

# Preparing DFG Proposals in L<sup>A</sup>T<sub>E</sub>X with `dfgproposal.cls`

Michael Kohlhase  
Computer Science, Jacobs University Bremen  
<http://kwarc.info/kohlhase>

November 3, 2010

## Abstract

The `dfgproposal` class supports many of the specific elements of a DFG Proposal. It is optimized towards collaborative projects. The package comes with an extensive example (a fake DFG proposal) that shows all elements in action.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>The User Interface</b>	<b>2</b>
2.1	Package Options . . . . .	2
2.2	Proposal Metadata and Title page . . . . .	2
2.3	Work Areas, Work Packages, and Tasks . . . . .	3
2.4	Tasks . . . . .	4
2.5	Gantt Charts . . . . .	4
2.6	Project-Related Papers . . . . .	4
2.7	Referencing and Hyperinking . . . . .	5
2.8	Final Report Infrastructure . . . . .	5
<b>3</b>	<b>Limitations and Enhancements</b>	<b>5</b>
<b>4</b>	<b>The Implementation</b>	<b>7</b>
4.1	Package Options and Format Initialization . . . . .	7
4.2	Proposal Metadata and Title Page . . . . .	8
4.3	Work Packages and Work Groups . . . . .	10
4.4	Tasks and Work Phases . . . . .	13
4.5	Project-Related Papers . . . . .	14
4.6	Referencing and Hyperinking . . . . .	14
4.7	The Work Package Table . . . . .	15
4.8	Gantt Charts . . . . .	17
4.9	Miscellaneous . . . . .	18

# 1 Introduction

Writing grant proposals is a collaborative effort that requires the integration of contributions from many individuals. The use of an ASCII-based format like L<sup>A</sup>T<sub>E</sub>X allows to coordinate the process via a source code control system like SUBVERSION, allowing the proposal writing team to concentrate on the contents rather than the mechanics of wrangling with text fragments and revisions.

The `dfgproposal` class supports many of the specific elements of a DFG Proposal. The package documentation is still preliminary, fragmented and incomplete. Please consult the example proposal `proposal.tex`, which comes with the package and shows the usage of the class in action. It is intended as a template for your proposal, but please bear in mind that the DFG guidelines may change, if in doubt, please consult the DFG guide for proposers [For].

The `dfgproposal` class is distributed under the terms of the LaTeX Project Public License from CTAN archives in directory `macros/latex/base/lppl.txt`. Either version 1.0 or, at your option, any later version. The CTAN archive always contains the latest stable version, the development version can be found at <https://svn.kwarc.info/repos/kwarc/doc/macros/forCTAN/dfgproposal> For bug reports please use the sTeX trac at <https://trac.kwarc.info/sTeX/> with component `dfgproposal`.

## 2 The User Interface

In this section we will describe the functionality offered by the `dfgproposal` class along the lines of the macros and environments the class provides. Much of the functionality can better be understood by studying the functional example `proposal.tex` (and its dependents) that comes with the `dfgproposal` package in conjunction with the proposer's leaflet of the DFG [For] (we have included it as `1_02e.pdf` for convenience into the package distribution).

### 2.1 Package Options

As usual in L<sup>A</sup>T<sub>E</sub>X, the package is loaded by `\documentclass[options]{dfgproposal}`, where [*options*] is optional and gives a comma separated list of options. The `dfgproposal` package takes the options `submit`, `noworkareas`, `public`, and `keys`.

`submit` The `submit` will disable various proposal management decorations which are enabled by default for submission.

`noworkareas` The `noworkareas` specifies that we do not want to structure our work plan into work areas (see section 2.3).

`noRAM` The `noRAM` specifies that we do not specify research assistant months in the effort tallies (see section 2.3).

`public` Finally, the `public` option allows to hide certain sensitive (e.g. financial) parts of the proposal. For this, the `dfgproposal` provides the environment. If the option `public` is set, the parts

`private` of the document between `\begin{private}` and `\end{private}` do not produce output. This is useful for producing public versions of the proposal that hide confidential parts. Note that both `\begin{private}` and `\end{private}` *have to be on lines of their own may not have any leading whitespace* otherwise an error occurs and L<sup>A</sup>T<sub>E</sub>X gives error messages that are difficult to comprehend. An alternative way to distinguish private and public sections are to use the `\ifpublic` `\ifpublic` conditional: `\ifpublic{3}\else{5}\fi` will result in “5” in the submitted draft and “3” in the public document.

`keys` The `keys` specifies that we want to see the values of various `keyval` arguments in the margin.

### 2.2 Proposal Metadata and Title page

`proposal` The metadata of the proposal is specified in the `proposal` environment, which also generates the title page and the first section of the proposal as well as the last pages of the proposal with

the signatures, enclosures, and references. The `proposal` environment should contain all the mandatory parts of the proposal text. The `proposal` environment uses the following keys to specify metadata.

- `title` • `title` for the proposal title (used on the title page),
- `instrument` • `instrument` for the instrument of funding that you would like to apply for,
- `thema` • `thema` for a concise (up to 140 characters) description of the topic of the proposal. This has to be in German and will be used by the DFG in internal communications and publications.
- `acronym` • `acronym` for the proposal acronym, possibly accompanied `acrolong` that explains it. The acronym will also be used in the page headings.
- `acrolong` • `acrolong` for the proposal acronym, possibly accompanied `acrolong` that explains it. The acronym will also be used in the page headings.
- `start` • `start` for the start date, and `months` for the length of the proposal in months. Both have to be specified to for the `dfgproposal` class to work.
- `months` • `months` for the length of the proposal in months. Both have to be specified to for the `dfgproposal` class to work.
- `discipline` • `discipline` for the academic discipline and `areas` for the research areas in that discipline.
- `PI` • `PI` to declare the principal investigator. For collaborative proposals we can use the `PI` key multiple times. The `dfgproposal` package uses the `dcm` package for representation of personal metadata, see [Koh10] or the file `proposal.tex` for details.
- `site` • Many collaborative proposals are shared between two institutions, which we can declare with the `site` key. As this changes the interface this should not be used for single-institution proposals. We will describe the setup for a single-site proposal below and point out the differences. The example `proposal.tex` is a two-site proposal.
- `pubspage` • The `pubspage` key can be used to give URIs of publication pages that are mentioned in the references sections as sources of publications of the proposers. Use one `pubspage` key per URI.

The Metadata section of the proposal concludes with a summary for academic laypeople, which will be used by DFG e.g. on their web sites. For Proposals written in English (as we assume in this documentation) most proposers also provide an English summary using the `Summary` environment.

## 2.3 Work Areas, Work Packages, and Tasks

DFG proposals have another part that is often highly stylized; the work plan. This is usually structured into “work packages” — i.e. work items that address a cohesive aspect of the proposed work. These work packages are usually consecutively numbered, have a title, and an associated effort estimation. As work packages are the “atomic” planning units, they are usually heavily cross-referenced. A well-written proposal usually contains a table giving an overview over the work packages and their efforts and a Gantt chart showing the temporal distribution of the proposed work to allow the reviewers to get a clear picture of the feasibility of the research and development proposed. But this picture is also essential during the development of a proposal (which the `dfgproposal` package aims to support), when the work packages (and their estimated efforts) usually change considerably. Therefore the `dfgproposal` class standardizes markup for work packages and automatically computes the work package table (which can be inserted into the table via the `\wpfig` macro) and the Gantt Chart (see Section 2.5).

`\wpfig`  
`workplan`

To achieve the automation, work plan is marked up by the `workplan` environment, which setups various internal counters and bookkeeping macros. It contains texts and `workpackage` environments for the work packages.

