
python-opc Documentation

Release 0.1.0

Steve Canny

September 10, 2015

1	Welcome	1
2	Documentation	3
3	OpcPackage objects	5
4	Part objects	7
5	_Relationship objects	9
6	Concepts	11
6.1	ISO/IEC 29500 Specification	11
6.2	Package contents	11
6.3	Pack URIs	11
6.4	Parts	11
6.5	Relationships	11
6.6	Content types	11
7	Contents	13
7.1	Content type constant names	13
7.2	Relationship type constant names	16
7.3	Design Narratives	18

Welcome

`python-opc` is a Python library for manipulating Open Packaging Convention (OPC) packages. An OPC package is the file format used by Microsoft Office 2007 and later for Word, Excel, and PowerPoint.

STATUS: as of Jul 28 2013 `python-opc` and this documentation for it are both work in progress.

Documentation

OpcPackage objects

Part objects

The `Part` class is the default type for package parts and also serves as the base class for custom part classes.

`_Relationship` objects

The `_Relationship` class ...

6.1 ISO/IEC 29500 Specification

6.2 Package contents

Content types stream, package relationships, parts.

6.3 Pack URIs

... A partname is a special case of pack URI ...

6.4 Parts

6.5 Relationships

... target mode ... relationship type ... rId ... targets

6.6 Content types

Contents

7.1 Content type constant names

The following names are defined in the `opc.constants` module to allow content types to be referenced using an identifier rather than a literal value.

The following import statement makes these available in a module:

```
from opc.constants import CONTENT_TYPE as CT
```

A content type may then be referenced as a member of `CT` using dotted notation, for example:

```
part.content_type = CT.PML_SLIDE_LAYOUT
```

The content type names are determined by transforming the trailing text of the content type string to upper snake case, replacing illegal Python identifier characters (dash and period) with an underscore, and prefixing one of these seven namespace abbreviations:

- **DML** – DrawingML
- **OFC** – Microsoft Office document
- **OPC** – Open Packaging Convention
- **PML** – PresentationML
- **SML** – SpreadsheetML
- **WML** – WordprocessingML
- no prefix – standard MIME types, such as those used for image formats like JPEG

BMP image/bmp

DML_CHART application/vnd.openxmlformats-officedocument.drawingml.chart+xml

DML_CHARTSHAPES application/vnd.openxmlformats-officedocument.drawingml.chartshapes+xml

DML_DIAGRAM_COLORS application/vnd.openxmlformats-officedocument.drawingml.diagramColors+xml

DML_DIAGRAM_DATA application/vnd.openxmlformats-officedocument.drawingml.diagramData+xml

DML_DIAGRAM_LAYOUT application/vnd.openxmlformats-officedocument.drawingml.diagramLayout+xml

DML_DIAGRAM_STYLE application/vnd.openxmlformats-officedocument.drawingml.diagramStyle+xml

GIF image/gif

JPEG image/jpeg

MS_PHOTO image/vnd.ms-photo

OFC_CUSTOM_PROPERTIES application/vnd.openxmlformats-officedocument.custom-properties+xml

OFC_CUSTOM_XML_PROPERTIES application/vnd.openxmlformats-officedocument.customXmlProperties+xml

OFC_DRAWING application/vnd.openxmlformats-officedocument.drawing+xml

OFC_EXTENDED_PROPERTIES application/vnd.openxmlformats-officedocument.extended-properties+xml

OFC_OLE_OBJECT application/vnd.openxmlformats-officedocument.oleObject

OFC_PACKAGE application/vnd.openxmlformats-officedocument.package

OFC_THEME application/vnd.openxmlformats-officedocument.theme+xml

OFC_THEME_OVERRIDE application/vnd.openxmlformats-officedocument.themeOverride+xml

OFC_VML_DRAWING application/vnd.openxmlformats-officedocument.vmlDrawing

OPC_CORE_PROPERTIES application/vnd.openxmlformats-package.core-properties+xml

OPC_DIGITAL_SIGNATURE_CERTIFICATE application/vnd.openxmlformats-package.digital-signature-certificate

OPC_DIGITAL_SIGNATURE_ORIGIN application/vnd.openxmlformats-package.digital-signature-origin

OPC_DIGITAL_SIGNATURE_XMLSIGNATURE application/vnd.openxmlformats-package.digital-signature-xmlsignature+xml

OPC_RELATIONSHIPS application/vnd.openxmlformats-package.relationships+xml

PML_COMMENTS application/vnd.openxmlformats-officedocument.presentationml.comments+xml

