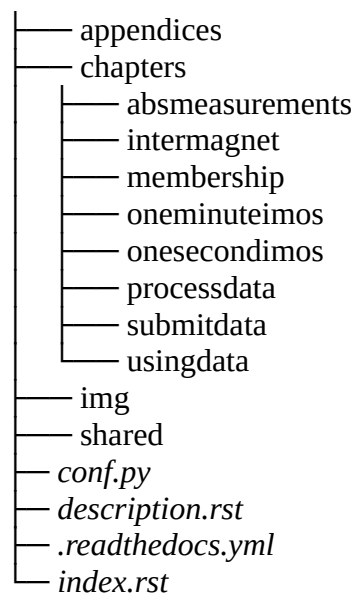


Guidelines SPHINX

This guideline contains a summary of the most important features and the way we use them to structure the PDF and HTML generation of the Intermagnet Technical Manual. It is not an explanation of sphinx or rst.

Folder layout



We can identify following folders

- The folder chapters will contain one folder for each chapter. The chapter directories group together the .rst files for each chapter.
- The folder img contains all image files used in the technical manual
- The folder shared contains some rst files that can be imported by all the other rst files
- The folder appendices that will contain all appendices rst files

On the root directory there are three files that are important to mention :

- `conf.py` is a config file for sphinx and can be used to change the way PDF or HTML is generated. It also is the place where modules/extensions can be configured and added to sphinx.
- `.readthedocs.yml` is a config file that contains directives for readthedocs to be able to autogenerate the needed formats.

- index.rst is the root rst file that contains the top table of contents for sphinx which is explained in the next section.

Table of contents

The index.rst contains the highest level of the table of contents tree

```
#####
INTERMAGNET Technical Reference Manual
#####

.. only:: html

    .. figure:: ./img/cover_intermagnet.png
       :align: center
       :alt: Intermagnet

    **version :** |version|

.. include:: ./description.rst

.. toctree::
   :caption: Table Of Contents
   :maxdepth: 5
   :numbered:

   chapters/intermagnet/toc_intermagnet
   chapters/oneminuteimos/toc_oneminuteimos
   chapters/onesecondimos/toc_onesecondimos
   ...

.. only:: html

    .. toctree::
       :maxdepth: 5

       appendices/...

.. raw:: latex

    \listoffigures
    \listoftables
```

Text 1: index.rst

There is one toctree element that will point to a chapter element with the name of that chapter preceded by toc_. This corresponds with an .rst file that contains the toctree of that chapter. This way of working was chosen to have a complete autonumbered and navigational chapter and sections in HTML and PDF. For the first chapters it points to chapters/intermagnet/toc_intermagnet.rst

```
.. _inter:

*****
Intermagnet
*****

.. toctree::
   :maxdepth: 4

   intro
   missionvision
   historystatus
   principles
   participation
   product
   condofuse
   management
```

Text 2: toc of first chapter

The structure of the table of contents determines the numbering and the order of each section.

To simplify and understand the way of working an rst file is chosen for each section on the highest level of the chapter. In this case we have a rst file for each section 1.1, 1.2, 1.3 etc

The indication in the top of each of these chapter table of contents `.. _inter:` is the way we can mark an identifier in sphinx to be used afterwards in referencing chapters and sections. More on that in the next section.

Sections and referencing

For each chapter we have of `toc_....rst` file. The title of the chapter is in this toc file as you can see in Text 2 the name of the first chapter is Intermagnet and it is surrounded by stars. This is an indication for sphinx that he has to interpret it as a chapter title.

In an `intro.rst` document we can use titles to have sections and sub sections

```
.. _inter_intro:
```

```
Introduction  
=====
```

Text 3: Sections

While doing so you will have the Introduction being numbered as 1.1 because of the order in the toc files. The `intro.rst` is placed on the first position so that this section will be identified as being the first section of the first chapter. Changing the order in the `index.rst` and `toc_intermagnet.rst` will change the numbering automatically. Further on you see in green `.. _inter_intro:` this is a logical name that can be used to reference to the introduction. This logical name needs to be set whenever we want to reference a section. The way this logical name is constructed is `.. _chapter_section:` for the chapter name we used `inter` and `intro` for the section name. This enables us to have a unique id that can be used everywhere in the overall document.

