AnyBlok / Attachment Documentation

Release 1.0.2

jssuzanne

Feb 24, 2018

Contents:

CHAPTER 1

Front Matter

Information about the AnyBlok / Attachment project.

1.1 Project Homepage

AnyBlok is hosted on github - the main project page is at https://github.com/AnyBlok/anyblok_attachment. Source code is tracked here using GIT.

Releases and project status are available on Pypi at http://pypi.python.org/pypi/anyblok_attachment.

The most recent published version of this documentation should be at http://doc.anyblok-attachment.anyblok.org.

1.2 Project Status

AnyBlok with attachment is currently in beta status and is expected to be fairly stable. Users should take care to report bugs and missing features on an as-needed basis. It should be expected that the development version may be required for proper implementation of recently repaired issues in between releases;

1.3 Installation

Install released versions of AnyBlok from the Python package index with pip or a similar tool:

```
pip install anyblok_attachment
```

Installation via source distribution is via the `setup.py` script:

```
python setup.py install
```

Installation will add the `anyblok` commands to the environment.

1.4 Unit Test

Run the test with nose:

```
pip install nose
nosetests anyblok_attachment/tests
```

1.5 Dependencies

AnyBlok / Attachment works with **Python 3.3** and later. The install process will ensure that AnyBlok is installed, in addition to other dependencies. The latest version of them is strongly recommended.

1.6 Author

Jean-Sébastien Suzanne

1.7 Contributors

Anybox team:

- Jean-Sébastien Suzanne

Sensee team:

- Franck Bret

1.8 Bugs

Bugs and feature enhancements to AnyBlok should be reported on the Issue tracker.

Contents

CHAPTER 2

Bloks

2.1 Blok attachment

class anyblok_attachment.bloks.attachment.**AttachmentBlok**(*registry*)

Bases: anyblok.blok.Blok

Add attachment in AnyBlok

author = 'Suzanne Jean-Sébastien'

conditional_by = []

conflicting_by = []

classmethod import_declaration_module()

name = 'attachment'

optional_by = []

classmethod reload_declaration_module(*reload*)

required = ['anyblok-core']

required_by = ['report']

version = '1.0.0'

2.1.1 Memento

The document historize the version:

```
doc = registry.Attachment.Document.insert(...)
assert doc.type == 'latest'
assert doc.previous_version is None

doc.data = {'other': 'data'}
```

```
assert doc.previous_version is not None
assert doc.previous_version.type == 'historized'
```

You can get:

- All document:

  ```
  registry.Attachment.Document.query().all()
  ```

- Only the latest:

  ```
  registry.Attachment.Document.Latest.query().all()
  ```

- Only the historized:

  ```
  registry.Attachment.Document.History.query().all()
  ```

Note: If one entry changed and a file is available then a historied version will be autmaticly created.

The historized version get the latest version of the document and a new version is added to the document

2.1.2 API doc

Attachment

class anyblok_attachment.bloks.attachment.attachment.**Attachment**

Bases: object

Name space for the attachment models

Declaration type Model

Registry name Model.Attachment

Tablename attachment

Inherit model or mixin

Documments

class anyblok_attachment.bloks.attachment.document.**Document**

Bases: object

Declaration type Model

Registry name Model.Attachment.Document

Tablename attachment_document

Inherit model or mixin

field name	Description
version	• default - <class **‘anyblok.column.NoDefaultValue’>** • Label - None • size - 64 • Field type - <class ‘anyblok.column.String’> • DB column name - None • nullable - False • Context: • primary_key - True • is crypted - False • foreign_key - None
data	• default - <class ‘dict’> • Label - None • Field type - <class ‘anyblok.column.Json’> • DB column name - None • Context: • is crypted - False • foreign_key - None
filename	• default - <class **‘anyblok.column.NoDefaultValue’>** • Label - None • size - 256 • Field type - <class ‘anyblok.column.String’> • DB column name - None • Context: • is crypted - False • foreign_key - None
type	• default - <class **‘anyblok.column.NoDefaultValue’>** • Label - None • size - 64 • selections: • history - History • latest - Latest • Field type - <class **‘anyblok.column.Selection’>** • DB column name - None • nullable - False • Context: • is crypted - False • foreign_key - None
previous_doc_version	• default - <class **‘anyblok.column.NoDefaultValue’>**

