

SitemapPathModule

SitemapPathModule

This input module determines the absolute path to a file-resource, relative to the current sitemap. For example, if the sitemap from which this module is used is located in `/here/is/my/sitemap.xmap`, the following results are returned:

accessor	result
{sitemap-path:}	/here/is/my
{sitemap-path:input.xml}	/here/is/my/input.xml
{sitemap-path:../something.xml}	/here/is/something.xml

This module is useful in situations where you cannot use the standard source-resolving mechanism on generators and transformers. It was written to access an XML-catalog in a sitemap-relative directory from a serializer, and probably has other uses as well.

The SitemapPathModule must be configured in `cocoon.xconf`:

```
<component-instance
  class="org.apache.cocoon.components.modules.input.SitemapPathModule"
  logger="core.modules.input" name="sitemap-path"/>
```

The SitemapPathModule is not in the Cocoon distribution, so you will need to compile it. The complete source is in the attachment [SitemapPathModule.java](#). The relevant part of the code is:

```
package org.apache.cocoon.components.modules.input;

import java.util.*;
import org.apache.avalon.framework.configuration.Configuration;
import org.apache.avalon.framework.configuration.ConfigurationException;
import org.apache.avalon.framework.thread.ThreadSafe;
import org.apache.cocoon.components.CocoonComponentManager;
import org.apache.cocoon.components.modules.input.AbstractInputModule;
import org.apache.cocoon.environment.ObjectModelHelper;
import org.apache.excalibur.source.Source;
import org.apache.excalibur.source.SourceResolver;
import org.apache.excalibur.source.impl.FileSource;

public class SitemapPathModule extends AbstractInputModule implements ThreadSafe {

    public Object getAttribute(String name, Configuration modeConf, Map objectModel)
        throws ConfigurationException {
        SourceResolver resolver = CocoonComponentManager.getCurrentEnvironment();
        FileSource source;
        try {
            source = (FileSource) resolver.resolveURI(name);
            return source.getURI().substring(5); // Without "file:" prefix.
            // return source.getFile().getCanonicalPath(); Does not work on Windows when turned into a file: URL
        } catch (Exception e) {
            throw new ConfigurationException(e.toString());
        }
    }

    public Iterator getAttributeNames(Configuration modeConf, Map objectModel)
        throws ConfigurationException {
        return (new java.util.ArrayList()).iterator();
    }

    public Object[] getAttributeValues(String name, Configuration modeConf, Map objectModel)
        throws ConfigurationException {
        List values = new LinkedList();
        values.add( this.getAttribute(name, modeConf, objectModel) );
        return values.toArray();
    }
}
```

This module was donated to the Cocoon community by [NicoVerwer], in the hope that it might be useful to others. If you have any remarks, suggestions, etc. please discuss them on the mailing lists.