master detail sections

```
      // ############################  END FORM DATA DEFINITION
####################################

    } else {
```

Once you have declared the field you intend to use, you must define it in the html layout of

the page.

```
      // ###########################  BEGIN FORM LAYOUT DEFINITION
##################################

      // create a container around the form
      Container c = new Container(pageState);
      c.title = "<big>Custom form DEMO</big> for task: " + task.getDisplayName();
      c.start(pageContext);
```

we create a container around the form

```
      // you can extract data to enrich your form using data from current task.
      // In this case we will extract missing days from current task
      String daysMissing = pageState.getI18n("UNSPECIFIED");
      if (task.getSchedule() != null && task.getSchedule().getEndDate() != null) {
        if (task.getSchedule().getValidityEndTime() > new Date().getTime()) {
          long missing = task.getSchedule().getValidityEndTime() - new Date().getTime();
          daysMissing = DateUtilities.getMillisInDaysHoursMinutes(missing);
        } else
          daysMissing = "<b>" + pageState.getI18n("OVERDUE") + "</b>";
      }
%>

  <%-- ---------------------- BEGIN TASK DATA ----------------------
       ---------------------- You can use the task recovered before to display cue data   --%>
    <br>
    <table border="0">
      <tr>
        <th colspan="2"> Some data from current task:</th>
      </tr>
      <tr>
        <td ><%=pageState.getI18n("RELEVANCE")%></td><td> <%=task.getRelevance()%></td>

      </tr><tr>
        <td> <%=pageState.getI18n("TASK_END")%></td>
        <td> <%=task.getSchedule() != null &&
                task.getSchedule().getEndDate() != null ?
                JSP.w(task.getSchedule().getEndDate()) : " - "%></td>

      </tr><tr>
        <td> <%=pageState.getI18n("TASK_REMAINING")%></td>
        <td> <%=daysMissing%></td>
      </tr><tr>
        <td> <%=pageState.getI18n("TASK PROGRESS")%></td><td>
          <%
            PercentileDisplay pd = TaskBricks.getProgressBarForTask(task, pageState);
            pd.toHtml(pageContext);
          %>
        </td>
      </tr>
    </table>
  <%-- -------------------  END TASK DATA -----------------  --%>
```

We know that in this context the main object is a task so we can use it to extract some data to enrich the form.

```
  <%-- -------------------  BEGIN HTML GRID ----------------- --%>
    <table border="0">
      <tr>
        <td colspan="4"><%designer.draw("RADIO", pageContext);%></td>
      </tr>
      <tr>
        <td><%designer.draw("COMBO", pageContext);%></td>
        <td><%designer.draw("STRING", pageContext);%></td>
      </tr>
      <tr>
        <td colspan="4"><%designer.draw("NOTES", pageContext);%></td>
      </tr>
      <tr>
        <td><%designer.draw("INTEGER", pageContext);%></td>
        <td><%designer.draw("DOUBLE", pageContext);%></td>
      </tr>
      <tr>
        <td><%designer.draw("DATE", pageContext);%></td>
        <td><%designer.draw("PERSON", pageContext);%>  
            <%designer.draw("BOOLEAN", pageContext);%>
       </td>
      </tr>
      <tr>
```

```
        <td><%designer.draw("FILE", pageContext);%></td>
      </tr>
    </table>
```

We call designer.draw for every declared field

```
    <table><tr><td><%designer.draw("DETAIL", pageContext);%></td></tr></table>
```

Then the master-detail

```
  <%-- -------------------  END HTML GRID ----------------- --%>
```

And the html grid is closed

```
      <%
        double testUseValues = 0;
        //sum of weights
        testUseValues  += designer.getEntry("INTEGER", pageState).intValueNoErrorCodeNoExc();
        testUseValues  += designer.getEntry("DOUBLE", pageState).doubleValueNoErrorNoCatchedExc();
      %>
      <hr>
        <b><big>Test of sum of stored values: <%=JSP.w(testUseValues)%></big></b>
```

We can add some computation on inserted values. We can eventually mix data from the form and data from the task.

```
      <%
      c.end(pageContext);
    }
   // ##############################  END FORM LAYOUT DEFINITION  ################################
  }
%>
```

That's all. "print" and "save" buttons are added automatically.

14.4 Custom wizards

Teamwork also provides the possibility of adding "custom wizards", which will be accessed by having an additional button on the menus and creating say a project from a set of templates. An example wizard is provided in the folder

```
[root]/applications/teamwork/plugins/moreExamples
```

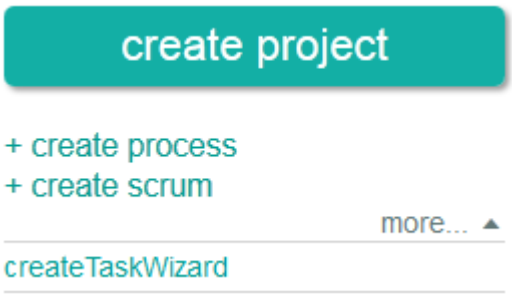
And is called "createTaskWizard.jsp". Move this file up by a folder, so you'll have it in the plugins folder, and the go to

admin -> forms and plugin -> reload plugins,

you'll get a new line:

WIZARD				
createTaskWizard	createTaskWizard	*delegated to impl*	createTaskWizard.jsp	/applications/teamwork/plugins/createTaskWizard.jsp

And a new button when accessing "projects":



Of course you will need to check out the details of the actions done in the code of the

wizard – someone in your team needs to be confident with Java web development.

14.5 Customize Excel exports

This is a bit advanced feature: you access it via *admin -> excel*, and each label serves the customization of the corresponding Teamwork section.

E.g. CUSTOM_EXPORT_EXCEL_ISSUE manages the customization of the "export issues to Excel" from the issue search / list page. Say you want to export the code and requester of the issues, which is not exported by default; the code is

```
{"code":"obj.code","requeter":"obj.assignedBy.name"}
```

Custom export excel

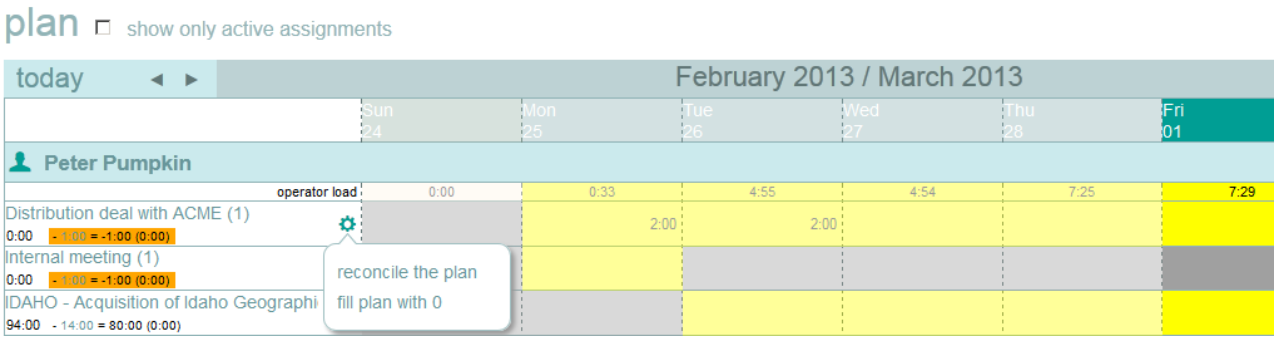
	code	application	EN
	CUSTOM_EXPORT_EXCEL_ISSUE	Teamwork	{"id":"obj.id","code":"obj.code","description":"obj.description","requester":"obj.assignedBy.name"}

Getting:

A	B	C	D
id	**code**	**description**	**requester**
17	A78	Use a browser integrated tool like Licorize http://licorize.com	Peter Pumpkin
18	A34	Find % cost on similar deals	Joe Kavalier
2		Write to Margaret Bourke-White for the opening pictures	Peter Pumpkin
4	B45	Write Bob Woodward for the "man bites dog" piece	Peter Pumpkin
5		Get Robert Capa for the cover	Peter Pumpkin
6		Get Peter Arnett interview the new cheerleaders for the college team	Peter Pumpkin

Similarly for the other Teamwork' main objects.

14.6 Custom features

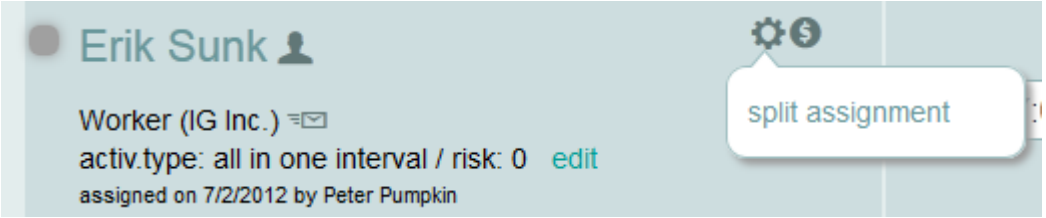


Since teamwork 5.2 Teamwork allows enabling / disabling of powerful customized parts, in such a way that also customers with customized parts can update Teamwork without danger of losing any customization.

Sections that can be currently enabled / disabled are:

CUSTOM_FEATURE_ASSIGNMENT_SPLIT

Enables the “split assignment” functionality



CUSTOM_FEATURE_FILL_PLAN_0

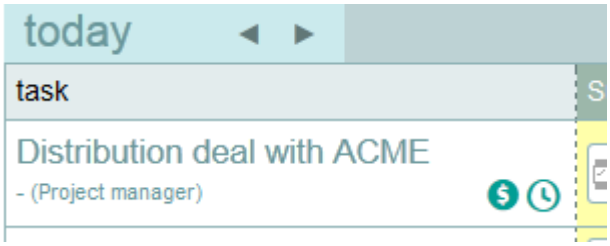
Adds a button “Fill plan at zero” on the plan, which fills unplanned parts at zero.

CUSTOM_FEATURE_RECONCILE_PLAN

Add a button on the plan view to reconcile planned with done.

CUSTOM_FEATURE_SHOW_EXP_ON_TIMESHEET

This enables the show expenses button on worklog week.



CUSTOM_FEATURE_SHOW_NOTES_ON_SIDEBAR

This makes the notes written in the Deliverables field be visible on every task tab.

CUSTOM_FEATURE_SHOW_UNDERPLANNED

In the plan view shows days where resources are underplanned.

CUSTOM_FEATURE_WORKLOG_FORM

This enables advanced customization of the worklog insertion part by linking an additional customization logic; contact Teamwork development if you need this kind of functionality.

14.7 Even more customization options

Teamwork team has several ways to further customize Teamwork and integrate it with other data sources without forcing the customer release to be a separate code branch, which would make application updates problematic. For example since 5.3 Teamwork supports “injecting” further, external, persistent classes. Contact Teamwork staff for further information.

15. Advanced business process

15.1 Why extend project management with business processes?

We use business process as synonymous with workflow.

There are several reasons for extending project management with business processes; in fact the two are presented as alternatives for meeting team organizational needs. With Teamwork 5 you get the advantages of both, in an *integrated* solution.

Project management, where processes are modeled with trees, dependencies and assignments, has several limits:

1. formal limits
2. rigidity
3. hard to maintain in time for the project manager
4. may result complex for the end user

For formal limits, we refer to http://www.workflowpatterns.com; just think of recurring phases for something hard to do with a tree. For 2 and 3, projects intrinsically lack the notion of “local evolution” (or expansion), which is natural in business processes. For 4, consider for example a step (a task) in a project which is assigned to a user only to get a signature from her; she may end up having tens of assignments only for a signature, and tens of task statuses to update and justify. Wouldn’t it be more natural that she got from the interface just some buttons to be pressed confirming that she signed, and that is all required from her to let the process proceed?

With business processes, we not only overcome such limits, but get more:

1. more flexibility and standardization of processes
2. easier to support change

15.1.1 Supporting change

In business processes, if you define a workflow, you can revise it anytime. Parallel instances of "version 1" of still running processes and new "version 2" ones should co-exist. In this sense business-process enhanced project management becomes compatible with "change management", so important in large organizations.

Now someone may be wondering: already the idea of introducing the complex sounding "project management" in our organization is scary, add "business processes", it sounds horribly complex. Well, we actually agree with those feeling. But in Teamwork it all comes down to very simple solutions: at least if you adopt Teamwork, you are not letting very expensive, complex, old and unusable software enter the organization, but something quite simple and that can be loved by the users, as a time and memory saver. If your process is say registering incoming orders, performing draft in several phases, and then getting feedback either by customers or from internal testing, well this is a typical business process, in particular if you have recurring test phases, and this is what you'll get from Teamwork.

15.1.2 Flowork and JBPM

Teamwork includes Open Lab’s Flowork module. This is a complete workflow management based on JBoss’ JBPM (http://www.jboss.com/products/jbpm).

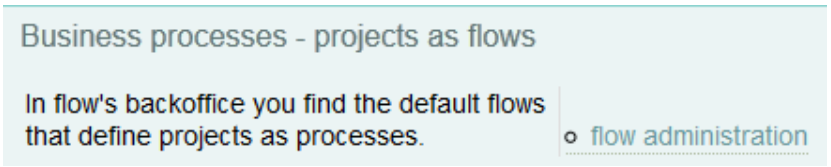
Flowork supports most used patterns in workflows (forks, joins, milestones, sub-processes, etc.).

We merge business processes and project management together, see chapter “3.10 Task - other ” for basic concepts.

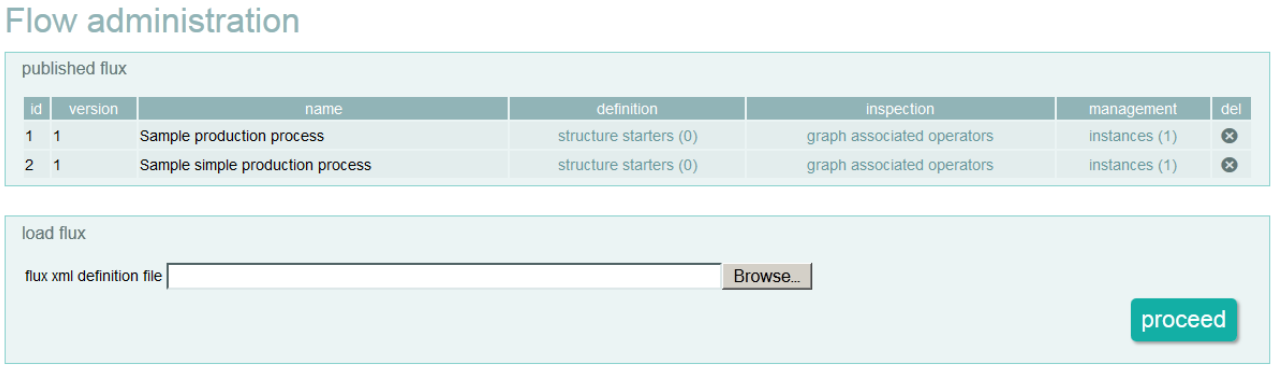
In this chapter we will see how to administer Flowork and some basics of JBPM. For a complete guide to JBPM see the JBoss site above.

15.2 Flow administration

Flowork can manage many kinds of flows at once. Every flow has a “definition”; a definition is a XML file written in JPDL syntax.



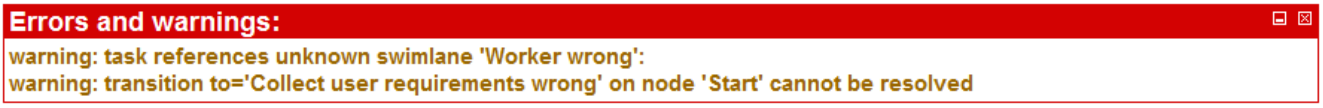
Go to admin page, and then follow the “flow admin link”:



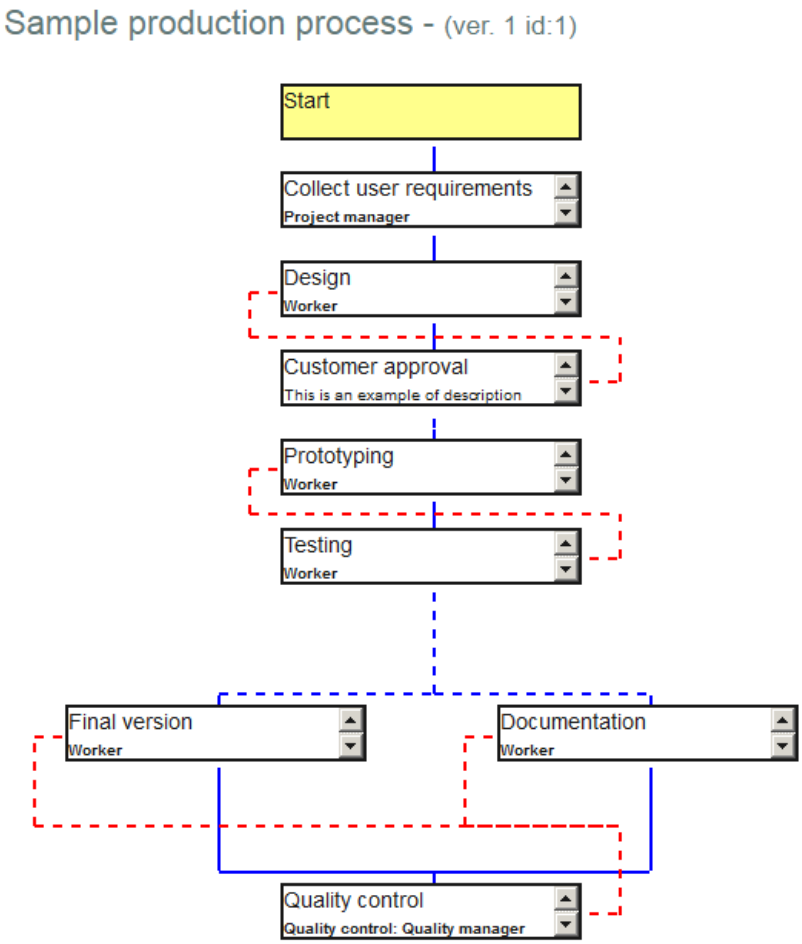
Flowork functionality is actually wider than what is used for the moment in Teamwork, we will examine features currently relevant for Teamwork.

In this page there is the list of published fluxes. Flowork supports more than a version per flux. If you load a revised version of the process, new instances will run with new version, while the already running process will continue with the old version.

If you want to upload a new process, select the file from your local drive and upload it. Flowork performs a validation of the process definition. In case of errors you will get an error message:



By clicking on “graph” you will get the detail view of how the process is composed:



By rolling over transitions you will highlight next steps.

Go back to the fluxes list. By clicking on “instances” you can manage running instances:

In the top part there is a list of running instances, in the lower part there is the list of the last 10 closed fluxes.

You normally won’t need any hand management: Teamwork manages instances for you.

Click on “log” to see the steps history:

log of instance 2 for flux Sample production process

steps done

role/actor	step	start	end	duration
Stakeholder/Customer	Collect user requirements	04 Jun 2012 17:37:08	13 Jun 2012 11:35:55	209:58
Project manager	Collect user requirements	04 Jun 2012 17:37:08	13 Jun 2012 11:28:28	209:51
Worker	Design	13 Jun 2012 11:35:56		192:36

steps to be done

role	step
Stakeholder/Customer	Customer approval
Worker	Prototyping
Worker	Testing
Worker	Final version
Worker	Documentation
Quality control	Quality control
Worker	Release
Worker	Baby sitting

print

From the previous page, you can examine the instance state by clicking on “instance state”:

instance state

instance flux management

Sample production process
(id: 2 ver. 1)

ID	name	resource	start	actions	proceed to
2	Design	not assigned	Monday, June 4, 2012 5:37 PM	STEP	Customer approval

Here you can “force” a step execution by clicking on the “step” button. Doing this you will “signal” the current token to proceed. This may have different behavior depending on how the flux is written. Normally you do not need to do it by hand.