class anyblok_attachment.bloks.attachment.document.**Latest**

Bases: `anyblok.model.Document`, `anyblok.mixin.ForbidDelete`

Declaration type Model

Registry name Model.Attachment.Document.Latest

Tablename attachment_document

Inherit model or mixin

- <class ‘anyblok.model.Document’>
- <class ‘anyblok.mixin.ForbidDelete’>

class anyblok_attachment.bloks.attachment.document.**History**

Bases: `anyblok.model.Document`, `anyblok.mixin.ReadOnly`

Declaration type Model

Registry name Model.Attachment.Document.History

Tablename attachment_document

Inherit model or mixin

- <class ‘anyblok.model.Document’>
- <class ‘anyblok.mixin.ReadOnly’>

Exceptions

exception anyblok_attachment.bloks.attachment.exceptions.**NoFileException**

Document has not got file entry

exception anyblok_attachment.bloks.attachment.exceptions.**ProtectedFieldException**

Try to update protected field

2.2 Blok wkhtml2pdf

class anyblok_attachment.bloks.wkhtml2pdf.**WkHtml2PdfBlok**(*registry*)

Bases: `anyblok.blok.Blok`

Add attachment in AnyBlok

author = 'Suzanne Jean-Sébastien'

conditional_by = []

conflicting_by = []

classmethod import_declaration_module()

name = 'wkhtml2pdf'

optional_by = []

classmethod reload_declaration_module(*reload*)

required = ['anyblok-core']

required_by = []

version = '1.0.0'

2.2.1 Memento

Define tools to convert HTML to PDF with wkhtmltopdf.

To use WkHtml2Pdf in your template is easy:

```
@register(Model.Attachment.Template)
class MyTemplate(Mixin.WkHtml2Pdf):
    ... # template configuration

    def render(self, data):
        html_content = ...
        return self.wkhtml2pdf(html_content)
```

You may define one or more configuration:

```
page_A4 = registry.Attachment.WkHtml2Pdf.Page.insert(
    label="A4", size="A4")
page_postal_label = registry.Attachment.WkHtml2Pdf.Page.insert(
    label="Postal carrier 1", height=80, width=120)
wkhtml2pdf = registry.Attachment.WkHtml2Pdf.insert(
    label="A4", page=page_A4, margin_top=20)
template = registry.Attachment.Template.insert(
    ...,
    wkhtml2pdf_configuration=wkhtml2pdf
)
doc = registry.Attachment.Document.insert(
    template=template, data=...)
doc.get_file()
```

2.2.2 API doc

Models

class anyblok_attachment.bloks.wkhtml2pdf.model.**WkHtml2Pdf**

Bases: object

Declaration type Model

Registry name Model.Attachment.WkHtml2Pdf

Tablename attachment_wkhtml2pdf

Inherit model or mixin

field name	Description
id	• `default` - <class **‘anyblok.column.NoDefaultValue’>** • `Label` - None • `autoincrement` - True • `Field type` - <class **‘anyblok.column.Integer’>** • `DB column name` - None • `nullable` - False • `Context:` • `primary_key` - True • `is crypted` - False • `foreign_key` - None
updated_at	• `Label` - None • `is auto updated` - True • `Field type` - <class **‘anyblok.column.DateTime’>** • `nullable` - False • `Context:`
javascript_delay	• `default` - 200 • `Label` - None • `Field type` - <class **‘anyblok.column.Integer’>** • `DB column name` - None • `nullable` - False • `Context:` • `is crypted` - False • `foreign_key` - None
local_file_access	• `default` - True • `Label` - None • `Field type` - <class **‘anyblok.column.Boolean’>** • `DB column name` - None • `Context:` • `is crypted` - False • `foreign_key` - None
grayscale	• `default` - False • `Label` - None • `Field type` - <class **‘anyblok.column.Boolean’>** • `DB column name` - None • `Context:` • `is crypted` - False • `foreign_key` - None
margin_right	• `default` - 10 • `Label` - None

cast_html2pdf (*prefix*, *html_content*)

