



TaskJuggler III

Project Management beyond Gantt Chart drawing

Chris Schlaeger
TaskJuggler Developer
cs@taskjuggler.org

Version: 1.0

Date: 2008-03-01

© 2008 Chris Schlaeger

Licensed under the GNU Free Document License 1.2





What is TaskJuggler?

- A new kind of Project Management Software
- Website: <http://www.taskjuggler.org>
- Supported Features include
 - Task and Resource Planning
 - Progress and Status tracking
 - Cost Accounting
 - Powerful reports
- It's not just a Gantt Editor
- Open Source software licensed under GPL v2
- Active user community from academia and industry
- Discussion forums are available on the website



TaskJuggler History

- Project was founded in 2001 by a small team of developers working for SUSE
- Until 2004 it was command line only
- New graphical user interface was released in March 2005
- TaskJuggler III rewrite started end of 2006 using Ruby as primary programming language
- First stable TaskJuggler III release is expected for Q4 2008



TaskJuggler 2.4.1

Interactive Demonstration of the following features:

- Launching TaskJuggler from the menu
- Creating a new Project from a template file
- Access to the manual and the F2 keyword help
- Editor Settings
- Auto-completion and indentation
- Date insertion and modification with CTRL-Shift-T
- Scheduling a project
- Handling syntax and scheduling errors
- Explanation of the list browsers and the editor and report tabs
- Loading of an existing Project
- Browsing of various reports
- Demonstration of keyboard navigation to all GUI elements



New Features in TaskJuggler III

- More custom attribute types (Strings, Dates, Numbers)
- More flexible accounting system
- Start->Start and End->End dependencies
- Support for RichText Strings
- Almost all attributes are now scenario specific
- Much improved HTML reports
- Single report generator for all reports
- New reference manual



Basic Components of a Project

- Project Header

```
project myProject    My Project    1.0
                2005-11-01 +3m
```

- Task Definitions

```
task prjstart    Project Start    {
    start 2005-11-05
}
task step1    Step 1    {
    depends prjstart
    duration 2w
}
```



Specifying Task Dependencies

- Task B depends on Task A

```
task a    Task A    { start 2005-11-01 }
task b    Task B    { depends a { onstart } }
```

- Task A precedes Task B

```
task a    Task A    { precedes b }
task b    Task B    { start 2005-11-01 }
```

- Relative and absolute Dependency Specifications

```
task p    Project   {
  task a    Task A    { start 2005-11-01 }
  task b    Task B    { depends !a }
  task c    Task C    { depends p.b }
}
```



Specifying a Task Duration

- Three methods of specifying a duration directly
 - Calendar time: `duration`
 - Working time: `length`
 - Resource time: `effort`
- Must be used with a unit
`min, h, d, w, m, y`
- Examples:
 - `duration 1.5y` (1.5 calendar years)
 - `length 2w` (10 working days)
 - `effort 2w` (10 resource days)
- Conversion factors: `yearlyworkingdays`,
`dailyworkinghours`



Defining Resources

- Resource definitions have similar format as task definitions
- Many attributes get inherited from enclosing scope.

```
resource team    Developer Team    {
    rate 300      # daily cost
    resource john    John Doe
    resource wilma   Wilma Flintstone
    resource paul    Paul McCartney  {
        rate 250
    }
    resource projector    Projector    {
        efficiency 0
    }
}
```



Assigning Resources to Tasks

- Simple case: 1 task, 1 resource

```
task foo    Foo Task    {  
    effort 5d  
    allocate joe  
}
```

- Allocating multiple resources

```
task foo    Foo Task    {  
    effort 10w  
    allocate wilma, paul, joe  
}
```



Assigning Resources to Tasks (Cntd.)

