

Metacat - Bug #2207

Advanced Search integration

09/26/2005 03:20 PM - Duane Costa

Status:	Resolved	Start date:	09/26/2005
Priority:	Normal	Due date:	
Assignee:	Duane Costa	% Done:	0%
Category:	metacat	Estimated time:	0.00 hour
Target version:	1.6	Spent time:	0.00 hour
Bugzilla-Id:	2207		

Description

Over the past year, the LTER Network Office has developed an Advanced Search web application that uses the Metacat client to run an advanced search on criteria such as subject, author, spatial, and taxon. In its current form, the Advanced Search interface exists as a separate web application, outside of the Metacat code base. The goal of this task is the integrate the Advanced Search web application with the Metacat code base, as described in more detail below.

Proposal to Integrate Advanced Search Capability with Metacat Distribution

The goal is to refactor the Query application so that major parts of it would be integrated with Metacat, while other parts of it could be customized for LTER-specific needs and maintained independent of Metacat.

1. Query Engine

The back-end Query Engine can be fully integrated with the Metacat code base. It contains search engine functionality that is generic to Metacat and should be relatively easy to factor out of the Query application. Once it is part of Metacat, it can be packaged as a library that can be distributed with the Query application. After it is integrated with Metacat, the Query Engine code can be maintained by all Metacat developers. If logical improvements or performance optimizations are made to the Query Engine code by the Metacat community, the LTER Query application will benefit from these improvements and optimizations because it will utilize the same code that Metacat utilizes.

2. Advanced Search Form

The Advanced Search Form is implemented as a JSP. It can be reimplemented to eliminate the Struts-based custom tags that it currently uses. JavaScript could be added to replace the Struts-generated JavaScript for client-side form validation; alternatively, form validation could be moved to the server side.

A LTER-customized version of the Advanced Search Form could be maintained in the Query Application. It would be nearly identical to the Metacat version, but it would contain additional input fields, such as a drop-down list that allows the user to restrict the search results to a particular LTER site.

If possible, a mechanism would be worked out to minimize the duplication of effort that would be required to maintain both the Metacat form and the custom LTER form and to keep them consistent.

3. Login Page, Simple Search Page, and Browse Page

These pages in the Query Application would not be integrated with Metacat, since equivalent functionality already exists in the default Metacat skin. These pages would continue to be part of the Query application. Since the Query application uses Struts to manage the functionality of these pages, we could continue to use Struts in the Query application, though we would not need Struts in Metacat.

We may be able to utilize the Metacat skin to provide this functionality for the Query application. Eventually, we may be able to fully migrate all of the functionality of the Query application to Metacat, though we would need to provide a way to extend the Metacat skin with LTER customizations.

The current browse capability of the Query Application is intended to be replaced by a browsable hierarchy of terms based on the work that is being initiated in the LTER working groups for Ontologies and Controlled Vocabularies. Since this work would be a valuable contribution to Metacat as well, it would be useful to integrate these new browse capabilities in the Metacat skin rather than restrict them to just the Query application.

Work Estimates:

Task	Effort (Weeks)
------	----------------

Phase One:

These tasks would fulfill Matt's requirements to integrate the Advanced Search capability with Metacat, while retaining the Query application as a LTER custom application that shares some of its software components with Metacat.

Refactor Query Engine. Add code to Metacat. Refactor Query Application to use Metacat library.	1
Refactor Advanced Search Form to eliminate Struts custom tags.	1
Implement generic Advanced Search Form and integrate it with Metacat Skin. Maintain custom LTER form in the Query Application as an extension to the Metacat form.	1

Phase Two:

This optional phase would deprecate the Query application as a separate entity, eliminating the duplication of effort needed to keep its advanced search functionality consistent with Metacat's. The time estimates for these tasks should be adjusted after Phase One is completed, since we will have a better understanding of the effort required at that point.

Fully migrate the Query Application to Metacat, allowing for LTER customizations within Metacat. Other organizations could use the LTER customizations in Metacat as a model for their own customizations. 2

On 9/6/2005, Mark Servilla wrote:

Hi Matt,

Attached, please find a brief statement of work proposed by Duane Costa

regarding the Advanced Query Interface for metacat. We consider the re-unification of the two applications to be a high-priority to the LTER, NCEAS, and the eco-community, and will begin the planning/work effort immediately. At your earliest convenience, please review the SOW and let us know if this is acceptable and/or if you have any questions/comments.

