Metacat - Bug #162

need harvest/batch load for metacat

10/24/2000 05:28 PM - Matt Jones

Status:	Resolved	**Start date:**	10/24/2000
Priority:	Normal	**Due date:**	
Assignee:	Duane Costa	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	1.4	**Spent time:**	0.00 hour
Bugzilla-Id:	162		

Description

The metacat server needs to be able to accept large numbers of metadata documents for insert and update from site metadata catalogs. This should be enabled either through a pull or push mechanism, so the pull (harvest) will need a registry service as well.

Related issues:

Blocks Utilities - Bug #323: Establish API for communication with src and dest	**In Progress**	**11/08/2001**
Blocks Utilities - Bug #324: perl implementation of harvester api	**In Progress**	**11/08/2001**

History

#1 - 04/09/2001 12:21 PM - Matt Jones

Decided that this feature will use a site-specific XML input filter that converts the site's metadata to XML format. Need to deal with synchronization issues (how to determine whether something is the "same" and thus not load it multiple times into metacat. Reassigned to Owen based on our discussion in Sanata Barbara.

#2 - 11/08/2001 11:38 AM - Matt Jones

Develop according to API defined in bug #323 and reference implementation in bug #324

#3 - 03/07/2003 11:07 AM - David Blankman

LTER will take over responsibility of this.

#4 - 03/31/2004 12:08 PM - Matt Jones

Duane, I'm preparing for a new release of metacat. Is this harvester code ready to ship? Please provide an update in this bug so we can decide. Thanks.

#5 - 04/01/2004 01:31 PM - Duane Costa

Harvester has demonstrated basic functionality. The following Harvester capabilities have been implemented:

(1) Read Harvester properties from a configuration file at startup.

(2) Read the HARVEST_SITE_SCHEDULE table.

(3) Determine whether a site is due to be harvested based on the scheduled date of its next harvest in the HARVEST_SITE_SCHEDULE table.

(4) Retrieve the Harvester Document List from a site's Harvest Document List URL.

(5) Validate the site's Harvester Document List using the harvesterDocument.xsd schema.
(See metacat/lib/harvester/harvesterDocument.xsd)

(6) Query the XML_DOCUMENTS table to determine whether Metacat already has the particular revision of the EML document that it intends to insert or update. More specifically, Harvester queries to determine

the highest revision that Metacat currently stores for a document.
This information is also recorded in a log entry so that it can
ultimately be included in a report that is generated to the site.

(7) Retrieve one or more EML documents from the site.

(8) Insert or update an EML document to Metacat.

(9) Log all Harvest operations in the HARVEST_LOG table.

(10) Log errors involving EML documents in the HARVEST_DETAIL_LOG table.

(11) Update the date of last harvest, and then schedule the date of
next harvest (calculated from the current date plus the
update frequency) in the HARVEST_SITE_SCHEDULE table.

(12) Generates and sends an email report to the site
administrator after each harvest. The email report summarizes the
list of documents that were harvested as well as any errors that
were encountered for a particular site. (Note: this item will be
completed and checked in within a day or two, so I am listing
it as completed.)

(13) Generates and sends an email report to the Harvester
Administrator after each harvest. The email report summarizes the
list of documents that were harvested as well as any errors that
were encountered for all sites that were harvested.

(14) Purge old entries from the HARVEST_LOG and HARVEST_DETAIL_LOG tables.
(By default, entries older than 90 days are purged, though this is a
configurable property.)

(15) (a) Harvester can be configured to run a new harvest every n number
of hours, where n is a whole number. This is managed with the
'period' property. For example:

```
period=24

will cause a new harvest to run once every 24 hours.
         Harvester calculates the amount of time it needs to sleep
         between harvest runs based on this value. The calculation is
         adjusted to account for the time spent on the last harvest
         run. In other words, the harvest runs are scheduled at a
         fixed rate (similar to the scheduleAtFixedRate() method in
         Java's Timer class).

(b) Harvester can be configured to delay its first harvest for n
         number of hours. For example, setting the delay property as
         follows:

delay=10

will cause Harvester to begin its first harvest 10 hours after
         it starts up.

(c) Harvester can be configured to stop running after a maximum
         number of harvest runs have been completed. For exmaple:

maxHarvests=30

will cause Harvester to shut down after 30 harvest runs.
         If, in this example, the period is set to 24 (period=24),
         then Harvester will run once per day for thirty days and
         then stop.
```

(16) A Harvester Registration Servlet has been implemented. The servlet
allows a site user to login and then set the following values for
a particular site:
Email Address (email reports for the site are sent to
this address)
Harvester Document List URL
Harvest Frequency (1-99)
Unit (days, weeks, or months)

Work Remaining: the following features have not yet been implemented:

(17) The xmltables.sql script must be modified to create the
Harvester tables.