There are now two ways of referencing it

- `:numref:`inter_intro`` resulting in anchor Section 1.1
- `:ref:`inter_intro`` resulting in anchor Introduction

The logical name should be unique over the whole document. The standard of `chapter_section_subsection` guarantees this uniqueness.

For the chapter itself we saw that in the `toc_intermagnet.rst` we putted the `.. _inter:` before the name of the title.

This logical name gives us the possibility to reference the chapter by an `:numref:`inter``. This will be transformed by sphinx into an anchor Section 1. However we want to have Chapter 1 as indication instead of sphinx default Section 1. For this we can easily indicate in the numref another name instead of Section by simply typing `:numref:`Chapter %s <inter>`` and this will result in Chapter 1. This way of working gives us a unique possibility to reference whatever chapter/section without typing the real numbers.

Shared Folder

The shared folder has as aim to put rst files that can be used by other rst files (different sections).

There is one file defined as variables.rst

```
.. |contact| replace:: secretary_intermagnet@gfz-potsdam.de
.. |intermagnet_website| replace:: http://www.intermagnet.org
```

Text 4: variables.rst

The variables.rst contains definition of variables and bind them to a value.

The way variables are defined is that you define a logical name here |contact| and define it as a concrete email address in this case secretary_intermagnet@gfz-potsdam.de. From now on, we can use the variable name |contact| in an rst file without typing in the email. If the email changes we just update the variables.rst and the email is correctly updated in all other rst files.

To be able to use the variables in an rst file you need to import the rst file.

```
.. _inter_cond_use:
Conditions Of Use
=====

.. include:: ../../shared/variables.rst

- A citation reference should be sent to the INTERMAGNET
  Secretary |contact| for inclusion in a publications list on the
  INTERMAGNET web site.
```

Text 5: Example of import

Whenever you refer to a variable they will be replaced by there literal definition.

Sometimes we don't want to have literal replacements with web links but we want a clickable word that will direct you to a particular website. For example [application form](#) and when we click on the application form we need to be directed to the website needed. The way this is done in rst is by splitting it up in the clickable word and the target.

```

chapter.rst
....
`application form`_ on the INTERMAGNET web site
....
....

targets.rst
....
.. _application form: https://www.intermagnet.org/imos/apply-eng.php

```

Text 6: logical name and target

To avoid the spreading of urls over the rst's a targets.rst was created. That contains only the target definition and while importing it, it can be used to reference it by typing the same name as the target and ending it with an underscore `application form`_

If websites change all targets and variables need to be checked and updated.

Directives of sphinx used

Directives are extensions to sphinx and most of them are described here :

<https://www.sphinx-doc.org/en/master/usage/restructuredtext/directives.html>

Although I quickly mention some of them used in the technical manual

.. tabularcolumns::

The way tables are described you can easily find out in restructured text manuals on the websites

We make sometimes use of an extra directive tabularcollumns

```

.. tabularcolumns:: |p{3.5cm}|p{9cm}|

.. table::
   :widths: auto
   :align: center

```

Text 7: tabular column

The tabular column directive has only influence on the way tables are produced in PDF format (latex). Here we specify explicitly the dimensions of each row because the automatic width do not give the best results in the PDF.

.. math::

The directive math gives basically the possibility to define mathematical formula on a similar way as latex does. You can find some examples of the usage in Section 6.5 of the intermagnet manual.

.. math::

$$\begin{aligned} V_0(t_1) &= Z(t_1) - V_{\text{var}}(t_1) \\ N_0(t_1) &= \sqrt{(H^2(t_1) - E_{\text{var}}^2(t_1)) - N_{\text{var}}(t_1)} \\ D_0(t_1) &= D(t_1) - \text{atan}\left\{\frac{E_{\text{var}}(t_1)}{N_0(t_1) + V_{\text{var}}(t_1)}\right\} \end{aligned}$$

the & is the place where mathematical functions need to be aligned on. (most of the things that work in latex will work here).

other directives used :

- ..note::
- ..figure::

These are straight forward and can be understood easily.

Appendices

Appendices needed to be treated separately because they are not known to html and only make sense in the PDF format.

The use of :numref: could not be applied to appendices because the numref continues numbering in 1,2,3,4 and for appendices we want A B C D E F

The way we decided to use the appendices is build upon different components to have the ability to easily manage the appendices and reference them without explicitly typing in the A B all over the place.