Sincerely,
Mark

On 9/6/2005, Matt Jones wrote:

Hi Mark,

Thanks. In general this statement of work looks great -- no real modifications on my part. It will be a great time-saver for many people who want an advanced search in their metacat installation, so I appreciate it.

A couple of brief comments for context:

1) We need to revise our login infrastructure. Right now metacat uses cookies for session state. That has worked well, and is pretty robust.

We also have a more recent javascript login for managing the cookies on the KNB and default skins -- this is incomplete and therefore broken, although it is what is used on the main KNB page. The problem is that the session information does not propagate across pages as one navigates through the app, especially in the EML pages. We know why, but haven't fixed it. It would be a good time to do so, so let us know if you have particular needs in your part of the login infrastructure.

2) We've wanted an effective browsable hierarchy of terms for KNB datasets, but there just isn't consensus on a controlled vocabulary. The one on the KNB skin is one I made up with feedback from Mark S and a few others. If you get consensus with LTER, we'd probably want to switch the KNB in general over to using it and creating a browsing interface that allows one to navigate it. So that might be another area for shared work.

3) I think the advanced search page needs a map to draw a box for spatial searches. We've found users simply don't know the lat/lon for their area of interest.

4) We are working on a spatial option for metacat for Kruger that allows the locations of data to be plotted against other GIS layers in a map and searched using spatial queries. This is in prototype now, but will be released hopefully with the next metacat release. Just a FYI in case a similar request has come your way.

5) The new SEEK web developer located at LNO (in process of hiring replacement for Tekell) will be working on a portal for access to EcoGrid data (both EML and DarwinCore). I hope we will be able to adapt your client interface so that it can be applied to the web services backend in EcoGrid, as I think the requirements are essentially the same.

Thanks. Looking forward to working on this with you.

Matt

On 9/23/05, Mark Servilla wrote:

Hi Matt,

Sorry for the delay in getting back to this matter. After discussing your comments with Duane, the only item that impacts our involvement directly is number 1 - the session management. It will be critical to work this problem to completion. With respect to number 3, the map UI, we would appreciate any suggestions in this area; short of installing a full-up ArcIMS/MapServer type of application. Is there a simple javascript version for such a map? A first version of this would require only the bounding lat/lon for the query. Thanks!

Sincerely,
Mark

On 9/23/05, Matt Jones wrote:

Mark,

Thanks for the followthrough. I agree that (1) needs to be worked out, and I'd like to see the GT4 GSI certificate stuff that NCSA did before we decide on a solution. We should probably take an approach that accomodates that if we're delving into the auth infrastructure.

Regarding (3), I think there are several open-source geospatial libs for doing this (we use one of them in Morpho). Maybe Duane could talk to John Harris (who is working on this stuff in metacat now) and Dan Higgins (who did the geospatial map in morpho) and come up with a proposal? My impression is that the java lib Dan used was pretty effective and easy to plug into morpho, and I think others have come about. GEON has one in their portal search client, so we might be able to borrow code or ideas from them. We definitely don't want an IMS for this part -- our needs are much simpler.

Matt

History

#1 - 10/21/2005 09:23 AM - Duane Costa

Advanced Search Design and Implementation Plan

The Advanced Search interface was originally designed and implemented at LNO as a Struts web application. However, to simplify its integration with Metacat, it has been re-implemented with JSPs and servlets, with all Struts being stripped out. It still retains some of the desirable design characteristics of a Struts web application in that it is based on a Model/View/Controller (MVC) architecture.

Design and implementation of the Metacat advanced search interface is proposed as follows:

1. Back End: Advanced Search Engine (Model)

The back end advanced search engine is implemented in Java. It is comprised of a package of related classes whose purpose is to:

- (A) Construct a complex PathQuery XML string based on form inputs stored in a bean class.
- (B) Send the PathQuery XML to the Metacat client.
- (C) Receive the search results from the Metacat client.
- (D) Transform the search results to HTML and store them in a request attribute where they can be displayed by a JSP.

The advanced search engine code will be stored in the following package:

```
package edu.ucsb.nceas.metacat.advancedsearch;
```

2. Front End: Advanced Search Form (View)

The advanced search form is implemented in JSP and will be integrated, as seamlessly as possible, with the existing set of Metacat skin JSPs. For example, just as there is an existing skin JSP called:

```
include_searchbox.jsp
```

that implements the simple search capability, there will be a new skin JSP called:

```
include_advancedsearch.jsp
```

that implements the advanced search capability.

