Batch Parsing XML Metadata for Cataloging GIS Data

Jeff Essic

Data Services Librarian

North Carolina State University

jeff_essic@ncsu.edu

2005 ESRI Education Users Conference



Problem

Need to create *GIS Lookup* Cataloging Records for multiple data layers contained within various data collections in a time efficient manner.

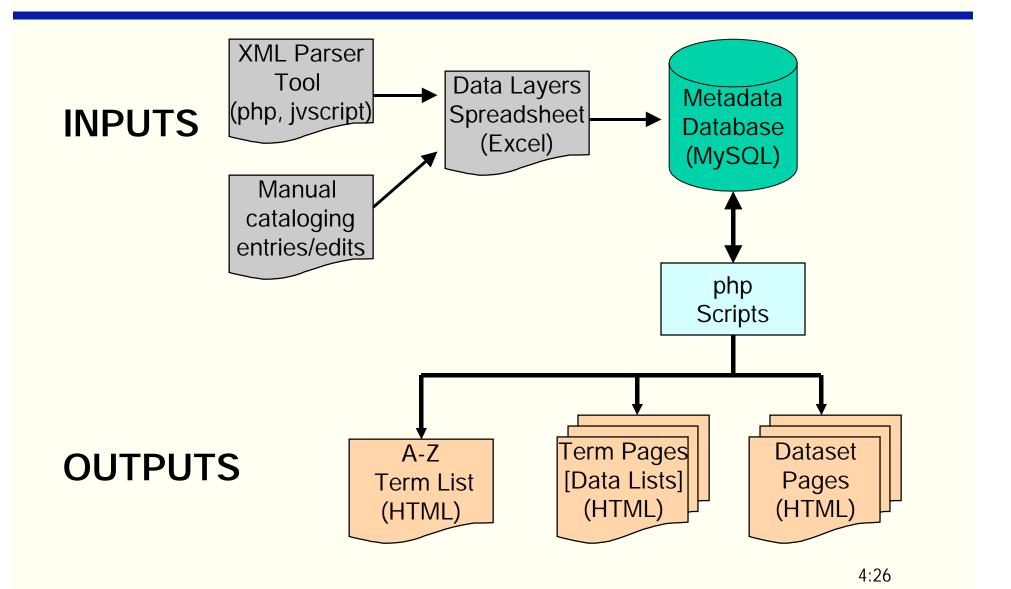
What is *GIS Lookup*?

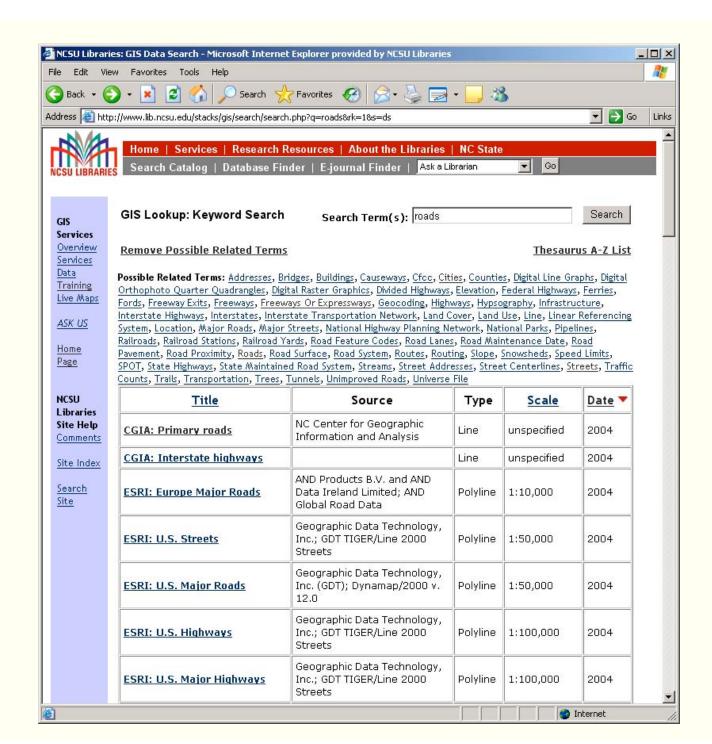
- Database of cataloged geospatial data layers
- Allows keyword and thematic searching
- Provides data layer identification, access instructions, related terms, description, source info, coordinate system info, etc.
- Links to data download, metadata, collection web pages, related data, web mapping, other docs.
- As of May 2005, contains 2027 records.

