How to create a PDF document for each tab

This How-To describes the generation of a PDF document for each group of documents that is defined by a tab.

Table of contents

1. Overview

This How-To describes the generation of a PDF document for each group of documents that is defined by a tab.

1. Intended Audience

Users who need to generate one printable document aggregated from a group of documents.

2. Purpose

By default Forrest generates a pdf file for each separate document of your project. As well you can create a pdf of the whole site. But sometimes it may be necessary to generate a pdf file out of selected tab, i.e. only for certain parts of the site.

3. Prerequisites

- Understand how to create project-specific sitemaps by following the Using Forrest document.

4. Steps

The procedure outlined below will define a project `sitemap.xmap` and create a new `pdf-tab.xmap`.

4.1. Create your project's main sitemap.xmap

If you do not have already a sitemap then create a new empty one in your `src/documentation` directory (or wherever ${project.sitemap-dir} points to).

4.2. Create another sitemap: pdf-tab.xmap

Like before create an empty sitemap and name it pdf-tab.xmap.

4.3. Edit project sitemap.xmap to mount pdf-tab.xmap

Your sitemap should look something like this.

```
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:pipelines>
    <map:pipeline internal-only="false">
      <map:match pattern="**.xml">
        <!-- pdf-tab definitions -->
        <map:match pattern="pdf-tab.xml">
          <map:mount uri-prefix="" src="pdf-tab.xmap"
            check-reload="yes" />
        </map:match>
        <!-- end of pdf-tab definitions -->
      </map:match>
    </map:pipeline>
  </map:pipelines>
</map:sitemap>
```

4.4. Edit the file pdf-tab.xmap

The `<map:match pattern="*.xml">` element should look like the following:

```
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:pipelines>
    <map:pipeline internal-only="false">
     <map:match pattern="*.xml">
        <map:generate src="cocoon://abs-linkmap"/>
          <map:transform type="xpath">
                  <map:parameter name="include" value="//*[@wholesite='true']"/>
                <map:parameter name="exclude" value="//*[@wholesite='false']"/>
          </map:transform>
          <map:transform src="resources/stylesheets/site2book.xsl" />
          <map:transform src="resources/stylesheets/aggregates/book2cinclude.xsl">
             <map:parameter name="title"
                value="{conf:project-name}: Still My Foo Site"/>
             </map:transform>
          <map:transform type="cinclude"/>
          <map:transform
src="resources/stylesheets/aggregates/doc2doc-uniqueids.xsl"/>
          <map:transform
src="resources/stylesheets/aggregates/docs2document.xsl"/>
          <map:serialize type="xml"/>
    </map:match>
   </map:pipeline>
  </map:pipelines>
</map:sitemap>
```

4.5. Edit your site.xml

> **Note:**
>
> Do not use directories with "." in it. Replace them by "_" e.g 1.2/ will not work in the aggregation. e.g. 1_2/ just works fine.

Add the following entry to your site.xml in the `<about>` element

```
...
<whole_foosite href="pdf-tab.html" label="sub site" />
```

Your site.xml should look like this ...

```
...
<about label="About">
  <index label="Index" href="index.html" description="Welcome to MyProj"/>
  <changes label="Changes" href="changes.html"
    description="History of Changes" />
  <todo label="Todo" href="todo.html" description="Todo List" />
  <whole_foosite href="pdf-tab.html" label="pdf-tab" />
</about>
...
```

This allows you to link to it via a `<link href="site:whole_foosite">` reference.

Add to every element that should be included in the pdf-tab.pdf the attribute `wholesite="true"`

```
<sample-wiki label="Wiki page" href="wiki-sample.html"
  description="wiki-sample" wholesite="true"/>
```

> **Note:**
>
> This attribute will be inherited by all children of the element. Do not use it in the parent element that contains the `<whole_foosite href="pdf-tab.html" label="pdf-tab" />` as the child (will cause a `stack overflow` if you do)!!!

4.6. Explanation of the operation

Line 4 of our example `<map:parameter name="include" value="//*[@wholesite='true']"/>` looks at your site.xml and will match every element containing the `wholesite="true"` attribute. For example, to use the "samples" tab ...

```
...
<samples label="Samples" href="samples/" tab="samples" wholesite="true">
...
</samples>
...
```

It matches **all** of the elements that contain `wholesite="true"` (in our example `<samples>` and its "children") for the content of the pdf file to be generated.

```
<samples label="Samples" href="samples/" tab="samples" wholesite="true">
 <sample2 label="Static content" href="sample2.html"
   description="More Samples" wholesite='false'/>
 <sample-wiki label="Wiki page" href="wiki-sample.html"
   description="wiki-sample" />
 <sample-ihtml label="ihtml page" href="ihtml-sample.html"
   description="Test iHTML page" />
</samples>
```

This example shows that you can as well exclude site(s) from the aggregation by using the `wholesite="false"` attribute. This attribute will be as well inherited by all children of the element.

Line 8 defines the title of the pdf file by taking the content of the project-name variable in `skinconf.xml` and adding some funny text: `<map:parameter name="title" value="{conf:project-name}: Still My Foo Site"/>`

5. Feedback and further development of this How-To

Please provide feedback about this document via the mailing lists.

In the future, this ability will probably be incorporated into the main Forrest process.