The following files will be added to the default skin in directory lib/style/skins/default:

include_advancedsearch.jsp Implements the advanced search form.

index_advanced.jsp Similar to index.jsp, but replaces the simple search box with the advanced search form. When user clicks on the ">> advanced search <<" link, this form replaces the index.jsp form.

advancedsearchresults.jsp Auxiliary JSP to display the advanced search results that are generated by the advanced search engine.

advancedsearchforward.jsp Auxiliary JSP that populates the bean with form input and forwards control to the advanced search servlet.

It will be necessary for other skins to integrate this code on an individual basis. This is probably a good thing, because it will allow us to work out any bugs first in the default skin, before other groups begin to integrate the code into their particular skins.

3. Advanced Search Servlet (Controller)

The advanced search servlet controls the interaction between the JSPs, the bean, and the advanced search engine. (This serves the same purpose as a Struts "action" class.) For simplicity, the servlet code resides in the same package as the advanced search engine:

```
package edu.ucsb.nceas.metacat.advancedsearch;
```

#2 - 11/29/2005 08:49 AM - Duane Costa

On 11/22/2005, Duane Costa wrote

Hi Matt,

I've checked in all my code to enhance the default skin with Advanced Search functionality. A demo can be viewed on my PC at:

<http://earth.lternet.edu:8080/knb/index.jsp>

Mark will be sending you a follow-up message shortly to discuss where we go from here with it.

Thanks,
Duane

On 11/28/2005, Mark Servilla wrote

Hi Matt,

Not too much to follow up on. The existing query interface code is not fully integrated into the Metacat distribution due to the login infrastructure issue. We (Duane) have taken the integration as far as we can until the session issue is resolved. We are wondering what your schedule looks like for this fix, and should we still remain open in the near future to complete the integration. I believe that you are still targeting a 1 Jan release for ESA? Thanks.

Sincerely,
Mark

On 11/28/2005, Duane Costa wrote

Just to expand on Mark's comments a bit: The Advanced Search code is fully integrated into the Metacat distribution with regard to the 'default' skin. But with regard to the 'lter' skin, we are still redirecting our Metacat interface to a separate 'Query' web application maintained at LNO. Our goal is to eventually base the 'lter' skin on a modified version of the 'default' skin in the Metacat distribution, but we're holding off on that until the login and session management bug in the 'default' skin is fixed.

Meanwhile, though, I've configured things such that the 'default' skin and our separate 'Query' web application share as much common code as possible. This has minimized the effort needed to maintain the two, but eventually we'd like to drop the 'Query' web application altogether and maintain the 'Iter' skin completely from within Metacat.

Duane

On 11/28/2005, Mark Servilla wrote

Matt,

One additional note...

After reviewing the start of this email thread, I realized that you may be waiting for input from the NCSA folks regarding the authentication infrastructure that was implemented for Metacat w/r to our grid pilot study. I've attached the only document I have - it's a brief technical summary of the GT4 Web Service implementation for Metacat. Please let me know if we need to take any action from our side to make this happen.

Sincerely,
Mark

On 11/28/2005, Matt Jones wrote

Hi Mark and Duane,

Sid has been working on new functional login infrastructure for the ESA skin, which I believe is working. This has not been transferred to the KNB skin, but I'm hoping we will before the release. I'll also try to get someone here to integrate the advanced search into the other skins that we use so that it is more universally accessible. We are still targeting a Jan 1 release. Thanks.

Matt

On 11/28/2005, Matt Jones wrote

Hi Mark,

Bill Baker at NCSA was very nice about keeping us updated on the GSI changes for metacat. He even arranged a conference call with me, Sid, and Jing to review what they changed and how it worked, and sent us the code in a nice package to integrate in with metacat. So the ball is in our court to add the GSI functionality, which I want to do, but probably not until after we get the ESA registry up and running. I'll probably ask John or Sid to work on this in the future.

Matt

#3 - 12/13/2005 01:52 PM - Duane Costa

Modified the advanced search servlet so that it no longer hard-codes the context to 'knb'. The servlet dynamically reads the context property value from the metacat.properties file.

Modified the advanced search form so that it no longer hard-codes the URL to the LiveMap 3.0 Java applet. The form now uses tokens to construct the URL to the applet on the metacat server.

Modified build.xml to protect the LiveMap 3.0 applet jar file from ant token filtering.

I believe that all work on this bug has now been completed. I will close out the bug. Work on migrating the advanced search interface from the default skin to other skins will be tracked in separate bug entries.

#4 - 12/13/2005 01:56 PM - Duane Costa

See above comment.

#5 - 03/27/2013 02:19 PM - Redmine Admin

Original Bugzilla ID was 2207