`workpackage`

The purpose of the `workpackage` environment is to mark up a fragment of text as a work package description and specify the metadata so that it can be used in the work package table and Gantt chart generation. The metadata is specified by the following keys:

- `id` • The `id` key is used to specify a label for cross-referencing the work package or work group, it must be document-unique.
- `title` • The `title` and `short` keys are used for the work package/group title. The short title is used in tables and should not be longer than 15 characters.
- `short` • The `title` and `short` keys are used for the work package/group title. The short title is used in tables and should not be longer than 15 characters.
- `RM` • In single-site proposals, the `RM` and `RAM` keys are used to specify the estimated efforts to be expended on research and development in this work package. Both are specified in person months. `RM` is used for “researcher months” (Wissenschaftlicher Mitarbeiter) and `RAM` for “research assistant months” (wissenschaftliche Hilfskraft).
- `RAM` • In single-site proposals, the `RM` and `RAM` keys are used to specify the estimated efforts to be expended on research and development in this work package. Both are specified in person months. `RM` is used for “researcher months” (Wissenschaftlicher Mitarbeiter) and `RAM` for “research assistant months” (wissenschaftliche Hilfskraft).

- `*RM` • In multi-site proposals, the `dfgproposal` package generates the keys  $\langle site \rangle RM$  and  $\langle site \rangle RAM$
- `*RAM` where  $\langle site \rangle$  is any site label declared via the `site` key in the top-level `proposal` environment. This can be used to specify the person months that the site spends on this work package (the value for work groups is automatically computed (remember to run  $\LaTeX$  twice for this)).
- `lead` • In multi-site proposals the `lead` key specifies the work package or work group lead, the value of this feature should be the short name of the respective partner.

It is often useful to group the work packages in a proposal further (especially for larger, collaborative proposals). This can be done via the `workarea` environment, which groups work packages. This environment takes the same keys as the `workpackage` environment, except for the efforts, which can be computed automatically from the work packages it groups.

As the author of the `dfgproposal` class likes more structured proposals, using work areas is the default, but the `dfgproposal` class can also be used with the `noworkareas` option for less structured (smaller) proposals.

## 2.4 Tasks

`tasklist` In the work packages we can list tasks that need to be undertaken with the `tasklist` environment.

`task` The individual tasks are marked up with the `task` environment. This takes a keyval argument with the keys `id` for identification, `start` for the start month (a natural number) of the task, and `len` for the length of the work phase. The `force` key that specifies the work intensity (this should be a real number between 0 and 1 to prevent overlaps). The main reason for specifying this metadata for tasks is to generate a Gantt chart (see Section 2.5). The `requires` key can be used to mark, up dependencies between tasks. If `requires=\taskin{\langle rid \rangle}{\langle wp \rangle}` is given in a task with `id=\langle t \rangle`, then task  $\langle rid \rangle$  in work package  $\langle wp \rangle$  must be completed for task  $\langle t \rangle$  to become possible. This key will draw an arrow into the gantt chart from the end of task  $\langle rid \rangle$  to  $\langle t \rangle$ . Note that dependencies should always point forward in time. Furthermore, note that the fact that dependencies always go from the end of the source to the beginning of the target work phase is intentional, if this does not meet your needs, then you should probably break a work phase into pieces that can be addressed separately.

`\taskref` Tasks can be referenced by the `\taskref` macro that takes two arguments: the work package identifier and the task identifier. As for work packages and work areas, there is a long reference variant with work package title: `\tasktref`. Finally, `\localtaskref` references a task in the local work package by the identifier in its argument.

If the work packages are not structured into tasks or the task structure is not adequate for structuring the work phases in a work package, then we can use the `\workphase` macro for this. The use of the `\workphase` package is deprecated.

## 2.5 Gantt Charts

Gantt charts are used in proposals to show the distribution of activities in work packages over time.

`gantt` A gantt chart is represented by the `gantt` environment that takes a on optional keyval argument.

`xscale` The keys `xscale` and `yscale` are used to specify a scale factors for the chart so that it fits on the page. The `step` key allows to specify the steps (in months) of the vertical auxiliary lines. Finally,

`yscale` the `draft` key specifies that plausibility checks (that can be expensive to run) are carried out.

`step` Note that the value does not have to be given, so `\begin{gantt}{draft,yscale=.5,step=3}` is a perfectly good invocation.

`draft`

`\ganttchart` Usually, the `gant` environment is not used however, since it is part of the macro that takes the same keys. This generates a whole Gantt chart automatically from the work phase specifications in the work packages. As above we have to run  $\LaTeX$  two times for the work phases to show up.

## 2.6 Project-Related Papers

Since August 2010, DFG requests list of “project-related papers” as section 2.2. We use the `\dfgprojpapersbiblatex` package to automate this. We only need to use `\dfgprojpapers{\langle keys \rangle}`, where  $\langle keys \rangle$

EdNote(1)

ist a comma-separated list of keys from the bibTeX database used in the proposal. Note that you will have to run `bibtex proposal1-blx.aux` to generate file `proposal1.bbl` that generates the bibliography list.<sup>1</sup>

## 2.7 Referencing and Hyperinking

The `dfgproposal` package extends the hyperlinking provided by the `hyperref` package it includes to work packages, work groups, . . . . Whenever these are defined using the `dfgproposal` infrastructure, the class saves the relevant information in the auxiliary file `<proposal>.aux`. This information can be referenced via the macro, which takes three arguments.

`\euref`

In a reference `\euref{<type>}{<id>}{<aspect>}` the first argument `<type>` specifies the type of the object (currently one of `wp`, `wa`, and `partner`) to be referenced, `<id>` specifies the identifier of the referenced object (it matches the identifier given in the `id` key of the object), and `<aspect>` specifies the aspect of the saved information that is referenced.

For a partner `<aspect>` can be one of `number` (partner number), `short` (partner acronym), `long` (official partner name), `nationality` (partner nationality).

For a work package `<aspect>` can be `number`, (the work package number), `label` (the label `WP $n$`  where  $n$  is the work package number for referencing), `title` (the work package title), `lead` the work package leader, `short` (a short version of the WP title for tables). For work groups we have the same aspects with analogous meanings. In all cases, the referenced information carries a hyperlink to the referenced object.

`\euRef`

The `\euRef` macro is a variant of `\euref` that also carries a hyperlink (if the `hyperref` package is loaded).

Note that since the referencable information is written into the `<proposal>.aux` file, it is available for forward references. However, it will only become available when the auxiliary file is read, so the proposal has to be formatted twice for references to be correct.<sup>2</sup>

EdNote(2)

Finally, the `dfgproposal` package supplies specialized reference macros for work packages and areas. The `\WPref` macro takes a work package identifier as an argument and makes a reference: `\WPref{<id>}` abbreviates `\euRef{wp}{<id>}{label}`. The `\WPtref` macro is similar, but also prints out the title: `\WPref{<id>}` abbreviates `\euRef{wp}{<id>}{label}: \euRef{wp}{<id>}{title}`.

`\WPref`

`\WPtref`

`\WApref`

`\WAtref`

Unless the `noworkareas` macro is set, we also have the variants `\WApref` and `\WAtref` for work areas.

## 2.8 Final Report Infrastructure

The `dfgproposal` package also gives an infrastructure for writing final reports of completed projects (see the file `finalreport.tex` in the package distribution). The `finalreport` environment has functionality analogous to the `proposal` environment. It takes the same metadata keys — making it easy to generate by copy/paste from the proposal — but adds the keys `fundedperiod` for specifying the funded period and `reportperiod` for the project period reported on (they can differ from each other). Furthermore, the key `key` can be used to specify the reference key (something like `K0 2428 47-11`) given to the project by DFG. Note that in the case of multiple proposers, you can use multiple instances of `key` to specify more than one reference key.