Cast html document to a pdf document

Parameters

- **prefix** – prefix use for the tempory document
- **html_content** – html file (bytes)

Return type bytes

Exception WkHtml2PdfException

class anyblok_attachment.bloks.wkhtml2pdf.model.**Page**

Bases: `object`

Define the Page size

Declaration type Model

Registry name Model.Attachment.WkHtml2Pdf.Page

Tablename attachment_wkhtml2pdf_page

Inherit model or mixin

field name	Description
width	• default - <class ‘anyblok.column.NoDefaultValue’> • Label - None • Field type - <class ‘anyblok.column.Integer’> • DB column name - None • Context: • is crypted - False • foreign_key - None
height	• default - <class ‘anyblok.column.NoDefaultValue’> • Label - None • Field type - <class ‘anyblok.column.Integer’> • DB column name - None • Context: • is crypted - False • foreign_key - None
label	• default - <class ‘anyblok.column.NoDefaultValue’> • Label - None • size - 64 • Field type - <class ‘anyblok.column.String’> • DB column name - None • nullable - False • Context: • primary_key - True • is crypted - False • foreign_key - None
size	• default - <class ‘anyblok.column.NoDefaultValue’> • Label - None • size - 64 • Field type - <class ‘anyblok.column.String’> • DB column name - None • Context: • is crypted - False • foreign_key - None
updated_at	• Label - None • is auto updated - True • Field type - <class ‘anyblok.column.DateTime’> • nullable - False • Context:
created_at	• Label - None

Mixins

class anyblok_attachment.bloks.wkhtml2pdf.mixin.**WkHtml2Pdf**

Bases: `object`

Declaration type Mixin

Inherit model or mixin

field name	Description
wkhtml2pdf_configuration	• `Label` - None • `info`: • `nullable` - False • `unique` - False • `remote_model` - **Model.Attachment.WkHtml2Pdf** • `primary_key` - False • `index` - False • `Field type` - <class **'anyblok.relationship.Many2One'>** • `index` - False • `Context`: • `primary_key` - False • `_remote_columns` - None • `model` - Model.Attachment.WkHtml2Pdf • `_column_names` - None • `unique` - False

check_if_file_must_be_generated(*document*)

Overwrite to check also if the configuration have changed

Exceptions

exception anyblok_attachment.bloks.wkhtml2pdf.exceptions.**WkHtml2PdfException**

Only id the page configuration is wrong

exception anyblok_attachment.bloks.wkhtml2pdf.exceptions.**PageValidityException**

Only id the page configuration is wrong

2.3 Blok report

class anyblok_attachment.bloks.report.**ReportBlok**(*registry*)

Bases: `anyblok.blok.Blok`

Add attachment in AnyBlok

author = 'Suzanne Jean-Sébastien'

conditional_by = []

conflicting_by = []

classmethod import_declaration_module()

name = 'report'

optional_by = []

classmethod reload_declaration_module(*reload*)

required = ['attachment']

required_by = ['report-format']

version = '1.0.0'

2.3.1 Memento

The main goal is to create document in function of template

Add a new template:

```
template = registry.Attachment.Template.insert(...)
```

Add a Document without file but with a Template:

```
document = registry.Attachment.Document.insert(template=template, ...)
document.generate_file()
```

> **Warning:** the `get_file` method may re-create the document with historization if the data have been changed

`Model.Attachment.Parser`

It is an unSQL model, the goal is to transform the data from `Model.Attachment.Document` to Json format to create the file:

```
@register(Model.Attachment.Parser)
class MyParser(Model.Attachment.Parser):

    @classmethod
    def serialize(cls, data):
        # return serialize data

    @classmethod
    def check_if_file_must_be_generated(cls, template, document):
        # return a Boolean to know if the file must be regenerate
```

`Model.Attachment.Template`

The template describe how to generate the document file in function of of the data in the document.