So basically it comes down to following compromise :

1. For each appendix needed we make an rst file containing everything of this appendix
2. For the html we have a toctree entry in the regular index.rst
3. For the PDF generation we need to add appendices to the conf.py
4. We will use shared variables to references towards the appendices without typing in the real label (located in appendices/appendices.rst)

The toctree for html appendices

```
.. only:: html
    .. toctree::
        :maxdepth: 5
        appendices/terminology
        appendices/observatories
        appendices/archivedataformats
        appendices/imagaddresses
        appendices/dataformats
        appendices/filters
        appendices/acknowledgements
```

Text 8: toctree appendices

The toctree is a part of the index.rst and is placed at the end of the index.rst.

The reference of the appendices are not indicated as numbered this means that within the html the appendices have no numbers or alphabetical letters assigned to them.

The order is however important and needs to be the same as used in the next sections being the PDF assignment in the conf.py and the shared variables appendices.rst

Latex appendices in conf.py

```
latex_appendices = ['appendices/terminology',
                    'appendices/observatories',
                    'appendices/archivedataformats',
                    'appendices/imagaddresses',
                    'appendices/dataformats',
                    'appendices/filters',
                    'appendices/acknowledgements'
                    ]
```

Text 9: latex appendices in conf.py

While assigning the appendices to the variable latex_appendices, the latex to PDF generator will make these rst's into appendices labeled A B C D etc

Due to the mechanism used, the ordered list in the conf.py needs to be exactly the same as in the list of the index.rst for the html to be able to correctly reference them. These two lists need to be maintained in sync.

Variables as references

We want to use in all rsts the same way to reference an appendix in PDF as in the HTML generated.

We prefer a reference see ‘Appendix A’ without typing A because when this identifier changes we need to change them everywhere we referenced it. So we accomplish that with a third list of appendices as variables.

```
.. |app_imag_term| replace:: :ref:`APPENDIX A <app_imag_term>`
.. |app_obs| replace:: :ref:`APPENDIX B <app_obs>`
.. |app_iaf| replace:: :ref:`APPENDIX C1 <app_iaf>`
.. |app_imag_dir| replace:: :ref:`APPENDIX C2 <app_imag_dir>`
.. |app_iyf| replace:: :ref:`APPENDIX C3 <app_iyf>`
.. |app_imag_cd| replace:: :ref:`APPENDIX C4 <app_imag_cd>`
.. |app_imag_addr| replace:: :ref:`APPENDIX D <app_imag_addr>`
.. |app_imag_imfv_2| replace:: :ref:`APPENDIX E1 <app_imag_imfv_2>`
.. |app_sat_cod| replace:: :ref:`APPENDIX E2 <app_sat_cod>`
.. |app_imag_imfv_1| replace:: :ref:`APPENDIX E3 <app_imag_imfv_1>`
.. |app_imag_ibf| replace:: :ref:`APPENDIX E4 <app_imag_ibf>`
.. |app_iaga_2002| replace:: :ref:`APPENDIX E5 <app_iaga_2002>`
.. |app_cdf| replace:: :ref:`APPENDIX E6 <app_cdf>`
.. |app_1min_filter| replace:: :ref:`APPENDIX F1 <app_1min_filter>`
.. |app_1sec_filter| replace:: :ref:`APPENDIX F2 <app_1sec_filter>`
```

Text 10: variables used to reference appendices in appendices.rst

These variables are aligned with the two other references (for HTML in index.rst and PDF in conf.py) and hard code the link between the A B C and the relative reference app_id_of_appendix_rst. So when you change the order in the latex appendices in conf.py it is sufficient to keep in sync :

- the appendices list in toctree appendices
- the variables in variables used to reference appendices in appendices.rst

This three files make it possible to reference appendices from whatever rst we want with the logical name linked to it . Whenever you want to reference appendices you need to import the variable file as described in the section on variables.

```
.. _inter_hist_status:
History And Status Of INTERMAGNET
=====

.. include:: ../../appendices/appendices.rst
```

At present the observatories shown in |app_obs| are transmitting daily over the Internet

Text 11: Example of import

This will result in PDF and HTML:

“At present the observatories shown in APPENDIX B are transmitting daily over the Internet”