# TaskJuggler 3.x - Project Management for Linux Users

## Project Management beyond Gantt Chart Drawing

Chris Schlaeger

chris@linux.com

Chemitzer Linux-Tage 2012

(english translation by Sebastian Tramp)

# TaskJuggler and Linux

- TaskJuggler was initially developed to support the development process of a Linux distribution
- It is still in use by some distributions such as Fedora
- It uses a simple project description language
- TaskJuggler works similar to a compiler or LaTeX
- It consists of a few tools which provide the main features and are integrated with other programs

# Ruby Runtime Environment

- TaskJuggler 3.x is written in Ruby
- The main functions are Ruby 1.8.7 compatible
- The server and scripting functions need at least Ruby 1.9.3
- We generally suggest to use Ruby 1.9.3 since it is 3 times faster than Ruby 1.8.7

# Gem Installation

- Ruby is available on any Linux distribution
- Unfortunately many distributions provide only Ruby 1.8
- RubyGem is the Ruby package management tool
- Ruby gems are OS and CPU architecture independent
- The installation is usually easy. A single command downloads the gem package as well as its dependencies and installs everything in you system:

```
gem install taskjuggler
```

# Ruby 1.9 Installation

- The current version is available at `ruby-lang.org`
  `http://www.ruby-lang.org/en/downloads/`
- decompress, configure and install

```
tar -Zxvf ruby-X.X.X-*.tar.gz
cd ruby-X.X.X-*
./configure --program-suffix=19
make
sudo make install
ln -s /usr/local/bin/ruby19 ${HOME}/bin/ruby
```

- `${HOME}/bin` has to be part of your `$PATH` variable
- Each major Ruby version has its own gem package collection

# Ruby 1.9 Installation

- The current version is available at `ruby-lang.org`
  `http://www.ruby-lang.org/en/downloads/`
- decompress, configure and install

```
tar -Zxvf ruby-X.X.X-*.tar.gz
cd ruby-X.X.X-*
./configure --program-suffix=19
make
sudo make install
ln -s /usr/local/bin/ruby19 ${HOME}/bin/ruby
```

- `${HOME}/bin` has to be part of your `$PATH` variable
- Each major Ruby version has its own gem package collection

# Ruby 1.9 Installation

- The current version is available at `ruby-lang.org`
  `http://www.ruby-lang.org/en/downloads/`
- decompress, configure and install

```
tar -Zxvf ruby-X.X.X-*.tar.gz
cd ruby-X.X.X-*
./configure --program-suffix=19
make
sudo make install
ln -s /usr/local/bin/ruby19 ${HOME}/bin/ruby
```

- `${HOME}/bin` has to be part of your `$PATH` variable
- Each major Ruby version has its own gem package collection

# Ruby 1.9 Installation

- The current version is available at `ruby-lang.org`
  `http://www.ruby-lang.org/en/downloads/`
- decompress, configure and install

```
tar -Zxvf ruby-X.X.X-*.tar.gz
cd ruby-X.X.X-*
./configure --program-suffix=19
make
sudo make install
ln -s /usr/local/bin/ruby19 ${HOME}/bin/ruby
```

- `${HOME}/bin` has to be part of your `$PATH` variable
- Each major Ruby version has its own gem package collection

# Online Help: `tj3man`

- Get help for a single TaskJuggler keyword

  `tj3man <keyword>`
- Use `-html` to load it in the browser

  `tj3man --html <keyword>`
- Get help for the online help tool

  `tj3man --help`

# Online Help: `tj3man`

- Get help for a single TaskJuggler keyword

  `tj3man <keyword>`
- Use `-html` to load it in the browser

  `tj3man --html <keyword>`
- Get help for the online help tool

  `tj3man --help`

# Online Help: `tj3man`

- Get help for a single TaskJuggler keyword

  `tj3man <keyword>`
- Use `-html` to load it in the browser

  `tj3man --html <keyword>`
- Get help for the online help tool

  `tj3man --help`

# The main program: `tj3`

- Calculate a project and create reports

  ```
  tj3 yourproject.tjp
  ```

- Projects can consist of more than one file

  ```
  tj3 yourproject.tjp reports.tji
  ```

# The main program: `tj3`

- Calculate a project and create reports

  ```
  tj3 yourproject.tjp
  ```

- Projects can consist of more than one file

  ```
  tj3 yourproject.tjp reports.tji
  ```

