pdfTEX and Optional Content in PDF

Martin Schröder

2008-06-30

Optional Content poses a new challenge for pdfTEX. We show how Optional Content works and discuss how pdfTEX does and could support it.

1 How Optional Content works

With PDF 1.5, Adobe introduced Optional Content (typically called "layers"), which allows parts of a PDF document to be viewed or hidden by authors or consumers. It works as follows:

1.1 On a page or in an XObject

- In the content stream parts of the stream are enclosed in a pair of **BDC** and **EMC** operators [1, p. 778f], where two arguments to **BDC** identify by name the OCG (see below) this part belongs to.[1]
- In the **/Resources** for the content stream [1, p. 128f] is a key **/Properties** with a dictionary.
- This **/Properties** dictionary maps every name used with **BDC** in the stream to an Optional Content Group (OCG).[2]
- Optional Content Group dictionaries (**/OCG**s) [1, p. 334f] provide a **/Name** that is displayed in the GUI.[3]

1.2 In the document

- The document catalog [1, p. 112f] of a PDF document with Optional Content has an entry **/OCProperties** pointing to an optional content properties dictionary.
- The optional content properties dictionary [1, p. 345f] lists all OCGs of the document and sets their initial visibility. It also allows alternate visibility configurations.

1.3 Rendering

The visiblity of OCGs is set through the **/OCProperties**. When a content stream is rendered, the OCG is found through the **/Properties** and its visibility is checked.

2 Usage

2.1 Using Optional Content in the document

One may want to make parts of the document optional, indentify the parts by name, and set their visibility rules.

[1] This is really marked content, a concept already introduced in PDF 1.2, but then intended only for applications and plugins. PDF 1.5 adds the OCGs and the part in the document, i. e. the GUI.

[2] We ignore Optional Content Membership dictionaries [1, p. 335f] here.

[3] We ignore **/Intent**s and **/Usage**s [1, p. 335f] here.

2.2 Including PDFs with Optional Content

One may want to include PDFs with Optional Content and then reproduce their **/OCProperties** in the resulting document. When multiple PDFs with Optional Content are included, the **/Properties** must be changed, unifying identical OCGs—otherwise one ends up with multiple OCGs with identical names. Also Optional Content in the document and in included PDFs may be merged.

3 Solutions in pdfTEX

3.1 Using Optional Content in the document

For using Optional Content in a document one may want a command for declaring an OCG and a command for switching between OCGs (and to no OCGs).

Currently this has to be done via `\pdfliteral` and controlled by TEX macros.

3.2 Including PDFs with Optional Content

An automatic solution for including PDFs with OCGs would automatically (with some control by the document) migrate the OCGs of the PDFs to the including document and treat them like OCGs in the including documents. OCGs with the same name would automatically be merged (this might be optional).

Since an automatic solution is difficult to combine with documents that want to control everything or already use `\pdfliteral`, we instead provide four commands:[4]

`\pdfocgsnumber` ⟨*object number*⟩
: Gives the number of OCGs of an already included PDF

`\pdfocgname` ⟨*object number*⟩ ⟨*number*⟩
: Gives the name of the OCG ⟨*number*⟩ of the image ⟨*object number*⟩

`\pdfocgobjnumber` ⟨*object number*⟩ ⟨*number*⟩
: Gives the object number of the OCG ⟨*number*⟩ of the image ⟨*object number*⟩; this is needed for reproducing the **/OCProperties** in the included document and can be used with `\pdfocgmerge`.

`\pdfocgmerge` ⟨*object number*$_1$⟩ ⟨*number*$_1$⟩ ⟨*object number*$_2$⟩
: Unifies the OCG ⟨*number*$_1$⟩ of the image ⟨*object number*$_1$⟩ with the OCG ⟨*object number*$_2$⟩ so that `\pdfocgobjectnumber` ⟨*object number*$_1$⟩ ⟨*number*$_1$⟩ and ⟨*object number*$_2$⟩ are identical, i. e. it replaces the object number of OCG ⟨*number*$_1$⟩ in the **/Properties** of ⟨*object number*$_1$⟩ with ⟨*object number*$_2$⟩. Afterwards an `\pdfocgobjnumber` ⟨*object number*$_1$⟩ ⟨*number*$_1$⟩ returns ⟨*object number*$_2$⟩. The result of `\pdfocgname` ⟨*object number*$_1$⟩ ⟨*number*$_1$⟩ is not changed.

The first OCG to be used with `\pdfocgname`, `\pdfocgobjnumber` and `\pdfocgmerge` has the number one. The generation of a new **/OCProperties** has to be done with `\pdfliteral`. There is currently no way to inquire about the **/OCProperties** or **/OCMD**s of the included PDF.

References

[1] Adobe Systems Incorporated. *Portable Document Format Reference Manual.* Adobe Systems Incorporated, San Jose, California, USA, version 1.6 (november 2004) edition, 2005.

[4] To help with this, all **/Properties** and OCGs of included PDFs are automatically made indirect objects.