

# PDF5-ML plug-in Features for practical use




Toshihiko Makita  
[tmakita@antenna.co.jp](mailto:tmakita@antenna.co.jp)  
Wybens Titus  
[wybens@antennahouse.com](mailto:wybens@antennahouse.com)

# Agenda

- What is PDF5-*ML* plug-in?
- What are the PDF5-*ML* features?
- Is PDF5-*ML* truly usable?

# What is PDF5-*ML* plug-in?

- PDF5-ML has been in development since 2012
  - <https://github.com/AntennaHouse/pdf5-ml> 
- PDF5-ML has more enhancements than it's predecessor PDF5 plug-in (released 2008)
- PDF5-ML doesn't use `xsl:attribute-set` declaration. All of the styles and language specific variables are defined in external XML called “Style Definition File”

# What are the PDF5-*ML* features?

- Multiple languages in one PDF
- Conditional variable & style definition
- Free format cover pages

# Multiple languages support

- Most of the PDF plug-in adopts map/@xml:lang as document's language and assign fonts by its value.
- PDF5-*ML* allows different @xml:lang in **ANY** level of topic
- Automatically selects appropriate fonts and styles based on @xml:lang respectively

# Lets test it with “Hello” worldwide!

<p>We say "Hello" in English.</p>

<p>We say "<ph xml:lang="ja-JP">こんにちは</ph>" in Japanese.</p>

<p>We say "<ph xml:lang="zh-CN">你好</ph>" in Chinese.</p>

<p>We say "<ph xml:lang="ko-KR">안녕하세요</ph>" in Korean.</p>

<p>We say "<ph xml:lang="th-TH">สวัสดีครับ</ph>" in Thai.</p>

<p>We say "<ph xml:lang="hi-IN">नमस्ते</ph>" in Hindi.</p>

# Formatting Conditions

- Set “en-US” to bookmap/@xml:lang
- Use DITA-OT 2.1.1 and AH Formatter V6.2 for PDF generation
- Disable auto font fallback

# Result: PDF5-*ML*

## Chapter 1 Multiple language test

### 1.1 "Hello" worldwide!

We say "Hello" in English.

We say "こんにちは" in Japanese.

We say "你好" in Chinese.


We say "안녕하세요" in Korean.

We say "สวัสดีครับ" in Thai.

We say "नमस्ते" in Hindi.



# Result: PDF2



## "Hello" worldwide!

---

We say "Hello" in English.

We say "□□□□□" in Japanese.

We say "□□" in Chinese.

We say "□□□□□" in Korean.

We say "□□□□□□□□□□" in Thai.

We say "□□□□□□" in Hindi.

# The formatting result

- In PDF5-*ML* all glyphs are shown in PDF with **NO** problem. It has **NO** formatting errors!
- In PDF2 non-Latin characters are shown in PDF as “ToFu” mark. It has **MANY** font errors reported. This is because PDF2 doesn't assign fonts based on each @xml:lang attribute
- PDF5-*ML* is better if you are intending to create multiple-language publication

# Conditional variable & style definition

- You can use following predefined attributes
  - **@output-type**: The usage of PDF(Default: WEB)
  - **@doc-type**: Document type (Free definition)
  - **@paper-size**: Paper size of PDF (Default: Letter)
- By using above, you can define conditional style or variable in “Style Definition File”
- Conditional definition enables stylesheet templates free from complex coding.

# Note icon definition by @output-type

```
<!-- For WEB: Use SVG -->
```

```
<variable name="Note_Icon"
```

```
output-type="web">url(note.svg)</variable>
```

```
<!-- For Color Print: Use Full Color PDF -->
```

```
<variable name="Note_Icon"
```

```
output-type="print-color">url(note.pdf)</variable>
```

```
<!-- For Monochrome Print: Use Monochrome PDF -->
```

```
<variable name="Note_Icon"
```

```
output-type="print-mono">url(note-mono.pdf)</variable>
```

# Template code

- Function ahf:getVarValue automatically selects the variable value by its name
- @output-type value is given via xsl:param
- Template can be independent from output-type

```
<!-- Note template -->
<xsl:template match="*[contains(@class,' topic/note ')]">
  ...
  <fo:external-graphic>
    <xsl:attribute name="src"
                  select="ahf:getVarValue('Note_Icon')"/>
  </fo:external-graphic>
  ...
</xsl:template>
```