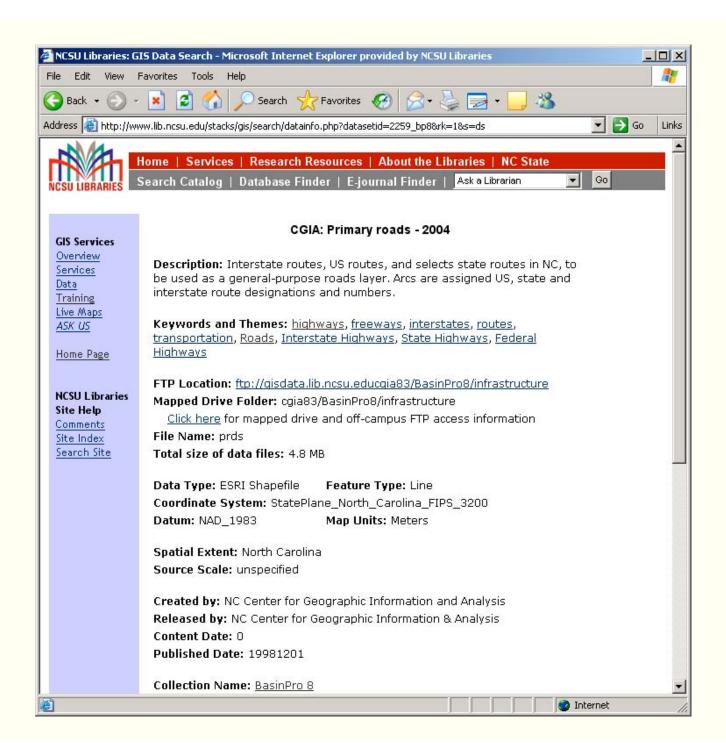
Note: The following screenshots were taken prior to implementing NCSU Libraries' 2005 website redesign.