`finalreport`

`fundedperiod`

`reportperiod`

`key`

## 3 Limitations and Enhancements

The `dfgproposal` is relatively early in its development, and many enhancements are conceivable. We will list them here.

1. macros cannot be used work package and work area titles. They really mess up our `\wpfig` automation. The problem is that they are evaluated too early, and our trick with making them undefined while collecting the parts of the table-rows only works if we know

<sup>1</sup>EDNOTE: automate this some more

<sup>2</sup>EDNOTE: need to document all the kinds of metadata that can be referenced!

which macros we may expect. We might specify all “allowable” macros in an optional key `protectmacro`, which is defined via

```
\define@key{wfig}{protectmacro}{\epandafter\let\csname #1\endcsname=\relax}
```

But I am not sure that this will work.

2. It would be great, if in the Gantt Charts, we could include some plausibility checks (for draft = not `submit` mode). I can see two at the moment:
  - calculating the effort (i.e. the weight of the black area) and visualizing it. Then we could check whether that is larger than the effort declared for the work package.
  - calculating (and visualizing) the monthly effort. That should be kind of even (or it has to be explained in the positions requested).
3. we currently do not have a way to relate PIs to `sites`, but we do not really need to.
4. it would be nice if we could count the characters in the places where lengths are restricted (`Abriss`, and `thema`). Then we could warn the authors.

If you have other enhancements to propose or feel you can alleviate some limitation, please feel free to contact the author.

## 4 The Implementation

In this section we describe the implementation of the functionality of the `dfgproposal` package.

### 4.1 Package Options and Format Initialization

We first set up the options for the package.

```
1 (*cls)
2 \newif\ifsubmit\submitfalse
3 \newif\ifpublic\publicfalse
4 \newif\ifkeys\keysfalse
5 \newif\ifwork@areas\work@areastrue
6 \newif\if@RAM\@RAMtrue
7 \DeclareOption{submit}{\submittrue}
8 \DeclareOption{public}{\publictrue\excludecomment{private}}
9 \DeclareOption{noworkareas}{\work@areasfalse}
10 \DeclareOption{noRAM}{\@RAMfalse}
11 \DeclareOption{keys}{\keystrue}
12 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
13 \ProcessOptions
```

Then we load the packages we make use of

```
14 \LoadClass[a4paper,twoside]{article}
15 \RequirePackage{dcm}
16 \RequirePackage{amssymb}
17 \RequirePackage{url}
18 \RequirePackage{graphicx}
19 \RequirePackage{colortbl}
20 \RequirePackage{xcolor}
21 \RequirePackage{rotating}
22 \RequirePackage{fancyhdr}
23 \RequirePackage{array}
24 \RequirePackage{xspace}
25 \RequirePackage{eurosym}
26 \RequirePackage{comment}
27 \RequirePackage{tikz}
28 \RequirePackage{paralist}
29 \RequirePackage{a4wide}
30 \renewcommand{\familydefault}{\sfdefault}
31 \RequirePackage[scaled=.90]{helvet}
32 \RequirePackage{textcomp}
33 \RequirePackage[USenglish,ngerman]{babel}
34 \RequirePackage{biblatex}
35 \RequirePackage[a4paper=true,bookmarks=true,linkcolor=blue,
36 citecolor=blue,urlcolor=blue,colorlinks=true,
37 breaklinks=true,bookmarksopen=true]{hyperref}
```

the `ed` package [Koh07] is very useful for collaborative writing and passing messages between collaborators or simply reminding yourself of editing tasks, so we preload it in the class. However, we only want to show the information in draft mode. Furthermore, we adapt the options for the `svninfo` package.

```
38 \ifsubmit
39 \RequirePackage[hide]{ed}
40 \RequirePackage[final,today]{svninfo}
41 \else
42 \RequirePackage[show]{ed}
43 \RequirePackage[eso-foot,today]{svninfo}
44 \fi
```

`private` We configure the `comment` package, so that it provides the `private` environment depending on the status of the `public` option.

```
45 \ifpublic\excludecomment{private}\else\includecomment{private}\fi
```

And we set up the appearance of the proposal. We want numbered subsections.

```
46 \setcounter{secnumdepth}{3}
```

We specify the page headings.

```
47 \newif\ifofpage\ofpagefalse
```

```
48 \fancyhead[RE,LO]{\dfg@gen@acronym}
```

```
49 \fancyhead[LE,RO]{page~\thepage\ifofpage~of~\euref@num{prop}{page}{last}\fi}
```

```
50 \pagestyle{fancyplain}
```

## 4.2 Proposal Metadata and Title Page

We define the keys for metadata declarations in the `proposal` environment, they park their argument in an internal macro for use in the title page. The `site` key is the most complicated, so we take care of it first: We need a switch `\if@sites` that is set to true when the `site` key is used. Furthermore `site=<site>` makes new keys `<site>RM` and `<site>RAM` (unless the `noRAM` option was set) for the `workpackage` environment and records the sites in the `\dfg@gen@sites` token register.

```
51 \newif\if@sites\@sitesfalse\let\dfg@gen@sites=\relax
```

```
52 \define@key{dfg@gen}{site}{\@sitestrue\@dmp{site=#1}%
```

```
53 \@ifundefined{dfg@gen@sites}{\xdef\dfg@gen@sites{#1}}{\xdef\dfg@gen@sites{\dfg@gen@sites,#1}}
```

```
54 \define@key{workpackage}{#1RM}{\dfg@def\wp@id{#1}{RM}{##1}}
```

```
55 \if@RAM\define@key{workpackage}{#1RAM}{\dfg@def\wp@id{#1}{RAM}{##1}}\fi}
```

similarly, the `PI` keys are registered in `\dfg@gen@PIs`.

```
56 \define@key{dfg@gen}{PI}{\@dmp{PI=#1}%
```

```
57 \@ifundefined{dfg@gen@PIs}{\xdef\dfg@gen@PIs{#1}}{\xdef\dfg@gen@PIs{\dfg@gen@PIs,#1}}
```

and the `pubspage` keys in `\dfg@gen@pubspages`.

```
58 \define@key{dfg@gen}{pubspage}{\@ifundefined{dfg@gen@pubspages}%
```

```
59 {\xdef\dfg@gen@pubspages{#1}}{\xdef\dfg@gen@pubspages{\dfg@gen@pubspages,#1}}
```

The rest of the keys just store their value.

```
60 \define@key{dfg@gen}{instrument}{\def\dfg@gen@instrument{#1}\@dmp{inst=#1}}
```

```
61 \define@key{dfg@gen}{title}{\def\dfg@gen@title{#1}}
```

```
62 \define@key{dfg@gen}{thema}{\def\dfg@gen@thema{#1}}
```

```
63 \define@key{dfg@gen}{acronym}{\gdef\dfg@gen@acronym{#1}\@dmp{acro=#1}}
```

```
64 \define@key{dfg@gen}{acrolong}{\def\dfg@gen@acrolong{#1}}
```

```
65 \define@key{dfg@gen}{discipline}{\def\dfg@gen@discipline{#1}}
```

```
66 \define@key{dfg@gen}{areas}{\def\dfg@gen@areas{#1}}
```

```
67 \define@key{dfg@gen}{start}{\def\dfg@gen@start{#1}}
```

```
68 \define@key{dfg@gen}{months}{\def\dfg@gen@months{#1}}
```

and now the ones for the final report

```
69 \define@key{dfg@gen}{fundedperiod}{\def\dfg@gen@fundedperiod{#1}}
```

```
70 \define@key{dfg@gen}{reportperiod}{\def\dfg@gen@reportperiod{#1}}
```

```
71 \define@key{dfg@gen}{key}{\@dmp{key=#1}%
```

```
72 \@ifundefined{dfg@gen@keys}{\xdef\dfg@gen@keys{#1}}{\xdef\dfg@gen@keys{\dfg@gen@keys,#1}}
```

and the default values, these will be used, if the author does not specify something better.

```
73 \def\dfg@gen@acro{ACRONYM}
```

```
74 \def\dfg@gen@months{???
```

```
75 \def\dfg@gen@title{Proposal Title}
```

```
76 \def\dfg@gen@instrument{Neuantrag auf Sachbeihilfe}
```

```
77 \def\dfg@gen@thema{??? Thema ???}
```

```
78 \long\def\deu#1{#1}
```