# Client and Server: `tj3client` and `tj3d`

- Start the server

  ```
  tj3d -w
  ```

- Query the server status

  ```
  tj3client status
  ```

- (Re-)load a project

  ```
  tj3client add yourproject.tjp
  ```

- Shutdown the server

  ```
  tj3client terminate
  ```

# Client and Server: `tj3client` and `tj3d`

- Start the server

  ```
  tj3d -w
  ```

- Query the server status

  ```
  tj3client status
  ```

- (Re-)load a project

  ```
  tj3client add yourproject.tjp
  ```

- Shutdown the server

  ```
  tj3client terminate
  ```

# Client and Server: `tj3client` and `tj3d`

- Start the server

  ```
  tj3d -w
  ```

- Query the server status

  ```
  tj3client status
  ```

- (Re-)load a project

  ```
  tj3client add yourproject.tjp
  ```

- Shutdown the server

  ```
  tj3client terminate
  ```

# Client and Server: `tj3client` and `tj3d`

- Start the server

  ```
  tj3d -w
  ```

- Query the server status

  ```
  tj3client status
  ```

- (Re-)load a project

  ```
  tj3client add yourproject.tjp
  ```

- Shutdown the server

  ```
  tj3client terminate
  ```

# The Project Description

- All project data is collected in one ore more text files
- Information is structured as *properties* which consists of *attributes*
- A single *property* has always the following structure

```
PROPERTY [ID] "NAME" [ { ATTRIBUTES } ]
```

- Elements surrounded by `[]` are optional

# The Properties

A project consists of the following *properties*

- `project`: the project head
- `accounts`: accounts for cost calculation
- `resources`: human and non-human resources
- `tasks`: the project tasks
- `reports`: reports to generate

All *properties* have *attributes*, e.g. the *property* `task` has the *attributes* `start`, `duration` as well as other properties.

# The Properties

A project consists of the following *properties*

- `project`: the project head
- `accounts`: accounts for cost calculation
- `resources`: human and non-human resources
- `tasks`: the project tasks
- `reports`: reports to generate

All *properties* have *attributes*, e.g. the *property* `task` has the *attributes* `start`, `duration` as well as other properties.

# The Properties

A project consists of the following *properties*

- `project`: the project head
- `accounts`: accounts for cost calculation
- `resources`: human and non-human resources
- `tasks`: the project tasks
- `reports`: reports to generate

All *properties* have *attributes*, e.g. the *property* `task` has the *attributes* `start`, `duration` as well as other properties.

# The Properties

A project consists of the following *properties*

- `project`: the project head
- `accounts`: accounts for cost calculation
- `resources`: human and non-human resources
- `tasks`: the project tasks
- `reports`: reports to generate

All *properties* have *attributes*, e.g. the *property* `task` has the *attributes* `start`, `duration` as well as other properties.

# The Properties

A project consists of the following *properties*

- `project`: the project head
- `accounts`: accounts for cost calculation
- `resources`: human and non-human resources
- `tasks`: the project tasks
- `reports`: reports to generate

All *properties* have *attributes*, e.g. the *property* `task` has the *attributes* `start`, `duration` as well as other properties.

# The Properties

A project consists of the following *properties*

- `project`: the project head
- `accounts`: accounts for cost calculation
- `resources`: human and non-human resources
- `tasks`: the project tasks
- `reports`: reports to generate

All *properties* have *attributes*, e.g. the *property* `task` has the *attributes* `start`, `duration` as well as other properties.

# The Project Head

- What is the name of the project?
- When do you start?
- How long is the project?

```
1 project "Example"  2012-03-01 +4m
```

# The Project Head

- What is the name of the project?
- When do you start?
- How long is the project?

```
1 project "Example"  2012-03-01 +4m
```

# Localization

- Date format and time zone localization
- Time format
- Currency unit and value representation

```
1 project "Example"  2012-03-01 +4m {
2   timezone "Europe/Berlin"
3   timeformat "%d.%m.%Y"
4   numberformat "-" "" "" "," 1
5   currencyformat "-" "" "" "," 0
6   currency "EUR"
7 }
```