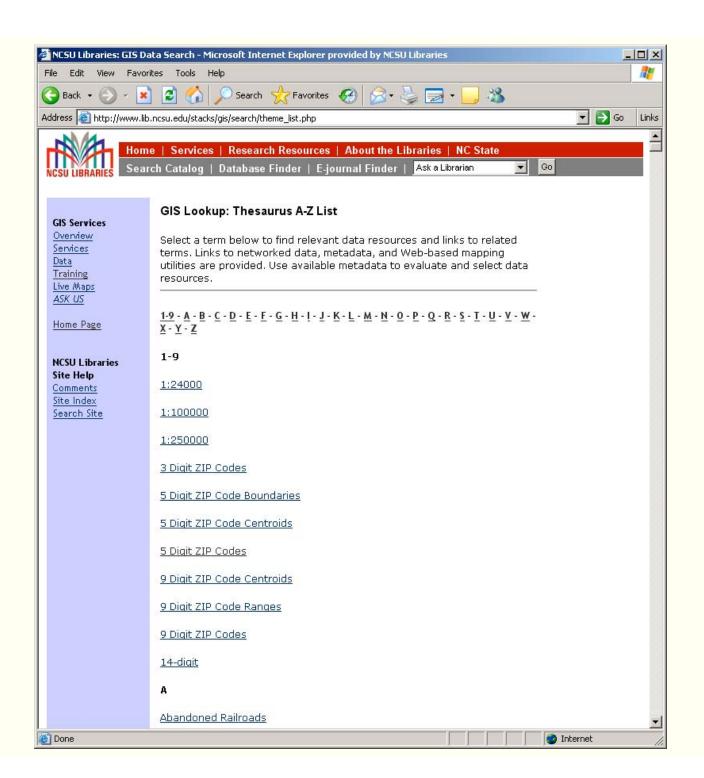
3:26

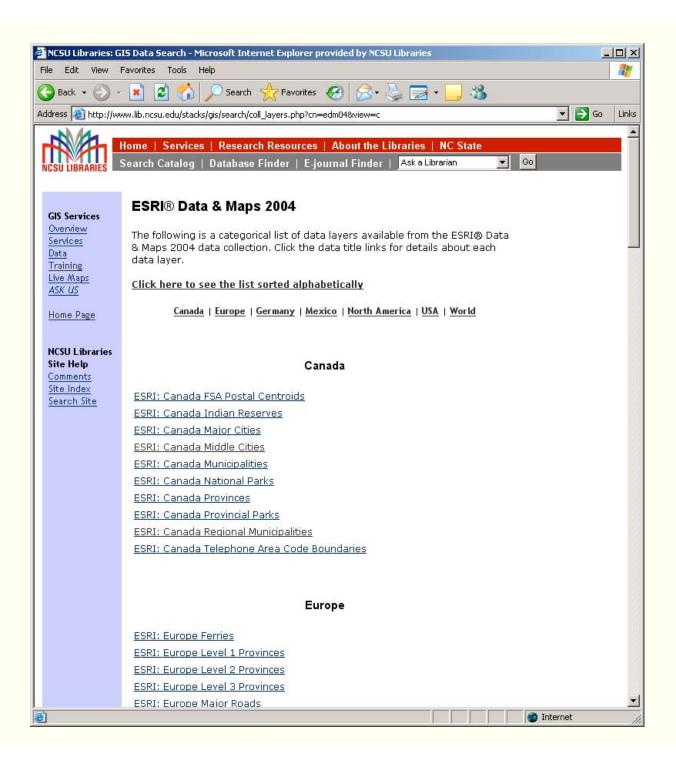
GIS Lookup Architecture

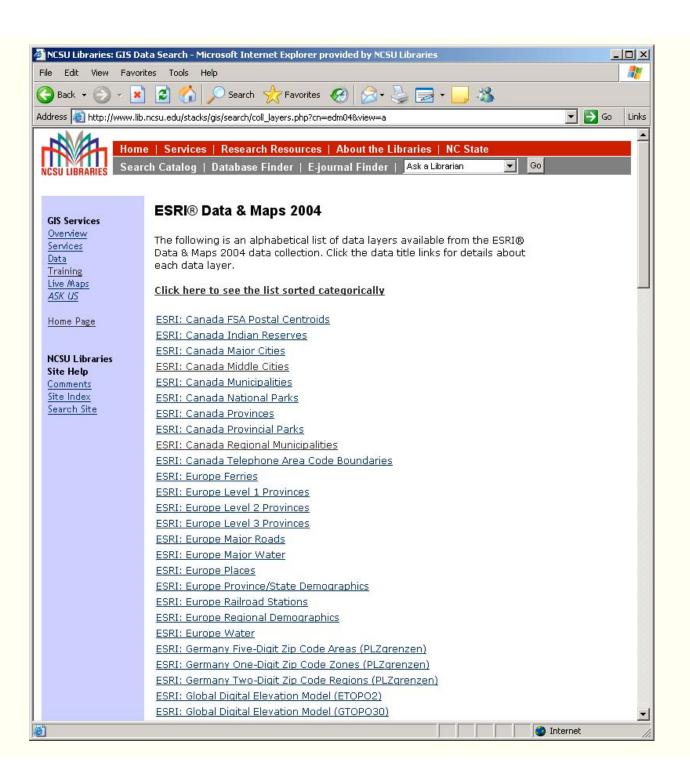












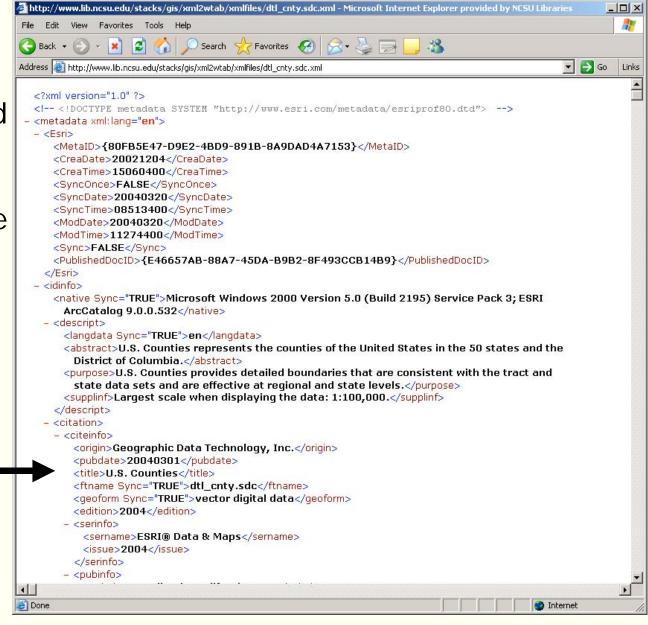
Solution

- Parse XML metadata files for element content needed for cataloging
 - Geospatial data are increasingly accompanied by XML metadata files, usually in FGDC format.
 - XML advantage: automated extraction of metadata elements needed for cataloging
 - Batch processes can very quickly extract elements from a large set of XML files
 - Extraction process must remain independent of any metadata structure standard

Goal:

Develop an automated process for harvesting selected metadata elements from multiple XML files.

