

The Apache Forrest xdocs document-v1.3 DTD

NOTICE: The content of this document doesn't make any sense at all.

This is a demonstration document using all possible elements in the current Apache Forrest xdocs document-v1.3.dtd

Table of contents

1 Sample Content.....	2
1.1 Using sections.....	4
1.2 Sections, the sequel.....	4
1.3 Showing preformatted source code.....	4
1.4 Using tables.....	5
1.5 Using figures.....	5
2 DTD changes.....	5
2.1 Changes since document-v12.....	5
2.2 Changes since document-v11.....	5

This is a legal notice, so it is **important**.

Copyright © 2002-2004 The Apache Software Foundation. All rights reserved.

Note:

This is a demonstration document using all possible elements in the current Apache Forrest xdocs document-v1.3.dtd (See the [DTD changes](#) section at the bottom.)

1. Sample Content

Hint: See the xml source to see how the various elements are used and see the [DTD documentation](#) (../docs/dtd-docs.html) .

This is a simple paragraph. Most documents contain a fair amount of paragraphs. Paragraphs are called <p>.

With the <p xml:space="preserve"> attribute, you can declare that whitespace should be preserved, without implying it is in any other way special.

This next paragraph has a class attribute of 'quote'. CSS can be used to present this <p class='quote'> in a different style than the other paragraphs. The handling of this quoted paragraph is defined in the <extra-css> element in the skinconf.xml.

```
Anyway, like I was sayin', shrimp is the fruit of the sea.
You can barbecue it, boil it, broil it, bake it, sautee it.
Dey's uh, shrimp-kabobs, shrimp creole, shrimp gumbo. Pan
fried, deep fried, stir-fried. There's pineapple shrimp,
lemon shrimp, coconut shrimp, pepper shrimp, shrimp soup,
shrimp stew, shrimp salad, shrimp and potatoes, shrimp
burger, shrimp sandwich. That- that's about it.
```

A number of in-line elements are available in the DTD, we will show them inside an unordered list ():

- Here is a simple list item ().
- Have you seen the use of the <code> element in the previous item?
- Also, we have <sub> and <sup> elements to show content above or below the text baseline.
- There is a facility to *emphasize* certain words using the elements.
- We can use <icon>s too.
- Another possibility is the element:

another feather

, which offers the ability to refer to an image map.

- We have elements for hyperlinking:

```
<link href="your-project.html">
```

Use this to [link](#) (your-project.html) to another document. As per normal, this will open the new document in the same browser window.

<link href="#section">

Use this to [link](#) to the named anchor in the current document.

<link href="your-project.html#images">

Use this to [link](#) (your-project.html#images) to another document and go to the named anchor. This will open the new document in the same browser window.

<jump href="your-project.html">

Use this to jump to another document and optionally go to a named anchor within that document. This will open the new document in the same browser window. So what is the difference between link and jump? The jump behaves differently, in that it will replace any frames in the current window. This is the equivalent of `<a . . .`

`target="_top">`

<fork href="your-project.html">

Use this to fork your webbrowser to another document. This will open the document in a new, unnamed browser window. This is the equivalent of `<a . . .`

`target="_blank">`

- Oh, by the way, a definition list `<dl>` was used inside the previous list item. We could put another
 - unordered list
 - inside the list item

Or even tables..	inside tables..
or inside lists, but I believe this liberty gets quickly quite hairy as you see.	

Table 1: A sample nested table

So far for the in-line elements, let's look at some paragraph-level elements.

FIXME (SN):

The `<fixme>` element is used for stuff which still needs work. Mind the `author` attribute!

Note:

Use the `<note>` element to draw attention to something, e.g. ...The `<code>` element is used when the author can't express himself clearly using normal sentences ;-)

Warning:

Sleep deprivation can be the result of being involved in an open source project. (a.k.a. the `<warning>` element).

Important

If you want your own labels for notes and warnings, specify them using the `label` attribute.

Apart from unordered lists, we have ordered lists too, of course.

1. Item 1
2. Item 2
3. This should be 3 if my math is still OK.

1.1. Using sections

You can use sections to put some structure in your document. For some strange historical reason, the section title is an attribute of the `<section>` element.

1.2. Sections, the sequel

Just some second section.

1.2.1. Section 2.1

Which contains a subsection (2.1).

1.3. Showing preformatted source code

Enough about these sections. Let's have a look at more interesting elements, `<source>` for instance:

```
// This example is from the book _Java in a Nutshell_ by David Flanagan.  
// Written by David Flanagan.  Copyright (c) 1996 O'Reilly & Associates.  
// You may study, use, modify, and distribute this example for any purpose.  
// This example is provided WITHOUT WARRANTY either expressed or implied.  
  
import java.applet.*;    // Don't forget these import statements!  
import java.awt.*;  
  
public class FirstApplet extends Applet {  
    // This method displays the applet.  
    // The Graphics class is how you do all drawing in Java.  
    public void paint(Graphics g) {  
        g.drawString("Hello World", 25, 50);  
    }  
}
```

CDATA sections are used within `<source>` elements so that you can write pointy brackets without needing to escape them with messy `<` entities ...

```
<pointy>  
  easy  
</pointy>
```

Please take care to still use a sensible line-length within your source elements.

1.4. Using tables

And now for a table:

heading cell	heading cell
data cell	data cell
Tables can be nested	<ul style="list-style-type: none">• and can include most other elements, like lists

Table 1: Table caption

Not much of attributes with `<table>`, if you ask me.

1.5. Using figures

And a `<figure>` to end all of this. Note that this can also be implemented with an `` element.

The fine Forrest logo

2. DTD changes

See the generated [DTD documentation](#) (`../docs/dtd-docs.html`) .

2.1. Changes since document-v12

All v1.2 docs will work fine as v1.3 DTD. The main change is the addition of a `@class` attribute to every element, which enables the "extra-css" section in the skinconf to be put to good use.

2.2. Changes since document-v11

doc-v12 enhances doc-v11 by relaxing various restrictions that were found to be unnecessary.

- Links ((`link|jump|fork`) and inline elements (`br|img|icon|acronym`) are allowed inside `title`.
- Paragraphs (`p|source|note|warning|fixme`), table and `figure|anchor` are allowed inside `li`.
- Paragraphs (`p|source|note|warning|fixme`), lists (`ol|ul|dl`), table, `figure|anchor` are allowed inside definition lists (`dd`) and tables (`td` and `dh`).
- Inline content (`strong|em|code|sub|sup|br|img|icon|acronym|link|jump|fork`) is allowed in `strong` and `em`.

This is a legal notice, so it is **important**.