# Localization

- Date format and time zone localization
- Time format
- Currency unit and value representation

```
1 project "Example"  2012-03-01 +4m {
2   timezone "Europe/Berlin"
3   timeformat "%d.%m.%Y"
4   numberformat "-" "" "" "," 1
5   currencyformat "-" "" "" "," 0
6   currency "EUR"
7 }
```

# Different Scenarios

- A scenario describes a variant of the project
- You can write and calculate as much scenarios as you want
- Scenarios can be nested
- Nested scenarios inherit all *attributes* but can overwrite them

```
1 project "Beispiel"  2012-03-01 +4m {
2   scenario plan "Plan" {
3     scenario real "Realität"
4   }
5   now 2012-03-17
6 }
```

# Different Scenarios

- A scenario describes a variant of the project
- You can write and calculate as much scenarios as you want
- Scenarios can be nested
- Nested scenarios inherit all *attributes* but can overwrite them

```
1 project "Beispiel"  2012-03-01 +4m {
2   scenario plan "Plan" {
3     scenario real "Realität"
4   }
5   now 2012-03-17
6 }
```

# Extend the Data Model

- *Properties* can have a set of *attributes*
- You can extend this set with your own *attributes*

```
1 project example "Example"  2012-03-01 +4m {
2   extend resource {
3     text Phone "Phone"
4   }
5   extend task {
6     reference Wiki "Wiki"
7   }
8 }
```

# Extend the Data Model

- *Properties* can have a set of *attributes*
- You can extend this set with your own *attributes*

```
1 project example "Example"  2012-03-01 +4m {
2   extend resource {
3     text Phone "Phone"
4   }
5   extend task {
6     reference Wiki "Wiki"
7   }
8 }
```

# Define Resources

- Add an employee to the project

```
1 resource karl "Karl Mustermann"
```

- Define a team

```
1 resource ateam "Das A-Team" {
2   rate 330.0
3   resource karl "Karl Mustermann"
4   resource erika "Erika Musterfrau"
5 }
```

- All nested resources re-use the *attribute* `rate`

# Define Resources

- Add an employee to the project

```
1 resource karl "Karl Mustermann"
```

- Define a team

```
1 resource ateam "Das A-Team" {
2   rate 330.0
3   resource karl "Karl Mustermann"
4   resource erika "Erika Musterfrau"
5 }
```

- All nested resources re-use the *attribute* `rate`

# Define Resources

- Add an employee to the project

```
1 resource karl "Karl Mustermann"
```

- Define a team

```
1 resource ateam "Das A-Team" {
2   rate 330.0
3   resource karl "Karl Mustermann"
4   resource erika "Erika Musterfrau"
5 }
```

- All nested resources re-use the *attribute* `rate`

# Other resources such as tools

- Defining resources that don't contribute to the "'effort"'

```
1 resource tool "The Tool" {
2   efficiency 0.0
3   rate 500.0
4 }
```

# The Project Task Structure

```
1 task "Phase 1" {
2   task "Step 1"
3   task "Step 2"
4 }
5 task "Phase 2" {
6   task "Step 1"
7   task "Step 2"
8 }
```

# Define Dependencies

```
1  task p1 "Phase 1" {
2    task s1 "Step 1"
3    task "Step 2" {
4      depends !s1 # relative ID
5    }
6  }
7  task p2 "Phase 2" {
8    task s1 "Step 1" {
9      depends p1.s1 # absolute ID
10   }
11   task "Step 2"
12 }
```

# Workload and Task Lengths

```
1  task p1 "Phase 1" {
2    task s1 "Step 1" {
3      duration 2d
4    task "Step 2" {
5      length 10d
6    }
7    task "Step 3" {
8      effort 5d
9      allocate karl
10   }
11 }
```

# Lists and Reports

- TaskJuggler supports differnt types of lists and reports
  - Task lists
  - Employee and resource lists
  - Calendar
  - Time sheet forms
  - Status report forms
- TaskJuggler can create lists in different formats
  - HTML
  - CSV
  - TaskJuggler Syntax
  - iCal

# Lists and Reports

- TaskJuggler supports differnt types of lists and reports
  - Task lists
  - Employee and resource lists
  - Calendar
  - Time sheet forms
  - Status report forms
- TaskJuggler can create lists in different formats
  - HTML
  - CSV
  - TaskJuggler Syntax
  - iCal