Data Layer Title



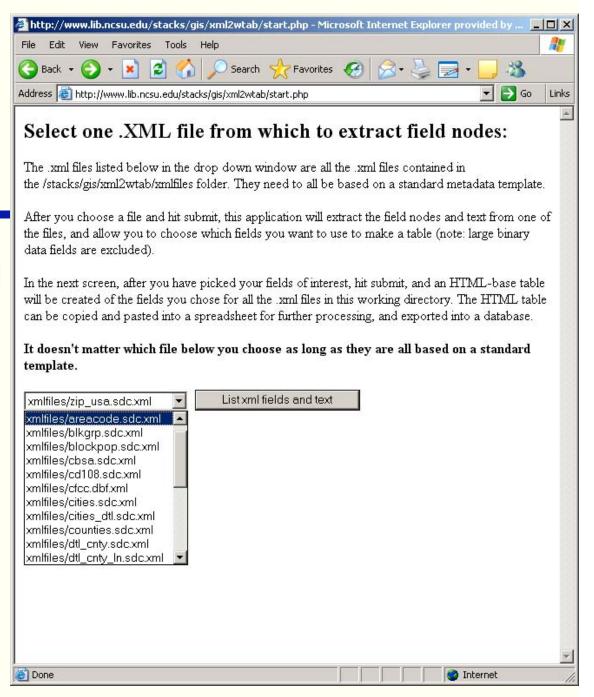
Existing Tools

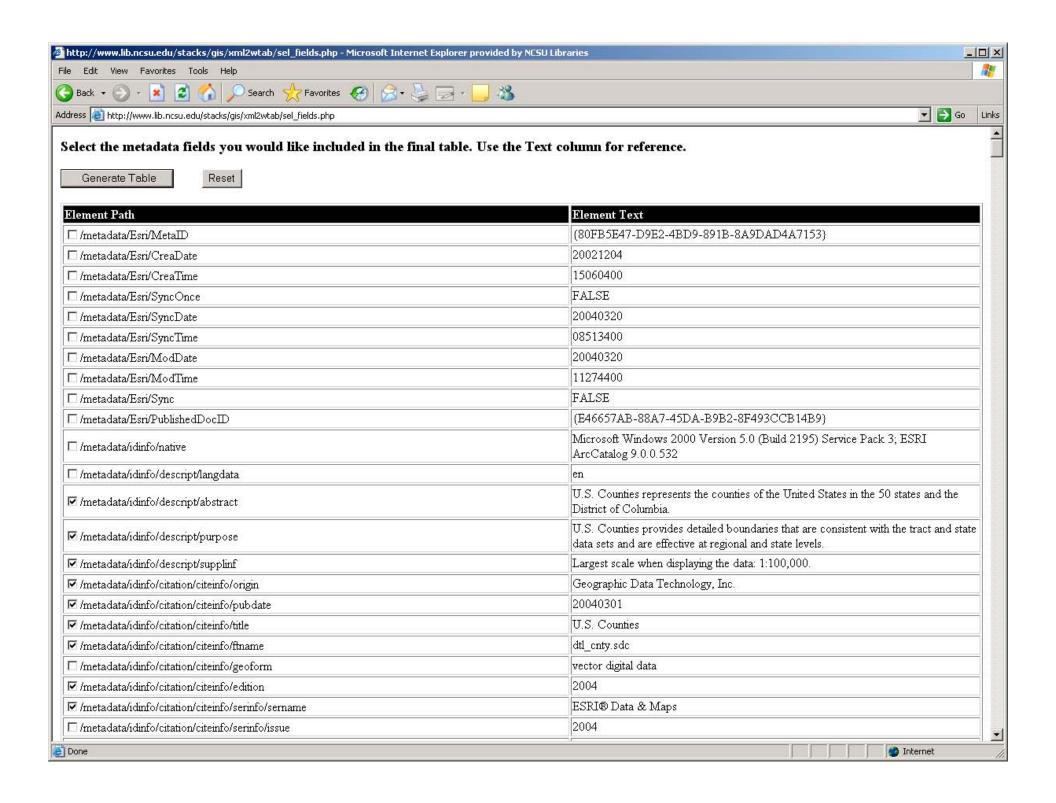
- Internet search for software tools, scripts, utilities, etc. did not yield any turnkey solutions.
 - Majority of tools were designed for XML file creation or did not offer structure independence flexibility.
 - However, php and javascript provide robust capabilities for interacting with XML