Now, we can build the title page and general information, we first start with the applicants table:

```

79 \def\dfg@applicants@table%
80 {\begin{tabular}{|l|*{\the@PIs}{l|}}\hline
81 \dcm@bitabline\dfg@gen@PIs{personaltitle}{ }{name}\\
82 \dcm@tabline\dfg@gen@PIs{academictitle}\\
83 \dcm@tabline\dfg@gen@PIs{birthdate}\\
84 &\multicolumn{\the@PIs}{|l|}{\textbf{Work Address\deu{ (Dienstanschrift)}}}\\\hline
85 \dcm@tabline\dfg@gen@PIs{affiliation}\\
86 \dcm@tabline\dfg@gen@PIs{workaddress}\\
87 \dcm@atabline\dfg@gen@PIs{Tel: }{worktel}\\
88 \dcm@atabline\dfg@gen@PIs{Fax: }{workfax}\\
89 \dcm@tabline\dfg@gen@PIs{email}\\
90 &\multicolumn{\the@PIs}{|l|}{\textbf{Private Address\deu{ (Privatanschrift)}}} \\
91 \dcm@tabline\dfg@gen@PIs{privaddress}\\
92 \dcm@atabline\dfg@gen@PIs{Tel: }{privtel}\\
93 \end{tabular}}

94 \def\dfg@applicants{\subsection{Applicants \deu{(Antragsteller)}}
95 \dfg@applicants@table}

96 \def\dfg@topic{\deu{\subsection{Topic (Thema)}}\dfg@gen@thema}

97 \def\dfg@area@field{%
98 \subsection{Research area and field of work \deu{(Fachgebiet und Arbeitsrichtung)}}
99 Scientific discipline: \dfg@gen@discipline\\
100 Fields of work: \dfg@gen@areas}

101 \newenvironment{dfg@titlepage}
102 {\thispagestyle{empty}\setcounter{page}{0}%
103 \begin{center}}
104 {\newcounter{@PIs}\@for{\@I:=\dfg@gen@PIs\do{\stepcounter{@PIs}}
105 \newcounter{@sites}\@for{\@I:=\dfg@gen@sites\do{\stepcounter{@sites}}
106 \begin{tabular}{c*{\the@PIs}{c}}
107 \dcm@tabline\dfg@gen@PIs{name}\\
108 \dcm@tabline\dfg@gen@PIs{affiliation}
109 \end{tabular}}\ [2cm]
110 \end{center}
111 \setcounter{tocdepth}{1}\tableofcontents\newpage\setcounter{page}{1}}

```

**proposal** The proposal environment reads the metadata keys defined above, and if there were no site keys, then it defines keys RM and RAM (unless the noRAM package option was given) for the workpackage environment.

```

112 \newenvironment{proposal}[1][ ]
113 {\ofpagetrue\setkeys{dfg@gen}{#1}
114 \if@sites\else
115 \define@key{workpackage}{RM}{\dfg@def{wp}\wp@id{RM}{##1}\@dmp{RM=##1}}
116 \if@RAM\define@key{workpackage}{RAM}{\dfg@def{wp}\wp@id{RAM}{##1}\@dmp{RAM=##1}}\fi
117 \fi
118 \begin{dfg@titlepage}
119 {\LARGE \dfg@gen@instrument}\ [2cm]
120 {\LARGE\textbf{\dfg@gen@title}}\ [3cm]
121 {\LARGE Acronym: {\dfg@gen@acronym}}\ [2cm]
122 {\large\today}\ [1em]
123 \end{dfg@titlepage}
124 \section{General Information \deu{(Allgemeine Angaben)}}
125 \dfg@applicants
126 \dfg@topic
127 \dfg@area@field
128 \subsection{Anticipated total duration \deu{(Voraussichtliche
129 Gesamtdauer)}}\dfg@gen@months\quad Months

```

```

130 \subsection{Application period \deu{(Antragszeitraum)}}\dfg@gen@start
131 \selectlanguage{USenglish}}
Now we come to the end of the environment:
132 {\thispagestyle{empty}\signatures
133 \section{List of Attachments \deu{(Verzeichnis der Anlagen)}}}
134 \begin{itemize}
135 \@for\@I:=\dfg@gen@PIs\do{%
136 \item Curriculum Vitae and list of publications for
137 \@nameuse{dcm@person@\@I @personaltitle}
138 \@nameuse{dcm@person@\@I @name}
139 (CD + printed version)}
140 \item Publications either cited in section~\ref{sec:ourpubs} or occurring in one of the
141 curricula vitae (CD only)
142 \end{itemize}
143 \dfg@def{prop}{page}{last}{\thepage}\ofpagefalse
144 \newpage
145 \printbibliography[heading=warnpubs]}

```

We only have to define the warnpubs heading constructor

```

146 \defbibheading{warnpubs}{\section*{References}%
147 \@ifundefined{dfg@gen@pubspages}
148 {\@latex@warning{No publication pages specified;
149 use the pubspage key in the proposal environment!}}
150 {Many of the proposers' publications are online at one of the following URIs:
151 \@for\@I:=\dfg@gen@pubspages\do{\par\noindent\url{\@I}}}}

```

The finalreport is very similar.

finalreport

```

152 \newenvironment{finalreport}[1][
153 {\ofpagetrue\setkeys{dfg@gen}{#1}
154 \begin{dfg@titlepage}
155 {\LARGE Final Project Report}\[\.2cm]
156 {\LARGE\textbf{\dfg@gen@title}}\[\.3cm]
157 {\LARGE Acronym: {\dfg@gen@acronym}}\[\.2cm]
158 {\large Reference number(s): \@ifundefined{dfg@gen@keys}{??:???? ?/?}{\dfg@gen@keys}}\[\.2cm]
159 {\large\today}\[\1em]
160 \end{dfg@titlepage}
161 \section{General Information \deu{(Allgemeine Angaben)}}}
162 \dfg@applicants
163 \dfg@topic
164 \subsection{Funding Period \deu{(F"orderzeitraum)}}\dfg@gen@fundedperiod
165 \subsection{Report Period \deu{(Berichtszeitraum)}}\dfg@gen@reportperiod
166 \selectlanguage{USenglish}}
167 {\dfg@def{prop}{page}{last}{\thepage}\ofpagefalse}

168 \newenvironment{Summary}{\subsection{Summary \deu{(Zusammenfassung)}}}{-}
169 \newenvironment{Abriss}%
170 {\selectlanguage{ngerman}\subsection*{Zusammenfassung (dt. Version)}}
171 {\selectlanguage{USenglish}}

```

### 4.3 Work Packages and Work Groups

\dfg@def This macro writes an \@dfg@def command to the current aux file and also executes it.

```

172 \def\dfg@def#1#2#3#4{%\@dfg@def{#1}{#2}{#3}{#4}%
173 \protected@write\auxout{}{\string\@dfg@def{#1}{#2}{#3}{#4}}

