# MS2 Notification APIs

### LabKey Server Version 11.1

Draft March 10, 2011

LabKey Server version 11.1 includes two server APIs and associated java-language wrappers that support automatic processing of MS spectra files as they are produced by the instrument, without operator intervention. This document describes their configuration and use.

## Overview

The LabKey Server Enterprise Pipeline is designed to be used in a shared file system configuration with the MS instrument. In this configuration data files are copied from the instrument to a directory shared with the LabKey Server and with its remote task runners. From the LabKey Server's perspective, this directory lives under the Pipeline root directory for a given folder. Once the raw data files are copied, the Pipeline web part can be used to manually select a search protocol and initiate search processing. Alternately, these notification APIs can be called by a batch processing step *after* the copy to the shared pipeline directory is complete.

* The **StartSearchCommand** initiates MS2 searching on one or more specified data files using a named, pre-configured search protocol. If a data file is not found in the specified location at the time this command is called, the search job will still be initiated and will enter a "File Waiting" status.
* The **FileNotificationCommand** tells LabKey Server to check for any jobs in a given folder that are in the File Waiting status. A File Waiting status is cleared if the specified file being waited for is found in the expected pipeline location. If the waited-for file is not present the File Waiting status remains until it is checked again the next time a FileNotificationCommand is called on that folder.

In addition, LabKey Server includes two wrapper classes to make these APIs easier to call from a batch file:

* The **MS2SearchClient** class takes data file and protocol information from a CSV file and uses it to call StartSearchCommand one or more times. The CSV file contents are also saved at the server using the SubmitAssayBatches API. MS2SearchClient is designed to be called in a batch file.
* The **PipelineFileAvailableClient** is a simple wrapper over FileNotificationCommand to enable calling from a batch file.

LabKey Server does not try to detect partially-copied files, so these APIs should be called at a time when there are no file copies in progress.

NOTE: As if the beta release, please keep your input mzXML or RAW file names to less than 50 characters. There are problems running some steps of the search processing if the original file length is too long. We are investigating some way to relax this limit.

## Using MS2SearchClient

MS2SearchClient is called from a batch file after copy operations are complete. It uses a config.properties and a .csv file as inputs and assumes the search protocol, search database, and assay definition have already been configured in the target folder on LabKey Server.

### Create an MS2 folder on LabKey Server

In the example, the MS2 folder name is **mstest**, and it is a top-level folder (i.e. a project). After creating the folder, go to ManageProject->Folder settings and check the box next to Study module, then press Update Settings. (This is needed for the Assay web parts).

### Configure the named search protocol

There are two ways to create a named search protocol:

1. Configure and run an actual search from the pipeline manager on an example .raw or .mzXML file. If you have submitted searches manually, you have already done this. The definition of a named search is written to a file under the pipeline root:  
     
   <pipeline root>/.labkey/protocols/<search engine>/<protocol name>.xml
2. Write the xml protocol definition file directly to the path specified above. This is useful if you have an established protocol in another folder or LabKey Server, and you want to use this file unchanged in a new folder with a new pipeline root. If you make a change in a protocol file (or use it with a different default.xml file) you should save that protocol under a different file name.

In the example files shown below, the protocol name is **Srch1**. Srch1 refers to a FASTA file that is accessible to the Pipeline Manager, but the name and location of this file does not affect the usage of the notification APIs.

### Create a CSV file

MS2SearchClient takes as an input argument a comma-separated value (CSV) text file whose first row contains the following property names:

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Description** | **Example Value** |
| FileName | the name of the .raw or .mzXML file to be processed | File11.mzXML |
| Path | the absolute path to folder where the .raw or .mzXML file to be processed has been copied. Use forward slashes. | C:/lktrunk/build/ deploy/files/ mstest/@files/test1/ |
| Sample | the sample identifier string of the physical sample put into the mass spec instrument from which the data file was produced. | sample11 |
| LabKeyFolder | the folder on LabKey Server where the run will be loaded | mstest |
| ProtocolToRun | the named search protocol discussed in the previous section | Srch1 |
| <UserProp1>, …<UserPropN> | any user-defined property name that matches a run property name in the Assay definition (next section) | abc |

Example CSV file:

CSVTest.csv

FileName,Path,Sample,LabKeyFolder,ProtocolToRun,UserProp1

File11.mzXML,C:/lktrunk/build/deploy/files/mstest/@files/test1/,sample11,mstest,Srch1,abc

### Configure the general-purpose Assay definition for run properties

The CSV file is interpreted as the instructions to initiate one or more search jobs. The contents of the CSV are saved into a general-purpose assay on LabKey Server. To create this assay definition,

1. Add an Assay List web part to the folder dashboard.
2. In this web part, select New Assay Design
   1. Type General
   2. Name TestX
   3. Delete the 2 pre-defined batch properties
   4. Run Properties: Press Infer Fields from file
   5. Choose CSV file created in previous step
   6. Delete the 4 pre-defined Data properties
   7. Save and Finish
3. Back on the dashboard, hover over the TextX link in the Assay List web part. Look at the status bar (lower left of browser) to find the rowId of the TestX assay. For example it may show   
     
   http://localhost:8080/labkey/assay/mstest/begin.view?rowId=86  
     
   In this case the AssayId would be 86. This will be used in the config.properties file.
4. For convenience you may want to add an Assay Batches web part on the dashboard, showing batches of TestX assays.