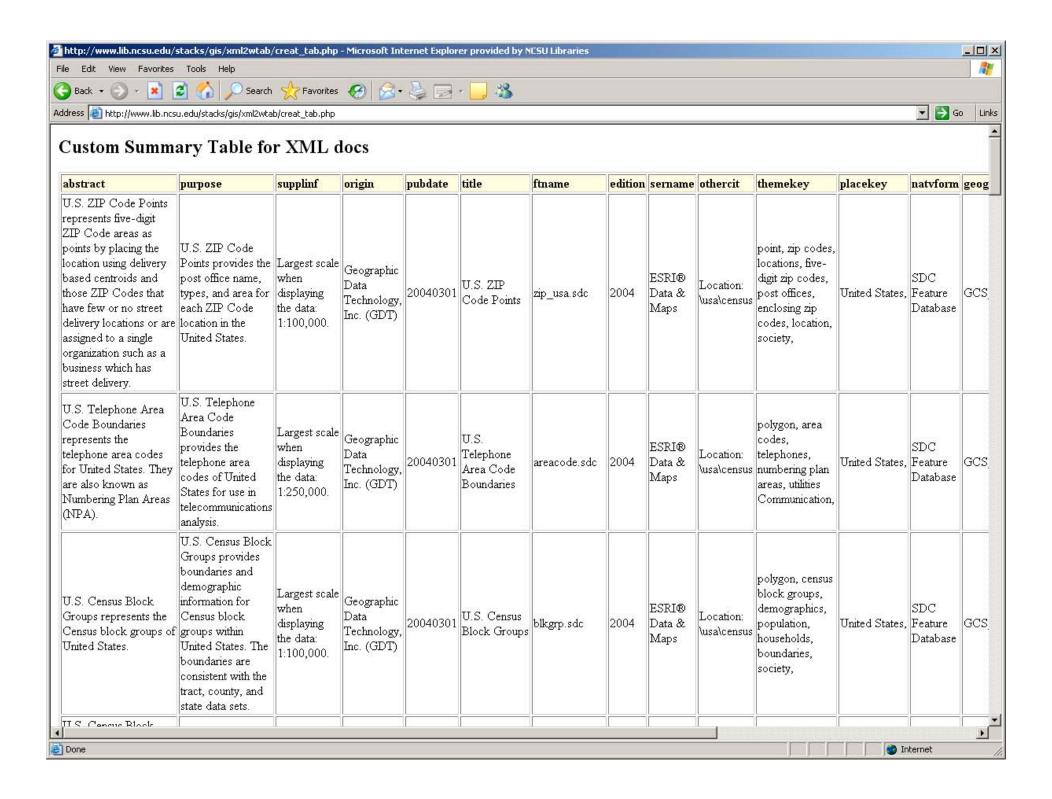
XML Parser Tool: Client Interface

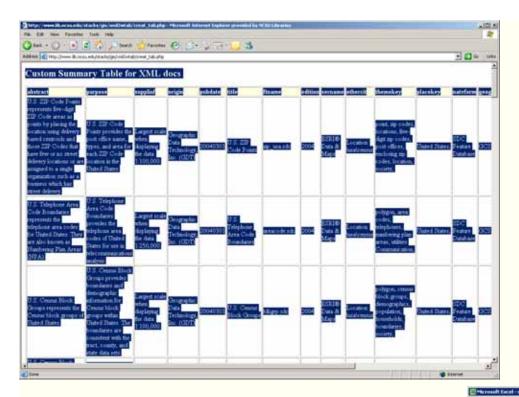
Files to parse are located in the /xmlfiles folder.

Choose any file from the drop-down menu in order to list the fields and sample text.