If you want to stop a process you can click on “cancel instance”. This will end the flux without completing steps, so you may get “strange” behaviors on the associated task. Usually when you force a process to end you will have to edit the associated task, closing it.

15.3 Creating your flow

Processes are defined by using JPDL definition language, we warmly suggest to read JBPM documentation to fully understand the module potential.

Teamwork uses processes in a particular way that make possible the linearization discussed in chapter “3.11 Business processes”. In particular Teamwork uses extensively “ActionHandlers” to synchronize fluxes and task trees.

Let’s examine the “sampleSimpleProjectProcess.xml”.

```
<process-definition name="Sample simple production process">
```

First the process name. It must be unique; two processes with the same name are considered versions of the same process.

```
  <swimlane name="Project manager">
    <assignment class="org.jblooming.flowork.defaultHandler.ActorAssignmentHandler"/>
  </swimlane>

  <swimlane name="Stakeholder/Customer">
```

```
    <assignment class="org.jblooming.flowork.defaultHandler.ActorAssignmentHandler"/>
  </swimlane>

  <swimlane name="Worker">
    <assignment class="org.jblooming.flowork.defaultHandler.ActorAssignmentHandler"/>
  </swimlane>
```

Then the swimlane declaration. A swimlane represents the “role” played by an “actor”. In terms of Teamwork “actors” are “resources” and when instantiating the process a resource will be assigne as actor for each node. Swimlane is a “local role”, so a swimlane’ name must correspond to existing local roles. In this example "Project manager", "Stakeholder/Customer", "Worker" are all existing roles.

```
  <event type="process-end">
    <action class="com.twproject.task.process.TaskProcessFluxEndHandler"/>
  </event>
```

Then we call TaskProcessFluxEndHandler when the process ends. When a process is instantiated, Teamwork creates a root task and a child for every process step. This handler is responsible for closing the root on the project associated to the flux.

```
  <start-state name="Start" siwmlane="Project manager">
   <transition to="Collect user requirements"/>
  </start-state>
```

Starting point definition, and first transition

```
  <task-node name="Collect user requirements">
    <task swimlane="Stakeholder/Customer"/>
    <transition to="Design"/>
    <event type="node-enter">
      <action class="com.twproject.task.process.TaskProcessTaskNodeEnterHandler"/>
    </event>
    <event type="node-leave">
      <action class="com.twproject.task.process.TaskProcessTaskNodeLeaveHandler"/>
    </event>
  </task-node>
```

This a real step of the process. Each “task-node” has a name that will be used as task name, assignments will be created for every swimlane involved in the step. Then there is the transition and two mandatory handlers, TaskProcessTaskNodeEnterHandler and TaskProcessTaskNodeLeaveHandler associated to node-enter and node-leave events respectively.

Both extend TaskProcessTaskNodeEventHandler. We will examine this class in detail.

```
  ----------------- see the complete flux on sampleSimpleProjectProcess.xml file -----------
  <end-state name="Done"/>

</process-definition>
```

Then the process end.

In order to understand how Teamwork and Flowork are related take a look to the TaskProcessTaskNodeEventHandler class:

```
public abstract class TaskProcessTaskNodeEventHandler implements ActionHandler {

  ProcessInstance processInstance;
  TaskNode taskNode;
  TaskProcess taskProcess;
  Task taskProcessRoot;
  Task taskOnStep;
  TeamworkOperator loggedOperator;

  public void execute(ExecutionContext executionContext) throws PersistenceException {

    try {
      processInstance = executionContext.getProcessInstance();
      taskNode = (TaskNode) executionContext.getNode();
      taskProcess = (TaskProcess) PersistenceHome.findUnique
                    (TaskProcess.class, "processInstance", processInstance);
      taskProcessRoot = taskProcess.getTask();

      // recover the taskOnStep by using the externalCode
      for (PerformantNodeSupport t: taskProcessRoot.getChildren()) {
        Task task=(Task)t;
        if ((taskNode.getId()+"").equalsIgnoreCase(task.getExternalCode())){
```

```
          taskOnStep=task;
          break;
        }
      }

      String loggedOperatorId = executionContext.getJbpmContext().getActorId();
      loggedOperator = (TeamworkOperator) PersistenceHome.findByPrimaryKey
                       (Operator.class, loggedOperatorId);
      doTheRealAction(executionContext);
    } catch (Throwable t) {
      Tracer.platformLogger.error(t);
    }
  }

  abstract void doTheRealAction(ExecutionContext executionContext);
}
```

This is an abstract class that retrieves the link between “task” and “process” that is modeled via “TaskProcess” object, fills attributes and calls doTheRealAction that is implemented on TaskProcessTaskNodeEnterHandler and TaskProcessTaskNodeLeaveHandler.

```
public class TaskProcessTaskNodeEnterHandler extends TaskProcessTaskNodeEventHandler {

  public void doTheRealAction (ExecutionContext executionContext) {
    try {
      if (!TaskStatus.STATUS_ACTIVE.equals(taskOnStep.getStatus())){
        // set task status to Active
        taskOnStep.changeStatusPersistAndPropagate(taskOnStep.getStatus(),
                    TaskStatus.STATUS_ACTIVE, "", new HashSet<Task>(), true, loggedOperator);
      }
    } catch (Throwable t) {
      throw new PlatformRuntimeException(t);
    }

  }
}
```

In this case when you are entering a step, the corresponding task’s status is set to “active”.

This is the minimal implementation, but if you want you can extends this classes to perform different actions. You can also add more than an action for every event.

- Is it possible to also define the description of a task, not just the name?
- Yes, of course. Here an example:

```
<task-node name="Customer approval">
    <description>
        This is an example of description usage in a task node
    </description>
    <task swimlane="Stakeholder/Customer">
        <description>
            worker approval
        </description>
    </task>
    <transition to="Design" name="Failed"/>
    <transition to="Prototyping" name="Passed"/>
    <event type="node-enter">
        <action class="com.twproject.task.process.TaskProcessTaskNodeEnterHandler"/>
    </event>
    <event type="node-leave">
        <action class="com.twproject.task.process.TaskProcessTaskNodeLeaveHandler"/>
    </event>
</task-node>
```

- The first "description" tag content is used for task’s description; the second one for the assignment's description.

Is it possible to give a default resource name that is proposed by teamwork when a process driven project is about to being created?

Not directly. Probably a solution could consist in developing a custom "ActorAssignmentHandler" that use some trick in the swimlane name and then by modifying the "createProcess" page.

In Teamwork 5 we added attributes to workflow descriptions that allow setting duration of tasks and estimations of assignments:

```
<task-node name="Design">
    <description>
        This task have a duration defined in flux definition duration:5
    </description>

    <task swimlane="Worker">
        <description>
            estimated:4:30
        </description>
    </task>
```

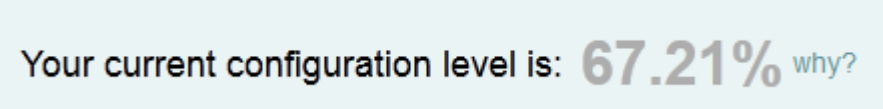
16. Configuration

16.1 Introduction

Teamwork is feature-rich and hence has an extensive configuration; the default configuration covers already a lot of needs, the only really necessary configuration concerns e-mail servers. But the point is that *if* you want more, you can get it.

Configuring Teamwork is a task for administrators.

For accessing the configuration page from the main menu select “admin”, getting

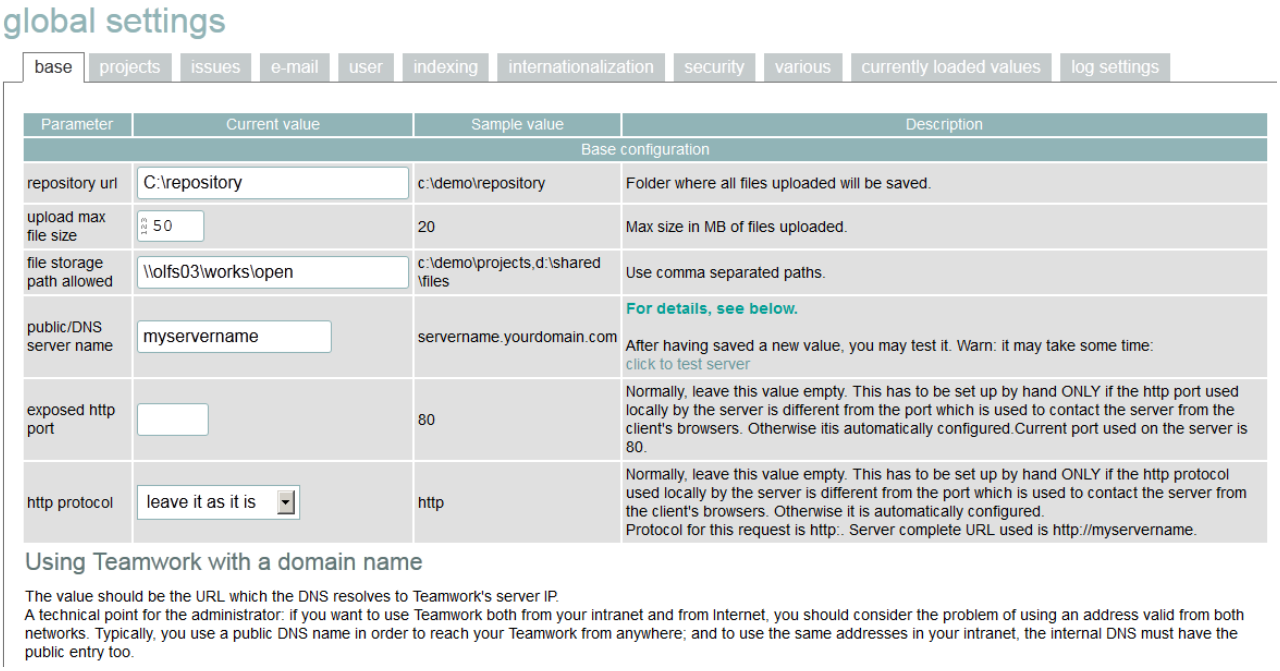


In the very last box there is your configuration level: a number that represents how many Teamwork’ features you are using. It is not necessary to get to a high number to use Teamwork, but if your level is under 30-40% you are probably missing something important.

This page is divided in five sections: *Basic system configuration, Teamwork behavior, Customization, Advanced configuration, Monitoring*.

16.2 Paths, network and security

The first two links will lead to the same configuration page:



First three lines are use to define paths. Parameters have a detailed description. The third parameter: “file storages paths allowed”, is intended to allow administrators to limit the creation of file storages starting with specified paths.

The following parameters are used for configuring how user access Teamwork through the

network. Proxy, firewall, https, NATting may make things incredibly complex in large organizations, but Teamwork gives a good support, notice https.

Back to the admin page, by clicking on “Security policies”:

base | projects | issues | e-mail | user | indexing | internationalization | security | various | currently loaded values | log settings

Parameter	Current value	Sample value	Description
Security			
enable redir after login	☑	yes / no	Whether to redir after login to a pending URL (may generate exception)
allow empty string psw	☑	yes / no	Password empty allowed
fld password expiry	0	0 (never)	Password expiry in days.
fld password min len	0	0 (not set)	Password minimal length.
logout time	0	29 (minutes)	After how many minutes of inactivity the application will log out the user. If 0, session will never expire; this is the suggested setting.

you can define Teamwork’ password and security policies.

Following “Ldap integration” link:

Authentication and LDAP			
◉ enable standard authentication	○ enable http authentication	○ enable ldap authentication	○ enable ldap authentication with fallback on standard

Teamwork supports three different types of authentication:

16.2.1 Standard authentication

In this case users and passwords are managed by Teamwork; it’s the default after setup.

Teamwork does not store the real password but only an hash, so you can only reset a password but not recovery it.

The login screen is supplied by Teamwork.

16.2.2 Http authentication

This means that the authentication is provided by the container (by default, Tomcat).

In this case Tomcat checks user credentials (eventually by a SSO) and then passes the authenticated user name to Teamwork.

Teamwork will get the authenticated user from the container context, search by its login name in its people list, and if the user is found and is an enabled one, it will not ask to login again in Teamwork.

So users MUST be present on Teamwork; in this case, Teamwork passwords are not used.

The login screen is supplied by the container.

For example, the distributed Tomcat has commented out in the server xml file various sources of authentication, JDBC, JNDI (and hence LDAP), memory etc..

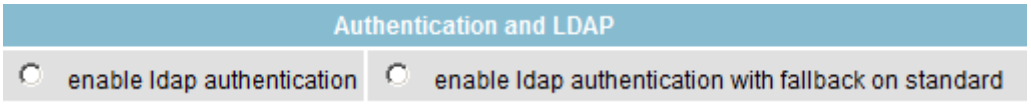
Details are in the container documentation, for example

http://tomcat.apache.org/tomcat-6.0-doc/realm-howto.html

or

http://www.caucho.com/resin-3.0/security/authorization.xtp

16.2.3 LDAP authentication / with fallback



In this case Teamwork will check user credentials with an LDAP server.

User credential are validated on a LDAP server and if validation goes well Teamwork checks if the user is in its archives and enabled, otherwise it will perform a standard authentication.

Login screen is supplied by Teamwork.

Notice that users' rights in Teamwork and in LDAP are totally disconnected.

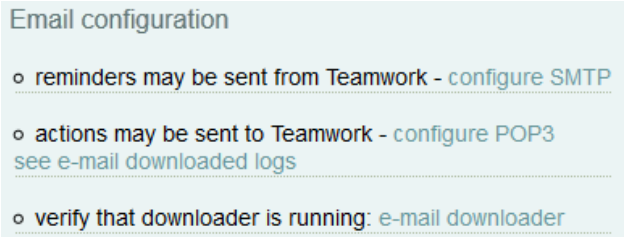
You can eventually import and/or schedule synchronization with an LDAP/AD server

By selecting "enable ldap auth." you will have to configure connection parameters. Go to section "16.8 LDAP/Active Directory" for details.

If you enable the fallback, failed LDAP authentication will fallback to Teamwork's, giving access to Teamwork in any case.

16.3 E-mail configuration

E-mail flow can go in two directions: from Teamwork to the users, and from users to Teamwork. Different configurations are needed for the two directions; you may activate one and not the other, at your choice. Of course having both directions is the ideal situation.



For all configurations, log in with administrator rights and go to admin page and follow the e-mail configuration link:

base | projects | issues | e-mail | user | indexing | internationalization | security | various | currently loaded values | log settings

Parameter	Current value	Sample value	Description
Mail			
application e-mail	test@example.com	teamworkmail@yourdomain.com	This address will be used to send messages, if SMTP is configured. If POP3 (or IMAP,...) is configured, this should be the e-mail address corresponding to the POP3 account. In order to send messages by mail, and receive email notifications, this must be set.
e-mail subject prefix		Teamwork notification:	This text will be added as prefix in the subject of each notification sent from Teamwork by e-mail.
SMTP - sending			
server SMTP	smtp.example.com	smtp.yourserver.com	In order to send messages by mail, and receive email notifications, this must be set. After having saved a new value, you may test it. Warn: it may take some time: click to test server
mail smtp port		25	If left empty, the standard port (25 or 465 for smtps) will be used.
use authenticated SMTP	☐		For details, see the "e-mail configuration" chapter of the **user guide**.
mail SMTP user			
mail SMTP password			
e-mail send protocol	smtp	smtp	Use smtps for example to use Gmail as SMTP server.
POP3/IMAP - receiving			
pop3 host	pop3.example.com	pop3.yourserver.com	In order to enable Teamwork to receive messages by mail, this must be set. After having saved a new value, you may test it. Warn: it may take some time: click to test server
pop3 user	test@example.com		In order to enable Teamwork to receive messages by mail, this must be set.
pop3 password	••••••••••		In order to enable Teamwork to receive messages by mail, this must be set. see e-mail downloaded logs
e-mail download protocol		pop3	Use pop3s for example to use Gmail as POP3 server.
port	-1	-1	Leave -1 for default values: e.g: pop3=110, pop3s=995, imap=143

16.3.1 Configuration of e-mail from Teamwork to users

This is quite simple, as it amounts to configuring “send e-mail” from the server where Teamwork is installed. This consists in setting an SMTP server, and an e-mail from which e-mails will be sent. These two parameters are all is generally needed to set up send e-mail from Teamwork.

server SMTP	smtp.example.com	smtp.yourserver.com
mail smtp port		25
use authenticated SMTP	☐	
mail SMTP user		
mail SMTP password		
e-mail send protocol	smtp	smtp

In case you use “authenticated SMTP”, a bit more parameters may be needed.

This done, users will receive e-mail only if they have an e-mail address set on their profile:

e-mail

ppolsinelli@example.com

url

16.3.2 Configuration of e-mail from users to Teamwork

In this direction, you have to create a new e-mail account, which will be used (exclusively) by Teamwork: Teamwork will connect to such account, and download and parse e-mail, just like your local e-mail client does.

pop3 host	pop3.example.com	pop3.yourserver.com
pop3 user	test@.example.com	
pop3 password	●●●●●●●●●●●	
e-mail download protocol		pop3
port	−1	-1

All configurations are extensively commented on the interface, which on save will also test the accessibility of the e-mail account.

In order to make this feature running you must configure it AND the e-mail downloader scheduler must is running. To check follow the “verify that downloader is running: e-mail downloader” link.

16.3.3 Using Gmail as SMTP and POP3 (IMAP) server

You can use Google’s Gmail service to receive and send e-mail from Teamwork.

You **absolutely must** use a NEW Gmail account for your Teamwork.

Here is an example configuration to use Gmail as server:

application e-mail	teamworktest@gmail.com
server SMTP	smtp.gmail.com
mail smtp port	
use authenticated SMTP	☑
mail SMTP user	teamworktest@gmail.com
mail SMTP password	[the password]
mail send protocol	smtps
pop3 host	pop.gmail.com
pop3 user	teamworktest@gmail.com
pop3 password	[the password]
e-mail download protocol	pop3s

Notice that for SMTP, smtps is used, and for pop3, pop3s is used.

Imap support is currently experimental: To connect to Gmail using the IMAP protocol instead of the POP3 protocol, simply change the host name "pop.gmail.com" to "imap.gmail.com" and change the protocol name "pop3s" to "imaps" in the above instructions.

Maintenance: Google Gmail engine will keep all messages available in the web interface, even when Java API has downloaded and “deleted” them from the inbox. This is nice because it is a backup, but you may get really a lot of e-mail in there, so maybe clear it once in a while.

16.3.4 Debugging e-mail configuration

To test sending e-mail from Teamwork to the clients - do these tests sequentially:

- check that you’ve set an SMTP server
- check that you’ve set an e-mail on your resources/users
- a simple way to test that it works is by doing a "send message" from docs&tools -> send message and checking "e-mail
- check that the scheduler is running (admin -> monitoring -> scheduler monitor)
- if your aim is to send appointments to you e-mail client, check that in your user options you have checked “send appointments to my e-mail client”

For errors check the email log in WEB-INF/log/email.log

To test receiving e-mail from clients in Teamwork - do these tests sequentially:

- check that you’ve set and checked a POP server

- check that you've set unique e-mails on your resources/users
- check that the scheduler is running (admin -> monitoring -> scheduler monitor)

For errors check the email log in WEB-INF/log/email.log

16.3.5 Customizing e-mail

You can customize the e-mail subject prefix by going to admin -> configure SMTP -> e-mail subject prefix.

In the case of assignment notifications, this get combined with the labels

ASSIGNMENT_NOTIFICATION

And

ASSIG_AS

Which you can change in the label editor.

16.4 Full-text indexing and ranking

Teamwork indexes text present in its objects, using Lucene as full-text indexer.

