XSPEnvironment

An XSP page executes within a specific environment within Cocoon. This environment is very similar to that of a JSP page or Java Servlet.

Default Variables

The boiler plate code created in the generation of the compiled form of an XSP page includes a number of default variables:

- context - ServletContext
- request - HttpServletRequest
- response - HttpServletResponse
- parameters - parameters defined in the sitemap

The content, request and response classes are all in the `org.apache.cocoon.environment` package

Note: is this subsequent to change?

For example, you could access sitemap parameters in your XSP page as follows:

```
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp">
  <my-root>
    <xsp:logic>String id = parameters.getParameter("id", "-1");</xsp:logic>
    <id><xsp:expr>id</xsp:expr></id>
  </my-root>
</xsp:page>
```

Helpers

There are a number of helper classes which contain utility code for XSP classes. These all live in the `org.apache.cocoon.components.language.markup.xsp` package.

These helpers can be referenced from your own code if required, or used through one or more of the BuiltInLogicsheets.

Much of the code in the created `XSPGenerator.generate()` method involves calls to two classes:

- `XSPObjectHelper`

Responsible for taking expressions and producing SAX events from them. The `xspExpr` element is overloaded to take a number of different object types (Java primitives, String, Collection, XMLizable, Node, Object). Therefore the expression in an `xsp:expr` element must return one of these types.

- `org.xml.sax.ContentHandler`

Methods are called on an instance of this object to create the SAX events passed downstream from this pipeline. Ultimately all the methods on the XSPObjectHelper do the same.

A single instance of an `AttributesImpl` object is used in passing parameters to startElement calls on the `ContentHandler`. The attributes are cleared after each call.

There are several other helpers that are likely to be encountered in XSP soure code:

- `XSPRequestHelper`

This helper class is used by the XSP Request LogicSheet, and provides methods for interacting with the HTTP request, and access to the session. Effectively it offers the same API as the HTTPRequest object in the servlet API.

- `XSPResponseHelper`

This helper class is used by the XSP Session Logicsheet. It provides methods to set and add HTTP headers as well as encoding URLs

- `XSPCookieHelper`

Used by Cookie Logicsheet to manipulate cookies.

Default Imports

There are a few standard imports in every XSP page, but these are the minimum required to make the class compile (i.e. for the boiler plate code).

The imports currently (6-Dic-2002) are:

- From **java.util**: `Date, Stack, List`
- From **java.io**: `File, IOException, StringReader`
- From **org.xml.sax**: `InputSource, SAXException, AttributesImpl`
- From **org.apache.avalon.framework.context**: `Context`

- From **org.apache.avalon.framework.component**: `Component`, `ComponentException`, `ComponentManager`, `ComponentSelector`
- From **org.apache.avalon.excalibur.datasource**: `DataSourceComponent`
- From **org.apache.cocoon**: `Constants`, `ProcessingException`
- From **org.apache.cocoon.components.parser**: `Parser`
- From **org.apache.cocoon.generation**: `Generator`
- From **org.apache.cocoon.components.language.markup.xsp**: `XSPGenerator`, `XSPObjectHelper`, `XSPRequestHelper`, `XSPResponseHelper`, `XSPSessionHelper`