

H2020 Proposal Class

A LaTeX class for writing EU RIA H2020 proposals.

Giacomo Indiveri
Institute for Neuroinformatics
University of Zurich and ETH Zurich
`giacomo@ini.uzh.ch`

October 4, 2026

Abstract

This class provides macros and commands for automating the creation of cross-referenced tables in Grant Proposals for the EU Research and Innovation Actions (RIA). The appropriate tables are created, updated and (re)sorted using the data entered in the class commands, across the document.

Contents

1	Introduction	2
1.1	Template document structure	2
1.2	Class Options	2
1.3	Class Requirements	3
2	Class commands	3
2.1	<code>\maketitle</code>	3
2.2	<code>\makewplist</code>	4
2.3	<code>\makedeliverablelist</code>	4
2.4	<code>\makemilestonelist</code>	4
2.5	<code>\makerisklist</code>	5
2.6	<code>\makesummaryofefforttable</code>	5
2.7	<code>\maketasklist</code>	5
2.8	<code>\makecoststable</code>	5
2.9	<code>\makelritable</code>	6

3	The workpackage environment	6
3.1	<code>\makewptable</code> command	7
3.2	The <code>wprojectives</code> environment	7
3.3	The <code>wpdescription</code> environment	7
3.4	The <code>wpdeliverables</code> environment	8
4	Labels and References	8
5	Conclusions	8

1 Introduction

The proposals submitted to the H2020 EU program have a very well defined structure, with well defined sections and several cross-references among sections and tables. The templates are available at the EU H2020 Participant Portal http://ec.europa.eu/research/participants/portal/desktop/en/funding/reference_docs.html#h2020-call_ptef-pt

The `h2020proposal` class implements a series of macros and commands to simplify the formatting of the document and automate the creation of the several cross-referenced tables required by the template. This class was inspired by the 2009 `fet-workpackage.sty` file developed by Dennis Goehlsdorf and by the 2008 EU proposal LaTeX template files developed by Elisabetta Chicca and Chiara Bartolozzi in 2009. It is the extension of the `ICTProposal` class, designed by Giacomo Indiveri to create large and complex Integrated-Project proposals of the past FP7 program.

1.1 Template document structure

The document structure should have a Cover page, an Table of Contents (for ICT proposals) or an Abstract (for FET proposals), and five main sections. The first three sections are: Excellence (Section 1.), Impact (Section 2.), and Implementation (Section 3.). The total length of these three sections, including the cover page, has strict page limits. The last two sections include a part dedicated to the description of the “Members of the consortium” (Section 4.) and one related to “Ethics and Security” (Section 5.) These two sections are not covered by page limit.

The `h2020proposal` class commands start with the data that needs to appear in the cover page, and most of the other commands manage the cross-referenced tables and lists that appear in Section 3. The other Sections do not require any special commands, and can be created using the standard LaTeX commands.

1.2 Class Options

The user can specify the option “draft”. In this case, guidelines from the EU *Guide for Applicants* are printed, latex labels are displayed on the side, and

page-breaks are added between work-packages.

1.3 Class Requirements

The `h2020proposal` class is based on (and requires) the extremely feature rich and configurable `memoir` class. Additional packages required by the `h2020proposal` class are:

- `coolstr` package
- `longtable` package
- `colortbl` package

The `h2020proposal` class also supports the automated creation of the required Gantt chart. For this, it is necessary to include the `pgfgantt` package.

By using different `memoir` class options, it is possible to configure and personalize most of the proposal aspects.

2 Class commands

2.1 `\maketitle`

The `\maketitle` command has been redefined to create the proposal title-page. It uses the data obtained from the following macros:

`\title{}` This macro requires one argument: the full title of the proposal.
E.g., `\title{Future Emerging Technologies}`

`\shortname{}` This macro require as argument the proposal short name.
E.g., `\shortname{FET}`

`\fundingscheme{}` This macro require one single argument, which should be either the text “Small or medium scale focused research project (STREP)”, or the text “Large-scale integrating projects (IP)”.

`\topic{}` The argument of this macro should be a text string with the specific Work-Program topic addressed.

`\coordinator{ }{ }` This macro requires three arguments. The first argument should be the full name of the coordinator, the second argument should be the coordinator’s email address, and the third argument should have the coordinator’s fax number.
E.g., `\coordinator{John Doe}{j.doe@null.com}{+41 44 11223344}`

`\participant{ }{ }` This macro requires three arguments as well. The first argument should be the full name of the participant’s institution, the second argument should be the institution short name, and the third argument should be the participant’s Country. There should be as many

`\participant{ }{ }{ }` macros as the number of participants. The first instance of this macro should be reserved for the coordinator.

E.g., `\participant{University of Zurich}{UZH}{Switzerland}`

`\titlelogo{ }` This is an optional macro to include an image as logo on the title page. It has two arguments: the first argument is the name (and path) of the image file, and the second argument is the image scaling factor.

