

Fine Tune Your DITA PDF Outputs Using CSS

Julien Lacour, Syncro Soft

julien_lacour@oxygenxml.com

Oxygen Webinar, July 15th 2020

© 2020 Syncro Soft SRL. All rights reserved.



Introduction

What is Oxygen Publishing Engine?



Publishing Engine

=



DITA-OT

+



WebHelp

+



PDF Chemistry

Introduction

Why Use the Oxygen Publishing Engine?

- For the advanced WebHelp Responsive output
- For the highly customizable PDF using HTML5 & CSS
- For the consistency between both plugins

Introduction

Why Use the Oxygen Publishing Engine?

X22000 Specs

Table 1. X22000 Specifications

Name	Spec Value	Notes
Cores	4	Three cores make up the first stage. The side cores are connected at the base and at the top of the center core's liquid oxygen tank. The four cores generate 25,489 kN ¹ (2599 tf ²) of thrust at liftoff.
Engines	37	Shortly after liftoff 19 of the engines are throttled down. After the side cores separate, the 19 engines throttle back up to full thrust.
Mini Engines	55	Inside each core is a cluster of 55 mini engines. These same engines power the Z11000, enabling efficiencies that make the X22000 the most cost-effective light launch vehicle in the world. With a total of 37 first-stage engines, the X22000 has engine-out capability that no other launch vehicle can match. Under most payload scenarios, it can sustain up to 8 unplanned engine shutdown at any point in flight and still successfully complete its mission.
Thrust at sea level	38297 kN	
Thrust in vacuum	44383 kN	
Height	78.9 m	Measured from lowest to highest point
Width	14.18 m	Measured from widest point
Mass	3,263 kg	

X22000 Specs

Table 1. X22000 Specifications

Name	Spec Value	Notes
Cores	4	Three cores make up the first stage. The side cores are connected at the base and at the top of the center core's liquid oxygen tank. The four cores generate 25,489 kN ³ (2599 tf ⁴) of thrust at liftoff.
Engines	37	Shortly after liftoff 19 of the engines are throttled down. After the side cores separate, the 19 engines throttle back up to full thrust.
Mini Engines	55	Inside each core is a cluster of 55 mini engines. These same engines power the Z11000, enabling efficiencies that make the X22000 the most cost-effective light launch vehicle in the world. With a total of 37 first-stage engines, the X22000 has engine-out capability that no other launch vehicle can match. Under most payload scenarios, it can sustain up to 8 unplanned engine shutdown at any point in flight and still successfully complete its mission.

Introduction - The Basics

- Elements in DITA are marked with a class attribute

```
<codeph class="topic/ph pr-d/codeph">
```

- The PDF transformation produces an intermediate HTML file keeping these attributes

```
<code class="topic/ph pr-d/codeph">
```

- Best practice is to use this class name inside CSS selectors

```
*[class ~="pr-d/codeph"] {  
    color: blue;  
}
```

Introduction

Customizing PDF Output Using CSS

- Using a transformation parameter - 'args.css'
- Using a publishing template file (.opt)
 - To go further: [Customizing PDF Output Using CSS](#)

Introduction

Debugging the CSS

- Using the “.merged.html” file
 - Using a browser: Right-click → Inspect
 - Using Oxygen: Right-click → Inspect Styles
- To go further: [Debugging the CSS](#)

Tables Customization

Problem: I want to control the table caption display.

Solution: I will use two transformation parameters:

- `table.title.placement`
- `table.title.repeat`

Tables Customization

Problem: I have a table with multiple columns, it bleeds over the page.

1st Solution: I can use the *@orient=land* DITA attribute

2nd Solution: I can use the `overflow-wrap:break-word` CSS property

Tables Customization

Problem: I want the last row border to be displayed at the end of the page even if my table continues.

Solution: I will use the `-oxy-borders-conditionality` property:

```
*[class~="topic/table"] {  
    -oxy-borders-conditionality: retain;  
}
```

To go further: [Styling Tables](#)

Preserve Space Elements Customization

Problem: I have a document with a `<pre>` element and content is bleeding outside the element.

Solution: I will use the `overflow-wrap:break-word` CSS property

- Advantage: Content fits and no extra hyphens are displayed
- Disadvantage: The copy/paste may lead to some split lines

Mark and Flag the Content

Problem: I want to mark the tracked content with change-bars.

Solution: I will use the following transformation parameter:

- show.changes.and.comments.as.changebars

Additionally, I can customize the change-bars

```
oxy-range-start[is-changebar]:before(100),  
.oxy-range-start[is-changebar]:before(100) {  
  -oxy-changebar-color: orange;  
  -oxy-changebar-width: 0.5pt;  
}
```

Mark and Flag the Content

Problem: I want to hide the added/deleted tracked content from the annotations callout.

Solution: I will use the following transformation parameter:

- `show.changed.text.in.pdf.sticky.notes.content`

Advantage: Possibility to use the PDF reader comment search.

Mark and Flag the Content

Problem: I want to mark a part of my content with change-bars and flags.

Solution: I will use a ditaval file that contains the following elements:

```
<revprop action="flag">
```

```
<prop action="flag">
```

To go further: [Flagging Content](#)

Publish a Single Topic

- Default transformation: DITA PDF - based on HTML5 & CSS
- Specific Transtype: 'pdf-css-html5-single-topic'

Publish a Single Topic

Problem: Both images and `<xref>` elements are not displayed.

Solutions:

- I will use the `fix.external.refs.com.oxygenxml` transformation parameter to display images
- I will use the `args.enable.root.map.key.processing` transformation parameter to process references

Disadvantage: Increase processing time

Oxygen Styles Basket

- Available at <https://styles.oxygenxml.com/>
- Export CSS or Publishing Template files
- See Results button for online preview

Advantage: Good starting point for CSS beginners

To go further: Check the “How To” sections from [CSS Customization](#)

Continuous Integration (Jenkins)

- Download Oxygen Publishing Engine at https://www.oxygenxml.com/publishing_engine/download.html
- Unzip the Publishing Engine on a Jenkins server or in a project that contains DITA files
- Configure the Jenkins build with commands such as:
 - `dita -f pdf-css-html5 -i map.ditamap -o output.dir`

THANK YOU!

Any questions?

Julien Lacour

julien_lacour@oxygenxml.com