- Allocating a team

```
task foo    Foo Task    {  
    effort 5d  
    allocate team  
}
```

- Mandatory allocations

```
task foo    Foo Task    {  
    effort 10w  
    allocate wilma  
    allocate projector { mandatory }  
}
```



Using Task Priorities to control the Scheduling

- The priority attribute controls the probability that a tasks gets the allocated resources
- The default priority is 500

```
task secUpds    Security Updates    {
    duration 2m
    allocate paul
    limits { dailymax 1h }
    priority 700
}
task calls      Handle customer calls {
    duration 2m
    allocate paul
    priority 300
}
```



Generating a Report

- Available Format Types
Interactive, HTML, CSV, Export
- Available Content Types
Task Reports, Resource Reports
- Definition of a Report

```
htmlresourcereport    ResourceList.html
```

```
htmltaskreport      ProjectOverview.html    {  
    columns no, name, start, end, chart  
}
```



Using Flags to mark Objects

- Use flags to mark certain groups of tasks.
- Flags must be declared before they can be used.
- Flag names are TaskJuggler IDs. They must consist only of letters, numbers and underscores.

```
flags important
```

```
task foo    Foo Task    {  
    flags important  
}
```



Filtering of Report Content

- Content of reports can be limited to the exact amount needed.
- Show only columns you are interested in
- Show only tasks you are interested in
`hidetask important`
- Show only resources you are interested in
`hideresource ~team`
- Sort content by up to 3 criteria
`sorttasks tree, start.up, id.up`
`sortresources plan.effort.up`



Excluding details from reports

- Limiting the report period

```
taskreport Task List {  
    period 2001-12-01 +2w  
}
```

- Excluding tasks or resources

```
hidetask <LOGICAL EXPRESSION>  
rolluptask <LOGICAL EXPRESSION>  
hideresource <LOGICAL EXPRESSION>  
rollupresource <LOGICAL EXPRESSION>
```




Logical Expressions for Filters

- Logical Filter expressions consist of flags, functions and operators
- Supported operators (subset):
 - & (and), | (or), ~ (not), > (larger), < (smaller)
- Expressions can be grouped with parentheses
- Many query functions supported

e.g. `isChildOf(ID)`, `isMilestone()`, `treeLevel()`

- Example:

```
taskreport    Task Overview (Important ones)    {  
    rolluptask (treelevel() > 1) & ~important  
}
```



Working with Macros

- Macros are somewhat flexible text fragments that can be inserted multiple times once they have been defined.
- Macro names must have at least one uppercase letter

- Definition of a Macro

```
macro allocateGrp [ allocate john, wilma ]
```

- Using a defined Macro

```
task foo    Foo Task    {  
    effort 20d  
    ${allocateGrp}  
}
```



Working with Macros (Cntd.)

- Parts of macros can be replaced during insertion time by using parameters.

- Definition of a Macro with parameters

```
macro defTask [ task ${1}    ${1} Task    ]
```

- Calling a Macro with parameters

```
${defTask    foo    }
```

```
${defTask    bar    }
```

- Result of the expanded Macros

```
task foo    foo Task
```

```
task bar    bar Task
```



Adding Information to Reports

- Adding a Headline

```
taskreport    Task List    {  
    headline  The ''tasks'' of my project  
}
```

- Adding a Caption

```
htmlresourcereport    Resources.html    {  
    caption    List of all the hard working men and  
              women on the project.  
    * Joe  
    * Marry  
    For more details see [ResourceDetails].  
}
```



User Defined Attributes

- The attribute set of tasks, resources and accounts can be extended by the user.
- There are several types of attributes available
 - Strings, Numbers, Dates and URLs
- User defined attributes do not impact the scheduling. They are for documentation and reporting purposes only.
- They can be used in all reports like the build-in attributes.
- User Defined attribute IDs must start with a capital letter



User Define Attributes (Cntd.)

```
project ca "Custom Attributes" "1.0" 2003-05-28 -
    2003-06-28 {
    extend task {
        reference MyLink "My Link"
        text MyText "My Text"
    }
}
```

```
task t "Task" {
    start 2003-05-28
    MyLink "http://www.taskjuggler.org" { label "TJ Web"
    }
    MyText "TaskJuggler is great!"
}
```



The End

Thanks for attending!

A copy of the slides is available from the
TaskJuggler web site at
<http://www.taskjuggler.org/>



© 2008 Chris Schlaeger

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts. A copy of the license is available at <http://www.fsf.org/licensing/licenses/fdl.txt>.

