

Metacat - Bug #5543

Can't upload a FGDC metadata file with a data file in SANParks skin

11/16/2011 02:55 PM - Jing Tao

Status:	Resolved	Start date:	11/16/2011
Priority:	Normal	Due date:	
Assignee:	Jing Tao	% Done:	0%
Category:	metacat	Estimated time:	0.00 hour
Target version:	2.0.0	Spent time:	0.00 hour
Bugzilla-Id:	5543		
Description			
<p>I tried to use the data package upload form to upload a FGDC file, it succeeded. However, if i uploaded a FGDC file with a data file, it failed:</p>			
<pre>edu.ucsb.nceas.metacat.client.MetacatException: <?xml version="1.0"?> <error> Error: problem reading multipart data: Corrupt form data: premature ending </error> at edu.ucsb.nceas.metacat.client.MetacatClient.upload(MetacatClient.java:711) at edu.ucsb.nceas.metacat.clientview.ClientViewHelper.handlePackageUpload(ClientViewHelper.java:453) at edu.ucsb.nceas.metacat.clientview.ClientViewHelper.handleClientRequest(ClientViewHelper.java:270) at edu.ucsb.nceas.metacat.clientview.ClientViewHelper.clientRequest(ClientViewHelper.java:180) at org.apache.jsp.style.skins.sanparks.SaeonUpload_jspx._jspService(SaeonUpload_jspx.java:108) at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70) at javax.servlet.http.HttpServlet.service(HttpServlet.java:717) at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:386) at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:313) at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:260) at javax.servlet.http.HttpServlet.service(HttpServlet.java:717) at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290) at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206) at org.apache.catalina.core.ApplicationDispatcher.invoke(ApplicationDispatcher.java:646) at org.apache.catalina.core.ApplicationDispatcher.doInclude(ApplicationDispatcher.java:551) at org.apache.catalina.core.ApplicationDispatcher.include(ApplicationDispatcher.java:488) at org.apache.jasper.runtime.JspRuntimeLibrary.include(JspRuntimeLibrary.java:968) at org.apache.jsp.style.skins.sanparks.upload_jsp._jspService(upload_jsp.java:312) at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70) at javax.servlet.http.HttpServlet.service(HttpServlet.java:717) at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:386) at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:313) at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:260) at javax.servlet.http.HttpServlet.service(HttpServlet.java:717) at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290) at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206) at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:233) at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:191) at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:470) at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:127) at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102) at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109) at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298) at org.apache.jk.server.JkCoyoteHandler.invoke(JkCoyoteHandler.java:190) at org.apache.jk.common.HandlerRequest.invoke(HandlerRequest.java:291) at org.apache.jk.common.ChannelSocket.invoke(ChannelSocket.java:776) at org.apache.jk.common.ChannelSocket.processConnection(ChannelSocket.java:705) at org.apache.jk.common.ChannelSocket\$SocketConnection.runIt(ChannelSocket.java:898) at org.apache.tomcat.util.threads.ThreadPool\$ControlRunnable.run(ThreadPool.java:690)</pre>			

History

#1 - 11/18/2011 05:58 PM - Jing Tao

After dug around the code, i figured out the problem is at the CoientViewHelper.java.

In handlePackageUpload method, this line will upload data:

```
getMetacatClient().upload(newDocId, fName, inputStream, Integer.MAX_VALUE);
```

It uses Integer.MAX_VALUE as the file size rather than the real file size. It worked. But now it causes the issue. I believe the change in MetacatClient client cause this.

In order to get the real file size, i tried to use the getFilePath method in FilePart class. Then to create a file object base on the file path and get the size from the file object. However, this method doesn't get the path, just get the file name. So i got java.io.FileNotFoundException.

Another way is to serialize the inputStream from FilePart object to a temporary file, then get file size from there. But I don't like this approach.

Do you guys have any suggestions?

#2 - 11/18/2011 07:51 PM - Matt Jones

Jing --

I'm not sure what the right solution is here, but here are some thoughts. First, trying to get the length of an incoming http stream is by definition difficult, but in a POST operation there should be a Content-length header that tells the length of the post body. This is not the length that you need, as you only need one part from the mime message, but its bigger than what you need. Second, the COS Mime multipart library is incredibly old, and we should seriously consider replacing it. On the client side for DataONE, we've switched to using the nicely maintained Apache HttpComponents implementation, particularly the MIME MultipartEntity class for representing our mime messages (<http://hc.apache.org/httpcomponents-client-ga/httpmime/apidocs/org/apache/http/entity/mime/MultipartEntity.html>). This provides a getContentLength() method that may provide the length you need. Another alternative is the javax.mail.MimeMultipart class, which can parse a data source and extract BodyParts form the message, which have a getSize() method. In both cases, the size methods may return a negative number if the stream has not been fully read or for some reason they can't figure out the length. In processing multipart messages, one can't really know the length of the parts until they've been read and mime-decoded.

So, in your case, you really need to know what ClientViewHelper is doing. If its creating a mime form encoded request, I think these approaches could work for you. If its parsing a request, they might too. Ben and Chris have been using this stuff in DataONE a bunch so you might run your question by them.

#3 - 11/21/2011 11:33 AM - ben leinfelder

We may not even have to require a known size -- Metacat isn't storing that information. Sure, the new DataONE apis use it, but this is a legacy Metacat API we are talking about. There's possible fix committed to MetacatClient to allow unknown content length. We'll see!

#4 - 11/21/2011 02:20 PM - Jing Tao

Ben committed a new version of cos.jar file and modified the MetacatClient.java to handle the file size being -1. In ClientViewHelper class, I set the file size -1. Then it works.

I am going to open a new bug to remind us to replace cos.jar file.

#5 - 03/27/2013 02:30 PM - Redmine Admin

Original Bugzilla ID was 5543