(18) JUnit tests have not yet been written to test the Harvester code.

(19) A Harvester Document List editor application is currently being
implemented by one of our graduate students (Saurabh Sood).
The editor allows a user to compose a Harvester Document List
that conforms to a schema (see file
metacat/lib/harvester/harvesterDocument.xsd) by entering values
in the expandable/collapsible nodes of a JTree.

```
We have been meeting with Saurabh on a weekly basis and
     monitoring his progress. He is about 75% finished with the
     implementation of the editor. I will need to review Saurabh's
     code, ask him to make any needed changes, and then commit his
     files when they are ready.
```

(20) User documentation has not yet been written.

Estimated time required to complete the remaining items:

```
Weeks to complete
(12) Email to site administrator            0.2
(17) xmltables.sql                          0.6
(18) JUnit tests                            0.8
(19) Harvester Document List Editor         1.4/0.4
(20) User documentation                     0.6
```

Note: The estimate for (19) is based on 1.4 weeks of Saurabh's
time and 0.4 weeks of Duane's time.

Total time estimate: 2.6 weeks (Duane), 1.4 weeks (Saurabh)

Other Possible Enhancements: The following would be useful
enhancements but they are probably not critical to the first
Harvester release:

(21) Harvester currently loads properties from a harvester.properties
file that it looks for in the metacat/lib/harvester directory.
A few of the properties duplicate those that appear in Metacat's
build.properties file. The harvester.properties should probably
be integrated into build.properties, which in turn would eliminate
the duplication.

(22) Harvester queries the XML_DOCUMENTS table directly. Instead, a
method named getHighestRevision() should be removed from Harvester
and added to the Metacat Client API. The method returns the highest
revision that Metacat currently has for a given document id, or
-1 if the document is not currently in Metacat.

(23) The Harvester Registration Servlet should probably be incorporated
into the Metacat home page. That is, when a user logs in to Metacat,
the user should be able to click a button that opens up the Harvester
Registration Servlet. In the current implementation, the servlet
has its own dedicated login page and it's a bit clunky, since the
user needs to enter their full LDAP distinguished name as the
username.

(24) Harvester has only been tested against the Oracle database. There's
nothing database-specific in the code, since Harvester uses JDBC,
but it has not yet been tested against Postgres or SQL Server.

Duane

#6 - 04/29/2004 11:09 AM - Duane Costa

Update on Work Remaining:

(12) Email to Site Administrator

Completed.

The email message content could be improved by adding more comprehensive
summary information at the top of the message. For example, provide a summary

of each site that was harvested from and the number of documents that were harvested from that site.

(17) SQL scripts

Five SQL scripts have been updated with the code that creates the three database tables used by Harvester. The modified scripts are:

```
metacat/src/xmltables.sql
  metacat/src/xmltables_postgres.sql
  metacat/src/xmltables-sqlserver.sql
  metacat/src/upgrade-db-to-1.4.sql
  metacat/src/upgrade-db-to-1.4_postgres.sql
```

Testing of these scripts has been completed on Oracle. Matt has agreed to test the scipts on PostgreSQL. I have tried to test the scripts on SQL Server but I'm encountering some problems; I'll continue trying.

Note also that there is currently no update-db-to-1.4 script for SQL Server.

(18) JUnit tests

An initial suite of 21 tests has been completed and the test classes have been checked in to directory:

```
metacat/test/edu/ucsb/nceas/metacattest/harvesterClient
```

The test suite has been run successfully on Windows and Linux.

These tests cover a lot of Harvester's functionality, though not every aspect of it. It was necessary to refactor portions of the Harvester code to accomodate the test cases. One lesson I've learned is that it is preferable to develop the JUnit tests in conjunction with code development rather than after the fact!

(19) Harvest List Editor

An initial version of the Harvest List Editor was submitted to me by our graduate student programmer, Saurabh Sood. In testing the editor, I found a number of areas that need improvement. I met with Saurabh today (4/29/04) and we agreed on a list of changes that he will make to the editor. Saurabh will need an additional two weeks to complete the changes I asked for.

(20) User Documentation

An initial draft of the user documentation has been checked in at:

```
metacat/docs/user/harvester.html
```

I've linked in the harvester.html page as one of the last pages on the Metacat Tour documentation. (Of course, if Matt feels that this isn't the right location for the Harvester documentation then I'll delink it from the Metacat Tour.)