Lucene is documented here:
http://lucene.apache.org/java/docs/index.html

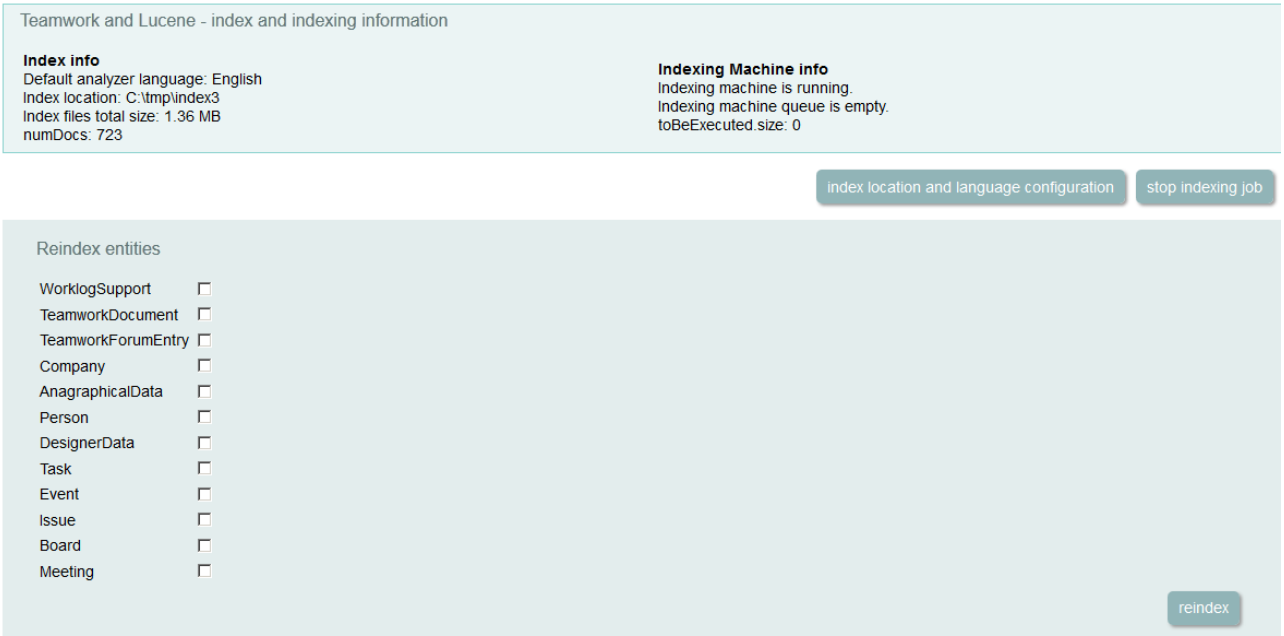
There are some parameters you can configure:

Parameter	Current value	Sample value	Description
Indexing configuration			
index path	C:\tmp\index3	c:\demo \index	Folder where all Lucene index files will be saved. You must restart the web app to make indexing effective. Reindex entities.
analyzer language	English	English	Language used by Snowball analyzer: examples are "English", "German", "Spanish", "French", "Italian". Click to test stemmer class. You must restart the web app to make this setting effective.

First of all the index files location, then the analyzer language. This analyzer is used to stem correctly your data. Stemming allows you to search "work" and find also "working", of course it works correctly when the language you are writing data and the stemmer match.

In any case Teamwork uses also exact matching in searches.

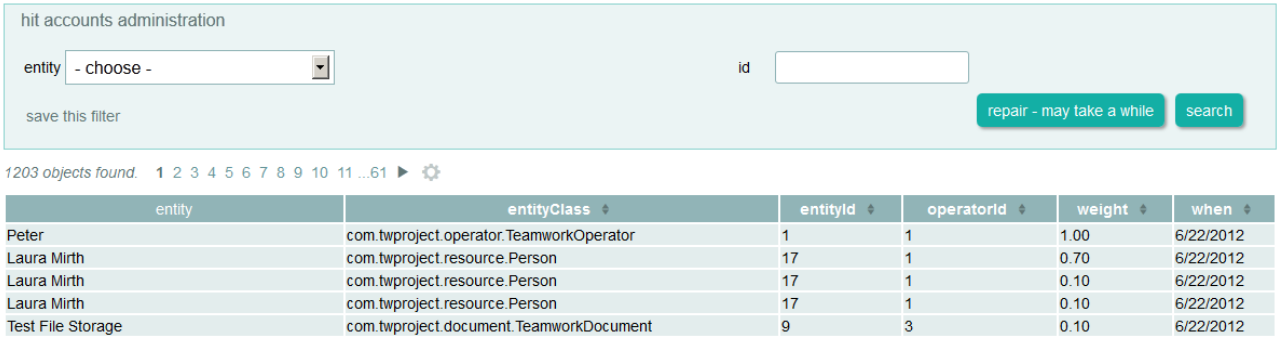
By following the "index management" link:



Here you can see the indexing machine status, stop the indexing job or force re-indexing.

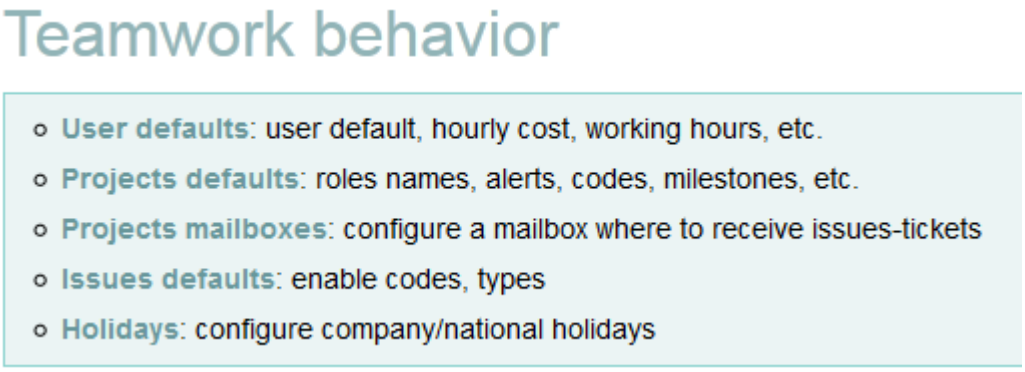
Hits management page

Hits are used by Teamwork to compute group rank. All "hit" data is de-normalized to maximize performance, so there is a scheduled action to "clean up" hits. You may use this page to "have a look" and eventually repair hits.



entity	entityClass	entityId	operatorId	weight	when
Peter	com.twproject.operator.TeamworkOperator	1	1	1.00	6/22/2012
Laura Mirth	com.twproject.resource.Person	17	1	0.70	6/22/2012
Laura Mirth	com.twproject.resource.Person	17	1	0.10	6/22/2012
Laura Mirth	com.twproject.resource.Person	17	1	0.10	6/22/2012
Test File Storage	com.twproject.document.TeamworkDocument	9	3	0.10	6/22/2012

16.5 Teamwork behavior



Teamwork has been built to works in different environments, countries, ways to work. In this section you can configure several of these aspects.

16.5.1 User defaults

base | projects | issues | e-mail | user | indexing | internationalization | security | various | currently loaded values

Parameter	Current value	Sample value	Description
Operator default options			
hour day start		08:00	Default start hour for agenda visualization.
hour day end		17:00	Default end hour for agenda visualization.
number of rows in lists	20	20	Default page size.
number of recent views tracked	10	10	
agenda visualization hour start (HH:MM)	09:00	09:00	Default start hour for working day.
agenda visualization hour end (HH:MM)	19:00	18:00	Default end hour for working day.
show user scores	☑	yes	Show user's scores whre possible.

Here are “defaults of default” or defaults when a user does not have her/his own.

16.5.2 Holidays

Configure shared holidays

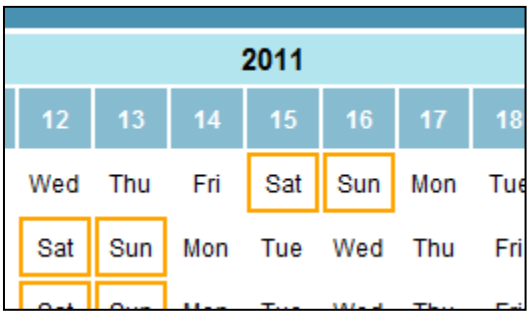
2012

month	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
January	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue
February	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed		
March	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
April	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	
May	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
June	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	
July	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue
August	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
September	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
October	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
November	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	
December	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon

Legenda: working day | variable holiday | fixed holiday | weekend

Easter this year is: Sunday, April 8, 2012 save

Using this page you can customize company’s holidays; check a cell if that day is holiday. Orange ones are not working day by configuration (Saturdays and Sundays).



These holidays are fixed, while oranges ones move year by year. You can customize differently each year.

16.5.3 Project defaults

Teamwork sometime uses default values:

base | projects | issues | e-mail | user | indexing | internationalization | security | various | currently loaded values | log settings

Parameter	Current value	Sample value	Description
Project management configuration			
default project manager role name	Project manager	Project manager	Used for finding a default project manager role in automated role setting. Change this value only if you are SURE of what you are doing.
default worker role name	Worker	Worker	Used for finding a default worker role in automated role setting. Change this value only if you are SURE of what you are doing. Create new area and roles.
milestone alert delta	3	3	How many days before the milestone deadline you want to get warnings.
hourly cost	40.00	20.00€	Default hourly cost on assignments; if set on the resource, this last is what will be proposed on the assignment editor when the assignee is chosen.
name of the scrum master role		Scrum Master	Name of the Scrum master role, used by the Scrum master wizard.
name of the team member role		Scrum Team	Name of the Scrum team member role, used by the Scrum master wizard. Create new area and roles.
total working hours per day	7:30	08:00	This will determine the default number of hour per day.
project import/export language	English	English	Default Microsoft localization for which projects will be exported and imported. Import a project.
progress 100%	☐	no	Set progress to 100% when closing a task.
use unique codes	☐	no	When true, codes on task, resources and eventually issues should be "unique" or empty.
generate codes	include year in code	no	Generate task codes as a progressive number based on the type
disable workflow	☐	no	When true the you will be able to create project as process.

default project manager role name	Project manager	Project manager
default worker role name	Worker	Worker
milestone alert delta	3	3
hourly cost	40.00	20.00€
name of the scrum master role		Scrum Master
name of the team member role		Scrum Team
total working hours per day	7:30	08:00
project import/export language	English	English
progress 100%	☐	no
use unique codes	☐	no
generate codes	include year in code	no
disable workflow	☐	no

Most relevant parameters are PM and worker role names, total working hour per day (that is used to convert days estimation in hours estimation) and milestone alert delta (how many days before the milestone date the “milestone approaching” event is raised).

Teamwork can generate unique codes from task types: if you select a task type, didn’t type a code, and enabled the “generate codes” option above.

Since version 4.7 there is also “disable workflow” that turns off the workflow module, simplifying the user interface:

disable workflow	☐	no	When true the you will be able to create project as process.

16.6 Customization

Customization

- Define **custom fields**: on tasks on issues on assignments on worklogs on costs
- Labels: change interface' labels
- Teamwork logo: change the default Teamwork logo on prints
- Pages: create/manage pages default home page: set a default entry page for each user.
- Templates: create/manage templates (aka page grids)
- Portlets: install/un-inmtall, restrict access, modify
- Forms and plugin: reload/manage plugins.
- Business processes: upload new processes, manage, control Teamwork's business process engine
- Extra features: enable/disable Teamwork customizations
- Excel: define export fields
- Colors default: change colors for agenda/plan/load special cases

16.6.1 Internationalization

base | projects | issues | e-mail | user | indexing | internationalization | security | various | currently loaded values | log settings

Parameter	Current value	Sample value	Description
Internationalization & agenda			
language and date format	American	en	
friday is working day	☑	yes	Whether Friday is a working day.
saturday is working day	☐	no	Whether Saturday is a working day.
sunday is working day	☐	no	Whether Sunday is a working day.
currency format	###,##0.00€	###,##0.00€	The currency format applied in costs. Do NOT change separators, only groupings! Money will appear like this: 1,234,567.00€ See detail.
print logo	printLogo.gif	printLogo.gif	teamwork The logo file must be put in the folder C:\develop\java\teamwork5\html\applications\teamwork\images

Here you can define the default interface language (every user can then pick a different one).

You can set some holydays, currency, date and time formats.

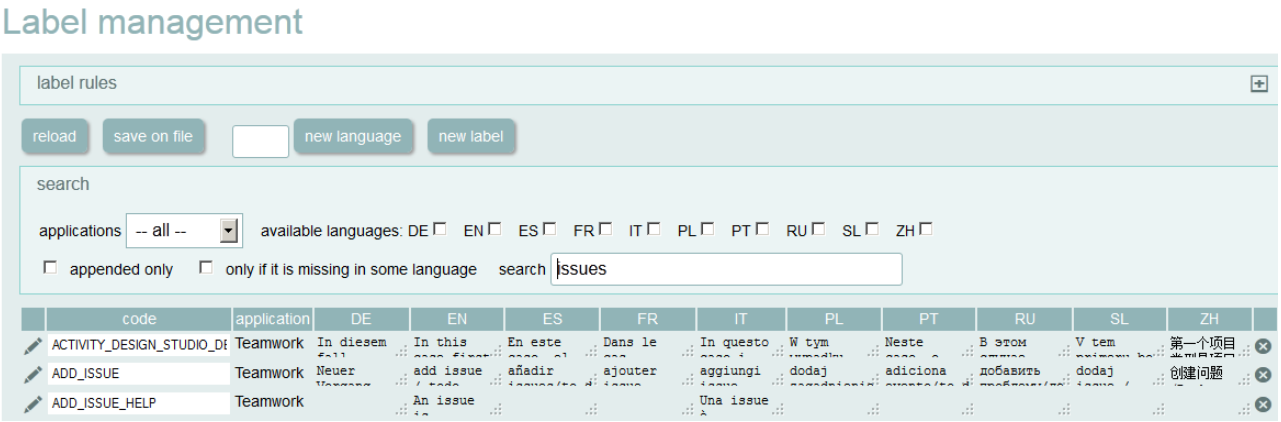
You can customize you reports by changing Teamwork logo. Insert here a new file name and copy the file in the suggested folder.

16.6.2 Customize labels

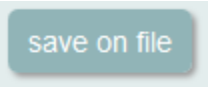
Teamwork supports label customization on-the-fly.

If you want to modify a label go to “labels” in the admin page:

Insert a label to search e.g.: “issues” and press “return” key:



Then modify one or more label, then press “save on file” button.



Labels modified by customers will be preserved by Teamwork updates.

Your work gets saved on the database, so web app updates will not overwrite it.

You can create your own language, but do not underestimate the effort: Teamwork’s labels are thousands. Details in next section.

16.6.3 Translate in a new language

Suppose you want to create an entirely new language for the user interface. Teamwork lets you create a new language and insert the label translation entirely from the web interface, and even “in context”, that is, when the label “edit”

mode is active, you can edit a label translation by clicking it in the web page where it is shown.

To create a new language, first you must login as administrator: for clarity, say you assign to your language the code “XZ”. Go to go to “labels” in the admin page, locate the "new language" button -> in the input write "XZ", click "new language" -> now in editing each entry you will have a "XZ" input column.

Now there are two problems:

Many labels: that when you create a new language translation, you have really a lot of labels to translate, so in place edit is not always the most comfortable way, and for some messages, like warnings sent by the application, not possible.

Contribute language: now if you want to send back the language to Teamwork developers (this contribution gets usually rewarded with free user licenses), the labels you have inserted are saved on your database.

Now we answer both problems with the same technical explanation: Teamwork distributes the common internationalization labels through textual files, those with .i18n extension. When the user inserts labels, those get saved on the database, so that web app updates

do not overwrite them. In order to get also your work to end up in .i18n files, you need to get to “development” modality, which is done by opening the config.properties file in WEB-INF, writing development=yes, and restarting Teamwork.

So for the “many labels” problem, now if you want to change several labels in one go, you can edit directly the internationalization files: search the files with the i18n extension in the web app.

And of course now you are able to send the internationalization back to Teamwork developers.

A quick way to do the translation is to use the export function in the i18n manager page, and work in Excel or similar. The import function wants the file in this format: Teamwork [TAB] code [TAB] language [TAB] translation, for example

Teamwork 01_GRAVITY_LOW FR bas

Teamwork 01_STATUS_OPEN FR ouvert

So if you decide to do a complete translation in a new language, go to admin -> labels, click “export”, open or save the file in Excel or what you prefer, add a column and start translating. Then you can import it with the import function, or send it back to Teamwork staff to be included in next release.

16.6.4 Portal/dashboard

See section “10 Portal/Dashboard”.

16.7 Advanced configuration

Most of these configurations have already been discussed before.

16.7.1 Teamwork security

See section “12 Security”

16.7.2 Business process / Flows

See section “15 Advanced business process”

16.7.3 Custom forms

See section “14 Plugins, custom fields/forms”.

16.8 LDAP/Active Directory

Native LDAP integration allows several operations:

1) user import “by hand”: there is a procedure for importing users from an LDAP server (including Microsoft Active Directory ©), which may be used even if authentication through LDAP is not enabled; it is of course necessary if it is enabled, because the users in Teamwork must exist in all cases.
2) user import from a scheduled job:
3) authentication from the LDAP server

Authentication and LDAP	
◉ enable ldap authentication	○ enable ldap authentication with fallback on standard

Since version 4.5, you can now enable LDAP with fallback to Teamwork internal authentication in case of LDAP login failure.

16.8.1 LDAP basic parameters

First of all, you must setup the LDAP basic parameters; go to “Ldap integration” from the administration page:

LDAP Parameters			
Parameter	Current value	Sample value	Description
ldap base dn	CN=Users,DC=open-lab,DC=com	CN=Users,DC=open-lab,DC=com	See the web page http://www.twproject.com/ldap.page.
ldap provider url	ldap://olfs01.open-lab.com	myserver.mydomain.com	
ldap security authentication	simple	simple	
ldap security principal	ppolsinelli@open-lab.com	testldap@mydomain.com	
ldap security credentials	●●●●●	somepwd	
retype ldap security credentials	●●●●●	confirm somepwd	
ldap config file	activeDirectory	activeDirectory	Select the type of LDAP server you are connecting to. Lists the ".properties" files in *C:\develop\java\teamwork5\html\commons\settings\ldap*.
ldap create users on login	area ☐ create		Whether any valid LDAP login will trigger a Teamwork user creation if the user is missing, in the area selected (area is required).
ldap enable ldap ssl	☐	no (default its false)	Enable SSL over LDAP (LDAPS protocol)

test LDAP connection go to LDAP imports

By checking LDAP radio button you will have to configure LDAP parameters.

Note: LDAP is a language with several dialects. Hence we provide out of the box some variants on the language, in different configuration files. The variants provided are:

- Active Directory (©Microsoft Corp.):
  in the file[web app root]/commons/settings/ldap/activeDirectory.properties
- Apache Directory Server:
  in the file[web app root]/commons/settings/ldap/apacheDirectory.properties
- OpenLdap:
  in the file[web app root]/commons/settings/ldap/openLdap.properties

One may add properties files here, and they will be available in the global configuration combo.

Notice also that both “1.0” and “2.0” LDAP queries should work.

A nice feature is:

ldap create users on login	area create		Whether any valid LDAP login will trigger a Teamwork user creation if the user is missing, in the area selected (area is required).

That allows Teamwork to create LDAP validated user at its first login.

Example configuration with Active Directory:

Authe

enable standard authentication | enable http authentication | enable ldap au

LD

Parameter	Current value
ldap base dn	CN=Users,DC=open-lab,DC=com
ldap provider url	ldap://olfs01.open-lab.com
ldap security authentication	simple
ldap security principal	awhirl@open-lab.com
ldap security credentials	•••••••••
retype ldap security credentials	•••••••••
ldap config file	activeDirectory
ldap create users on login	area Production create

Example configuration with Apache Directory:

Auther

enable standard authentication | enable http authentication | enable ldap au

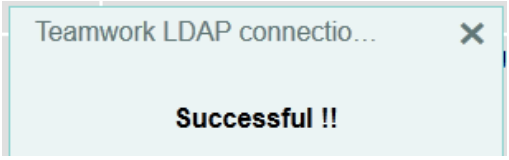
LD

Parameter	Current value
ldap base dn	ou=users,ou=system
ldap provider url	ldap://olpc016:10389
ldap security authentication	simple
ldap security principal	uid=admin,ou=system
ldap security credentials	••••••
retype ldap security credentials	••••••
ldap config file	apacheDirectory
ldap create users on login	area Production create

Example configuration with OpenLDAP:

Parameter	Current value
ldap base dn	dc=open-lab,dc=com
ldap provider url	ldap://192.168.0.157:389
ldap security authentication	simple
ldap security principal	cn=admin,dc=open-lab,dc=com
ldap security credentials	•••••
retype ldap security credentials	•••••
ldap config file	openLdap

Once you have inserted values you can check the configuration by using the test button.

