Commands to batch process MODS and DC records in the RO@M repository

MRB
Thu 02-Nov-2017

1. Commands to batch extract MODS records from the repository

- cURL command to output the PIDs for all Solr documents that have the Solr field mods_titleInfo_title_ms and thus have MODS datastreams (i.e., will retrieve PIDs for all Fedora objects in RO@M with MODS datastreams)

    -
      ```
      curl -s "http://roam.macewan.ca:8080/solr/select/?q=mods_titleInfo_title_ms%3A*&version=2.2&start=0&rows=999999&indent=on&fl=PID" | grep "str name=\"PID\"" | sed "s/<str name=\"PID\">//g;s/<\/str>//g" | sed -e 's/^[ \t]*//;s/[ \t]*$//' | sed "s/<\/doc>//g" | sort | tee solr_pids.txt
      ```
- Unix one-liner command that will retrieve the MODS datastreams from a list of PIDs for Fedora objects, and rename the files by replacing the colon with an underscore after the namespace and adding an underscore followed by the datastream ID after the PID (i.e., adding "_MODS" after the PID); note: this renaming is so the filenames are formulated properly so the Drush push datastream script will replace each MODS datastream in the correct Fedora object

    -
      ```
      for PID in `cat solr_pids.txt`; do curl http://roam.macewan.ca/islandora/object/$PID/datastream/MODS/download > $PID"_MODS.xml"; mv $PID"_MODS.xml" `echo $PID"_MODS.xml" | sed s/:/_/`; done
      ```

2. Commands to batch cleanup MODS and DC records

- Cleanup operations on the exported MODS and DC records consist of three phases:
    - (1) run the XSLT fix_mods.xsl stylesheet to fix the validation errors as well as structural/ conceptual errors in the MODS records (XSLT 2.0 stylesheet so need to use Saxon-HE)
        - Saxon-HE command

            -
              ```
              for FILE in *; do java -Xmx1024m -jar "/home/brundin/saxon9he.jar" $FILE ../transforms/fix_mods.xsl > ../mods-fix/$FILE; done
              ```
    - (2) run the XSLT cleanup_mods.xsl stylesheet to remove all of the empty elements and empty attributes in the MODS records (XSLT 1.0 stylesheet so can use xsltproc – need to remove kludge introduced in XSLT to address xalan-java bug, otherwise use xalan-java instead of xsltproc [if using xsltproc, delete the “doctype-public="yes"” attribute from the <xsl:output> element in the cleanup_mods.xsl XSLT stylesheet])
        - xsltproc command

            -
              ```
              for FILE in *; do xsltproc ../transforms/cleanup_mods.xsl $FILE > ../mods-rev/$FILE; done
              ```
        - xalan command

            -
              ```
              for FILE in *; do java -jar xalan.jar -IN $FILE -XSL ../transforms/cleanup_mods.xsl > ../mods-rev/$FILE; done
              ```
    - (3) run the XSLT mods_to_dc.xsl stylesheet to create full bibliographic DC records, i.e., right now the DC records just have dc:title (from the Fedora label) and dc:identifier (from the Fedora object’s PID), and this transformation will provide various mappings from MODS elements and attributes to the relevant DC elements (XSLT 1.0 stylesheet so can use xsltproc)

- xsltproc command

    -
      ```
      for FILE in *; do xsltproc ../transforms/mods_to_dc.xsl $FILE > ../dc-rev/$FILE; done
      ```
- need to rename the DC files in the "dc-rev" directory such that the substring "_MODS" is changed to "_DC"; Unix command to rename the files

    -
      ```
      ls -1 | while read file; do new_file=$(echo $file | sed s/\_MODS/_DC/g); mv "$file" "$new_file"; done
      ```

3. Commands to batch validate MODS and DC records

- xmllint command to batch validate a directory of MODS records

    -
      ```
      xmllint --noout --schema http://www.loc.gov/standards/mods/v3/mods.xsd *_MODS.xml 2>&1 | tee xmllint_mods.txt
      ```
- xmllint command to batch validate a directory of DC records

    -
      ```
      xmllint --noout --schema http://www.openarchives.org/OAI/2.0/oai_dc.xsd *_DC.xml 2>&1 | tee xmllint_dc.txt
      ```

4. Commands to batch format and indent (pretty print) MODS and DC records

- xmllint command to format and indent (pretty print) a directory of MODS records

    -
      ```
      for FILE in *; do XMLLINT_INDENT='    ' xmllint --format --encode utf-8 $FILE > ../mods-new/$FILE; done
      ```
- xmllint command to format and indent (pretty print) a directory of DC records

    -
      ```
      for FILE in *; do XMLLINT_INDENT='    ' xmllint --format --encode utf-8 $FILE > ../dc-new/$FILE; done
      ```

5. Commands to batch reingest and replace the MODS and DC datastreams

- Upload MODS and DC records into two separate directories on the roam.macewan.ca server
    - /home/brundin/roam/mods
    - /home/brundin/roam/dc
- Ensure MODS and DC record filenames are correctly formulated for the Drush script, which they should be; the formulation is namespace_rest-of-PID_DATASTREAM-ID.xml, e.g., gm_149_MODS.xml and gm_149_DC.xml
- Make directory "logs" as a subdirectory in the directory "roam"

    -
      ```
      mkdir /home/brundin/roam/logs
      ```
- Need to run the Drush command from the root of the drupal7 directory tree, i.e.,
    - /var/www/html/drupal7
- To push the MODS files to the correct Fedora objects, run the following Drush command:

    -
      ```
      drush islandora_datastream_crud_push_datastreams --user=brundin --datastreams_source_directory=/home/brundin/roam/mods --datastreams_mimetype=application/xml --datastreams_label="MODS Record" --datastreams_crud_log=/home/brundin/roam/logs/crud-push-mods.log
      ```
- To push the DC files to the correct Fedora objects, run the following Drush command:

    -
      ```
      drush islandora_datastream_crud_push_datastreams --user=brundin --datastreams_source_directory=/home/brundin/roam/dc --datastreams_mimetype=application/xml --datastreams_label="DC Record" --datastreams_crud_log=/home/brundin/roam/logs/crud-push-dc.log
      ```