```

We first define keys for work groups (if we are in an IP).

```
174 \ifwork@areas
175 \define@key{workarea}{id}{\def\wa{id#1}\@dmp{id=#1}}
176 \define@key{workarea}{title}{\dfg@def{wa}\wa{id{title}}{#1}}
177 \define@key{workarea}{short}{\dfg@def{wa}\wa{id{short}}{#1}}
178 \fi
```

work packages have similar ones.

```
179 \define@key{workpackage}{id}{\def\wp{id#1}\@dmp{id=#1}}
180 \define@key{workpackage}{title}{\dfg@def{wp}\wp{id{title}}{#1}}
181 \define@key{workpackage}{lead}{\dfg@def{wp}\wp{id{lead}}{#1}\def\wp@lead{#1}\@dmp{lead=#1}}
182 \define@key{workpackage}{short}{\dfg@def{wp}\wp{id{short}}{#1}}
```

We define the constructors for the work package and work group labels and titles.

```
183 \def\wp@mk@title#1{Work Package {#1}}
184 \def\wp@label#1{WP{#1}}
185 \ifwork@areas
186 \def\wa@label#1{WA{#1}}
187 \def\wa@mk@title#1{Work Area {#1}}
188 \fi
```

The wa and wp counters are for the work packages and work groups

```
189 \ifwork@areas\newcounter{wa}\newcounter{wp}[wa]\else\newcounter{wp}\fi
190 \newcounter{allwp}
```

`\update@*` update the list `\@wps` of the work packages in the local group and the list `\@was` work groups for the staff efforts table: if `\@wps` is undefined, then initialize the comma-separated list, otherwise extend it.<sup>3</sup>

EdNote(3)

```
191 \def\update@wps#1{\@ifundefined{@wps}{\xdef\@wps{#1}}{\xdef\@wps{\@wps,#1}}}
192 \def\update@tasks#1{\@ifundefined{@tasks}{\xdef\@tasks{#1}}{\xdef\@tasks{\@tasks,#1}}}
193 \def\update@deps#1{\@ifundefined{task@deps}{\xdef\@task@deps{#1}}{\xdef\@task@deps{\@task@deps,#1}}}
194 \ifwork@areas\def\update@was#1{\@ifundefined{@was}{\xdef\@was{#1}}{\xdef\@was{\@was,#1}}}\fi
```

with these it is now relatively simple to define the interface macros.

`workpackage` The `workpackage` environment collects the keywords, steps the counters, writes the metadata to the aux file, updates the work packages in the local group, generates the work package number `\wp@num`.

```
195 \newcounter{wp@RM}
196 \if@RAM\newcounter{wp@RAM}\fi
197 \newenvironment{workpackage}[1][\%
198 {\setkeys{workpackage}{#1}\stepcounter{wp}\stepcounter{allwp}\%
199 \let\@tasks=\relax\%
200 \edef\wp@num{\ifwork@areas\thewa.\fi\thewp}\%
201 \dfg@def{wp}\wp{id{label}}{\wp@label\wp@num}\%
202 \dfg@def{wp}\wp{id{number}}{\thewp}\%
203 \dfg@def{wp}\wp{id{page}}{\thepage}\%
204 \update@wps\wp{id}\%
205 \edef\wp@num{\ifwork@areas\thewa.\fi\thewp}\%
206 \dfg@def{wp}{\wp{id}{num}}{\thewp}\%
```

Now we have to consider two cases, if we have sites, then we make a header table.

```
207 \if@sites\%
208 {\let\G@refundefinedtrue=\relax\let\@latex@warning=\relax\%
209 \xdef\sites@line{Site}\xdef\efforts@line{Effort}\%initialize lines
210 \setcounter{wp@RM}{0}\%
211 \if@RAM\setcounter{wp@RAM}{0}\fi}
```

---

<sup>3</sup>EDNOTE: with the current architecture, we cannot have work areas that do not contain work packages, this leads to the error that `wps` is undefined in `endworkplan`

```

212 \@for\@site:=\dfg@gen@sites\do{%
213 \xdef\sites@line{\sites@line&\@site\ifx\@site\wp@lead~(lead)\fi}%
214 \xdef\efforts@line{\efforts@line&\euref\wp@id\@site{RM}\if@RAM+\euref\wp@id\@site{RAM}\fi}%
215 \edef\@RM{\euref@num\wp@id\@site{RM}}\addtocounter{wp@RM}{\@RM}%
216 \if@RAM\edef\@RAM{\euref@num\wp@id\@site{RAM}}\addtocounter{wp@RAM}{\@RAM}\fi}%
217 \dfg@def{wp}\wp@id{RM}{\thewp@RM}%
218 \if@RAM\dfg@def{wp}\wp@id{RAM}{\thewp@RAM}\fi%
219 \bigskip\noindent\begin{tabular}{|l|l|l|*{\the@sites}{c|}}\hline%
220 \textbf{\wp@mk@title{\wp@num}}&\sites@line\\\hline%
221 \textsf{\dfg@target{wp}\wp@id{\euref{wp}\wp@id{title}}} &\efforts@line\\\hline%
222 \end{tabular}\smallskip\par\noindent\ignorespaces%
223 \else% if@sites

```

When we do not have sites, we can make things much simpler: we just generate a subsection

```

224 \subsubsection*{\wp@mk@title{\wp@num}: \dfg@target{wp}\wp@id{\euref{wp}\wp@id{title}}
225 (\euref@safe{wp}\wp@id{RM} RM\if@RAM+\euref{wp}\wp@id{RAM} RAM\fi)}%
226 \fi}%
227 {\@ifundefined{tasks}\dfg@def{wp@id}{task}{ids}\@tasks}

```

**workarea** the `workarea` environment for work groups is almost the same, but we also have to initialize the work package counters. Also, the efforts can be computed from the work packages in this group via the `wa@effort` counter

```

228 \newcounter{dfg@RM}\if@RAM\newcounter{dfg@RAM}\fi
229 \ifwork@areas
230 \newcounter{wa@RM}\if@RAM\newcounter{wa@RAM}\fi\newcounter{wa@wps}
231 \newenvironment{workarea}[1] []
232 {\setkeys{workarea}{#1}
233 \let\@wps=\relax
234 \stepcounter{wa}
235 \dfg@def{wa}\wa@id{label}\wa@label\thewa}
236 \dfg@def{wa}\wa@id{number}\thewa}
237 \dfg@def{wa}\wa@id{page}\thepage}
238 \update@was{\wa@id}
239 \dfg@def{wa}\wa@id{num}\thewa}
240 \setcounter{wa@RM}{0}\if@RAM\setcounter{wa@RAM}{0}\fi\setcounter{wa@wps}{0}
241 \if@sites
242 \@for\@site:=\dfg@gen@sites\do{%
243 \edef\@wps{\euref@aux\wa@id{wp}{ids}}
244 \@for\@wp:=\@wps\do{%
245 \edef\@RM{\euref@num\@wp\@site{RM}}
246 \if@RAM\edef\@RAM{\euref@num\@wp\@site{RAM}}\fi
247 \addtocounter{wa@RM}{\@RM}\addtocounter{dfg@RM}{\@RM}
248 \if@RAM\addtocounter{wa@RAM}{\@RAM}\addtocounter{dfg@RAM}{\@RAM}\fi}}
249 \else
250 \edef\@wps{\euref@aux\wa@id{wp}{ids}}
251 \@for\@wp:=\@wps\do{%
252 \edef\@RM{\euref@num{wp}\@wp{RM}}
253 \if@RAM\edef\@RAM{\euref@num{wp}\@wp{RAM}}\fi
254 \stepcounter{wa@wps}
255 \addtocounter{wa@RM}{\@RM}\addtocounter{dfg@RM}{\@RM}
256 \if@RAM\addtocounter{wa@RAM}{\@RAM}\addtocounter{dfg@RAM}{\@RAM}\fi}
257 \fi
258 \dfg@def{wa}\wa@id{RM}\thewa@RM
259 \dfg@def{dfg}{all}{RM}\thedfg@RM
260 \if@RAM
261 \dfg@def{wa}\wa@id{RAM}\thewa@RAM
262 \dfg@def{dfg}{all}{RAM}\thedfg@RAM
263 \fi
264 \subsection*{\wa@mk@title\thewa}: {\dfg@target{wa}\wa@id{\euref{wa}\wa@id{title}}}}