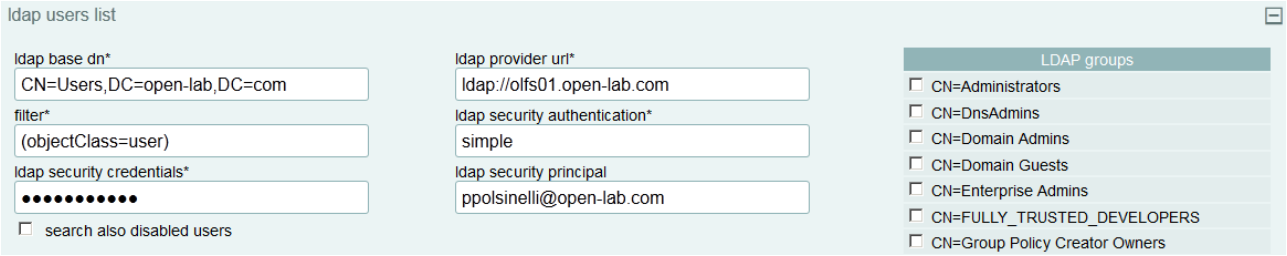


In order to enable authentication you MUST have users created in Teamwork. You may proceed by importing them manually or by scheduling an import.

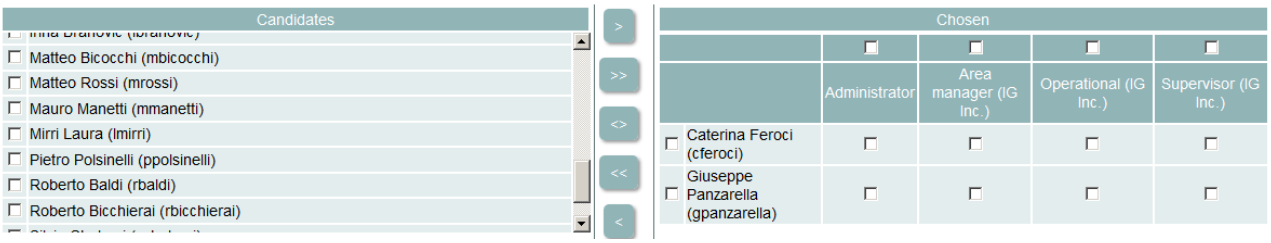
16.8.2 Importing users by hand

There is a comfortable procedure for importing users by hand, which also lets you configure the imported users rights from the point of view of Teamwork: if you've set up the LDAP parameters, then go to admin page an follow "LDAP integration - import users".

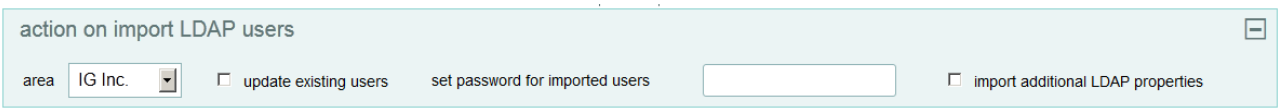
Here you can select the CN groups in which to search users, and once found some, pick those you want to import.



For every picked user, you can decide whether to make it a Teamwork administrator, or set on her/him other area-global roles.



LDAP roles are not mapped into Teamwork as the business logic behind them is quite different; customized behavior can be developed on demand.



Users will be put on the area you pick.

"update existing users" will update non security related data on existing users.

"set password for import users": this is the Teamwork password that will be set on imported users, in case LDAP authentication is off. If leaved empty, a password equal to the login name will be set.

16.8.3 Additional properties

The eventually mapped additional properties are listed in a ldap.properties file, in

[web app root]/commons/settings/ldap/[the chosen one].properties

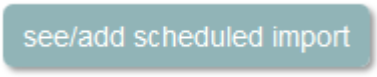
The sample ones mapped are:

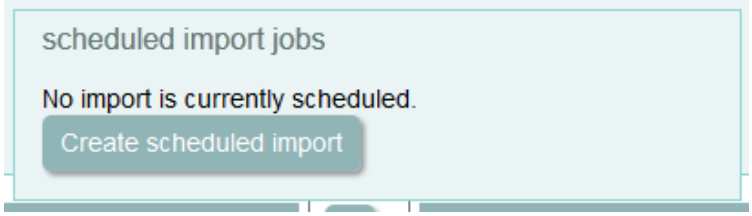
```
PHONE=telephoneNumber
COUNTRY=co
STATE=st
CITY=l
ZIP=postalCode
MOBILE=mobile
```

You can add your own, compliant with your LDAP dialect.

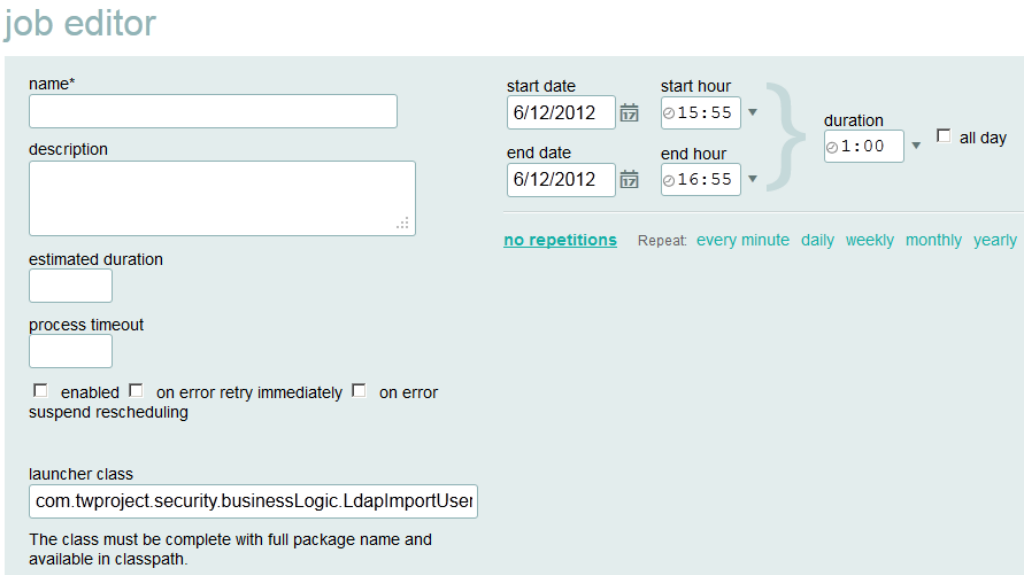
16.8.4 Scheduling user import

From LDAP user import click on button





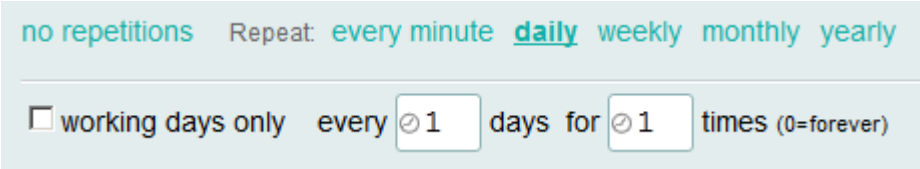
Click on "create schedule"



First box contains data about job scheduling the right box data about your LDAP.

default pwd: this is the Teamwork password that will be set on imported users, in case LDAP authentication is off. If leaved empty, a password equal to the login name will be set.

This is the setting for having it run every day:



16.8.5 LDAP F.A.Q:

HELP! I can't login into Teamwork anymore!

1) You may have enabled LDAP authentication, but didn't import any user. Proceed as follows:
2) stop Teamwork
3) go to [your root]webapps/ROOT/commons/settings, open the file global.properties, remove the property AUTHENTICATION_TYPE=ENABLE_LDAP_AUTHENTICATION
4) restart Teamwork

We login with our LDAP accounts, but nobody is administrator any more.

If you imported the users "by hand", not with the scheduled job, you should have selected the "administrator" checkbox for at least one user. To fix this, you must temporarily disable LDAP authentication, as in the FAQ above, enter with the original Teamwork administrator login, enable the administrator checkbox on some users, and then re-enable LDAP authentication.

HELP! I've setup LDAP parameters and successfully imported the users, but they can't login!

You may have forgot to set LDAP as authentication modality: log in with the original Teamwork administrator login, go to tools -> administration -> global settings, select the LDAP authentication radio:

Authentication and LDAP			
○ enable standard authentication	○ enable http authentication	◉ enable ldap authentication	○ enable ldap authentication with fallback on standard

and then save.

17. Monitoring

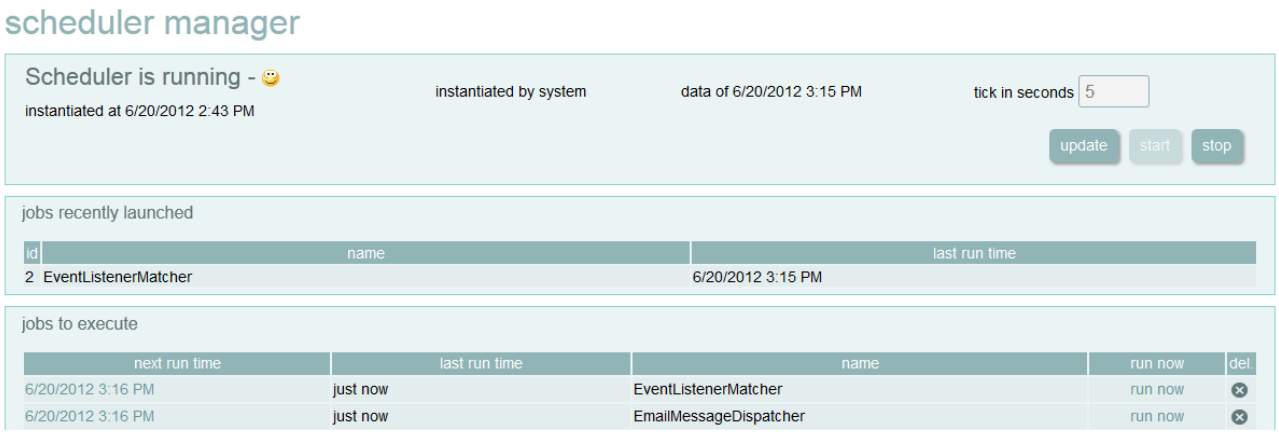
17.1 Introduction

Teamwork is a rich application with many features. Considering how many tasks it performs, it requires little maintenance.

17.2 Scheduler

There are some activities in Teamwork that are executed during batch processes. Teamwork includes a powerful scheduler that is used internally but could also be used to schedule some additional customer tasks.

Click on “scheduler monitor”:



Here you can see if the scheduler is running. By default Teamwork scheduler is launched when the application starts.

You can stop the engine and restart it. By default the scheduler perform a check of waiting jobs every 5 seconds.

.

“jobs recently launched” contains (usually nothing) jobs that are currently running.

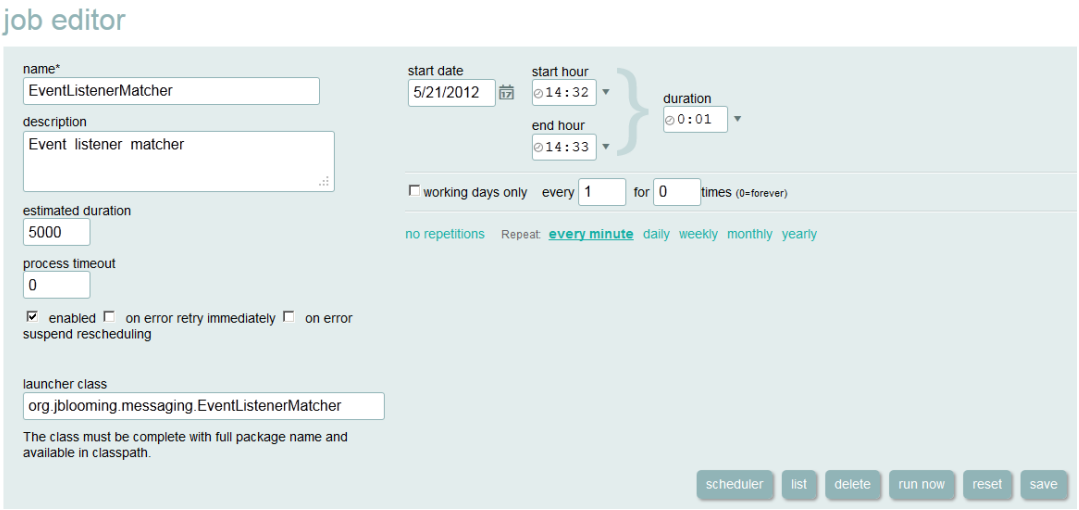
The bottom box contains jobs that are waiting to be launched. Next and last run times are both displayed.

You can force a job execution by clicking on

button.

Teamwork’s scheduler is built to run also in cluster environments. One node only will run a task, inhibiting the other.

You can change timing of a job by editing it:



You will have different kinds of schedule: minutes, single, daily, weekly and so on. For example,

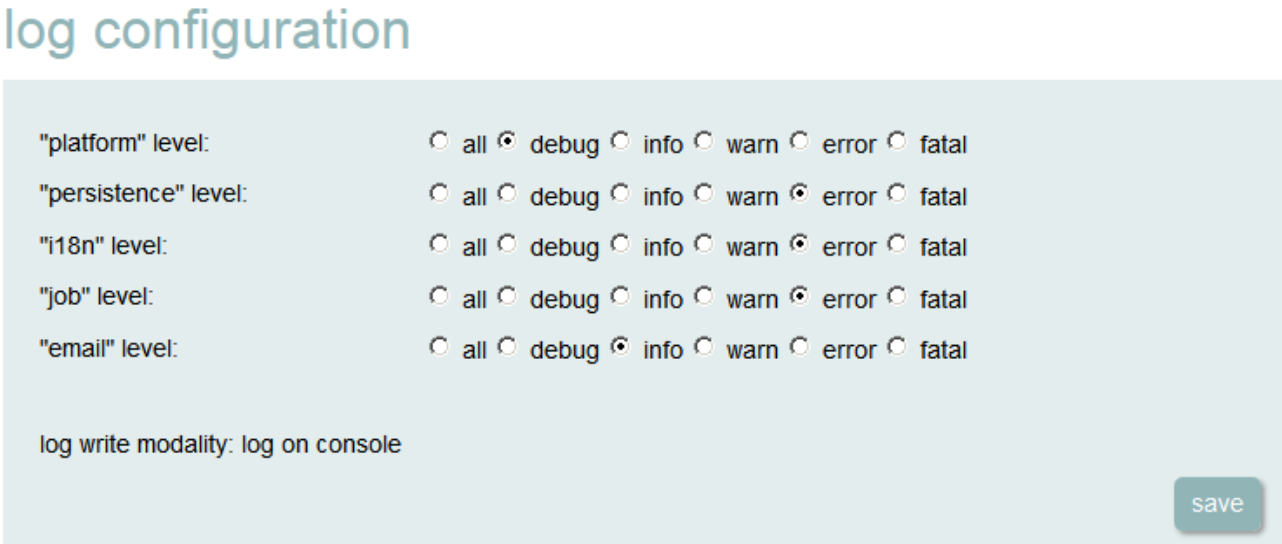
means every 180 minutes forever.

In the right part you must specify a class. In some cases you may have to insert parameters.

17.3 Logging

Teamwork logs events using log4j. Logs are really useful when there is some trouble… (admin -> log levels).

You can customize logging levels:



You can also view logs or get them it zipped.

Logs are located in

[your root]webapps/ROOT/WEB-INF/log

17.4 Messaging system

We have already described Teamwork's event/subscription capability. Now we will describe shortly how this engine works.

When a user subscribes listening an event, say "issue created", it is creating an object called "listener".

A listener refers a Teamwork object using class and id, has the event it is listening for (called "command"), has a validity period, has a user as owner, may be listening for children object too, etc. .

Usually there are lots of listeners (thousands after some months of real usage).

When something relevant happens, an "event" is generated and inserted in the event queue.

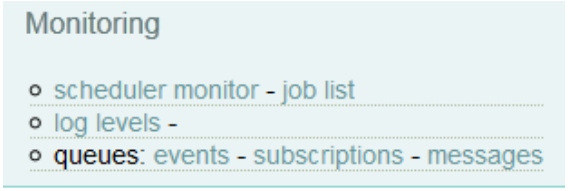
A scheduled job ("EventListenerMatcher") checks (by default every minute) for each event if there is a subscription listening for. If there is nothing listening, the event is removed.

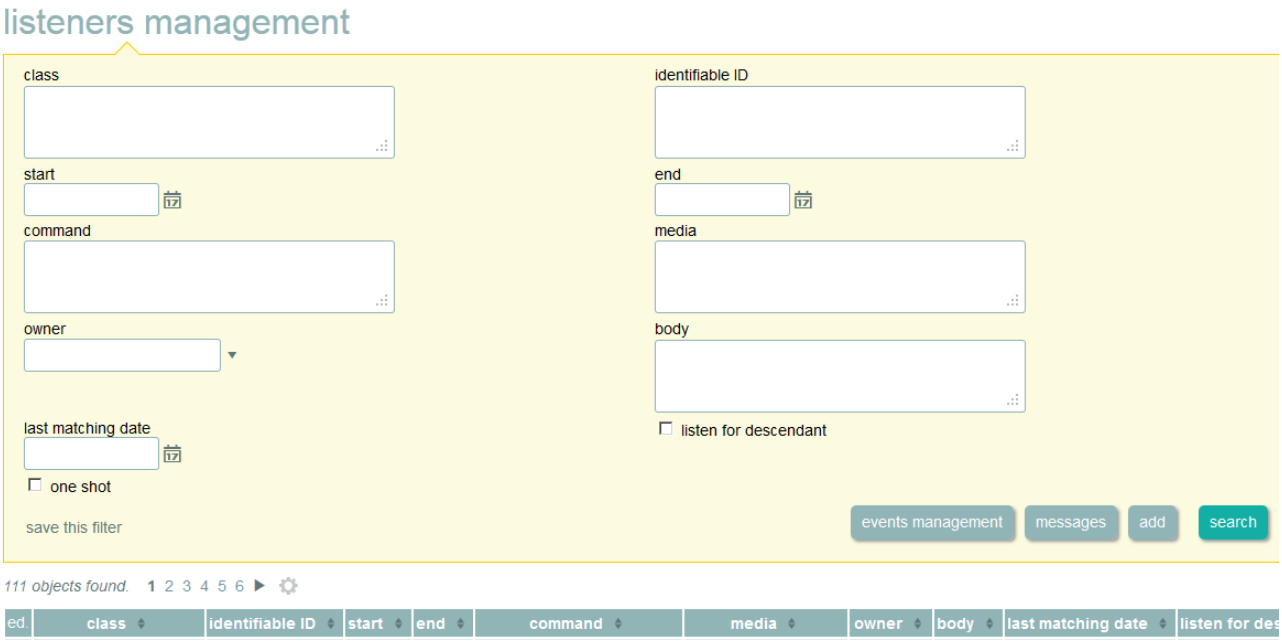
If there is a subscription a message is generated by integrating "listener" and "event" data; the message is inserted in the message queue. Then the event is removed.

Usually the event queue is empty or contains only a few records.

Once the message is queued it is available for dispatching: "EmailMessageDispatcher" and "StickyMessageDispatcher" are two scheduled jobs that send messages to their respective media channels. Messages for the "news" channel rest in the message list until they are read by the user, then removed. So the message queue should contain elements for "news" channel only.

