

Substitution of Custom Maven Components

Substitution of Custom Maven Components

We're already starting to see an emerging need to customize Maven, by adding new types of artifact resolvers, project builders, etc. Since we've taken so much care to design Maven as component-oriented software, we need to open up those component-oriented aspects.

State of the Art

Currently, if a user wants to add a custom artifact handler to Maven, the cleanest way to do it is to write the resolver along with a new `components.xml`. This `components.xml` must define a new component descriptor that overrides the default one. After that, the user will have to stash this resolver jar into a new directory, most likely under someplace like `${maven.home}/custom`. Finally, they have to add a new line to the `${maven.home}/bin/m2.conf` file, to add the new directory to the front of the classpath, like this:

```
load ${maven.home}/custom/*.jar <-- this one has to come first, to override.
load ${maven.home}/lib/*.jar
```

Where can we go from here?

It would be **much** nicer to have the ability to simply dump the new resolver jar into `${maven.home}/lib`, and change the selection priority for the `ArtifactResolver` to use the new implementation instead of the default one...this would happen in a master config XML file, maybe. It would also allow the new resolver to delegate to the default implementation without redefining its descriptor (right now, we'd be overriding the default implementation's component descriptor, effectively wiping it out of the container).