E.g., `\titlelogo{logo.png}{0.25}`

2.2 `\makewplist`

This command creates the work package (wp) list required as Table 3.1b in the EU proposal template. It is a table with a list of work packages with their details (*e.g.*, wp leader, wp start-month, wp end-month, *etc.*). The list is compiled automatically by using the data defined in each `workpackage` environment (see Section 3). As this data is typically defined later in the document, the `h2020proposal` class creates an auxiliary file (`<jobname>.lwp`) and uses it at the second round of compilation to generate the table. In order to comply with the table numbering provided by the template, it is necessary to redefine the `table` counter before issuing this command: *e.g.*

```
\renewcommand{\thetable}{\thesection\alph{table}}
\makewplist
```

2.3 `\makedeliverablelist`

This command creates the List of Deliverables Table 3.1c required by the EU template. As for the wp list, it uses the data defined (typically later) in each `workpackage` environment. The required information is saved in the auxiliary file `<jobname>.ldl>` and used in subsequent compilation runs to create the table. This command sorts the table rows by delivery date.

2.4 `\makemilestonelist`

This command creates the List of Milestones Table 3.2a required by the EU template. Milestones need to be defined before this command is issued, by using the `\milestone` macro:

`\milestone[ ]{ }{ }{ }` This macro has four arguments. The first (optional) argument is the milestone delivery month; the second argument is the name of the milestone; the third argument is the means of verification, and the last argument is the list of work packages involved. The list of work packages can use the latex `\ref` command.

Example: `\milestone[24]{Completed algorithmic model}{Software package released and validated}{WP\ref{wp:modeling}}`

2.5 `\makerisklist`

This command creates the Critical risks for implementation Table 3.2b required by the EU template. Critical risks relating to project implementation need to be defined before this command is issued, by using the `\criticalrisk` macro:

`\criticalrisk{}{}{}` This macro has three arguments. The first argument is the description of the risk; the second is the list of work packages involved, and the third should describe any risk mitigation measures, and the actions planned for it. The list of work packages involved can use the latex `\ref` command.

Example: `\criticalrisk{The dedicated chip sent to fabrication is not functional.}{WP \ref{wp:testwp1}}{Resort to Software simulations}`

2.6 `\makesummaryofefforttable`

This command is typically issued after defining the various work packages (via the `workpackage` environment). It uses the data defined in each `workpackage` environment, saves it to the auxiliary file `<jobname>.lse` and creates the table in subsequent compilation runs. The WP leader is identified in the table by showing the relevant person-months figure in bold (as specified in the EU template).

2.7 `\maketasklist`

This command is not required by the EU template, but it can be useful to have an overview of who is responsible for which task, in which WP. The command creates an ordered list of tasks with Task ID, PI name (in case it is defined in the `\wptask` macro), start and end dates, and task title. The required information is saved in the auxiliary file `<jobname>.ltk>` and used in subsequent compilation runs to create the table.

2.8 `\makecoststable`

This command creates a summary of costs table, for each participant, if the sum of the costs for “travel”, “equipment”, and “goods and services” exceeds 15% of the personnel costs for that participant. In order to enter the data for these tables it is necessary to use the following macros:

`\costsTravel{}{}{}` This macro accepts three arguments. The first is the participant short name; the second has to be a number indicating the total amount of euro requested for travel expenses; and the third argument is a text string describing the justification for the requested budget.

`\costsEquipment{}{}{}` Similar to the previous macro, this one accepts three arguments: the participant short name, the total amount of euro requested for equipment expenses, and a text string describing the justification for the requested budget.

that the participant specified is the wp leader. *E.g.*

```
\personmonths*{UZH}{24}
```

```
\personmonths{ETHZ}{18}
```

(the UZH partner is the wp leader and will use 24 person months, while ETHZ is an additional wp participant that will use 18 person months).

3.1 \makewptable command

This command creates the top table in the wp environment with all the info specified using the macros described above. The WP leader data is shown in boldface.

Three additional environments need to be defined within the `workpackage` environment, to complete the wp description: the Objectives (`wprojectives`), the Description of work (`wpdescription`), and the Deliverables (`wpdeliverables`) environments.

3.2 The wprojectives environment

This environment has no specific macros. It is useful for creating a framed paragraph, in which the user can write arbitrary text. *E.g.*

```
\begin{wprojectives}
  This work package has the following objectives:
  \begin{enumerate}
    \item To develop ....
    \item To apply this ....
    \item etc.
  \end{enumerate}
\end{wprojectives}
```

3.3 The wpdescription environment

This environment creates a separate framed paragraph, in which the details of the wp are specified. Within this environment it is possible to define multiple Tasks, using the `\wptask` macro:

`\wptask{}{}{}{}{}{}` This macro has 6 arguments. The first argument specifies the name of the Task leader responsible for the task using the short name defined with the `\participant` macro. The second argument specifies the list of contributor partners. The third and fourth arguments specify task start and end date respectively. The fifth argument represents the Task title, while the last one is text with a short description of the task.

The `\wptask` macro creates an ordered list of tasks within the `wpdescription` environment and stores the data relative to the WP tasks. The information specified here is used also by the `\maketasklist` macro, which creates an ordered list of tasks to show which PI is responsible for which task.

The class is still in a beta-testing stage. Please do not distribute. If you have suggestions or find bugs and problems, please report them to Giacomo Indiveri giacomo@ini.uzh.ch.

Copyright and license

The `h2020proposal` Class

Copyright 2015 ©, Giacomo Indiveri

This latex class is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

`h2020proposal.cls` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with `h2020proposal`. If not, see <http://www.gnu.org/licenses/>.

Disclaimer

Use the original source and the <http://ec.europa.eu/> documentation for reference. We make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability or availability with respect to the original template. In no event will we be liable for any loss or damage including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising out of, or in connection with, the use of this template and/or class.