

# Amherst: Render fedora metadata resources as MODS/XML

<b>Title (Goal)</b>	Amherst - MODS XML transformation service
<b>Primary Actor</b>	Developer
<b>Scope</b>	Component
<b>Level</b>	
<b>Author</b>	<a href="#">Unknown User (acoburn)</a>
<b>Story (A paragraph or two describing what happens)</b>	There is a need to express resource metadata as MODS/XML for external harvesters. This would involve a transformation service that converts the RDF descriptive metadata into MODS XML. In principle, this can be done with camel using the xslt component along with an external XSLT resource.

A sample OSGi-based implementation is available here: <https://github.com/acoburn/repository-extension-services>

## Web Resource interaction

This service would expose an HTTP endpoint to convert fedora resources into a MODS/XML or DC/XML representation.

## Deployment or Implementation notes

This service would be deployed separately from fedora, possibly on a separate machine. I envision that this would be implemented as a combination of OSGi services and camel routes that can be deployed in any OSGi container, written in Java and Blueprint XML. The implementation would require access to Fedora's HTTP API, fedora's event stream and possibly a local linked data fragments server.

The primary challenge would be in dereferencing external URIs. For example, a fedora resource that points to an external URI (e.g. via dc:contributor). I would like not to copy those remote triples into fedora (they should remain externally managed), but I would like to cache them locally. To support this, a process that listens to fedora's event stream could detect such external URIs, and if they contain RDF, those assertions could be dereferenced and stored in a linked data fragments server or local triplestore. Then, when the XML document is constructed, dereferencing those values would not require access to the remote resource, rather the triples could be generated from the local cache.

## API-X Value Proposition

The primary use of this service in the context of API-X would be to allow for service discovery and deployment.