The `Model.Attachment.Template` can be used directly an Template type class must be add with polymorphism:

```
@register(Model.Attachment)
class Template:

    def get_template_type(self):
        res = super(MyTemplateType, self).get_template_type()
        res.update({'MyTemplateType': 'My type label'})
        return res


@register(Model.Attachment.Template)
class MyTemplateType(Model.Attachment.Template):
    TYPE = 'MyTypeOfTemplate'

    uuid = UUID(primary_key=True, nullable=False, binary=False,
                foreign_key=Model.Attachment.Template.use('uuid').options(ondelete=
↪'cascade'))
    # other field need for the template

    def render(self):
        # generate a file
        return file
```

or on the same table:

```
@register(Model.Attachment)
class Template:

    def get_template_type(self):
        res = super(MyTemplateType, self).get_template_type()
        res.update({'MyTemplateType': 'My type label'})
        return res


@register(Model.Attachment.Template,
          tablename=Model.Attachment.Template)
class MyTemplateType(Model.Attachment.Template):
    TYPE = 'MyTypeOfTemplate'

    def render(self):
        # generate a file
        return file
```

2.3.2 API doc

Common

`anyblok_attachment.bloks.report.common.`**`format_path`**(*path*)

Return the absolute path in function of the blok

Parameters **`path`** – the path to format

Returns the absolute path

Exception PathException

Parser

`class` `anyblok_attachment.bloks.report.parser.`**`Parser`**(*model*)

Bases: `object`

Base parser to serialize the data for templating

Declaration type Model

Registry name Model.Attachment.Parser

Tablename attachment_parser

Inherit model or mixin

classmethod check_if_file_must_be_generated(*template*, *document*)

Return a boolean to know if the file must be generate or not

Parameters

- **template** – template intance
- **document** – latest document instance

Returns bool

classmethod serialize(*model*, *data*)

Serialize the data to be understanding by template

Parameters

- **model** – an anyblok model need to serialize
- **data** – json dict

Returns json dict

Template

class anyblok_attachment.bloks.report.template.**Template**

Bases: `object`

Base Model template to define the main configuration between templating system

Declaration type Model

Registry name Model.Attachment.Template

Tablename attachment_template

Inherit model or mixin

field name	Description
template_from	• default - path • Label - None • size - 64 • selections - get_template_from • Field type - <class ‘anyblok.column.Selection’> • DB column name - None • nullable - False • Context: • is crypted - False • foreign_key - None
filename	• default - {doc.uuid}-{doc.version}-{date} • Label - None • size - 64 • Field type - <class ‘anyblok.column.String’> • DB column name - None • nullable - False • Context: • is crypted - False • foreign_key - None
updated_at	• Label - None • is auto updated - True • Field type - <class ‘anyblok.column.DateTime’> • nullable - False • Context:
parser_model	• default - Model.Attachment.Parser • Label - None • size - 64 • Field type - <class ‘anyblok.column.String’> • DB column name - None • Context: • is crypted - False • foreign_key - None
template_path	• default - <class ‘anyblok.column.NoDefaultValue’> • Label - None • size - 64 • Field type - <class ‘anyblok.column.String’> • DB column name - None • Context: • is crypted - False • foreign_key - None
type	• default - <class ‘any-

check_if_file_must_be_generated(*document*)
Return True if the file must generate

Parameters **document** – attachment document instance

Return type bool

create_file_for(*document*)
Create a file and add it in the attachment document

Parameters **document** – attachment document instance

get_parser()
return the template file to compute

classmethod get_parser_from()
Give the location of the parser

get_template()
return the template file to compute

classmethod get_template_from()
Give the location of the template

classmethod get_template_type()
Give the tempate type

render(*data*)
Return the file create by the templating engine

Parameters **data** – the serialized data

update_document(*document*, *file_*, *data*)
Update the document to fill file and content