PML_COMMENT_AUTHORS application/vnd.openxmlformats-officedocument.presentationml.commentAuthors+xml

PML_HANDOUT_MASTER application/vnd.openxmlformats-officedocument.presentationml.handoutMaster+xml

PML_NOTES_MASTER application/vnd.openxmlformats-officedocument.presentationml.notesMaster+xml

PML_NOTES_SLIDE application/vnd.openxmlformats-officedocument.presentationml.notesSlide+xml

PML_PRESENTATION_MAIN application/vnd.openxmlformats-officedocument.presentationml.presentation.main+xml

PML_PRES_PROPS application/vnd.openxmlformats-officedocument.presentationml.presProps+xml

PML_PRINTER_SETTINGS application/vnd.openxmlformats-officedocument.presentationml.printerSettings

PML_SLIDE application/vnd.openxmlformats-officedocument.presentationml.slide+xml

PML_SLIDESHOW_MAIN application/vnd.openxmlformats-officedocument.presentationml.slideshow.main+xml

PML_SLIDE_LAYOUT application/vnd.openxmlformats-officedocument.presentationml.slideLayout+xml

PML_SLIDE_MASTER application/vnd.openxmlformats-officedocument.presentationml.slideMaster+xml

PML_SLIDE_UPDATE_INFO application/vnd.openxmlformats-officedocument.presentationml.slideUpdateInfo+xml

PML_TABLE_STYLES application/vnd.openxmlformats-officedocument.presentationml.tableStyles+xml

PML_TAGS application/vnd.openxmlformats-officedocument.presentationml.tags+xml

PML_TEMPLATE_MAIN application/vnd.openxmlformats-officedocument.presentationml.template.main+xml

PML_VIEW_PROPS application/vnd.openxmlformats-officedocument.presentationml.viewProps+xml

PNG image/png

SML_CALC_CHAIN application/vnd.openxmlformats-officedocument.spreadsheetml.calcChain+xml

SML_CHARTSHEET application/vnd.openxmlformats-officedocument.spreadsheetml.chartsheet+xml

SML_COMMENTS application/vnd.openxmlformats-officedocument.spreadsheetml.comments+xml

SML_CONNECTIONS application/vnd.openxmlformats-officedocument.spreadsheetml.connections+xml

SML_CUSTOM_PROPERTY application/vnd.openxmlformats-officedocument.spreadsheetml.customProperty

SML_DIALOGSHEET application/vnd.openxmlformats-officedocument.spreadsheetml.dialogsheet+xml

SML_EXTERNAL_LINK application/vnd.openxmlformats-officedocument.spreadsheetml.externalLink+xml

SML_PIVOT_CACHE_DEFINITION application/vnd.openxmlformats-officedocument.spreadsheetml.pivotCacheDefinition+xml

SML_PIVOT_CACHE_RECORDS application/vnd.openxmlformats-officedocument.spreadsheetml.pivotCacheRecords+xml

SML_PIVOT_TABLE application/vnd.openxmlformats-officedocument.spreadsheetml.pivotTable+xml

SML_PRINTER_SETTINGS application/vnd.openxmlformats-officedocument.spreadsheetml.printerSettings

SML_QUERY_TABLE application/vnd.openxmlformats-officedocument.spreadsheetml.queryTable+xml

SML_REVISION_HEADERS application/vnd.openxmlformats-officedocument.spreadsheetml.revisionHeaders+xml

SML_REVISION_LOG application/vnd.openxmlformats-officedocument.spreadsheetml.revisionLog+xml

SML_SHARED_STRINGS application/vnd.openxmlformats-officedocument.spreadsheetml.sharedStrings+xml

SML_SHEET application/vnd.openxmlformats-officedocument.spreadsheetml.sheet

SML_SHEET_METADATA application/vnd.openxmlformats-officedocument.spreadsheetml.sheetMetadata+xml

SML_STYLES application/vnd.openxmlformats-officedocument.spreadsheetml.styles+xml

SML_TABLE application/vnd.openxmlformats-officedocument.spreadsheetml.table+xml

SML_TABLE_SINGLE_CELLS application/vnd.openxmlformats-officedocument.spreadsheetml.tableSingleCells+xml

SML_USER_NAMES application/vnd.openxmlformats-officedocument.spreadsheetml.userNames+xml

SML_VOLATILE_DEPENDENCIES application/vnd.openxmlformats-officedocument.spreadsheetml.volatileDependencies+xml

SML_WORKSHEET application/vnd.openxmlformats-officedocument.spreadsheetml.worksheet+xml