The documentation describes what Harvester does, as well as the roles of the Harvester Administrator and the Site Contact and what each person needs to do for Harvester to function.

#7 - 06/15/2004 10:12 AM - Duane Costa

Update on Work Remaining:

(17) SQL Scripts

Still need testing on PostgreSQL and SQL Server.

(19) Harvest List Editor

Completed.

Documentation for the Harvest List Editor has been checked in to metacat/docs/user/harvestListEditor.html. A link in the Harvester documentation (harvester.html) points to the Harvest List Editor documentation.

(20) User Documentation

Completed

metacat/docs/user/harvester.html

(21) Integrate Harvester properties with Metacat

Completed.

All Harvester properties are loaded from the metacat.properties file. They now use the Options class from the utilities module rather than Java's built-in Properties class.

(22) Add new query method to the Metacat Client

Not yet implemented. Harvester currently uses a direct query of the XML_DOCUMENTS table to determine a document's current revision.

(23) Improve the login page for the Harvester Registration servlet

The user interface for the login page has been improved. It is similar to the login page on the Metacat web interface. Users only need to enter the uid portion of their LDAP account name and select their affiliation by clicking a radio button. (This differs slightly from the Metacat web interface which uses a drop-down list for selecting the user's affiliation.) As with the Metacat web interface, the "dc=ecoinformatics,dc=org" portion of the LDAP account name is automatically appended to form the distinguished name.

I have not integrated the Harvester Registration Servlet directly into the Metacat web interface. One aspect I am unsure about is how to accomplish this when there are many different skins for the web interface. Would the logic need to be repeated separately for each skin, or is there a way to implement the logic in one place that makes it common to all skins?

However, I'm no longer certain that it's really necessary to integrate the Harvester Registration servlet into the Metacat web interface, since the servlet now has an improved login interface of its own which is very much like the one in the Metacat web interface.

(24) Testing Harvester on PostgreSQL and SQL Server

Not yet completed.

Summary

At this point, all major work on the first version of the Harvester implementation has been completed.

Prior to the Metacat 1.4. release, additional testing is needed on the other two databases (PostgreSQL and SQL Server) for Harvester itself (24) as well as for the SQL scripts (17). Since Harvester uses JDBC, I have no reason to believe that there is anything vendor-specific, though this should still be verified with testing.

Although (22) would be a more elegant way for Harvester to query Metacat about document revisions, the simple direct query of the XML_DOCUMENTS table is working fine, so I don't think this is a release critical issue.

The Harvester Registration login page has been improved (23). At some future point it may be desirable to integrate it with the Metacat web interface if an appropriate implementation for this can be devised.

#8 - 09/21/2004 03:46 PM - Saurabh Garg

Closing as testing with postgres has been completed and harvester works with both postgres and oracle.

#9 - 09/21/2004 08:18 PM - James Brunt

Yahoo!

#10 - 04/04/2005 03:45 PM - Duane Costa

The following bug fixes and enhancements to the Harvester code have been completed and will be included in Metacat 1.5:

Bug Fixes:

- Modify property values of harvester registration servlets to match the servlet-mapping URL values in web.xml. The old values used the servlet class names. This worked in Tomcat 4 but seems to break in Tomcat 5 on Windows. The new values use the servlet-mapping URL values. This should work in both Tomcat 4 and Tomcat 5.

- Re-implement logic to prune old log entries from the HARVEST_LOG and HARVEST_DETAIL_LOG tables. The old logic caused integrity constraint violations in the database because it tried to delete parent records from HARVEST_LOG prior to deleting child records from HARVEST_DETAIL_LOG.

Enhancements:

- Implement a new HarvesterServlet for running Harvester as a servlet. This eliminates the need to run Harvester in a terminal window. By default, the HarvesterServlet is commented out in lib/web.xml.tomcat(3,4,5). The user documentation will be modified to instruct Harvester administrators to uncomment the HarvesterServlet entry.

- Minor enhancement to support multiple email addresses for harvester administrator and site contact. Each address is separated by a comma or semicolon.

- Increase number of rows in Harvest List Editor from 300 to 1200.

- Changed default maxHarvests value to 0. Added logic to ignore maxHarvests value when it is set to 0 or a negative number. This allows Harvester to run indefinitely without shutting down after reaching a maximum number of harvests. The previous default value of 30 would cause Harvester to terminate after 30 harvests.

#11 - 03/27/2013 02:13 PM - Redmine Admin

Original Bugzilla ID was 162