Parameters

- **document** – attachment document instance
- **file** – bytes file
- **data** – serialized data

Document

class anyblok_attachment.bloks.report.document.**Document**
Bases: object

Declaration type Model

Registry name Model.Attachment.Document

Tablename attachment_document

Inherit model or mixin

field name	Description
template	• Label - None • info: • primary_key - False • unique - False • remote_model - Model.Attachment.Template • index - False • Field type - <class 'anyblok.relationship.Many2One'> • index - False • Context: • primary_key - False • _remote_columns - None • model - Model.Attachment.Template • _column_names - None • unique - False

class anyblok_attachment.bloks.report.document.**Latest**

Bases: object

Declaration type Model

Registry name Model.Attachment.Document.Latest

Tablename attachment_document

Inherit model or mixin

Exceptions

exception anyblok_attachment.bloks.report.exceptions.**TemplateException**

Simple exception if miss Template data

exception anyblok_attachment.bloks.report.exceptions.**PathException**

Simple exception if it is not a valid path for Template

exception anyblok_attachment.bloks.report.exceptions.**RenderException**

Simple exception for all error in the render method

2.4 Blok report-format

class anyblok_attachment.bloks.format.**ReportBlok**(*registry*)

Bases: anyblok.blok.Blok

Add attachment in AnyBlok

author = 'Suzanne Jean-Sébastien'

conditional_by = []

conflicting_by = []

classmethod import_declaration_module()

name = 'report-format'

optional_by = []

classmethod reload_declaration_module(*reload*)

required = ['report']

required_by = []

version = '1.0.0'

2.4.1 Memento

Simple python templating formatage

template file exemple:

```
<!doctype html>
<html>
    <head>
        <title>{title}</title>
    </head>
    <body>
        {description}
    </body>
</html>
```

add the document and create file:

```
template = registry.Attachment.Template.Format.insert(
    template_path='path/of/the/python.template',
    contenttype='text/html',
    filename='mypage.html')
document = registry.Attachment.Document.insert(
    template=template,
    data={'title': 'My page', 'description': 'Hello world !!'}
)
document.get_file()
```

>>>:

```
{
    'contenttype': 'text/html',
    'file': b'<!doctype html>\n<html>\n    <head>\n        <title>My page</title>'
            b'\n    </head>\n    <body>\n        Hello world !!\n    </body>\n<'
            b'/html>\n',
    'file_added_at': datetime.datetime(2018, 1, 4, 9, 22, 56, 922346, tzinfo=
↪<DstTzInfo 'CET' CET+1:00:00 STD>),
    'filename': 'mypage.html',
    'filesize': 131,
    'hash': b'\xf0\xe2?\xebk\xb4\x15\x0f\xb2\x9cT\x08\xee#\x02\xe2\xbe\xa16\x8d'
            b'\xc8\xcda\x91;\xba2\x9c\x9dUF\x10'}

}
```

2.4.2 API doc

Template

class anyblok_attachment.bloks.format.template.**Template**

Bases: `object`

Declaration type Model

Registry name Model.Attachment.Template

Tablename attachment_template

Inherit model or mixin

Format

class anyblok_attachment.bloks.format.template.**Format**

Bases: `anyblok.model.Template`

Simple python format templating

Declaration type Model

Registry name Model.Attachment.Template.Format

Tablename attachment_template_format

Inherit model or mixin

- <class 'anyblok.model.Template'>

field name	Description
uuid	• default - <class 'anyblok.column.NoDefaultValue'> • Label - None • Field type - <class 'anyblok.column.UUID'> • DB column name - None • nullable - False • Context: • primary_key - True • is crypted - False • foreign_key - Model.Attachment.Template => uuid
contenttype	• default - <class 'anyblok.column.NoDefaultValue'> • Label - None • size - 64 • Field type - <class 'anyblok.column.String'> • DB column name - None • nullable - False • Context: • is crypted - False • foreign_key - None