TIFF image/tiff

WML_COMMENTS application/vnd.openxmlformats-officedocument.wordprocessingml.comments+xml

WML_DOCUMENT_GLOSSARY application/vnd.openxmlformats-officedocument.wordprocessingml.document.glossary+xml

WML_DOCUMENT_MAIN application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml

WML_ENDNOTES application/vnd.openxmlformats-officedocument.wordprocessingml.endnotes+xml

WML_FONT_TABLE application/vnd.openxmlformats-officedocument.wordprocessingml.fontTable+xml

WML_FOOTER application/vnd.openxmlformats-officedocument.wordprocessingml.footer+xml

WML_FOOTNOTES application/vnd.openxmlformats-officedocument.wordprocessingml.footnotes+xml

WML_HEADER application/vnd.openxmlformats-officedocument.wordprocessingml.header+xml

WML_NUMBERING application/vnd.openxmlformats-officedocument.wordprocessingml.numbering+xml

WML_PRINTER_SETTINGS application/vnd.openxmlformats-officedocument.wordprocessingml.printerSettings

WML_SETTINGS application/vnd.openxmlformats-officedocument.wordprocessingml.settings+xml

WML_STYLES application/vnd.openxmlformats-officedocument.wordprocessingml.styles+xml

WML_WEB_SETTINGS application/vnd.openxmlformats-officedocument.wordprocessingml.webSettings+xml

XML application/xml

X_EMF image/x-emf

X_FONTDATA application/x-fontdata

X_FONT_TTF application/x-font-ttf

X_WMF image/x-wmf

7.2 Relationship type constant names

The following names are defined in the `opc.constants` module to allow relationship types to be referenced using an identifier rather than a literal value.

The following import statement makes these available in a module:

```
from opc.constants import RELATIONSHIP_TYPE as RT
```

A relationship type may then be referenced as a member of `RT` using dotted notation, for example:

```
rel.reltype = RT.SLIDE_LAYOUT
```

The relationship type names are determined by transforming the trailing text of the relationship type string to upper snake case and replacing illegal Python identifier characters (the occasional hyphen) with an underscore.

AUDIO <http://schemas.openxmlformats.org/officeDocument/2006/relationships/audio>

A_F_CHUNK <http://schemas.openxmlformats.org/officeDocument/2006/relationships/aFChunk>

CALC_CHAIN <http://schemas.openxmlformats.org/officeDocument/2006/relationships/calcChain>

CERTIFICATE <http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/certificate>

CHART <http://schemas.openxmlformats.org/officeDocument/2006/relationships/chart>

CHARTSHEET <http://schemas.openxmlformats.org/officeDocument/2006/relationships/chartsheet>

CHART_USER_SHAPES <http://schemas.openxmlformats.org/officeDocument/2006/relationships/chartUserShapes>

COMMENTS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/comments>

COMMENT_AUTHORS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/commentAuthors>

CONNECTIONS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/connections>

CONTROL <http://schemas.openxmlformats.org/officeDocument/2006/relationships/control>

CORE_PROPERTIES <http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties>

CUSTOM_PROPERTIES <http://schemas.openxmlformats.org/officeDocument/2006/relationships/custom-properties>

CUSTOM_PROPERTY <http://schemas.openxmlformats.org/officeDocument/2006/relationships/customProperty>

CUSTOM_XML <http://schemas.openxmlformats.org/officeDocument/2006/relationships/customXml>

CUSTOM_XML_PROPS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/customXmlProps>

DIAGRAM_COLORS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramColors>

DIAGRAM_DATA <http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramData>

DIAGRAM_LAYOUT <http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramLayout>

DIAGRAM_QUICK_STYLE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/diagramQuickStyle>

DIALOGSHEET <http://schemas.openxmlformats.org/officeDocument/2006/relationships/dialogsheet>

DRAWING <http://schemas.openxmlformats.org/officeDocument/2006/relationships/drawing>

ENDNOTES <http://schemas.openxmlformats.org/officeDocument/2006/relationships/endnotes>

EXTENDED_PROPERTIES <http://schemas.openxmlformats.org/officeDocument/2006/relationships/extended-properties>

EXTERNAL_LINK <http://schemas.openxmlformats.org/officeDocument/2006/relationships/externalLink>

FONT <http://schemas.openxmlformats.org/officeDocument/2006/relationships/font>

FONT_TABLE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/fontTable>

FOOTER <http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer>

FOOTNOTES <http://schemas.openxmlformats.org/officeDocument/2006/relationships/footnotes>