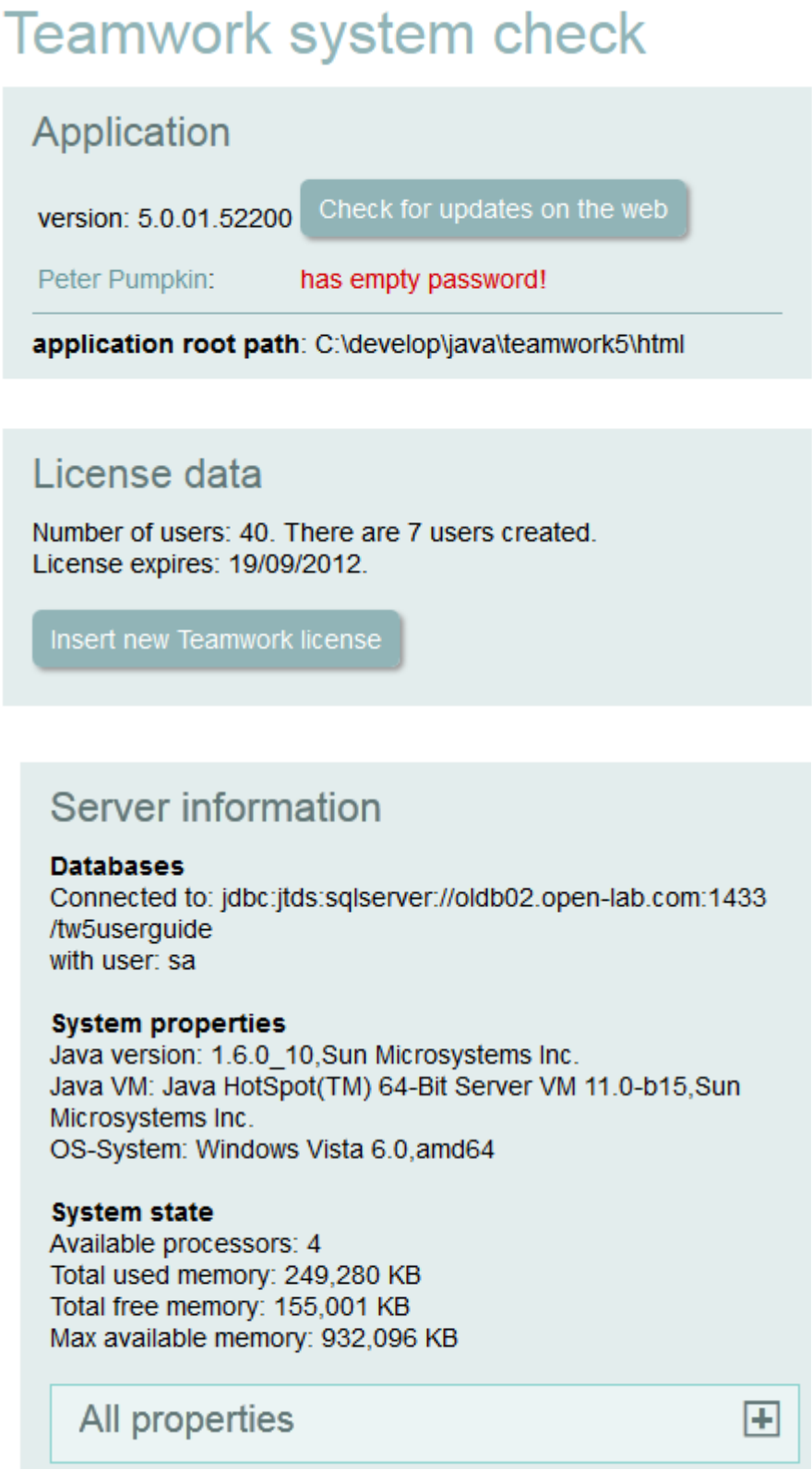
Teamwork provides list/search and edit pages for subscriptions, events and messages. Usually you do not need to operate in pages that are intended for monitoring purposes only (admin ->):





17.5 System check

This page performs a check of system properties, user's password, license; it may be useful to check your general Teamwork status, and also communicate with us in case of support:



17.6 Tree checks

Teamwork uses intensively data structured as trees. As SQL does not generally support

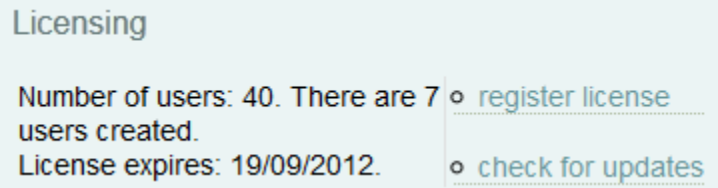
recursion, some key data is also de-normalized to increase performance. Occasionally something could go wrong (black-outs, system crashes, network failures etc.) and this de-normalized data may get corrupted.

Go to admin -> tree checks.

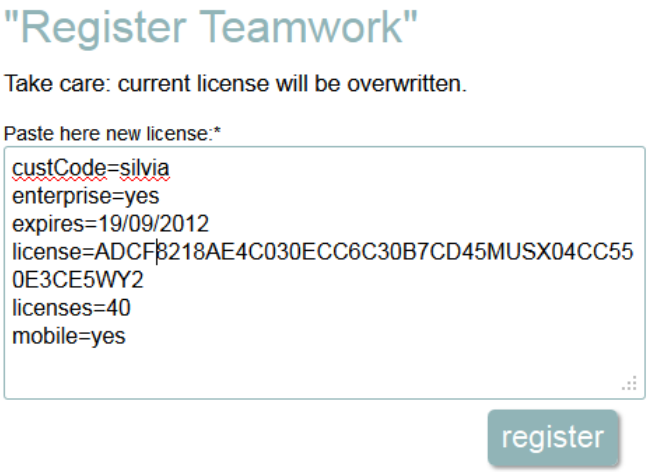
This utility will perform a check of “trees” and eventually will rebuild data correctly.

17.7 Licensing

Here you have the number of licensed users and the number counted for licenses.



Here you can register a new license:



or check online for updates:

17.8 Backup

Teamwork is composed by the application (with its configuration), the database, and the repository.

In order to perform a complete backup you must backup these three “entities”.

17.8.1 Application backup

In order to backup the application, you must copy the application root path (visible on “17.5 System check” page) in a safe place.

By doing this all your files, configurations and license will be saved.

17.8.2 Database backup

Teamwork is database independent; every database has its own backup procedures. Some need that you copy files only, but most require special procedures.

17.8.3 Repository backup

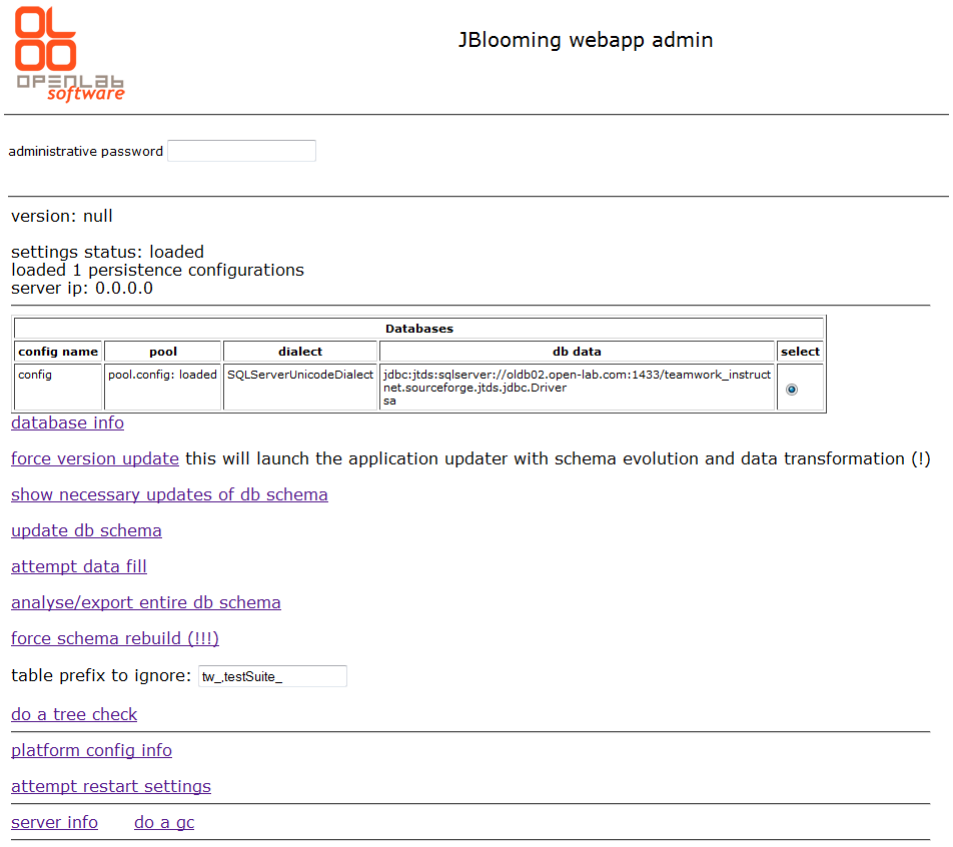
Repository contains uploaded files. The folder repository is specified in section “16.2 Paths, network and security”. Copy its content in a safe place.

17.9 Low level administration page

Teamwork is provided with a low level administration page that helps when your application doesn't starts-up correctly. DO NOT USE THIS PAGE IF DO NOT ASKED BY SUPPORT.

This page does not check security on Teamwork's data and can be potentially dangerous. This is why it comes disabled by default. If you need to use it (we may ask you to do this for support/debug purposes), go to the folder [root]/commons/administration and rename the file "admin.rename" to "admin.jsp".

This page looks like:



To perform actions, a password is required; by default is "domagic". You can change this password by editing

```
## admin page psw - optional
adminPassword=domagic
```

in [root]/WEB-INF/config.properties file.

17.10 Performance

Teamwork's installer will set some default configuration of the web server and database usage – which must be fine tuned for each particular environment.

This section assumes some technical IT competence.

17.10.1 Java memory usage

Teamwork is a Java application, and it runs in Java web servers – by default Tomcat. Java is hungry for memory; fortunately server memory is really cheap these days.

The server where Teamwork is running should have at least 2GB of memory, so that Teamwork can run in at least 1GB.

> If you want to go beyond the 1GB limitation, you should not use Windows servers, as these have notoriously contiguous memory limitations in running Java. Use a 64 bit Linux server and you will have no limitations about the amount of memory you can give to Java.

It must be said that for most usage scenarios 1GB of memory for Teamwork is more than sufficient.

A typical symptom that memory configuration must be tuned is when you get "java.lang.OutOfMemoryError: Java heap space" errors. In this case the most likely culprit

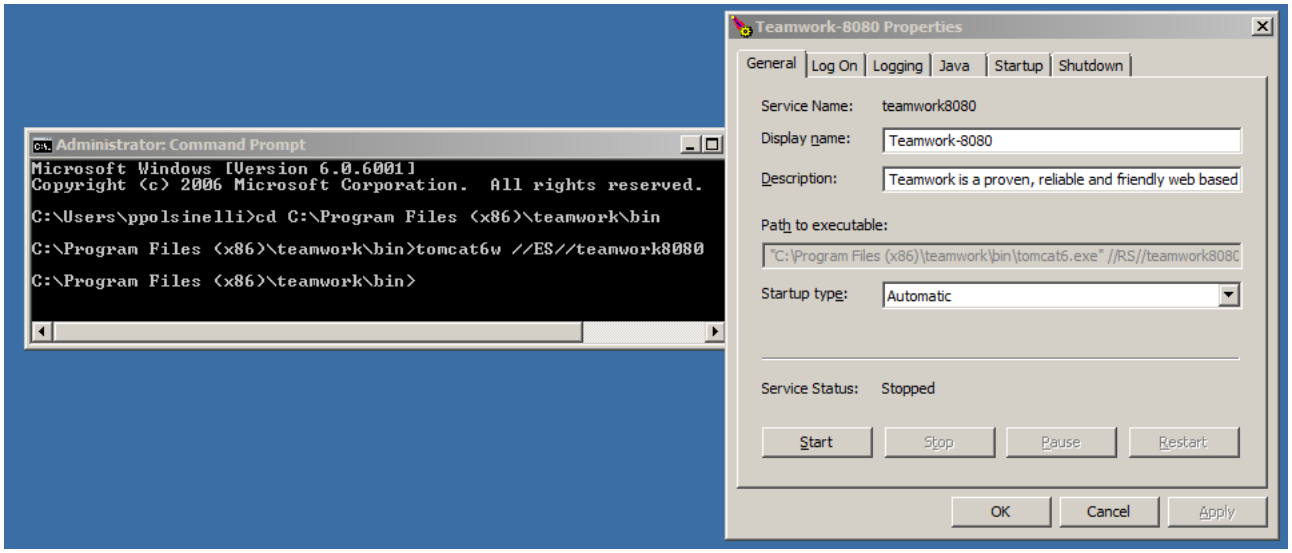
The best way to be sure that the memory configuration changes are in use in Teamwork is to launch the "system check" page (admin -> system check) and verify the memory parameters there reported. For example, before:

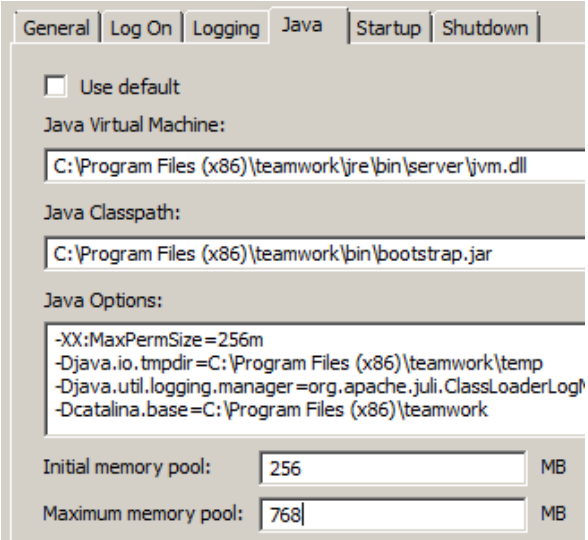
System state
Available processors: 4
Total used memory: 279,680 KB
Total free memory: 199,064 KB
Max available memory: 504,896 KB

To do the changes: instructions for different OS follow.

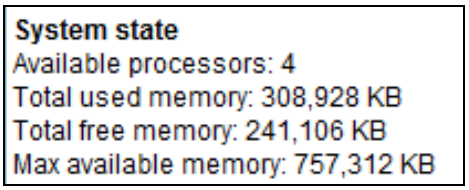
On a Windows server: open a DOS command interface:

```
Administrator: Command Prompt
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\ppolsinelli>cd C:\Program Files (x86)\teamwork\bin

C:\Program Files (x86)\teamwork\bin>tomcat6w //ES//teamwork8080
```





Changed maximum heap to 768MB, and after restarting the Teamwork service:



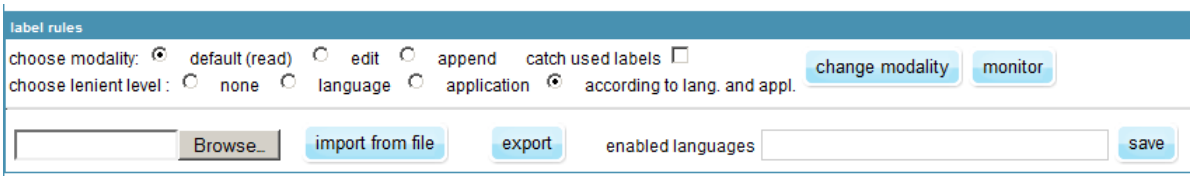
That's it!

On Linux: open the teamworkLinux.sh script, and change the -Xms128m -Xmx384m settings to a higher range.

On OSX: if you are launching by script by hand, open the teamworkOsx.sh script, and change the -Xms64m -Xmx128m settings to a higher range. If you are launching Teamwork as service, change in [teamwork root folder]/launchdScript.sh the line "export JAVA_OPTS=-Xmx512M $JAVA_OPTS" (if it is #export JAVA_OPTS=-Xmx512M $JAVA_OPTS, remove the #).

17.10.2 Disable unused languages

A way to reduce memory usage and speed up the interface is by disabling unused languages in your Teamwork instance.



If you log as administrator and go to the label management section, open the “label rules” container (it is closed by default), and say if you want to have only English as language, type EN in the enabled languages field and select SAVE.

17.10.3 Enabling more database connections

By default Teamwork uses a database connection pool, called Proxool (http://proxool.sourceforge.net/).

This pool has a built-in monitor, which you can enable by modifying the provided web.xml (WEB-INF/web.xml). Add among the servlets configuration

```
<servlet>
        <servlet-name>proxool</servlet-name>
        <servlet-class>org.logicalcobwebs.proxool.admin.servlet.AdminServlet</servlet-class>
</servlet>
```

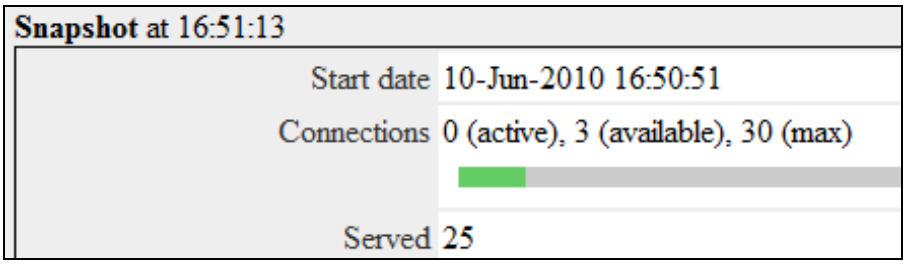
And in the servlets mapping

```
<servlet-mapping>
        <servlet-name>proxool</servlet-name>
        <url-pattern>/proxool</url-pattern>
</servlet-mapping>
```

Restart Teamwork, and call

[your Teamwork URL]/proxool

Getting:



To enable more db connections, you have to set in config.properties the poolMaxSize property:

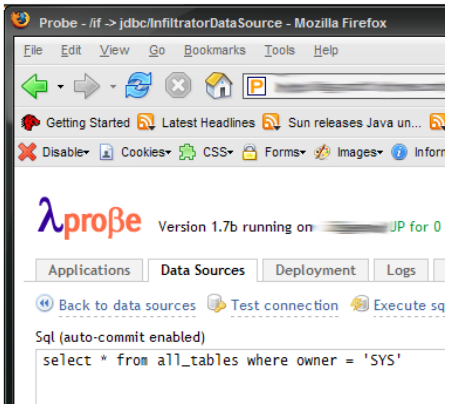
```
##  connection pooling: optionally set max size
poolMaxSize=50
```

17.10.4 Clustering the database

Teamwork uses a very simple JDBC connection to “a database”. The entry point provided can of course be clustered: for this you must refer to your database specific documentation.