```

```
265 {\@ifundefined{@wps}{\dfg@def\wa@id{wp}{ids}\@wps}\dfg@def\wa@id{wp}{count}\thewa@wps}\fi
```

**workplan** The `workplan` environment sets up the accumulator macros `\@wps`, `\@was`, for the collecting the identifiers of work packages and work groups. At the end of the workplan description it writes out their content to the aux file for reference.

```
266 \newenvironment{workplan}%
267 {\ifwork@areas\let\@was=\relax\else\let\@wps=\relax\fi}%
268 {\@ifundefined{task@deps}{\dfg@def{all}{task}{deps}{\task@deps}}
269 \ifwork@areas
270 \@ifundefined{@was}{\dfg@def{all}{wa}{ids}\@was}
271 \else
272 \@ifundefined{@wps}{\dfg@def{all}{wp}{ids}\@wps}
273 \fi
274 \ifwork@areas\dfg@def{all}{wa}{count}{\thewa}\fi
275 \ifwork@areas\dfg@def{all}{wp}{count}{\theallwp}\fi
276 \def\dfg@target#1#2#3{\hypertarget{#1@#2@target}{#3}}
```

## 4.4 Tasks and Work Phases

**tasklist**

```
277 \newenvironment{tasklist}
278 {\begin{compactenum}}{\end{compactenum}}
```

The next step is to

```
279 \def\task@label#1{T#1}
```

We define the keys for the task macro

```
280 \define@key{task}{id}{\def\task@id{#1}\@dmp{id=#1}}
281 \define@key{task}{start}{\def\task@start{#1}\@dmp{start=#1}}
282 \define@key{task}{len}{\def\task@len{#1}\@dmp{len=#1}}
283 \define@key{task}{force}{\def\task@force{#1}\@dmp{f=#1}}
284 \define@key{task}{requires}{\@requires\task@id{#1}\@dmp{req=#1}}
285 \define@key{task}{title}{\def\task@title{#1}\@dmp{title=#1}}
```

then we define an auxiliary function that gives them sensible defaults and sets the internal macros.

```
286 \def\task@set#1{\edef\task@id{task\thetask@all}
287 \def\task@force{1}
288 \let\task@start=\relax\let\task@len=\relax
289 \setkeys{task}{#1}}
```

**tasklist**

```
290 \newenvironment{task}[1]{}
291 {\@task{#1}\item[\dfg@target{task}{\taskin\task@id\wp@id}{\task@label{\thetask@wp}}}%
292 \@ifundefined{task@title}{\textbf{\task@title}}
293 {}}
```

**\@task** The `\@task` macro is an internal macro which takes a bunch of keyword keys and writes their values to the aux file.

```
294 \newcounter{task@all}\newcounter{task@wp}[wp]
295 \newcount\task@@end
296 \def\@task#1{\stepcounter{task@all}\stepcounter{task@wp}
297 \task@set{#1}
298 \@ifundefined{task@start}%
299 {\protect\G@refundefinedtrue\@latex@warning{start of task #1 undefined}}%
300 {\dfg@def{task}{\taskin\task@id\wp@id}{start}\task@start}
301 \@ifundefined{task@len}%
302 {\protect\G@refundefinedtrue\@latex@warning{length of task #1 undefined}}%
303 {\dfg@def{task}{\taskin\task@id\wp@id}{len}\task@len}
```

```

304 \dfg@def{task}{\taskin\task@id\wp@id}{force}\task@force
305 \dfg@def{task}{\taskin\task@id\wp@id}{label}{\task@label\thetask@wp}%
306 \dfg@def{task}{\taskin\task@id\wp@id}{number}{\thetask@wp}%
307 \dfg@def{task}{\taskin\task@id\wp@id}{page}{\thepage}%
308 \update@tasks{\taskin\task@id\wp@id}
309 \def\workphase#1{\@task{#1}}

```

`\localtaskref`

```

310 \def\localtaskref#1{\euRef{task}{\wp@id @#1}{label}}

```

`\taskref`

```

311 \def\taskin#1#2{#2@#1}
312 \def\taskref#1#2{\WPref{#1}\euRef{task}{#1@#2}{label}}
313 \def\tasktrf#1#2{\WPref{#1} (Task \euRef{task}{#1@#2}{number})}
314 \newcounter{ganttd@deps}
315 \def\@requires#1#2{\stepcounter{ganttd@deps}%
316 \edef\dep@id{taskdep\theganttd@deps}%
317 \dfg@def{taskdep}\dep@id{from}{\taskin{#1}\wp@id}%
318 \dfg@def{taskdep}\dep@id{to}{#2}%
319 \update@deps\dep@id}

```

## 4.5 Project-Related Papers

`\dfgprojpapers`

We first define a bibLaTeX bibliography heading that does not create headers, then we generate a subsection with a `refsection` (this makes a separate bibliography for this section) and activate the keys via `\nocite`. Then we just print the bibliography with the empty header we created before.

```

320 \defbibheading{empty}{}
321 \def\dfgprojpapers#1{%
322 \subsection{Project-Related List of Publications}\label{sec:ourpubs}
323 \begin{refsection}
324 \nocite{#1}
325 \printbibliography[heading=empty]
326 \end{refsection}}

```

## 4.6 Referencing and Hyperinking

`\@dfg@def`

This macro stores the value of its last argument in a custom macro for reference.

```

327 \def\@dfg@def#1#2#3#4{\expandafter\gdef\csname #1@#2@#3\endcsname{#4}}

```

`\euRef`

```

328 \def\euRef#1#2#3{\@ifundefined{#1@#2@#3}%
329 \protect\G@refundefinedtrue\@latex@warning{#3 for #1 #2 undefined}??}%
330 \csname #1@#2@#3\endcsname}}%
331 \def\euRef@aux#1#2#3{\@ifundefined{#1@#2@#3}{??}{\csname #1@#2@#3\endcsname}}%
332 \def\euRef@num#1#2#3{\@ifundefined{#1@#2@#3}{0}{\csname #1@#2@#3\endcsname}}%
333 \def\euRef@safe#1#2#3{\@ifundefined{#1@#2@#3}{-}{\csname #1@#2@#3\endcsname}}%

```

`\euRef`

```

334 \def\euRef#1#2#3{\@ifundefined{#1@#2@#3}%
335 \protect\G@refundefinedtrue\@latex@warning{#3 for #1 #2 undefined}??}%
336 \hyperlink{#1@#2@target}{\csname #1@#2@#3\endcsname}}

```

`\W*ref`

```

337 \def\WPref#1{\euRef{wp}{#1}{label}}
338 \def\WPtref#1{\euRef{wp}{#1}{label}: \euRef{wp}{#1}{title}}
339 \ifwork@areas

```

```

340 \def\WAref#1{\euRef{wa}{#1}{label}}
341 \def\WAtref#1{\euRef{wa}{#1}{label}: \euRef{wa}{#1}{title}}
342 \fi

```

\dfgcount

```

343 \def\@count #1{\ifcase #1 zero\or one\or two\or three\or four\or five\or six\or seven \or
344 eight\or nine\or ten\or eleven \or twelve\else#1\fi}
345 \def\dfgcount#1#2{\@count{\euref@num{#1}{#2}{count}}}

