

## XML Server

XML Server returns Innovative WebPAC search results as XML data.

### **Say what?**

You can take live search results *or* any Review File and display the data however you please. More importantly down the road, you can integrate catalog data with other systems. XML Server provides somewhat of a search API into the catalog. Even our users can integrate our catalog into *their* own applications...craziness! Just the data, please.

### **Doesn't Z39.50 do that?**

Sort of, but it's sooo Z39.50. Who wants to use it? It's difficult to work with and limited in function. Besides, Z39.50 itself is moving to the Web and XML via SRU/SRW.

XML is more developer-friendly. It's also open and non-proprietary, but it's gained much wider acceptance. You can take advantage of an ever growing set of open data, tools, and the expertise of library and non-library people alike. It has become the standard for data-sharing. The tools for doing cool stuff with it are becoming simpler all the time.

### **What will it output?**

It outputs a set of bib records along with, if specified, attached record data.

### **What about review files?**

Same thing and it's not limited to only the review files designated as Featured Lists.

If it's a review file of patron records you get a list of patron record numbers. Combine that with Patron API and you have the potential for some interesting applications.

### **Examples?**

No examples with the patron stuff yet, that's still a dream.

The primary reason we purchased XML Server was to make the state publications collection more accessible to search engines and other services. Just as importantly though is enabling our data and our services to meet our users, our customers, where they are.

Jason recently put up a "service" that accepts an XML Server (xmlopac) query along with some other optional parameters, queries Arthur, and transforms the XML response into the requested XML format. How about an [RSS feed of your catalog search](#)? Currently, the available formats are RSS, A9's OpenSearch, and Dublin Core.

That was the first step. It's simple and useable by the whole cluster.

We're expanding the service so that items can be filtered by item location and cover art from Amazon.com can be added to the output. We'll use these live feeds to feature new books, state publications, and other items on our web site.

Online at <http://arthurweb.pbwiki.com/XMLServer%3A%20About>

## Patron API

Patron API, given a patron record number or the unique identifier, returns patron record information over the web to a client with an authorized IP address.

In other words, you configure the IP address of the server that you want to have access to Patron API. It should be a server used only by authorized applications or library staff.

You send a URL request to Arthur that includes either a single patron record number or the established unique identifier. Arthur returns the patron record data as HTML text to the client.

### ***What's it for?***

Patron API is just that, another interface to patron data. The interface allows other systems to take advantage of the information stored in the patron database.

It is used mostly for authenticating patrons for access to another system, although it's not limited to that.

Missouri State Library will initially use Patron API to provide a live data source for EZ Proxy authentication instead of manually exporting data from Millennium to EZ Proxy. Ultimately, we see the Arthur patron database as being our single store of user information. We currently have multiple stores of users for various services. Patron API and XMLServer will allow us to get the most advantage out of our data.

### ***But we get patron loads from the campus admin office.***

Then you might not need Patron API. Or, if you're currently loading the same data separately, once for Arthur and once for your proxy authentication, using Patron API could eliminate that additional loading.

Missouri State Library can only obtain very limited data about state employees from the Office of Administration. Therefore, we manually enter all of our patrons in Millennium and it's our best data source for authenticating users for other services. Also, our patron data may change fairly frequently.

### ***How do I make Patron API talk to EZ Proxy?***

EZ Proxy has built-in support for the Innovative Patron API. See <http://www.usefulutilities.com/support/usr/iii.html>.

### ***What about requesting multiple patron records at once?***

Not in a single request. You must make a separate request for each patron record. Simple.

Online at <http://arthurweb.pbwiki.com/Patron%20API%3A%20About>