17.10.5 Clustering the web server

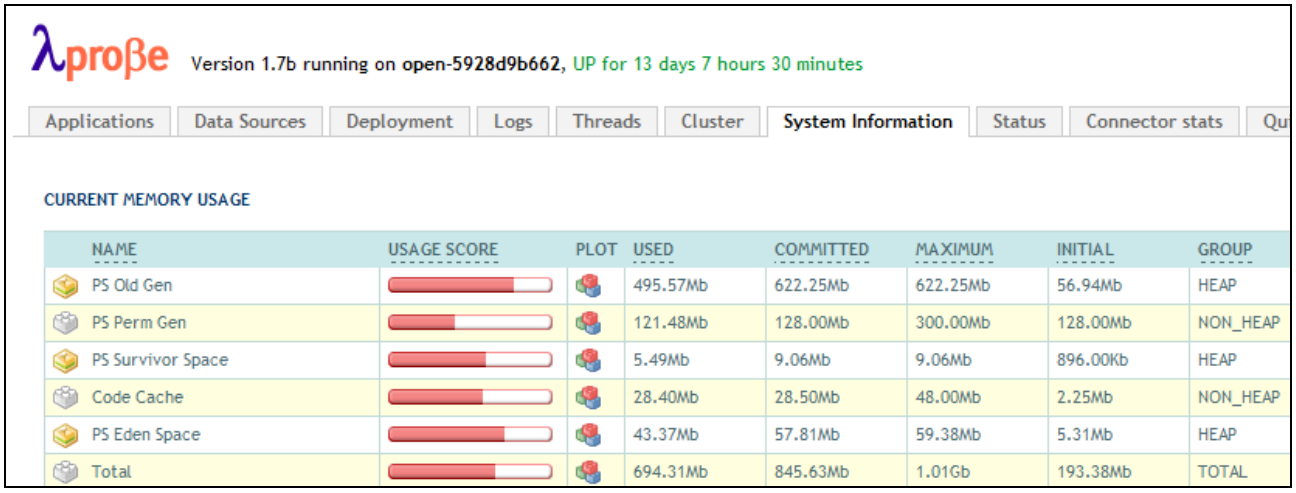
Teamwork can be clustered also as application: in case of load balance, so called “sticky sessions” must be enabled. The only conflict that could be raised in case of application clustering is in the scheduled jobs, but fortunately Teamwork has a built in mechanisms that enabled only one node in the cluster for job scheduling. Of course we assume that all nodes in the application cluster will be pointing to the same database entry point instance (eventually clustered too).



17.10.6 Monitoring memory usage

We often used the excellend Lambda Probe application. Find all details here:

http://www.lambdaprobe.org



NAME	USAGE SCORE	PLOT	USED	COMMITTED	MAXIMUM	INITIAL	GROUP
PS Old Gen			495.57Mb	622.25Mb	622.25Mb	56.94Mb	HEAP
PS Perm Gen			121.48Mb	128.00Mb	300.00Mb	128.00Mb	NON_HEAP
PS Survivor Space			5.49Mb	9.06Mb	9.06Mb	896.00Kb	HEAP
Code Cache			28.40Mb	28.50Mb	48.00Mb	2.25Mb	NON_HEAP
PS Eden Space			43.37Mb	57.81Mb	59.38Mb	5.31Mb	HEAP
Total			694.31Mb	845.63Mb	1.01Gb	193.38Mb	TOTAL

18. Other sources of information

This user guide is not the only source of information concerning Teamwork: the site Frequently Asked Questions (F.A.Q.), blog, forum, and feedback service are all sources that can be used.

The F.A.Q. can be found here:

http://answers.twproject.com/tags/faq/

18.1 What is in the blog

In the blog you may get some information about

- Relationship between methodologies and Teamwork
- What is currently being developed
- Details on Teamwork internal workings

A couple of examples:

A complete Gantt editor in the browser
http://blog.twproject.com/2012/06/15/a-complete-gantt-editor-in-the-browser/

Smarter search and recent object functionality
http://blog.twproject.com/2009/02/20/smarter-search-and-recent-object-functionality

Teamwork and multilinguism
http://blog.twproject.com/2009/01/13/teamwork-and-multi-language-management

18.2 What is in the forum

In the forum you may ask questions concerning usage, configuration and point out possible bugs. There are already about 300 questions answered. See it here:

http://answers.twproject.com

18.3 Direct contact

Direct support via e-mail or phone is reserved to customers with a support contract.

Company's HQ:
Open Lab
Via Venezia 18b
50121 Florence - Italy

Direct contact:
E-mail: info@twproject.com
Fax: +39 055 582451
Phone: +39 055 5522779
Time zone: 'Central European Time'
(6 hours ahead of New York)

19. Installation

You don't need to read this guide to install Teamwork: basically, you just need to download and follow the installer to get it running. Here you get detailed instructions for more detailed integrations and installations in complex environments.

19.1 System Requirements

Being a Java application, Teamwork runs **everywhere**, and by construction it runs on any relational database. It is a *server* application, so it runs as web server on a server, and it has to be installed only on the server. From all other machines, it is sufficient to point to the server with a browser, like Firefox, Chrome, Microsoft Explorer or Safari.

It needs a relational database where to write data. Almost all relational databases are supported; the installer supports natively MySql, SQL Server, Oracle, HsqlDB, and PostgreSQL. The provided HsqlDB is very fast. For installing on a database not listed above, you need to provide the JDBC driver and connection data (contact Open Lab if you need support on this).

The installer provides Java for the server (version 6), Apache Tomcat web server (version 6), and database drivers; this makes a default installation very easy: it consists in clicking a series of "next" on the installer. Technical staff can customize every aspect of how teamwork runs, like changing the web server, clustering the database access, and so on.

We strongly advise to take a look at the forum for updated FAQs and a lively discussion:

http://answers.twproject.com

19.2 Install using the graphical installer

Teamwork installation guide is also in video form, and can be seen or downloaded at

http://www.twproject.com

as "simple install" and "advanced install".

This document is an integration of the information provided in the videos.

19.2.1 Starting it up

To proceed with installation just follows these steps:

1) download the installer application for your platform from http://www.twproject.com/download.page

2) On a Linux box with a graphical interface, give execute permissions to the downloaded file and launch it from a console; both on Linux and OSX systems, launch it as root user. On Windows or OSX, just double click the file. Follow the install instructions.

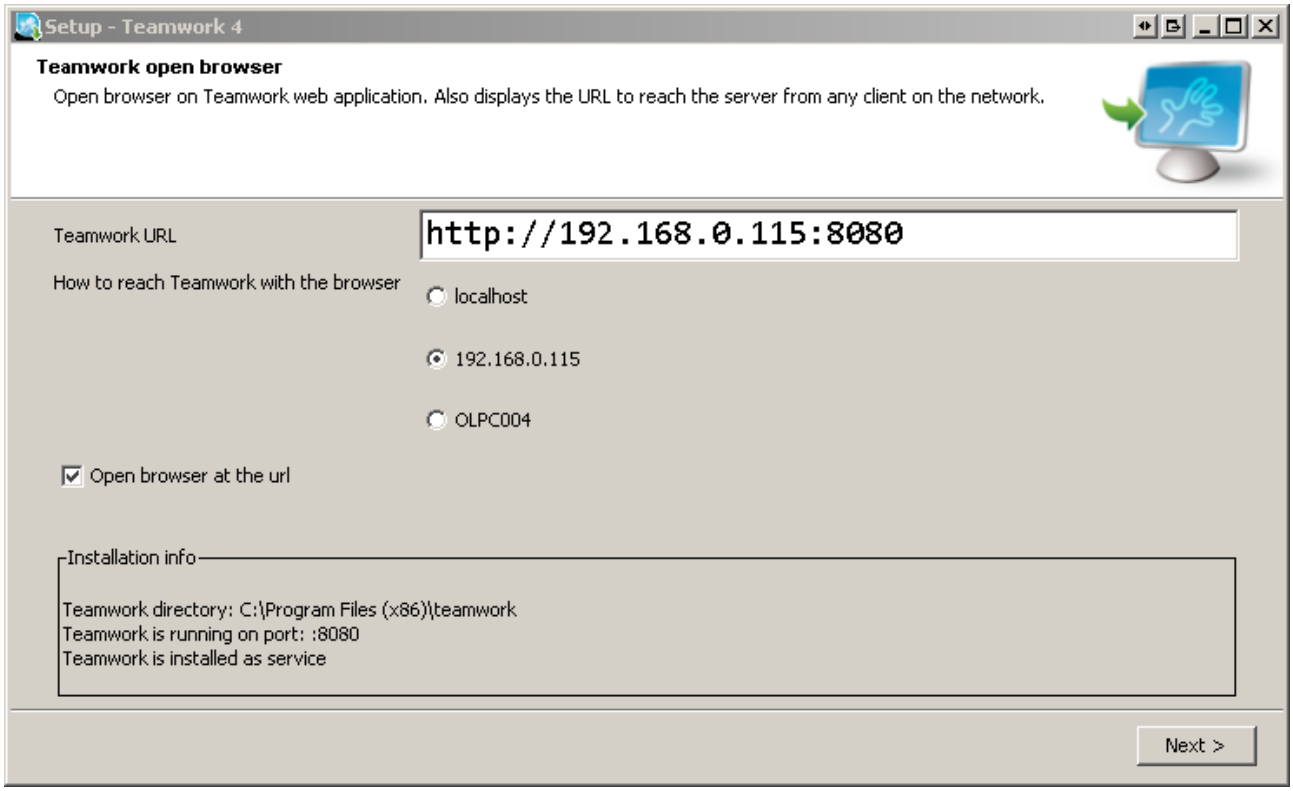
19.2.2 Installation completed correctly

At the end of installation, if the Tomcat and browser checkboxes are selected, Tomcat is launched by the installer as a system service or process.

To launch Teamwork by hand launch “teamworkLinux.sh start” (on Linux; or teamworkOsxStart.sh, for MacOSX, or teamworkWinStart.bat for Windows), and actually the best way is to install it as service.

19.2.3 Accessing from other machines

In the final installation screen,



Teamwork tries to hint as ways to reach the installation through the browser from other machines. This may depend also on your internal naming and DNS service.

19.2.4 Updating

Just download the latest installer, it will detect the previous version and update it.

19.2.5 Removing Teamwork / Uninstalling

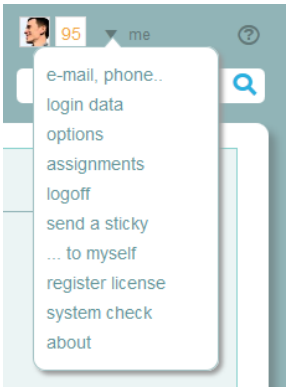
To uninstall, use the uninstall executable that is on the root of the installation; on Windows and OSX it will also remove the service.

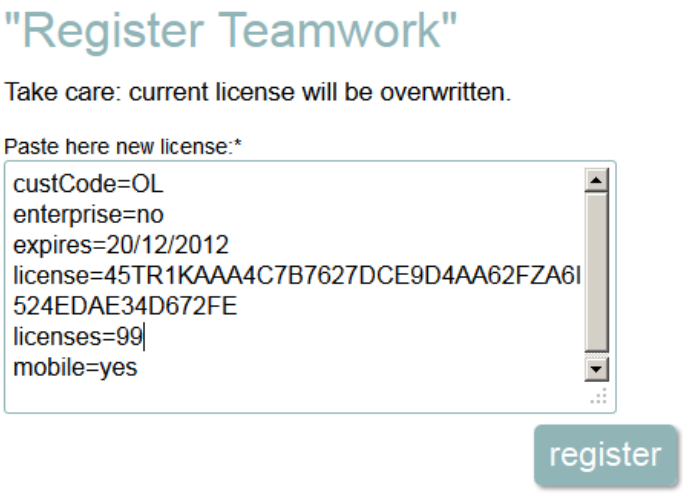
How to remove Teamwork’s service?

To remove the installed service on Windows and OSX’s, just launch the uninstaller executable in Teamwork’s installation folder; this of course will also remove the Teamwork application.

19.2.6 Inserting a non expiring license

Log in as administrator, go to admin -> register license (bottom right), a popup opens





where to paste the license.

19.2.7 Installation failed

Using the installer

If it is a new installation, and not just the web app (the default case), the installation log file installation.log is in

[installation folder]/.install4j

e.g. defaults on English Windows

C:\Program Files\teamwork\.install4j

or

C:\Program Files (x86)\teamwork\.install4j

If you are doing an update or installing as web app, the log file will be in

WEB-INF/.install4j

If from the log file you can't understand what the problem is, contact us through the web site and furum, see "19.9 Appendix: How to ask for help".

19.3 Hand/custom install

19.3.1 Copying a graphical installation

It may be necessary to install by hand because of your network or server settings, or because you are using an operating system which is not Windows, Linux or OSX, e.g. Solaris. If possible, do the graphical installation on some client machine that can access the database of the server, so that the installer can configure for you db access, license, http port and the other settings; doing all this by hand can be not trivial; then copy on the server.

The videos cover several cases: one is that you have already a web server running and a Java JDK installed, and you want to add Teamwork as a web app. The simplest way is to install using the installer on a client machine, with access to real database, as the installer will:

1) create the database schema
2) fill sample data
3) create a global.properties file with SMTP and similar settings
4) create a config.properties with the JDBC connection parameters
5) create a lic.properties file with the license data supplied

As all the files created are fine on any O.S., just copy the resulting webapp from webapps/ROOT in the webapp folder for Teamwork. Otherwise you will need to create all these by hand. A sample complete global.properties can be found as

webapps/ROOT/commons/settings/sampleGlobal.properties

19.3.2 Complete installation by hand

See this FAQ online:

http://answers.twproject.com/questions/3494/faq-installing-teamwork-5-by-hand

19.4 The structure of the obtained web server and application

If you are using the default installation, you will obtain a web application structure, here described.

19.4.1 Java

Java will be distributed in the JRE and JDK folders (only a small part of JDK is actually distributed).

19.4.2 The web server

Tomcat 6 is on the root of the distribution.

19.4.3 The database

By default, using the HSQLDB database, it is composed of the files in the webapps/ROOT/WEB-INF/data folder.

19.4.4 Optimizing performance

See *17.10 Performance*.

Transferring data from one database to another

Do not assume that this is a trivial operation: given the esistence of referential integrity, it often isn't. There are specific tools for data transfer from on database to another, refer to the producer's documentations.

Complete backup

Before any maintenance operation, and also periodically:

1. Backup your database

2. Backup the entire web application: TEAMWORK_HOME/webapps/ROOT

3. Backup your repository, where files get uploaded: if you have not customized this folder you can find it in TEAMWORK_HOME/teamworkRepository.

19.4.5 Starting and Stopping Teamwork

On Windows: stop and start the service

On Linux: use the teamworkLinux.sh with the parameters "stop", and the "start"

19.5 Platform specific requirements

19.5.1 Linux systems

Here Teamwork will not be installed as a service. You will need to set it up differently in function of the distribution.

19.5.2 Windows Vista/XP/7/8

Take care that the Personal Firewall is either off or allows Java to run.

19.5.3 Mac OSX

The “dmg” file does not bundle a Java Runtime Environment (JRE); it is assumed that version 5 or above of JRE is installed. The script that launches Teamwork on end is teamworkOsxStart.sh; if any “java.net.BindException: Permission denied:8080” error appears, you likely have a permission problem. If you prefer not to login as root, but are administrator, just use

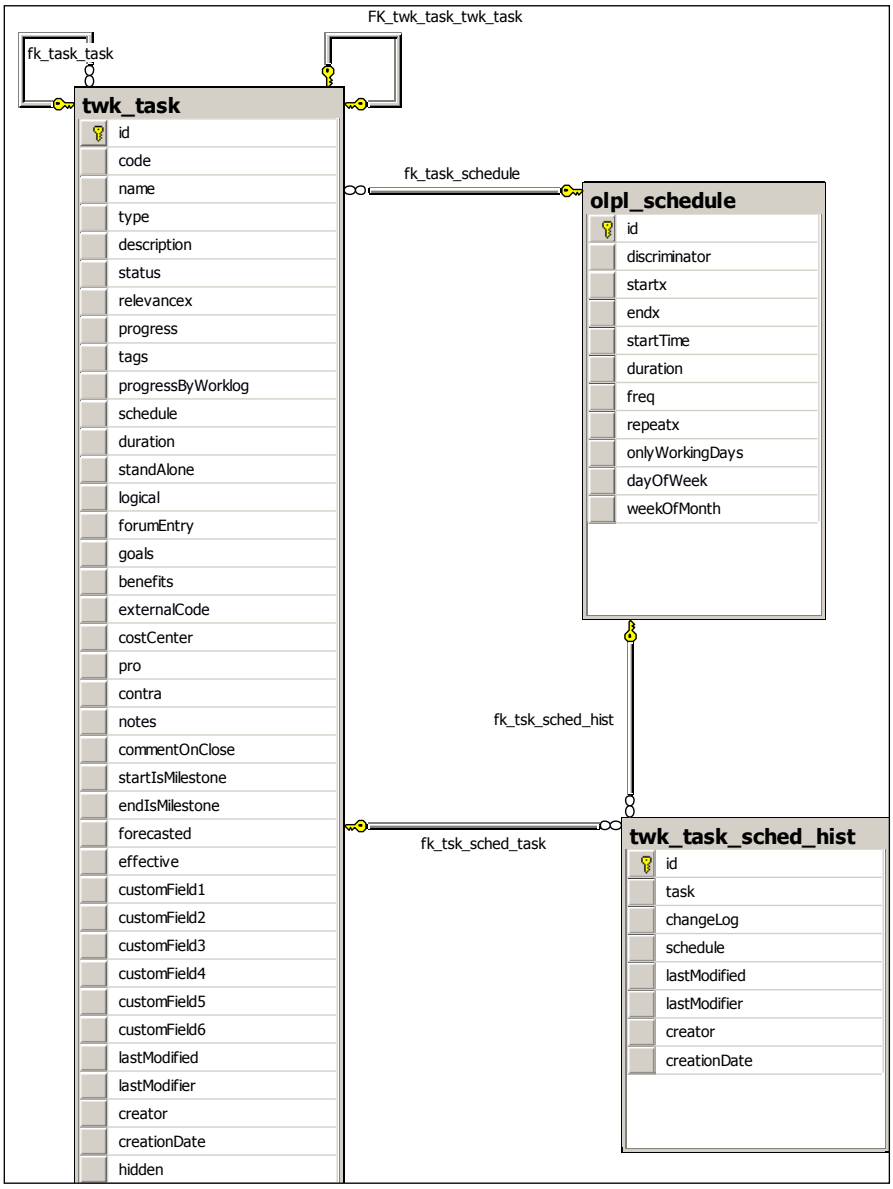
sudo ./ teamworkOsxStart.sh

19.6 Database structure

Here we outline Teamwork’s database relational structure. We assume familiarity with relational concepts. For readability and in order to give the essential information, we don’t cover complete schemas. E.g. relationship to the area and operator table are usually removed.

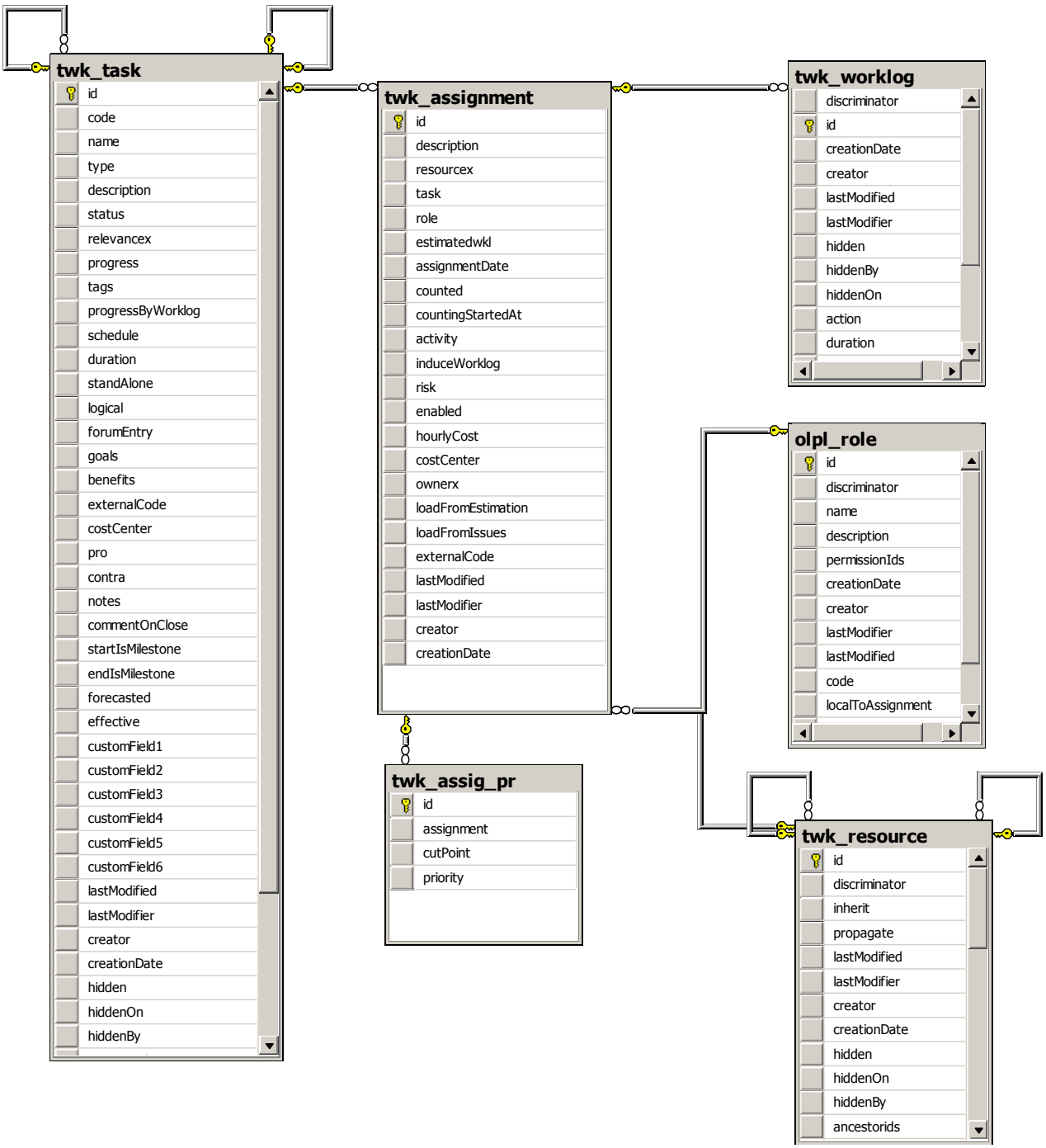
19.6.1 Basic task data and dates

Basic task data is in the twk_task table. Notice the “parent” column that is a foreign key that points to the same table, and determines the task position in the tree. The recursive values of the parents are also stored (denormalized) in the “ancestorids” field for performance reasons.



