CreatingStandardWebPresence

Commons Standard Web Presence

The Problem

Currently, each Commons component is entirely responsible for its own web site. In addition, the "home" Commons site is maintained independently of the components themselves. The following problems result from this arrangement.

- Different components use different visual styles for their individual sites. This contributes to an apparent lack of cohesion within the Commons project, both to end users and to others in the Apache community. It also makes it more difficult for users to navigate through the Commons site, since there is no unified look and feel.
- Different components use different technologies for their individual sites. In addition to increasing the barrier to entry for individuals to contribute to multiple components, this complicates the process of creating a unified Commons "home" site.
- The "home" site does not always reflect the state of Commons as whole. This is essentially a side effect of the component sites and the "home" site being entirely independent; for any change to the component site, a developer has to manually reflect any appropriate changes in the "home" site.
- Not every component has a web presence, or even any web mention at all. Unfortunately, this is currently the case for several Commons Sandbox components. As a result, these components are virtually unknown, severely hampering their ability to form a community and subsequently develop into a full Commons component.

The Proposal

The Commons community has deemed it a worthwhile goal to create a common looking web presence across all of the Commons components. To accomplish this goal, a common web site generation and deployment process will be developed and employed by all the Commons components. The final process will meet the following requirements:

- The site for each component (proper or sandbox) can be generated and deployed independently of those for other components.
- All component sites include the same global navigation links to important Commons and Apache assets.
- Each component provides quick navigation links to the other Commons components on which it depends.
- The main Commons site has the same global navigation links as the components themselves.
- The main Commons site provides quick navigation links to each and every Commons component.

The Plan

Use of Maven

Of all the tools available to generate websites, Maven has been chosen to create the Commons components' sites. The primary reasons for this choice are:

- It is extremely versatile at creating websites and provides a lot of project content without much effort.
- It provides the means to work on the components individually or as a group.
- Many Jakarta Commons components are using Maven already for their build process, include site generation.
- Many Jakarta Commons committers and contributors have solid working knowledge of Maven and have expressed a desire to help with the standarization

Step 1. Mavenize all Commons Components

Some Commons components already have a Maven generated site. Most, however, do not use Maven for site generation or deployment. The first step to reaching a common looking web presence is to mavenize all the components under the Commons moniker.

(PROPOSED) Step 2. Define a Standard "Interface" which every Component site should implement.

Each component should have a series of common pages. Such as "description", "examples", "FAQ", "roadmap".

Each component should use a xdocs/navigation.xml for their menus. Make use of the top-menus and bottom-menus xml entities to include the common menus, for example: http://svn.apache.org/viewvc/commons/proper/dbcp/trunk/xdocs/navigation.xml?view=co

Each component should have a xdocs/download.xml file. It should give an overview of all available resources.

- releases (version+date)
- links to mirror of latest binary/source
- archive for older releases
- nightly builds
- cvs access

For example: http://svn.apache.org/viewvc/commons/proper/dbcp/trunk/xdocs/downloads.xml?content-type=text%2Fplain&view=co

Each component should have a project.xml extending project-base.xml For example: http://svn.apache.org/viewvc/commons/proper/dbcp/trunk/project.xml?content-type=text%2Fplain&view=co

Each component should have a project.properties file that includes the following: maven.xdoc.date=bottom, maven.xdoc.poweredby.image=maven-feather.png For example: http://svn.apache.org/viewvc/commons/proper/dbcp/trunk/project.properties

Each component should have a build.xml file that can be used by gump. Not using the maven jars but the gump dependencies. (only required for proper components?)

(PROPOSED) Step 3. Reorganize the content of main commons site

TODO

How to deploy

- Use Maven 1.0.
- commons/commons-build/ `maven -Dmaven.username=foo site:deploy` updates the main Commons site.
- Checkout commons-build and your component, go to your component directory and run `maven -Dmaven.username=foo site:deploy` and your component site is updated.

Current Status

TBD (List status, w.r.t. this proposal, of each component in proper and sandbox.)

Commons Proper Components

- Attributes - Maven site.
- BeanUtils - ⚠ Has Maven site, but **not default**.
- Betwixt - Maven site.
- Cactus - ⚠ **Promoted**.
- Chain - Maven site.
- CLI - Maven site.
- Codec - Maven site.
- Collections - ⚠ Has Maven site, but **not default**.
- Configuration - Maven site.
- Daemon - Maven site.
- DBCP - Maven site.
- DbUtils - Maven site.
- Digester - ⚠ Has Maven site, but **not default**.
- Discovery - ⚠ Has Maven site, but **not default**.
- EL - ⚠ **No Maven site**.
- Email - Maven site.
- Feedparser - ⚠ Has Maven site, but **site is still under sandbox**.
- FileUpload - Maven site.
- HttpClient - Maven site.
- IO - Maven site.
- Jelly - Maven site.
- Jexl - Maven site.
- JXPath - Maven site.
- Lang - ⚠ Has Maven site, but **not default**.
- Latka - Maven site.
- Launcher - Maven site.
- Logging - ⚠ Has **minimal** Maven site, but **not default**.
- Math - Maven site.
- Modeler - ⚠ Has Maven site, but **not default**.
- Net - Maven site.
- Pool - Maven site.
- Primitives - Maven site.
- Resources - Maven site.
- Transaction - Maven site.
- Validator - Maven site.
- VFS - Maven site.

Commons Sandbox Components

- Benchmark - ⚠ **No site**.
- Compress - Maven site.
- Contract - Maven site.
- Exec - Maven site.
- I18N - Maven site.
- ID - Maven site.
- Javaflow - Maven site.
- JCI - Maven site.
- OpenPGP - ⚠ **No site**.
- Pipeline - Maven site.

- Proxy - Maven site.
- SCXML - Maven site.
- Servlet - ⚠ **No site**.

Commons Dormant Components

- Cache - Maven site.
- Clazz - Maven site (needs update).
- Codec multipart
- Convert
- Events - Maven site.
- Filters
- Functor - Maven site.
- Grant
- Graph2
- Http
- Jex
- JJar - Maven site.
- JPath
- JRCS - Maven site.
- JUX
- Mapper - Maven site.
- Messenger - Maven site.
- Pattern
- Periodicity
- Reflect
- Rupert
- Scaffold - Maven site.
- Services
- SimpleStore
- TBM
- Test
- SQL - Maven site.
- Threading
- ThreadPool - Maven site.
- Workflow - Maven site.
- XMLIO
- XMLUnit
- XO

Commons Miscellaneous Information

- AltRMI - moved to incubator
- amend - not a component
- armi - old AltRMI code
- BZip2 - code moved to Compress
- cadastre - removed
- cjan - removed
- HiveMind - Maven site (offline) - moved out of Commons
- Joran
- Naming
- Proposal
- Util