GLOSSARY_DOCUMENT <http://schemas.openxmlformats.org/officeDocument/2006/relationships/glossaryDocument>

HANDOUT_MASTER <http://schemas.openxmlformats.org/officeDocument/2006/relationships/handoutMaster>

HEADER <http://schemas.openxmlformats.org/officeDocument/2006/relationships/header>

HYPERLINK <http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink>

IMAGE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/image>

NOTES_MASTER <http://schemas.openxmlformats.org/officeDocument/2006/relationships/notesMaster>

NOTES_SLIDE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/notesSlide>

NUMBERING <http://schemas.openxmlformats.org/officeDocument/2006/relationships/numbering>

OFFICE_DOCUMENT <http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument>

OLE_OBJECT <http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject>

ORIGIN <http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/origin>

PACKAGE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/package>

PIVOT_CACHE_DEFINITION <http://schemas.openxmlformats.org/officeDocument/2006/relationships/pivotCacheDefinition>

PIVOT_CACHE_RECORDS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/spreadsheetml/pivotCacheRecords>

PIVOT_TABLE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/pivotTable>

PRES_PROPS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/presProps>

PRINTER_SETTINGS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings>

QUERY_TABLE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/queryTable>

REVISION_HEADERS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/revisionHeaders>

REVISION_LOG <http://schemas.openxmlformats.org/officeDocument/2006/relationships/revisionLog>

SETTINGS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/settings>

SHARED_STRINGS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/sharedStrings>

SHEET_METADATA <http://schemas.openxmlformats.org/officeDocument/2006/relationships/sheetMetadata>

SIGNATURE <http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/signature>

SLIDE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/slide>

SLIDE_LAYOUT <http://schemas.openxmlformats.org/officeDocument/2006/relationships/slideLayout>

SLIDE_MASTER <http://schemas.openxmlformats.org/officeDocument/2006/relationships/slideMaster>

SLIDE_UPDATE_INFO <http://schemas.openxmlformats.org/officeDocument/2006/relationships/slideUpdateInfo>
STYLES <http://schemas.openxmlformats.org/officeDocument/2006/relationships/styles>
TABLE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/table>
TABLE_SINGLE_CELLS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/tableSingleCells>
TABLE_STYLES <http://schemas.openxmlformats.org/officeDocument/2006/relationships/tableStyles>
TAGS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/tags>
THEME <http://schemas.openxmlformats.org/officeDocument/2006/relationships/theme>
THEME_OVERRIDE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/themeOverride>
THUMBNAIL <http://schemas.openxmlformats.org/package/2006/relationships/metadata/thumbnail>
USERNAMES <http://schemas.openxmlformats.org/officeDocument/2006/relationships/usernames>
VIDEO <http://schemas.openxmlformats.org/officeDocument/2006/relationships/video>
VIEW_PROPS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/viewProps>
VML_DRAWING <http://schemas.openxmlformats.org/officeDocument/2006/relationships/vmlDrawing>
VOLATILE_DEPENDENCIES <http://schemas.openxmlformats.org/officeDocument/2006/relationships/volatileDependencies>
WEB_SETTINGS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/webSettings>
WORKSHEET_SOURCE <http://schemas.openxmlformats.org/officeDocument/2006/relationships/worksheetSource>
XML_MAPS <http://schemas.openxmlformats.org/officeDocument/2006/relationships/xmlMaps>

7.3 Design Narratives

Narrative explorations into design issues, serving initially as an aid to reasoning and later as a memorandum of the considerations undertaken during the design process.

7.3.1 Semi-random bits

partname is a marshaling/serialization concern.

partname (pack URI) is the addressing scheme for accessing serialized parts within the package. It has no direct relevance to the unmarshaled graph except for use in re-marshaling unmanaged parts or to avoid renaming parts when the load *partname* will do just fine.

What determines part to be constructed? Relationship type or content type?

Working hypothesis: Content type should be used to determine the type of part to be constructed during unmarshaling.

Content type is more granular than relationship type. For example, an image part can be any of several content types, e.g. jpg, gif, or png. Another example is RT.OFFICE_DOCUMENT. This can apply to any of CT.PRESENTATION, CT.DOCUMENT, or CT.SPREADSHEET and their variants.

However, I can't think of any examples of where a particular content type may be the target of more than one possible relationship type. That seems like a logical possibility though.