Contents

CHAPTER 3

CHANGELOG

3.1 1.0.2 (2018-02-24)

- [REF] Anyblok 0.17.0 changed setter to add application and application groups, So I had to adapt the existing to use new setter

3.2 1.0.1 (2018-01-11)

- [FIX] `Mixin.WkHtml2Pdf` check also if the configuration changed for **Template.check_if_file_must_be_generated**

3.3 1.0.0 (2018-01-10)

- [ADD] **attachment** blok: stock versionned file
- [ADD] **report** blok: create versionned file from template
- [ADD] **report-format** blok: template type
- [ADD] **wkhtml2pdf** blok: convert html to pdf in the template

CHAPTER 4

Mozilla Public License Version 2.0

4.1 1. Definitions

4.1.1 1.1. “Contributor”

Means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

4.1.2 1.2. “Contributor Version”

Means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution.

4.1.3 1.3. “Contribution”

Means Covered Software of a particular Contributor.

4.1.4 1.4. “Covered Software”

Means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

4.1.5 1.5. “Incompatible With Secondary Licenses”

Means:

- **That the initial Contributor has attached the notice described in Exhibit B** to the Covered Software; or
- **That the Covered Software was made available under the terms of version 1.1** or earlier of the License, but not also under the terms of a Secondary License.

4.1.6 1.6. “Executable Form”

Means any form of the work other than Source Code Form.

4.1.7 1.7. “Larger Work”

Means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

4.1.8 1.8. “License”

Means this document.

4.1.9 1.9. “Licensable”

Means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

4.1.10 1.10. “Modifications”

Means any of the following:

- **Any file in Source Code Form that results from an addition to, deletion from,** or modification of the contents of Covered Software; or
- Any new file in Source Code Form that contains any Covered Software.

4.1.11 1.11. “Patent Claims” of a Contributor

Means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

4.1.12 1.12. “Secondary License”

Means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

4.1.13 1.13. “Source Code Form”

Means the form of the work preferred for making modifications.

4.1.14 1.14. “You” (or “Your”)

Means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

4.2 2. License Grants and Conditions

4.2.1 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- **Under intellectual property rights (other than patent or trademark)** Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- **Under Patent Claims of such Contributor to make, use, sell, offer for sale,** have made, import, and otherwise transfer either its Contributions or its Contributor Version.

4.2.2 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

4.2.3 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- For any code that a Contributor has removed from Covered Software; or
- **For infringements caused by: (i) Your and any other third party's** modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- **Under Patent Claims infringed by Covered Software in the absence of its** Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

4.2.4 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

4.2.5 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

4.2.6 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

4.2.7 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

4.3 3. Responsibilities

4.3.1 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

4.3.2 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- **Such Covered Software must also be made available in Source Code Form, as** described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- **You may distribute such Executable Form under the terms of this License, or** sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

4.3.3 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

4.3.4 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

4.3.5 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4.4 4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

4.5 5. Termination

4.5.1 5.1.

The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

4.5.2 5.2.

If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

4.5.3 5.3.

In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

4.6 6. Disclaimer of Warranty

Warning: Covered Software is provided under this License on an "as is" basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

4.7 7. Limitation of Liability

Warning: Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

4.8 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

4.9 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

4.10 10. Versions of the License

4.10.1 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

4.10.2 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

4.10.3 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

4.10.4 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

4.11 Exhibit A - Source Code Form License Notice

```
This Source Code Form is subject to the terms of the Mozilla Public
License, v. 2.0. If a copy of the MPL was not distributed with this file,
You can obtain one at http://mozilla.org/MPL/2.0/.
```

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

Note: You may add additional accurate notices of copyright ownership.

4.12 Exhibit B - "Incompatible With Secondary Licenses" Notice

```
This Source Code Form is "Incompatible With Secondary Licenses", as defined
by the Mozilla Public License, v. 2.0.
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index