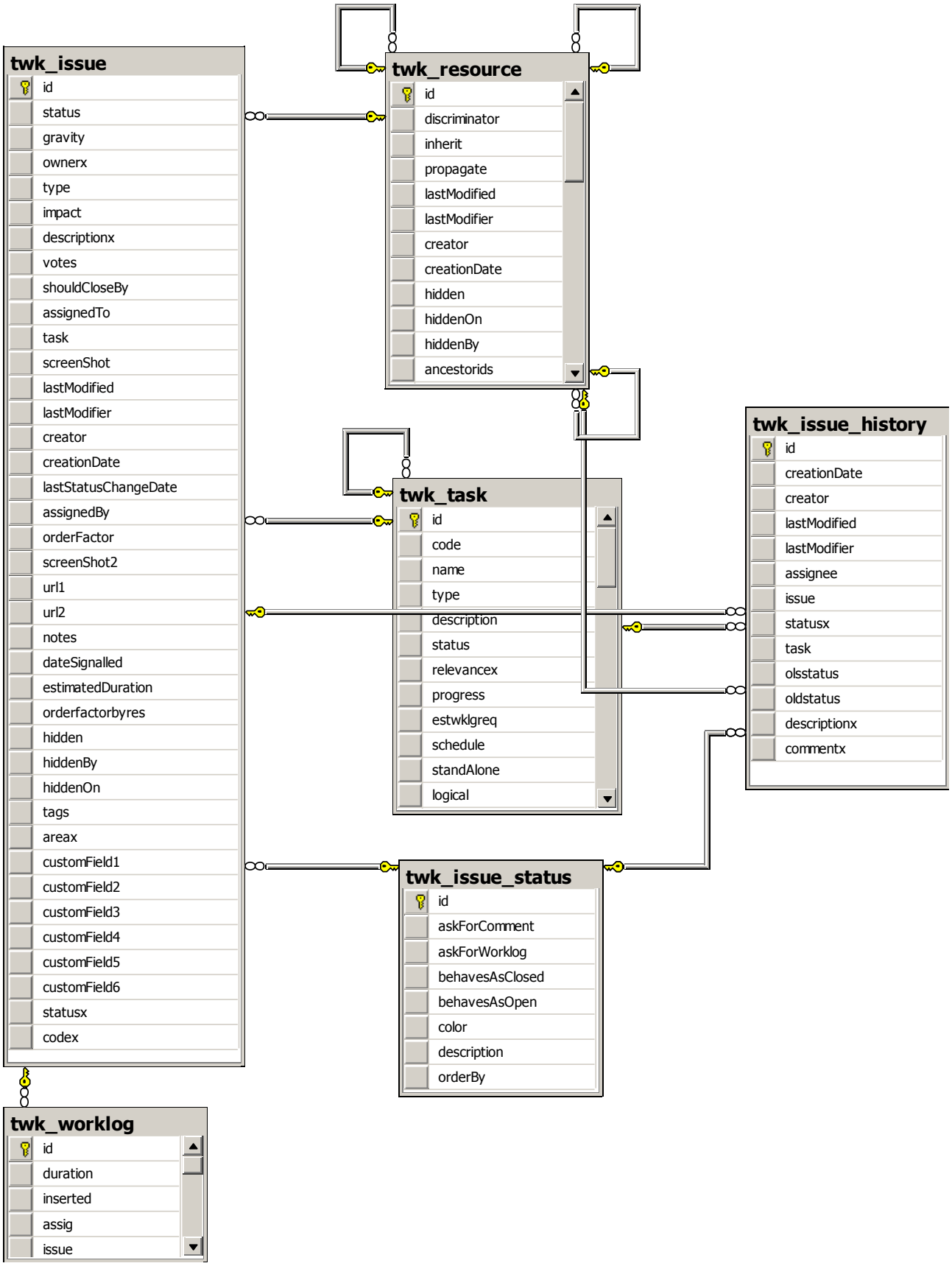
Project / task dates: the start, end and duration of the task are found by a join to the table olpl_schedule, to the filds startx, endx and duration respectively.

19.6.2 Task and assignments



The most important relation connecting tasks with people is the assignment. The twk_assignment table is a relational table that connects tasks to resources (people) and roles. Also all worklog inserted in Teamwork is “on” an assignment. So to query the worklog on a task, you will have to join to assignments and from there join to worklogs.

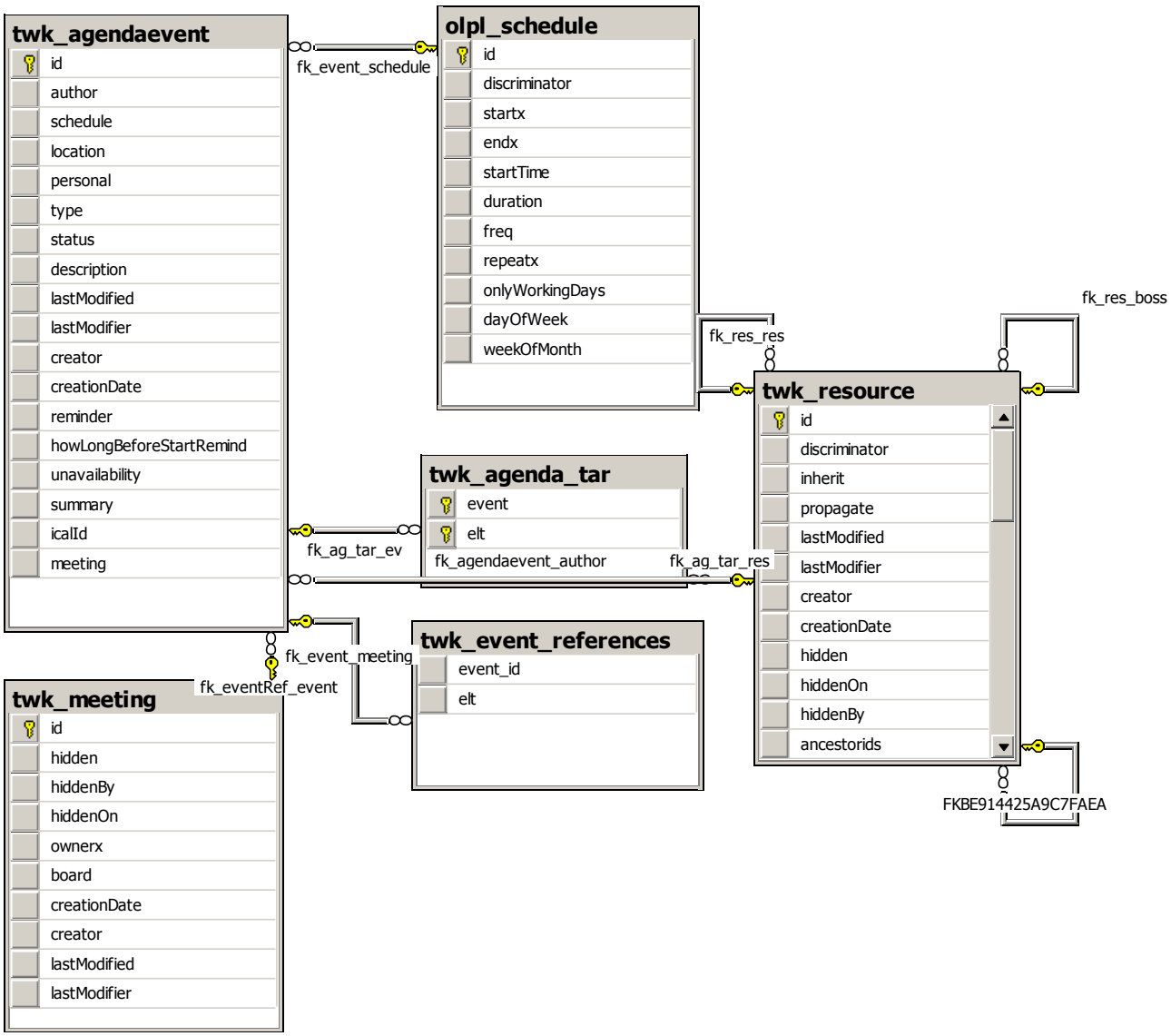
19.6.3 Issues



Issues can be without resource – just on a task, or without task – and then are a todo on a resource. But normally have a resource and a task, which indirectly most often determines an assignment, and that is where the worklog goes.

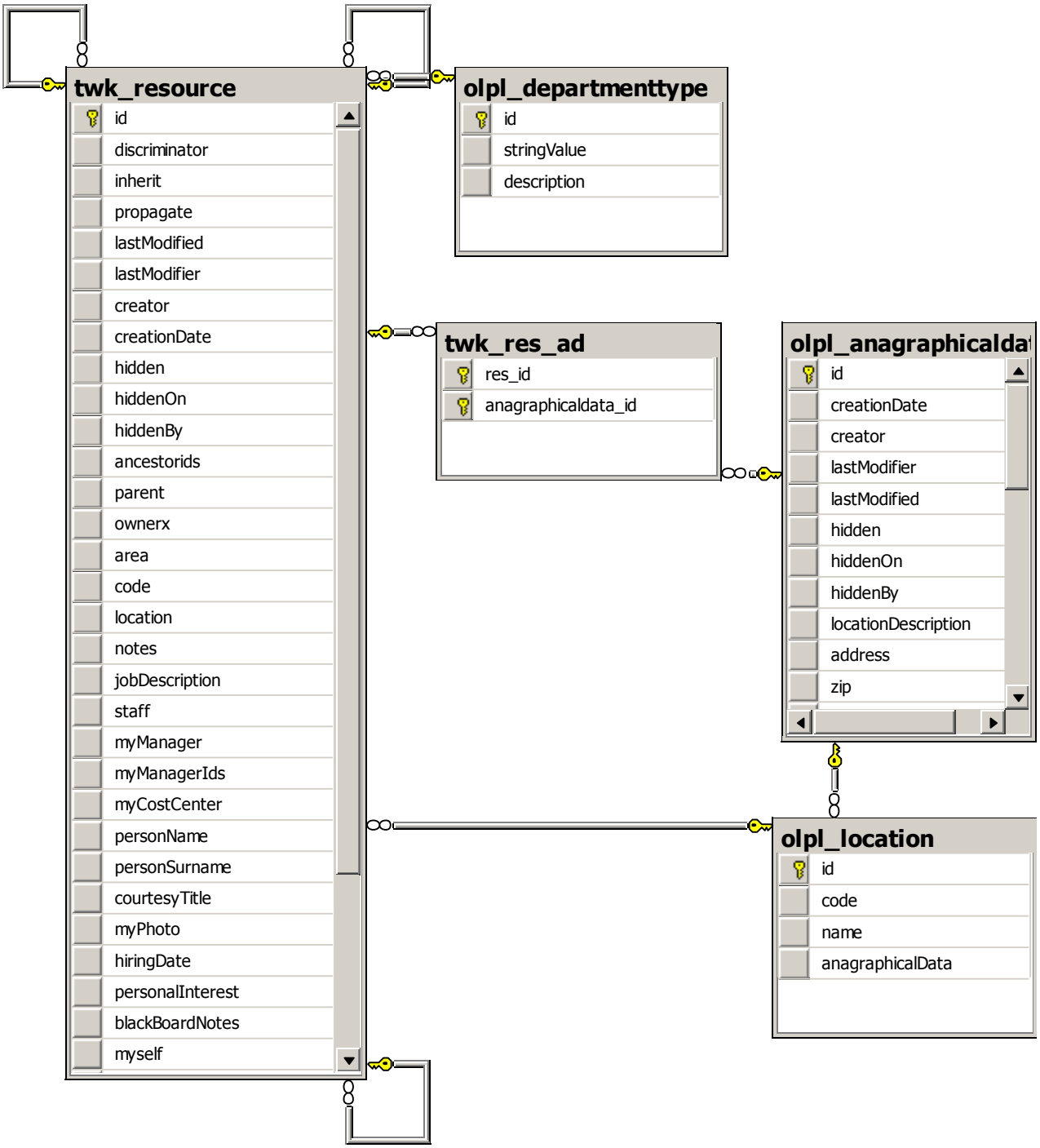
Notice that there is a denormalized reference to worklogs, and the table issue_history where the history of changes to the issue is kept.

19.6.4 Agenda



Every item in the agenda has a schedule and a set of participants. The schedule is directly joined, the participants are collected by joining through the twk_agenda_tar. If eventually the agenda event is also a meeting, there is a foreign key to the twk_meeting table.

19.6.5 Resource



Resources are obviously involved in most Teamwork tables; here we just show the direct dependencies on this table. Notice that the resource basic data (such as “e-mail”) is kept in a join though twk_res_ad to olpl_anagraphicaldata.

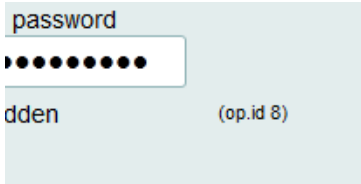
19.6.6 What is the logged user?

Teamwork logged users are “double” entities: they are “resources”, so that assignments can be done on them, and users of the system, so they can log in.

So a user that logs in is a record in the olpl_operator table, that is in 1-to-1 relationship with a record in the twk_resources table.

The resource record id is shown in the web interface in the resource editor general tab;

The operator record id in the security/login tab.



In almost all cases what you need to filter records is the resource record id: on issues, worklogs, assignments.

An exception is the subscription engine.

19.6.7 Filter by area?

Filtering by area is trivial, as most objects have an area or areax column for which you can filter.

19.7 First access

When you access Teamwork you have to authenticate using the login screen:

The default user login is administrator with no password:

LOGIN	administrator
PASSSWORD	[empty]

Set the password as soon as possible!

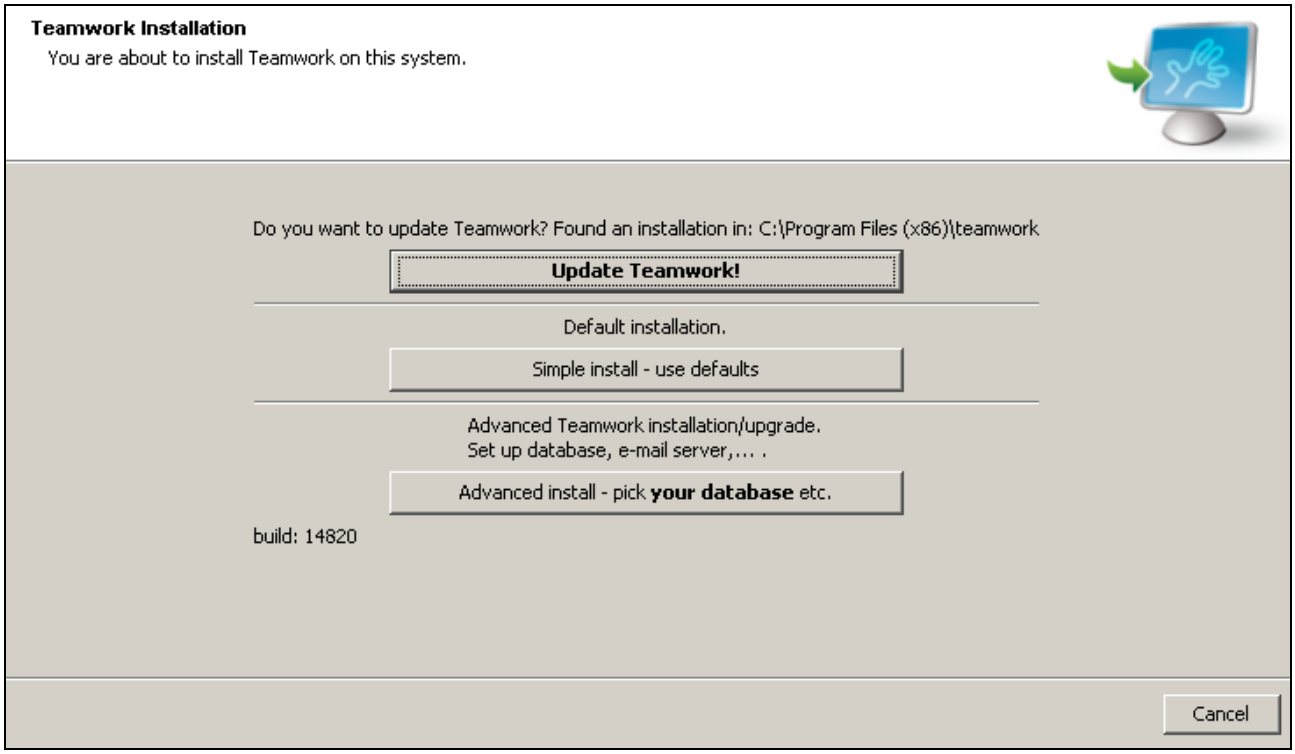
In order to integrate Teamwork with HTTP and/or LDAP authentication, see section “16.8 LDAP/Active Directory”.

19.8 Upgrading or reinstalling the application

For upgrading from alpha or beta versions, please contact us directly.

19.8.1 Upgrading using the installer

The installer contains a complete upgrade procedure:



19.8.2 Upgrading by hand

The only files that should not be replaced, eventually synchronized, for an upgrade are:

- [ROOT]/WEB-INF/lic.properties
- [ROOT]/WEB-INF/config.properties
- [ROOT]/commons/settings/global.properties

In order to upgrade by hand

1) make a complete backup of your application and data
2) stop the application server (Tomcat or other)
3) delete the directory WEB-INF/lib
4) extract the zip or tar.gz or rpm files, merging with the existing files. Take care that the pre-existing folders must be merged with the new ones, which is default on Windows, but not on Linuxes and OSXs.
5) remove the line SETUP_DB_UPDATE_DONE=yes from the global.properties file
6) restart the application server

The application on restart after upgrade does also a database upgrade.

19.8.3 Upgrading from 3.1 or 3.2.0 versions to 3.2.1 and following

Starting from version 3.2.1, Teamwork's startup procedure has changed.

Before 3.2.1: Teamwork started up on the first request: the file

…/commons/settings/settings.jsp

was loaded in background, and this in turn loaded

…/WEB-INF/config.properties

which contained JDBC connection data.