# Task Lists

```
1 taskreport "Tasks" {
2   formats html
3   hidetask ~isleaf()
4   sorttasks plan.end.up
5 }
```

# Employee Lists

```
1 resourcereport "Employees" {
2   formats html
3   sortasks plan.id.up
4   columns no, name, email
5 }
```

# Task lists with employees

```
1 taskreport "Tasks" {
2   formats html
3   hidetask ~isleaf()
4   sortasks plan.effort.up
5   hideresource 0
6   columns no, name, weekly
7 }
```

# Text Reports

- Text reports consists of up to five customizable text fields
- `header`, `left`, `center`, `right`, `bottom`

```
1 textreport "Report" {
2   formats html
3   left "Left Margin"
4   center "Center Area"
5   right "Right Margin"
6 }
```

# Text Reports

- Text reports consists of up to five customizable text fields
- `header`, `left`, `center`, `right`, `bottom`

```
1 textreport "Report" {
2   formats html
3   left "Left Margin"
4   center "Center Area"
5   right "Right Margin"
6 }
```

# *RichText* Markup

- Most text attributes are interpreted as *RichText* markup
- TaskJuggler uses a subset of MediaWiki's markup ...
- ... and extends this subset for
  - text colors,

    ```
    <fcol:green>Green</fcol>
    ```
  - navigation menus,

    ```
    <[navigator id='my_menu']>
    ```
  - value queries and

    ```
    <-query ...->
    ```
  - inline reports

    ```
    <[report id='my_report']>
    ```

# Compound Reports

```
1  taskreport r1 "" {
2    columns name, chart
3  }
4  resourcereport r2 "" {
5    columns name, phone
6  }
7  textreport "Report" {
8    formats html
9    left "<[report id='r1']>"
10   right "<[report id='r2']>"
11 }
```

# HTML Navigation Menus

```
1  navigator menu
2  textreport "" {
3    header "<[navigator id='menu']>"
4    formats html
5    taskreport "Tasks" {
6      columns name, start, end, chart
7      hideresource 0
8    }
9    resourcereport "Employees" {
10     columns name, email
11   }
12   purge formats
13 }
```

# Project Management Cycle

After starting the project, the following tasks should be re-run on a regularly basis:

- Ask for done workload for current tasks
- Ask for workload or time left for current tasks
- Update project description based on this data
- Freeze project history

# `timesheets` and `statussheets`

- Report of done and left workload and task times can be organized semi-automatically with `timesheetreports`
- Hierarchical status reports be organized semi-automatically with `statussheetreports`
- This topic is too complex for these slides but is documented well in the user documentation.

# Freeze the Project History

- After upgrading the project description with the reported data, the project history should be friezed.

```
tj3 --freeze yourproject.tjp
```

- This will create or update the following two files:
  - `yourproject-header.tjp`
  - `yourproject-bookings.tjp`
- These files must be included to the project with `include`

# Freeze the Project History

- After upgrading the project description with the reported data, the project history should be friezed.

  ```
  tj3 --freeze yourproject.tjp
  ```

- This will create or update the following two files:
  - `yourproject-header.tjp`
  - `yourproject-bookings.tjp`
- These files must be included to the project with `include`

# Freeze the Project History

- After upgrading the project description with the reported data, the project history should be friezed.

  ```
  tj3 --freeze yourproject.tjp
  ```

- This will create or update the following two files:
  - `yourproject-header.tjp`
  - `yourproject-bookings.tjp`
- These files must be included to the project with `include`

# Track Key Values with `tracereport`

- With `tracereports` you can track important key values over the whole project lifetime.
- Key values are exported to CSV explicitly on demand
- `tj3 -add-trace yourproject.tjp`
- Can be used to create burndown charts
- `tracereports` is the main feature of the upcoming version (3.2)

# Follow Up Links

- TaskJuggler on the web: `http://taskjuggler.org`
- Online Documentation: `http://taskjuggler.org/tj3/manual/`
- Questions?

# Follow Up Links

- TaskJuggler on the web: `http://taskjuggler.org`
- Online Documentation: `http://taskjuggler.org/tj3/manual/`

- Questions?

# Follow Up Links

- TaskJuggler on the web: `http://taskjuggler.org`
- Online Documentation: `http://taskjuggler.org/tj3/manual/`

- Questions?