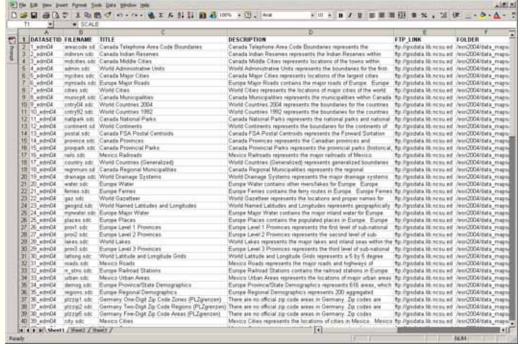


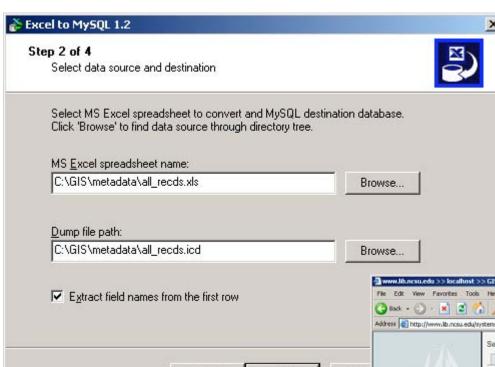




"Select All" of the output HTML table and copy it to the system clipboard...

...paste contents into a spreadsheet document and clean up as necessary.



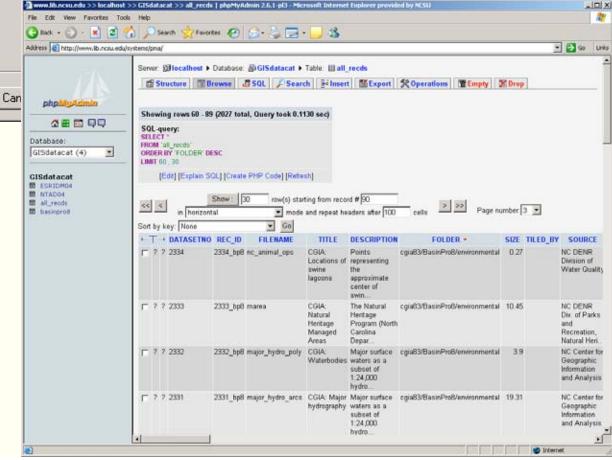


Next >

Convert data layer records in spreadsheet document into SQL INSERT statements for database...

...Import INSERT statements into database, and perform final edits if needed.

< Back



How it works – Script Backend

- Six Files, ~ 10 KB
- config.php:
 - Specifies the /xmlfiles folder as where the XML files are located.
 - Specifies the file type to parse (XML)
- start.php:
 - Gives basic instructions
 - Reads the list of XML files to a dropdown list
 - Users selects one of the files to use for listing metadata fields and content

sel_fields.php

- Calls a javascript to load the selected XML file using Microsoft.XMLDOM
- Calls a javascript which loads xslstyle_x2w.xsl
- Calls a javascript to display the results of xslstyle_x2w.xsl

xslstyle_x2w.xsl

- xsl template that generates a checkbox, element pathname, and element content for each metadata field in the selected XML file.
- Since XML files may contain binary data (e.g. preview graphics), the xsl does not select nodes with > 10,000 characters
- Passes the selected element paths to creat_tab.php

xsl_tabstyle.xsl

- Output table stylesheet
- Dynamically written by creat_tab.php
- Instructs output table to contain the value of each selected element
- Is executed for each XML file by creat_tab.php
- Must be a pre-existing file (can be zero length) with write permissions. Content is overwritten whenever creat_tab.php is executed.

creat_tab.php

- Generates the output table stylesheet template and writes it on-the-fly to xsl_tabstyle.xsl
- If multiple "keyword" fields exist, concatenates them into one comma separated field
- Creates output table headers and structure
- Loops through all XML files in the xmlfolder, calling a javascript to extract data from the selected elements, as specified by the xsl_tabstyle.xsl template
- Outputs element names and content for all XML files in an HTML table

Time Efficiency

 Cataloging NC BasinPro 8 took a span of about 5 weeks for approximately 80 data records.

 Cataloging ESRI Data and Maps 2004, which has about 130 records, took 3 days.

Limitations

- Limited by XML metadata content, formation
- Designed and tested to work only in Internet Explorer 6 - javascript relies on an IE ActiveX object call to work. Does not work in Firefox.

Future Applications

- Use for batch parsing any set of XML files
- Will be exploring using this tool for extracting layer information from WMS / WFS capability files

Code Distribution

Issues pertaining to distribution of the code have not been decided as of June 3, 2005, the date of submission of this presentation to ESRI. Contact jeff_essic@ncsu.edu for later status information.