This jsp-based structure was built in order to give the possibility to customers/developers of "injecting" deeply customized settings without class recompilation. This has revealed to be a scarcely used feature, so we removed this, and made startup more robust.

Since 3.2.1: Teamwork is started on web app creation by a servlet configured as "load on startup". The servlets is TeamworkLoader, which reads configuration from

…/WEB-INF/config.properties

JDBC configuration inclusive.

Hence if proceeding by hand, JDBC connection data must be copied from twdb.properties into config.properties, which is what the installed tries to do in upgrade cases; but it is bound to fail in case of customized paths.

Note: We found out that Tomcat by default does not correctly encode URLs: in order to get correct encodings, you must set

```
URIEncoding="UTF-8"
```

In the connector configuration for example:

```
<Connector port="##DEFAULT_PORT##" maxHttpHeaderSize="8192"
               maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
               enableLookups="false" redirectPort="8443" acceptCount="100"
               connectionTimeout="20000" disableUploadTimeout="true" URIEncoding="UTF-8"/>
```

So the Tomcat provided in this latest release has such parameter, but on existing Tomcat you must set it by hand.

19.8.4 Upgrading to 4.5

Several JARS have been updated, added and removed. If they are present these JARs should be deleted by hand from WEB-INF/lib:

commons-collections-2.1.1.jar

commons-logging-1.0.4.jar

poi-3.0.1-FINAL-20070705.jar

jcaptcha-all-1.0-RC3.jar

Other jars (with same name) will be overwritten by an updated version.

In case you are using sources, the same thing must be done in your development project.

19.8.5 Upgrading to version 5

Two points are to be considered when upgrading from version 4 to 5:

1. Most of the Jars have been updated; if updating by hand, remember to clear the

WEB-INF/lib folder before updating (the installer will try to do it if you use that instead)

2. The Java JDK required is version 6 (version 7 is fine too). See this FAQ on how to upgrade the JDK:

   http://answers.twproject.com/questions/3112/faq-upgrading-java-jdk

   Note that the installer will attempt to update the lib folder by itself but it cannot update the JDK – YOU MUST DO IT YOURSELF.

Once the Teamwork web app starts, it will auto-update Teamwork schema and data (remember backups:

http://answers.twproject.com/questions/3115/faq-how-to-do-a-complete-teamwork-backup

).

19.9 Appendix: How to ask for help

INFORMATION:

- whether it is an installation or usage problem
- version of Teamwork you are using (e.g. 4.7 build 13.066)
- operating system and database you are using
- web server (e.g. Tomcat 6.0) you are using, and whether you are using your own, or the one provided in the download

FOR INSTALLATION PROBLEMS:

- if you are doing an installation, whether you are using the graphical installer, and if not, which archive you are using

FOR USAGE PROBLEMS:

- all the information found on the “system check” page, if available:
- tools -> administration -> debug system check
- check out [..]WEB-INF/log/platform.log for an error reported there for the intended page
- if a page doesn't seem to work, do "view source" of the html of the page and post the eventual error found
- check out the JavaScript errors of the page

LOG AND CONFIG FILES:

send us by mail, eventually cleared of sensitive data, and possibly zipped:

1) [..]commons/settings/global.properties file
2) [..]WEB-INF/log/platform.log
3) [..]WEB-INF/config.properties

If the platform.log file is huge, stop the web server, backup it, clean it, reproduce the error and send that file.

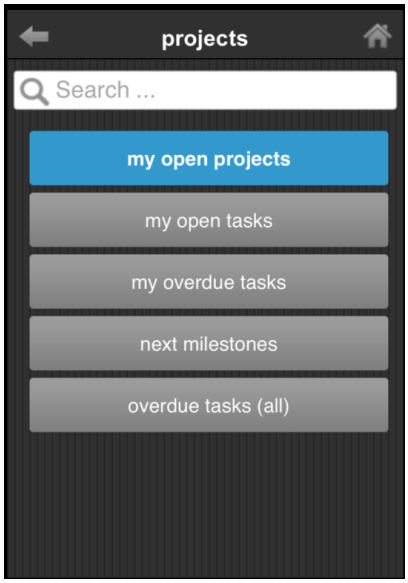
20. Teamwork Mobile

Teamwork Mobile is a new interface designed for accessing Teamwork with mobile devices. It allows reading all Teamwork data and insertion/editing of issues, worklogs and events in the agenda.

You won't need to install anything on your mobile: it is simply HTML5, which also facilitates updates. So if your users use different kinds of mobile devices, say some have iPhones, some Android devices and some Blackberries, all of them can use Teamwork mobile.

The mobile module can be used only if you have a running Teamwork instance updated to 4.7 – it is not a stand-alone application.

20.1 What you can do

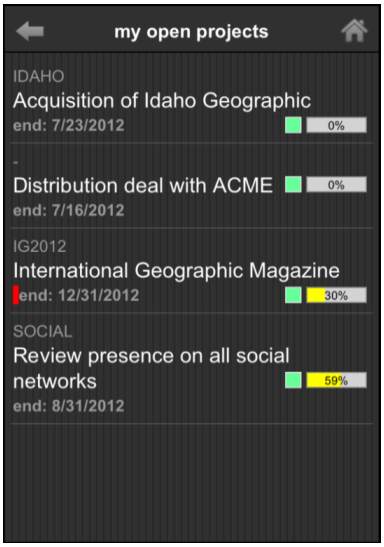


Access all your Teamwork projects, issues, worklogs, events. Add and close issues, record worklogs, add events. Read documents. And a nice last touch, you can access Teamwork's file storages, so Teamwork is acting as a gateway for you mobile access to your intranet documents.

20.2 Where it works

Teamwork mobile works on the following devices: iPhone 3/4, iPad, and on updated Android and Blackberry (do test your device on the demo).

20.3 How to try it



Just load up the web browser on your mobile and visit the online demos:

http://www.twproject.com/onlinedemo.page

Then with your mobile device just go to this URL:

http://demous.twproject.com/mobile

or

http://demo.twproject.com/mobile

respectively.

For a better user experience add the Teamwork page to your

Home screen.

20.4 How to buy

The mobile module is not included in your Teamwork 5 license, also not included in unlimited users licenses.

The mobile module has a fixed price, 990 Euro per server – 490 Euro if you are using yearly licenses. So you can buy it at a fixed price and use it with all the Teamwork accounts from your Teamwork installation. The mobile license module does not expire.

You buy it here: http://www.twproject.com/licensePrices.page

The license you will receive after purchase (usually within two working days at most) must substitute your existing Teamwork license.

All new demo licenses generated on Teamwork web site include the mobile module.

21. Teamwork from 1 to 5

21.1 A very short story of Teamwork releases

Mid 2001	First version of Teamwork, “Teamwork 1” is realized in Microsoft ASP and used in Open Lab
End 2002	Teamwork 2 development is started in Java/JSP
Mid 2004	Teamwork 2 released
7th January 2005	Teamwork goes open source, under LGPL license; remains such until June, 2006 (18 months): it’s a hit application on Sourceforge, with 38.000 downloads
April 2005	Development of Teamwork 3 is started
June 2005	Teamwork 2 final release
6th June, 2006	Teamwork 3 is out (15 months development)
November 2006	Teamwork 3.1 released
March 2007	The online Teamwork service is available JOLT Awards
June 2007	Teamwork 3.2.released
September 2008	Last Teamwork 3 release: 3.2.9
February 2009	Teamwork 4.0 is out
January 2010	Teamwork 4.3: multi Gantt support, import from Bugzilla
February 2010	Teamwork 4.4: public project pages, Chinese interface
June 2010	Teamwork 4.5: most of Teamwork has been extended and improved. Still, a free upgrade.
December 2010	Teamwork 4.6: released Mobile module
June 2011	Teamwork 4.7: released Issue Kanban / Organizer
March 2012	Teamwork 4.8: Added Issue Planner
July 2012	Teamwork 5 released and also the free online Gantt editor
November 2012	Teamwork 5.1 with new trees and improved model for staffing
March 2013	Teamwork 5.2 with stronger support for customizations and Japanese interface.
October 2013	Teamwork 5.3 supports document uploads via drag’n’drop in the browser and comments on issues.

21.2 Teamwork conceptual evolution leading to version 5

Teamwork's design and user interface has evolved in time, adapting to the evolution of web based tools and user needs and habits.

Version 3. In version 3 Teamwork was a solid and standard web application, before web applications were widely used for management.

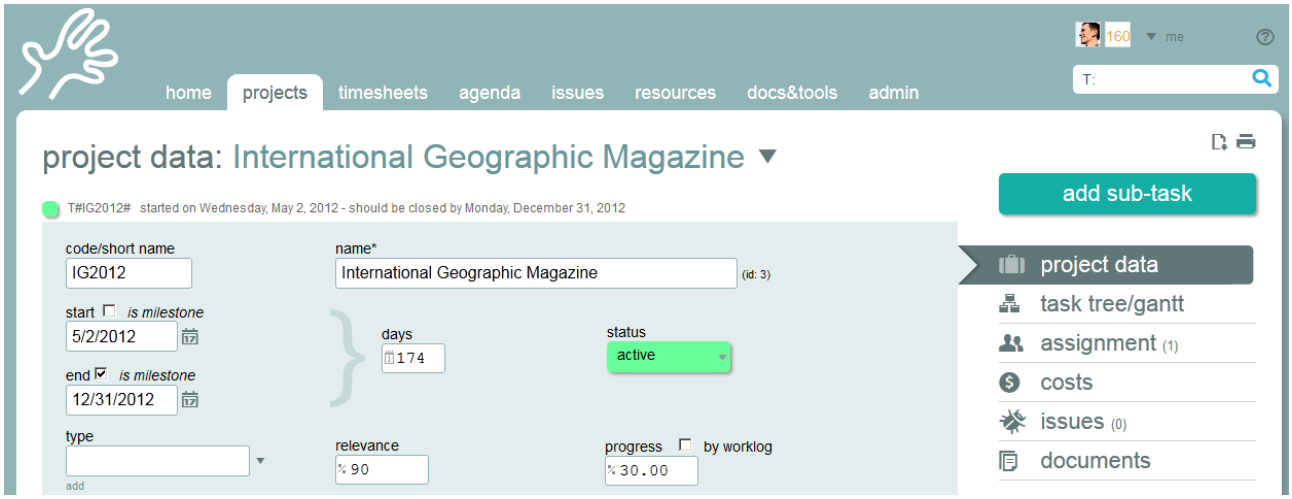
Version 4. With Version 4 we introduced gamification elements and more explicit functions. We added scores, skins, prizes and badges years before gamification became a buzzword.

Version 5. The idea of version 5 has been to get a clean interface, using large, nice web fonts, whitespace and an unbounded bottom of page, assuming that users are used to scrolling. We removed tabs, creating more readable dedicated pages to sections.

We removed gamification layer and skin customization, in order to get a clean and more controlled design. We shortened click paths with overlay layers. We sped up pages using data denormalization when necessary.

We identified users with their images, making it easier to "see" who is working where.

In version 5 we have more functions and less buttons. We hope you like it!

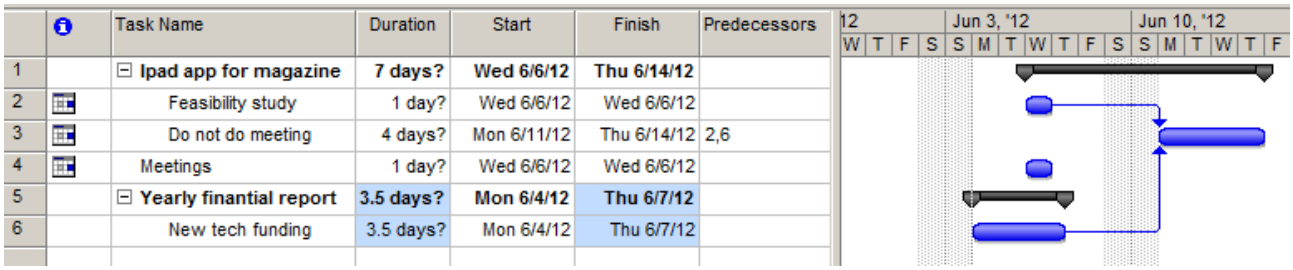


21.3 What's new in version 5

Teamwork version 5 has a new user interface and several new functionalities. The work model is mostly unchanged from version 4.

- Introduced Gantt-like editor

Also supports dependencies between any two tasks, not just between brother tasks.



	Task Name	Duration	Start	Finish	Predecessors
1	Ipad app for magazine	7 days?	Wed 6/6/12	Thu 6/14/12	
2	Feasibility study	1 day?	Wed 6/6/12	Wed 6/6/12	
3	Do not do meeting	4 days?	Mon 6/11/12	Thu 6/14/12	2,6
4	Meetings	1 day?	Wed 6/6/12	Wed 6/6/12	
5	Yearly finantial report	3.5 days?	Mon 6/4/12	Thu 6/7/12	
6	New tech funding	3.5 days?	Mon 6/4/12	Thu 6/7/12	

getting

	code	name	start	end	dur.	dep.
1	-	IG_iPad	6/4/2012	6/14/2012	9	
2	5	Yearly finantial report	6/4/2012	6/7/2012	4	
3	6	New tech funding	6/4/2012	6/7/2012	4	
4	1	Ipad app for magazine	6/6/2012	6/14/2012	7	
5	2	Feasibility study	6/6/2012	6/6/2012	1	
6	4	Meetings	6/6/2012	6/13/2012	6	
7	3	Do not do meeting	6/8/2012	6/13/2012	4	5,3

- Introduced Issue planner

- Assignment enriched with assignment specific costs (expenses)

- Statistics graphs improved and extended

    A line graph on the task editor which shows total worklogs / issues done.

- Heuristics on project progress / state

- Date and number formats relative to user profile (on top of user interface language, which was always customizable)

- Denormalization to achieve speed

- Speed improved in general

- Conceptual simplification of planning and estimation and more fine grained tools

- Shorter click paths with layers

- Amazon S3 file storages

- Custom reports on all Teamwork sections

- Agenda and meeting completely redesigned

- Workgroups visibility / association with images

- Technical: libraries updated to Hibernate 4.

- Released Teamwork Gantt, a free online tool for creating Gantts, you find it here:

http://gantt.twproject.com/

Of course Gantts created online can be imported in Teamwork 5 – just go to projects and select “import from Teamwork Gantt”.

- Flow management has been extended:

- for example worklog estimated in the time assignment can be set. Also added attributes to workflow descriptions that allow setting duration of tasks and estimations of assignments.
- clone tasks can clone also the assignment for the business process
- when writing flows you can automate assignments by “instantiator” and project manager:

```
<task-node name="Boss Approval">
    <task>
        <assignment class="com.twproject.task.process.ScaleToBossAssignmentHandler">
            <role>Project Manager</role>
            <estimated>1:50</estimated>
        </assignment>
    </task>
```

The classes to check are
com.twproject.task.process.InstantiatorSwimlaneAssignmentHandler
com.twproject.task.process.ScaleToBossAssignmentHandler

N.B. Complete Teamwork sources (under NDA) are available for free to all customers.

More news can be found on Teamwork’s blog: http://blog.twproject.com/

22. Open Lab products and initiatives

Open Lab produces Teamwork and other products. It also sponsors several on line initiatives. See http://www.open-lab.com/

Licorize is an online service that integrates Delicious – like browser tagging in a to-do manager context. See http://licorize.com .

23. Index

A

Active Directory · **227**
agenda · **151**
  configuration · 157
  day · 156
  month · 157
  recurrent events · 152
  unavailability · 151
  week · 151
Agile · 54, 66
  Scrum · 54
Apache Directory Server · 227
apache tomcat · 245
areaname · 178
areas · **184**
  security · 183
  usage example · 187
assignee · 65, **66**, **69**, **72**
assignment · 65
  activity · 66
  enabled · 66
  risk · 66
authentication
  http · 214
  ldap · 215
  standard · 214

B

backup · **238**
  application · 238
  database · 238
  repository · 238
Basecamp · 89
board · **119**
buddy · 36
budget · 74
bugs · 103
bulk management
  issues · 110
business process · 93
  advanced · 206
  create your own · 209
Business processes
  JBPM · 93

C

channel · *Vedi*
code
  resources · 44
  task · 55
company · 43, **49**
contact list · 43
copy/move
  task · 99
cost · 73
counters · 192, 193
custom
  portlet customization · 178
custom fields · **196**
custom filters · 51, 192
custom forms
  create your own · 198
customize label · 223

D

dashboards · **172**
  customization · 173
date
  shortcuts · 190
default
  date format · 48
  home page · 48
  language · 48
  skin · 48
  working days · 56
deliverables · 58
department · 43, **49**
dependencies · 56
diary · 91
discussion point · 162
discussion points · 161
document management · **163**
duration · 57

E

e-mail
  adding documents to tasks · 120
  adding issues to tasks · 121
  adding to-dos · 122
  configuration · 215
  manage by · **120**
estimation
  issue · 105
  issues · 66
event · 235

F

file server · 163
file storages · 163
file system · 165
firewall · 214
flowork · 206
fork · 93
forum · 91
FS · 165
full-text · **219**

G

google calendar · 151, **159**

H

hibernate · 93
holidays · **221**
home page
  customization · 173
HsqlDB · 245

I

iCal · 151, **157**
Ingres · 245
install
  by hand · 247
  installer · **245**
internationalization · 223
Issues · **103**

J

java · 245
jbpm · 93
JBPM · 207
join · 93
jpdl · 93
JPDL · 207

L

LDAP · **46**, **215**, **227**, **256**
  additional properties · 230
  F.A.Q. · 231
  import users · 229
license
  register new licence · **237**
licensing · 237
listener · 235
lock
  document · 169
log4j · 234
Logging · **234**
logs
  system · 234
  tasks · 92

M

MacOSX · 246
mandatory field · 44
meeting · 161
  minute · 161
  report · 162
merge
  issues · 111
message · 235
messaging system · **235**
milestone · 57, 93
  configuration · 69
  flow · 93
move to task
  issues · 111
mpx · 87
MS Project · 86
my colleagues · 51
MySql · 245

N

NAS · 163
nat · 214
news
  company news · 172

O

OpenLdap · 227
operator
  work load · 124
Oracle · 245
organigram · 43
outlook · **120**, 151, **157**

P

paging bar · 51, 99, 109
password · 46
  empty · 33
  forgotten · 35
permission · **186**
plan
  task · **127**
plugin
  dissection · 199
portal · **172**, **226**
portlets · 173
  creation · 175
progress · 58
progressive number · 55
project defaults · 222
propagation · 58

Q

QBE · 52, 189
query by example · 189

R

recurrent event · 152
relevance · 58
repository · **169**
re-scheduling
  issues · 108
resource · **43**
  editor · 43
  hidden · 46
risk · 66
role
  global role · 46
  local role · 65
roles · **185**

S

Scheduler · **233**
search
  box · 188
  issues · 108
  shortcuts · 188
skeleton · 56, 100
snapshots · 82
SQL Server · 245
status
  issues · 105
steps
  execution · 209
  history · 209
subscriptions · 48, 68
subversion · 142, 165
SVN · 142, 165

system check · 236
system requirements · 245

T

tags
  tasks · 56
task · **54**
  cost · 73
  definition · **54**
  deleting · 63
  duration · 57
  milestone · 57
  search · 98
  status · 56
  summary bar · 59
  template · 56
  templates · **100**
template
  dashboard · 178
tickets · 103
time counters · **139**
time recording · **137**
timesheets
  week · **137**
to-dos · 103
tree checks · **236**

U

upgrading
  by hand · 257
  installer · 257
upload
  document · 169
user defaults · 221
users
  disabled · 47

V

VCard · 45

W

work
  load · 124
  plan · **127**
  priority · **129**
work log · 58
  analysis · 147
  counters · **139**
  issues driven · **140**
  misplaced · 70
  missing · 146
  monitoring · **145**
  overflow · 70
  svn · 142
workflow · 93, 206, 207
workgroup · **117**
working days · 56

Z

zip · 166