```

## 4.7 The Work Package Table

\dfg@lead

```

346 \def\dfg@lead#1{\@ifundefined{wp@#1@lead}%
347 {\protect\G@refundefinedtrue\@latex@warning{lead for WP #1 undefined}??}%
348 {\csname wp@#1@lead\endcsname}}

349 \def\wa@style#1{\cellcolor{lightgray}{\textbf{#1}}}
350 \def\wp@style#1{#1}

```

wptable

```

351 \if@RAM
352 \newenvironment{wptable}%
353 {\begin{tabular}{|l|l|l|l|l|}\hline%
354 \textbf{\ifwork@areas{WA/P}\else{WP}\fi}&
355 \textbf{Title}&
356 \textbf{p.}&
357 \textbf{RM} &
358 \textbf{RAM}\\\hline\hline}
359 {\end{tabular}}
360 \else
361 \newenvironment{wptable}%
362 {\begin{tabular}{|l|l|l|l|l|}\hline%
363 \textbf{\ifwork@areas{WA/P}\else{WP}\fi}&
364 \textbf{Title}&
365 \textbf{p.}&
366 \textbf{RM}\\\hline\hline}
367 {\end{tabular}}
368 \fi

```

EdNotes (w)fig 4

```

369 \define@key{wfig}{size}{\def\wfig@size{#1}\@dmp{size=#1}}
370 \def\@sw#1{\begin{sideways}#1\end{sideways}}
371 \newcommand{\wfig}[1][1]{\setkeys{wfig}{#1}}
372 \newcount\@@@RM\if@RAM\newcount\@@@RAM\fi
373 {\gdef\@wp@lines{}%initialize
374 \let\@tabularnewline\relax\let\@hline\relax\let\@cellcolor\relax% so they
375 \let\wa@style\relax\let\wp@style\relax \let\@sw\relax\let\textbf\relax% do not
376 \let\G@refundefinedtrue=\relax\let\@latex@warning=\relax\let\hyperlink=\relax% bother us
377 \gdef\wfig@headline{\textbf{\ifwork@areas{WA/P}\else{WP}\fi}&\textbf{Title}&\textbf{p.}}
378 \if@sites
379 \@for\@site:=\dfg@gen@sites\do{\xdef\wfig@headline{\wfig@headline&\@sw{\@site\ RM}\if@RAM&\@sw{\@site\ RAM}}
380 \else
381 \xdef\wfig@headline{\wfig@headline &\@sw{RM}\if@RAM&\@sw{RAM}}\fi}
382 \fi
383 \edef\@@was{\euref@safe{all}{wa}{ids}}%

```

<sup>4</sup>EDNOTE: The computation can be distributed much more efficiently (by intermingling the counter advances with the row creation), but this works now

```

384 \@for\@wa:=\@was\do{% iterate over the work areas
385 \xdef\@wa@line{\wa@style{\euRef{wa}\@wa{label}}&
386 \wa@style{\@ifundefined{wa@\@wa @short}{\euref{wa}\@wa{title}}{\euref{wa}\@wa{short}}}&
387 \wa@style{\euref{wa}\@wa{page}}}}
388 \if@sites
389 \@for\@site:=\dfggen@sites\do{\@@RM=0\if@RAM\@@@RAM=0\fi
390 \edef\@wps{\euref@safe\@wa{wp}{ids}}%
391 \@for\@wp:=\@wps\do{% compute the WA efforts
392 \advance\@@@RM by \euref@num\@wp\@site{RM}%
393 \if@RAM\advance\@@@RAM by \euref@num\@wp\@site{RAM}\fi}
394 \xdef\@wa@line{\@wa@line&\wa@style{\the\@@@RM}\if@RAM&\wa@style{\the\@@@RAM}\fi}}
395 \else
396 \@@@RM=0\if@RAM\@@@RAM=0\fi
397 \edef\@wps{\euref@safe{all}{wp}{ids}}%
398 \@for\@wp:=\@wps\do{% compute the efforts
399 \advance\@@@RM by \euref@num{wp}\@wp{RM}%
400 \if@RAM\advance\@@@RAM by \euref@num{wp}\@wp{RAM}\fi}
401 \xdef\@wa@line{\@wa@line&\wa@style{\euref{wa}\@wa{RM}}}
402 \if@RAM&\wa@style{\euref{wa}\@wa{RAM}}\fi}%
403 \fi
404 \xdef\@wp@lines{\@wp@lines\@wa@line\tabularnewline\hline}% add the line for the workarea
405 \edef\@wps{\euref@safe\@wa{wp}{ids}}%
406 \@for\@wp:=\@wps\do{% iterate over its work packages
407 \xdef\@wp@line{\euRef{wp}\@wp{label}}&
408 \@ifundefined{wp@\@wp @short}{\euref{wp}\@wp{title}}{\euref{wp}\@wp{short}}&
409 \euref{wp}\@wp{page}}
410 \if@sites
411 \@for\@site:=\dfggen@sites\do{%
412 \xdef\@wp@line{\@wp@line&\wp@style{\euref\@wp\@site{RM}}\if@RAM&\wp@style{\euref\@wp\@site{RAM}}\fi}}%
413 \else
414 \xdef\@wp@line{\@wp@line&\wp@style{\euref{wp}\@wp{RM}}\if@RAM&\wp@style{\euref{wp}\@wp{RAM}}\fi}%
415 \fi
416 \xdef\@wp@lines{\@wp@lines\@wp@line\tabularnewline\hline}}
417 \gdef\@totals{}
418 \if@sites
419 \@for\@site:=\dfggen@sites\do{\@@RM=0\if@RAM\@@@RAM=0\fi
420 \edef\@was{\euref@safe{all}{wa}{ids}}%
421 \@for\@wa:=\@was\do{%
422 \edef\@wps{\euref@safe\@wa{wp}{ids}}%
423 \@for\@wp:=\@wps\do{% iterate over the work areas
424 \advance\@@@RM by \euref@num\@wp\@site{RM}%
425 \if@RAM\advance\@@@RAM by \euref@num\@wp\@site{RAM}\fi}}
426 \xdef\@totals{\@totals & \textbf{\the\@@@RM}\if@RAM& \textbf{\the\@@@RAM}\fi}}
427 \else
428 \@@@RM=0\if@RAM\@@@RAM=0\fi
429 \edef\@was{\euref@safe{all}{wa}{ids}}%
430 \@for\@wa:=\@was\do{\edef\@wps{\euref@safe\@wa{wp}{ids}}}%
431 \@for\@wp:=\@wps\do{% iterate over the work areas
432 \advance\@@@RM by \euref@num{wp}\@wp{RM}%
433 \if@RAM\advance\@@@RAM by \euref@num{wp}\@wp{RAM}\fi}}
434 \xdef\@totals{\&\the\@@@RM\if@RAM &\the\@@@RAM\fi}
435 \fi}
436 \begin{figure}[ht]\centering%
437 \if@RAM
438 \begin{tabular}{|l|l|l|*{\the@sites}{|r|r|}}\hline
439 \else
440 \begin{tabular}{|l|l|l|*{\the@sites}{|r|}}\hline
441 \fi

```



```

442 \wpfig@headline\\\hline\hline
443 \@wp@lines\hline
444 \multicolumn{3}{|c|}{\textbf{totals}}\@totals\\\hline
445 \end{tabular}
446 \caption{Work Areas and Work Packages [R{\if@RAM(A)\fi}M $\widehat{=} $ Researcher \if@RAM(Assistant)\fi Months]
447 \end{figure}}

```

## 4.8 Gantt Charts

Gantt Charts are done with help of the the `tikz` package. The `gantt` environments pick up on the declared duration of the proposal in months stored in the `\dfg@gen@months` macro.

