Project Guidelines

1. The xml.apache.org Project Guidelines

This document defines the guidelines of the *xml.apache.org* project. It includes definitions of the various categories of membership, who is able to vote, how conflicts are resolved by voting, and the procedures to follow for proposing and making changes to the codebase of the Project.

> **Note:**
>
> Note that our sister project, jakarta.apache.org, is formulating a number of other project-wide guidelines that the xml community may wish to copy or modify. In Sept-2001 there are also a number of discussions on the general@xml.apache.org and commons-dev@xml.apache.org mailing lists about other cross-project guidelines that we hope will standardize our projects and make developing and using them easier. Stay tuned!

This is a living document. Changes can be made by the Project Management Committee.

- Roles and Responsibilities Defines the recognized roles in the project.
- Communication Defines how users and developers communicate.
- Decision Making Defines how action items are proposed and voted on.
- Source Repositories Defines how the Project's source code is organized and developed.
- Project Management Defines the roles and responsibilities of the Project Management Committee (PMC).
- Our Mission / Charter A copy of the Charter document for the xml.apache.org Project.

2. Website update guidelines

As XML-oriented projects, so our website is also generated from XML files. While many similar websites may use servlets or a system like Cocoon to generate their websites on-the-fly from XML documents, we currently checkin source .xml files of the documentation and then generate a static set of .html files for most of this website.

Briefly: the entire xml.apache.org website is stored in CVS repositories. Both the source .xml documents and the generated .html docs are checked in, and the actual website itself is simply a checked-out copy of the repository.

While each subproject's individual areas may be managed separately (by that subproject's committers) the top-level of the website should be managed through the xml-site repository.

Thus to change any of the files at the root of http://xml.apache.org you should first check out the xml-site repository from cvs.apache.org

Note:

The website is gradually being converted to Forrest. Whilst this conversion is occuring, there are two methods for updating content, dependent on what part of the site is being updated.

2.1. Updating the top level site

The top level of the site is created using Forrest. To update, committers with access to xml-site should check out this module, and make the content changes to the files found in *xml-site/src/documentation/content/xdocs*

There are two methods for updating the content on xml.apache.org.

The first method is to manually update the contents of the cvs xml-site module. (After checking in the content changes made.)

```
cvs checkout xml-forrest
cd xml-forrest
./build.sh
export PATH=`pwd`/build/dist/shbat/bin
cd ..
cvs checkout xml-site
cd xml-site
forrest
cp -r build/site/* targets/
cvs add <new files>
cvs commit targets
```

The second method is to use the *forrestbot*.

```
Log in to http://forrestbot.cocoondev.org
Select 'xml-site' from the drop-down list
Click 'refresh', wait for the site to rebuild
Click 'publish'
```

Note:

To stop random people triggering resource-intensive rebuilds, the page is password-protected. Committers wishing to use this facility should contact the forrest-dev mailing list.

2.2. Updating Forrest based sub-project websites

To use Forrest install it as described, cd in the directory of your project and run "forrest".

Publish the HTML/PDF result of the build into the xml-site/targets module (copy the results of ./build/site to xml-site/targets and commit changes). The sources of the top-level xml.apache.org website are in the xml-site/src/documentation directory.

2.3. Updating non-Forrest sub-project websites

To update a part of the website (that does *not* use Forrest) do this:
Checkout the xml-site module (you only actually need a few subdirs, but the whole thing is quite large).
Make updates to the .xml files in the xml-site/sources/website directory.
Go to the xml-site/sources directory locally and run the relevant shell or bat script.
Check that the .html structure in your local targets directory is built correctly.
Checkin to CVS both your updated sources/ and targets/ trees as needed,
Either login to xml.apache.org and cd to /www/xml.apache.org and do an update of directories there as needed,
Or send an email to infrastructure<at>apache.org and ask for your website to be added to the cronjob for automatic update.