Metacat - Bug #191

Override protocol handling for http URLs

04/09/2001 12:41 PM - Matt Jones

Status:	Resolved	**Start date:**	04/09/2001
Priority:	Immediate	**Due date:**	
Assignee:	Matt Jones	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	Beta2	**Spent time:**	0.00 hour
Bugzilla-Id:	191		

Description

Metacat currently uses a socket system for uploading data files because of an inefficiency in how http urls are handled by the default protocol handler. Need to investigate overrriding that protocol handler so that this problem is not encountered.

Related issues:

Blocked by Metacat - Bug #240: data file upload does not allow the client to ...	**Resolved**	**06/20/2001**
Blocked by Metacat - Bug #194: manage files by accession# on file system	**Resolved**	**04/09/2001**

History

#1 - 07/19/2001 02:50 AM - Matt Jones

There are concomitant changes in Morpho to produce these types of file uploads. Morpho now uses the HTTPClient library for http communication, which allows streaming data between client and server (Sun's http protocol handler buffers all data before sending, which limits data transfer size to memory available). No changes in metacat were needed to accomodate this switch of protocol handlers on the client side; all changes were due to the need for an alternative data encoding (other than URL encoding) that efficiently supports binary files).

Modified Metacat to support large data file uploads. This is accomplished by supporting a new content type for data sent to metacat:
multipart/form-data
which allows multiple files to be sent in a standard MIME format. The MetacatServlet.handleGetOrPost() method now checks the incoming content type and sends multipart/form-data encoded submissions to MetacatServlet.handleMultipartForm(), which in turn delegates to MetacatServlet.handleUploadAction(). When a client wants to load a data file to metacat, now morpho creates a "multipart/form-data" encoded stream that contains an "action" parameter set to the value "upload", a "docid" parameter set to the value of the accession number for the file, and a "datafile" part that contains the actual file contents. Because of a bug / shortcoming in the com.oreilly.servlet library that I am using to parse this mime encoding, the "action" and "docid" paramters MUST come before the "datafile" part or metacat won't find the paramters.

The provided docid is inserted into xml_documents (after checking that it meets expectations [unique, rev 1, etc]). Then the data file recieved from the "upload" action is stored on the filesystem using the Accession number provided in the docid. See DocumentImpl.registerDocument for details.

This functionality completely replaces the socket-based functionality found in DataFileUploadInterface and DataFileServer. It would still be nice to abstract away the file storage functions so that they can be either written to the filesystem or other locations (like the db ina BLOB) when they are recieved in this new encoding format.

The cos.jar file is now needed for parsing the new encoding.

#2 - 07/19/2001 03:03 AM - Matt Jones

And also...

Because "multipart/form-data" is the standard format produced by web browsers
when doing a "file upload", this change makes it so that we can use a standard
web client to upload data to Metacat. Pretty cool!

#3 - 03/27/2013 02:13 PM - Redmine Admin

Original Bugzilla ID was 191