We define the keys for Gantt tables

```

448 \newif\ifgantt@draft
449 \define@key{gantt}{xscale}{\def\gantt@xscale{#1}}
450 \define@key{gantt}{yscale}{\def\gantt@yscale{#1}}
451 \define@key{gantt}{step}{\def\gantt@step{#1}}
452 \define@key{gantt}{size}{\def\gantt@size{#1}}
453 \define@key{gantt}{draft}[true]{\gantt@drafttrue}
454 \newif\ifgantt@RMdisclaimer\gantt@RMdisclaimerfalse
455 \define@key{gantt}{RMdisclaimer}[true]{\gantt@RMdisclaimertrue}

```

Then we define an auxiliary function that provides defaults for these keys and sets the internal macros.

```

456 \def\gantt@set#1{\gantt@draftfalse\def\gantt@xscale{1}\def\gantt@yscale{.35}\def\gantt@step{3}
457 \setkeys{gantt}{#1}}

```

Finally, the Gantt Chart environment itself.

`gantt`

```

458 \newenvironment{gantt}[1] []
459 {\gantt@set{#1}
460 \@ifundefined{gantt@size}{}{\csname\gantt@size\endcsname}
461 \def\gantt@wps{\euref@num{all}\wp}{count}}
462 \newdimen\gantt@ymonths
463 \gantt@ymonths=\gantt@wps cm
464 \advance\gantt@ymonths by .5cm
465 \begin{tikzpicture}[xscale=\gantt@xscale,yscale=\gantt@yscale]
466 \draw[xstep=\gantt@step,gray,very thin] (0,0) grid (\dfg@gen@months,\gantt@wps);
467 \foreach \x in {0,\gantt@step,...,\dfg@gen@months} \node at (\x,\gantt@ymonths) {\x};}
468 {\end{tikzpicture}}

```

In this we have used the macro that does the actual painting.

```

469 \newdimen\gantt@ymid\newdimen\gantt@yinc\newdimen\gantt@xend
470 \newcommand{\@action}[5]{%\message{@action: id=#1, line=#2, start=#3, dur=#4, int=#5.}
471 \gantt@ymid=#2 cm\gantt@yinc=\gantt@yscale cm
472 \gantt@xend=#3 cm\advance\gantt@xend by #4 cm
473 \advance\gantt@ymid by \gantt@yinc
474 \fill (#3,#2) rectangle +(#4,#5);
475 \node (#1@left) at (#3,\gantt@ymid) {};}
476 \node (#1@right) at (\gantt@xend,\gantt@ymid) {};}
477 \def\@dependency#1#2{\draw[->,line width=2pt,color=red] (#1@right) -- (#2@left);}

```

`\ganttchart` This macro iterates over the work areas, their work packages, and finally their work phases to use the internal macro `\@action`. All of this in the `gantt` setting.

```

478 \newcommand{\ganttchart}[1] []{\begin{figure}[ht]\centering
479 \begin{gantt}[#1]
480 \newcounter{taskwps}\newcount\@@line
481 \edef\@@was{\euref@safe{all}\wa}{ids}}

```

```

482 \for\@wa:=\@was\do{% iterate over work areas
483   \edef\@wps{\euref@safe\@wa{wp}{ids}}
484 \for\@wp:=\@wps\do{% iterate over work packages
485   \stepcounter{taskwps}
486   \@@line=\gantt@wps\advance\@@line by -\thetaskwps
487   \edef\@tasks{\euref@safe\@wp{task}{ids}}
488   \node at (-1/\gantt@xscale,\@@line) [above=-2pt] {\euref@wp\@@wp{label}};
489   \for\@task:=\@tasks\do{% iterate over work phases
490     \@action\@@task\@@line
491     {\euref@num{task}\@@task{start}}
492     {\euref@num{task}\@@task{len}}
493     {\euref@num{task}\@@task{force}}}}}}
494 \edef\@@deps{\euref@safe{all}{task}{deps}}
495 \for\@@dep:=\@@deps\do{\@dependency{\euref@safe{taskdep}\@@dep{from}}{\euref@safe{taskdep}\@@dep{to}}}
496 \ifgantt@draft
497   \newcounter{gantt@month}
498   \newcount\@@e\newdimen\gantt@effort\newdimen\gantt@plus
499   \@whilenum\thegantt@month<\dfg@gen@months\do{% step over months
500     \gantt@effort=0cm
501     \edef\@@was{\euref@safe{all}{wa}{ids}}
502     \for\@wa:=\@@was\do{% iterate over work areas
503       \edef\@wps{\euref@safe\@wa{wp}{ids}}
504       \for\@wp:=\@wps\do{% iterate over work packages
505         \edef\@tasks{\euref@safe\@wp{task}{ids}}
506         \for\@task:=\@tasks\do{% iterate over work phases
507           \@@e=\euref@num{task}\@@task{start}
508           \advance\@@e by \euref@num{task}\@@task{len}
509           \ifnum\thegantt@month<\euref@num{task}\@@task{start}\else
510             \ifnum\thegantt@month<\@@e
511               \gantt@plus=\euref@num{task}\@@task{force}cm
512               \advance\gantt@effort by \gantt@plus\fi\fi
513               \fill (\thegantt@month,-5) rectangle +(1,\gantt@effort);}}}}
514         \stepcounter{gantt@month}}\fi
515 \end{gantt}
516 \ifgantt@RMdisclaimer
517 \caption{Overview Work Package Activities}
518 \else
519 \caption{Overview Work Package Activities \if@RAM(the table only shows efforts
520   for the junior researchers (RM))\fi}
521 \fi\label{fig:activities}
522 \end{figure}}

```

## 4.9 Miscellaneous

The next macro is generally useful to put a comment at the end of the line, possibly making a new one if there is not enough space.

`\signatures`

```

523 \newcommand{\signatures}{\section{Signatures \deu{(Unterschriften)}}
524 \quad\quad\quad\quad. \quad\quad\quad\quad. \quad\quad\quad\quad\quad\quad[2ex]
525 \strut\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad
526 \dcm@pers@ref\@p{personaltitle}~\dcm@pers@ref\@p{name}\hfill}}

```

`\@dmp`

```

527 \def\@dmp#1{\ifkeys\marginpar{#1}\fi}

```

`\texteuro`

```

528 \def\texteuro{\officialearrowleft\xspace}

```

## Change History

v0.4		v0.7	
General: copied over from eustrep.dtx . . . .	1	General: added final report infrastructure .	1
v0.5		added noRAM option . . . . .	1
General: added gantt charts . . . . .	1	added paper list infrastructure . . . . .	1
v0.6			
General: added task referencing system . . .	1		

## References

- [For] Deutsche Forschungsgemeinschaft. *Guidelines and Proposal Preparation Instructions*. URL: [http://www.dfg.de/forschungsfoerderung/formulare/download/1\\_02.pdf](http://www.dfg.de/forschungsfoerderung/formulare/download/1_02.pdf) (visited on 10/27/2010).
- [Koh07] Michael Kohlhase. *Editorial Notes for L<sup>A</sup>T<sub>E</sub>X*. Self-documenting L<sup>A</sup>T<sub>E</sub>X package. Comprehensive T<sub>E</sub>X Archive Network (CTAN), 2007. URL: <http://www.ctan.org/get/macros/latex/contrib/ed/ed.pdf>.
- [Koh10] Michael Kohlhase. *dcm.sty: An Infrastructure for marking up Dublin Core Metadata in L<sup>A</sup>T<sub>E</sub>X documents*. Self-documenting L<sup>A</sup>T<sub>E</sub>X package. Comprehensive T<sub>E</sub>X Archive Network (CTAN), 2010. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/dcm/dcm.pdf>.