### Write the config.properties file

MS2SearchClient looks for a config.properties file in the current directory by default; it can be pointed to this file in a different directory by a command line parameter. The config file contains property and value pairs in the format

<property name 1>: <value>

<property name 2>: <value>

MS2SearchClient expects to find the following properties in the config.properties file:

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Description** | **Example Value** |
| baseServerURL | address for connecting to LabKey Server | http://localhost:8080/labkey |
| username | Sign-in name for the LabKey Server, an email address | testuser@labkey.com |
| password | Sign-in password for the LabKey Server | 123my$password |
| pipelineRoot | The path to the pipeline root for the target folder in which the MS2 runs will be processed. Use forward slashes. This value will be compared with the Path value in the CSV file to calculate a relative path to the data file | C:/lktrunk/build/ deploy/files/ mstest/@files/ |
| assayId | The rowId of the assay instance that will store the properties in the CSV file (See above for how to find this number) | 86 |
| searchEngine | The name of the MS2 search engine to use. Supported values are XTandem, Sequest, and Mascot. | Sequest |
| debug | (Optional) If true, print verbose HTTP connection debugging information | false |

If other properties are present in the config.properties file they will be ignored. The config.properties file should be written to the same directory as the batch file that invokes MS2SearchClient. If the config.properties is located in a different directory, it can be used by passing the optional parameter [-config=<path to config.properties>] as described in the next section,

### Create a batch file to call MS2SearchClient

The command line to invoke the MS2SearchClient is as follows

java -Done-jar.main.class=org.labkey.remoteapi.test.MS2SearchClient   
 -jar labkey-client-api-11.1Dev-complete.jar   
 <path to CSV file> [<path to CSV file 2>,…]

[-config=<path to config.properties>]

The only required user-specified parameter in this command is the <path to the CSV file>, which specifies the complete path to the file using the same conventions as the pipelineRoot property of the config.properties file and the path property in the CSV file.

Here is an example:

C:/lktrunk/build/deploy/files/mstest/@files/test1/CSVTest.csv

Note also that MS2SearchClient is packaged in the jar named

**labkey-client-api-11.1Dev-complete.jar**

The java command line expects to find that jar in the current directory or a directory on the classpath.

A typical batch file that contains the call to MS2SearchClient would first perform a copy operation of the CSV file to the path specified on the command line. Combining these two steps in a single batch file would look like this:

**startMS2Search.bat**

copy CSVtest.csv C:/lktrunk/build/deploy/files/mstest/@files/test1/CSVTest.csv

java -Done-jar.main.class=org.labkey.remoteapi.test.MS2SearchClient   
 -jar labkey-client-api-11.1Dev-complete.jar   
 C:/lktrunk/build/deploy/files/mstest/@files/test1/CSVTest.csv

## Using PipelineFileAvailableClient

**Config.properties**

The second wrapper function has many fewer options. It needs a config.properties file that is almost identical to the config.properties used by MSSearchClient. In fact the two utilities can share the same config.properties file by adding just the "folder" property to the set of properties described above. The assayed and searchEngine properties are only used by MSSearchClient and ignored by PipelineFileAvailableClient

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Description** | **Example Value** |
| baseServerURL | Same description and example values as shown under MSSearchClient | |
| username |
| password |
| debug |
| folder | Path of the LabKey Server container to which the notification should be directed | mstest |

**Command line**

PipelienFileAvailableClient needs no user-specified parameters. There is one optional parameter to point to the config.properties file, needed only if the config file is not in the current directory To invoke PipelienFileAvailableClient in a batch file, use the following:

java -Done-jar.main.class=org.labkey.remoteapi.test.PipelineFileAvailableClient   
 -jar labkey-client-api-11.1Dev-complete.jar

[-config=<path to config.properties>]

## Automation scenario

Suppose the first .raw file (and the .CSV file are available from the mass spectrometer at the same time. Another .raw file is expected to be written out by the mass spectrometer within the hour. Both of these files are included in the .CSV file.

The sequence of batch commands might look something like this:

copy file11.RAW C:\lktrunk\build\deploy\files\mstest\@files\test1\file11.RAW

copy CSVtest.csv C:\lktrunk\build\deploy\files\mstest\@files\test1\CSVTest.csv

java -Done-jar.main.class=org.labkey.remoteapi.test.MS2SearchClient   
 -jar labkey-client-api-11.1Dev-complete.jar   
 C:/lktrunk/build/deploy/files/mstest/@files/test1/CSVTest.csv

At this point the job for file11.RAW should be running. The job was started using the information in the CSVTest.csv file. The second search job, also specified in the .CSV file, is in the "Wait for Files" state, waiting for file12.raw to be available.

When the second file is ready, it is copied to the pipeline directory:

copy file12.RAW C:\lktrunk\build\deploy\files\mstest\@files\test1\file12.RAW

Running the following tells LabKey Server to check for files that are pending.

java -Done-jar.main.class=org.labkey.remoteapi.test.PipelineFileAvailableClient   
 -jar labkey-client-api-11.1Dev-complete.jar

[-config=<path to config.properties>]

### Cleanup LabKey folder to rerun tests

If you want rerun analyses using the same file names in the same folder more than once , you need to clean up all of the artifacts generated by running the notify commands. These artifacts are:

1. The assay batch(es) representing the saved properties from the CSV. These can be found and deleted from either the Assay List or Assay Batches web part, by selecting and clicking Delete.
2. The MS2 Experiment run(s) created by the search process. These can be deleted by selecting them and clicking Delete in the MS2 Experiment Runs grid on the folder dashboard.
3. The data files containing the results of each of the steps of the search process. These can be found from the Pipeline Manager "Process and Import Data" view. Find the .mzXML file(s) that were the basis for the MS2 Experiment runs you deleted. In this directory you should see a subfolder with the name of the search engine, such as "sequest". Expanding that folder, you should see a folder named by the MS2 search protocol name, for example "Search1", Delete this folder and its contents.
4. Return to the dashboard, and select the "Show: All" for pipeline jobs. Select the Completed jobs representing the history of the analysis done previously, and press delete.