There are examples of where a relationship type (customXml for example) are used to refer to more than one part type (Additional Characteristics, Bibliography, and Custom XML parts in this case). In such a case I expect the unmarshaling and part selection would need to be delegated to the source

part which presumably would contain enough information to resolve the ambiguity in its body XML. In that case, a `BasePart` could be constructed and let the source part create a specific subclass on `after_unmarshal()`.

When properties of a mutable type (e.g. list) are returned, what is returned should be a copy or perhaps an immutable variant (e.g. tuple) so that client-side changes don't need to be accounted for in testing. If the return value really needs to be mutable and a snapshot won't do, it's probably time to make it a custom collection so the types of mutation that are allowed can be specified and tested.

In `PackURI`, the `baseURI` property does not include any trailing slash. This behavior is consistent with the values returned from `posixpath.split()` and is then in a form suitable for use in `posixpath.join()`.

Design Narrative – Blob proxy

Certain use cases would be better served if loading large binary parts such as images could be postponed or avoided. For example, if the use case is to retrieve full text from a presentation for indexing purposes, the resources and time consumed to load images into memory is wasted. It seems feasible to develop some sort of blob proxy to postpone the loading of these binary parts until such time as they are actually required, passing a proxy of some type to be used instead. If it were cleverly done, the client code wouldn't have to know, i.e. the proxy would be transparent.

The main challenge I see is how to gain an entry point to close the zip archive after all loading has been completed. If it were reopened and closed each time a part was loaded that would be pretty expensive (an early version of `python-pptx` did exactly that for other reasons). Maybe that could be done when the presentation is garbage collected or something.

Another challenge is how to trigger the proxy to load itself. Maybe blob could be an object that has file semantics and the read method could lazy load.

Another idea was to be able to open the package in read-only mode. If the file doesn't need to be saved, the actual binary objects don't actually need to be accessed. Maybe this would be more like read-text-only mode or something. I don't know how we'd guarantee that no one was interested in the image binaries, even if they promised not to save.

I suppose there could be a "read binary parts" method somewhere that gets triggered the first time a binary part is accessed, as it would be during `save()`. That would address the zip close entry point challenge.

It does all sound a bit complicated for the sake of saving a few milliseconds, unless someone (like Google :) was dealing with really large scale.

Design Narrative – Custom Part Class mapping

```
pkg.register_part_classes(part_class_mapping)

part_class_mapping = {
    CT_SLIDE: _Slide,
    CT_PRESENTATION: _Presentation
    ...
}
```

Design Narrative – Model-side relationships

7.3.2 Might it make sense to maintain XML of .rels stream throughout life-cycle?

No. The primary rationale is that a partname is not a primary model-side entity; partnames are driven by the serialization concern, providing a method for addressing serialized parts. Partnames are not required to be up-to-date in the model until after the `before_marshall()` call to the part returns. Even if all part names were kept up-to-date, it

would be a leakage across concern boundaries to require a part to notify relationships of name changes; not to mention it would introduce additional complexity that has nothing to do with manipulation of the in-memory model.

always up-to-date principle

Model-side relationships are maintained as new parts are added or existing parts are deleted. Relationships for generic parts are maintained from load and delivered back for save without change.

I'm not completely sure that the always-up-to-date principle need necessarily apply in every case. As long as the relationships are up-to-date before returning from the `before_marshall()` call, I don't see a reason why that choice couldn't be at the designer's discretion. Because relationships don't have a compelling model-side runtime purpose, it might simplify the code to localize the pre-serialization concern to the `before_marshall()` method.

7.3.3 Members

rId

The relationship identifier. Must be a unique `xsd:ID` string. It is usually of the form `'rId%d' % {sequential_int}`, e.g. `'rId9'`, but this need not be the case. In situations where a relationship is created (e.g. for a new part) or can be rewritten, e.g. if `presentation->slide` relationships were rewritten on `before_marshall()`, this form is preferred. In all other cases the existing `rId` value should be preserved. When a relationship is what the spec terms as *explicit*, there is a reference to the relationship within the source part XML, the key of which is the `rId` value; changing the `rId` would break that mapping.

The **sequence** of relationships in the collection is not significant. The relationship collection should be regarded as a mapping on `rId`, not as a sequence with the index indicated by the numeric suffix of `rId`. While PowerPoint observes the convention of using sequential `rId` values for the slide relationships of a presentation, for example, this should not be used to determine slide sequence, nor is it a requirement for package production (saving a `.pptx` file).

reltype

A clear purpose for `reltype` is still a mystery to me.

target_mode

target_part

target_ref

- `genindex`
- `modindex`
- `search`