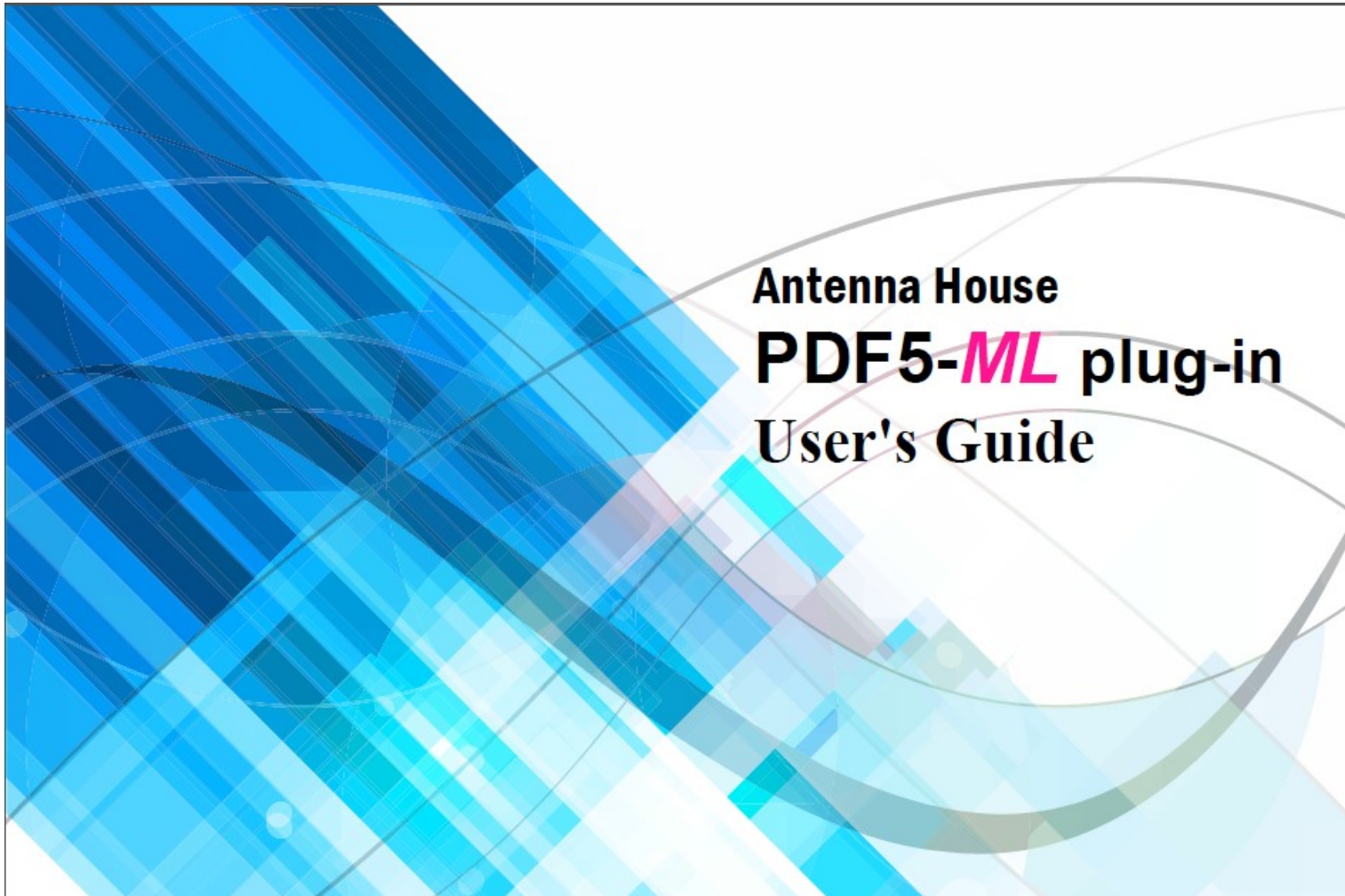
# Free format cover pages

- Originally cover pages did not fit with the DITA topic idea.
- Usually a plug-in developer used up lots of efforts to make cover page templates.
- PDF5-*ML* offers the practical solution by writing XSL-FO property in topic authoring.

# Example of cover page

```
<!--Set title to absolute position by fo:prop attribute-->
<conbodydiv fo:prop="position:absolute;top:40mm;right:10mm;
                width:40%;height:130mm;">
<!--Set title's FO property by fo:prop attribute-->
<section>
  <p fo:prop="font-family:Franklin Gothic Demi Cond;font-size:24pt;">
    Antenna House</p>
  <p fo:prop="font-family:Arial;font-weight:bold;
            font-size:32pt;">PDF5-fo:prop="color:deeppink;">ML</i>
  <ph fo:prop="font-size:28pt;">plug-in</ph></p>
  <p fo:prop="font-family:Times New Roman;font-weight:bold;
            font-size:28pt;">User's Guide</p>
</section>
</conbodydiv>
```

# The formatted result





# Is PDF5-*ML* truly usable?

*Yes, indeed!*

- PDF5/PDF5-*ML* has been developed for over 7 years
- Has far simpler structure and more features than PDF2
- Used in actual PDF plug-in development worldwide

# Questions?