SPASE Toolkit
Version 2.0.0

Table of Contents

Overview

The SPASE (Space Physics Archive Search and Extract) Toolkit contains a set of command-line applications which can be used to generate, validate, referentially check, use and organize resource descriptions written in SPASE XML. The toolkit is written in Java.

Installation

The SPASE toolkit is written entirely in Java and should run on any system with a Java Runtime Environment (JRE). Any required extensions are included in the SPASE toolkit installation package. The toolkit is packaged as a self-contained executable "jar" file.

System Requirements

- Java JRE 1.5 or higher

Installation

- Download the SPASE toolkit jar file from the SPASE web site.
  The most recent release can be found at:

```
http://www.spase-group.org/tools/toolkit/spase-tools-2.0.0.jar
```

- Place the jar file in any directory that you want.

Tools

All tools are written in Java and are part of the "org.spase.tools" package. Each tool has a common name which can be used with the executable jar package. A tool can run with a command like:

```
[node] java -jar spase-tools-2.0.0.jar [toolname] [tooloptions]
```

Each tool also has a Java class which can be called by other programs. Tool class names are in proper case.

Collator

Separate each SPASE resource description in a file into a separate file stored in a folder tree according to the Resource ID.

Optionally recursively scan a directory for all files with a given extension
and process each file.

Usage:
```
collator [options] file
        -or-
java org.spase.tools.Collator [options] file
```

Options:

-b,--base <arg>	Base path for collated output (default: .).
-h,--help	Dispay this text
-k,--check	Check files, but do not write collated output.
-r,--recurse	Recursively process all files starting at path.
-v,--verbose	Verbose. Show status at each step
-x,--ext <arg>	File name extension for filtering files when processing folders (default: .xml)

Acknowledgements
    Development funded by NASA's VMO project at UCLA.

Example
 Suppose a file (example.xml) contains three SPASE descriptions like:

```
<Spase>
   <Version>2.0.0</Version>
   <NumericalData>
      <ResourceID>spase://VMO/NumericalData/GeoTail/LEP/PT60S</ResourceID>
         ... Details omitted ...
   </NumericalData>
   <NumericalData>
      <ResourceID>spase://VMO/NumericalData/GeoTail/MGF/PT60S</ResourceID>
         ... Details omitted ...
   </NumericalData>
   <NumericalData>
      <ResourceID>spase://VMO/NumericalData/GeoTail/CPI/PT60S</ResourceID>
         ... Details omitted ...
   </NumericalData>
</Spase>
```

Then a command like:

```
collator example.zip
```

will generate three files with the path names of:

```
./VMO/NumericalData/GeoTail/LEP/PT60S.xml
./VMO/NumericalData/GeoTail/MGF/PT60S.xml
./VMO/NumericalData/GeoTail/CPI/PT60S.xml
```

With each file containing the description for the corresponding resource.

Downloader

Obtains a list of URLs associated with a resource by querying a registry server, then downloads and packages all the source files. The collection of files is packaged into a zip file and written to the output file.

Usage:

      org.spase.tools.Downloader [options] id

Options:

-b, --startdate	The start date of the desired time span.
-e, --stopdate	The stop date of the desired time span.
-h,--help	Display this text
-c, --check <arg>	Check a download package.
-o,--output <arg>	Output filename (default: resource.zip).
-s,--service <arg>	The URL to the registry service which generates download packages (default: http://www.spase-group.org/registry/downloader).
-v,--verbose	Verbose. Show status at each step

Acknowledgements

Example
To download all Granules associated with the resource with the SPASE ResourceID of:

```
spase://VMO/NumericalData/AMPTE_UKS/Plasma/SWI_PT5S
```

use the command:

```
downloader -o example.zip spase://VMO/NumericalData/AMPTE_UKS/Plasma/SWI_PT5S
```

which will collect all data files (granules) associated with the resource, package them in a zip file and write the file to "example.zip".

To download only granules that span a time range use a command like:

```
downloader -o example.zip spase://VMO/NumericalData/AMPTE_UKS/Plasma/SWI_PT5S \
        -b 1990-12-01T00:00:00 -e 2000-01-01T00:00:00
```

which will collect only those data files (granules) which contain data between 1990-12-01T00:00:00 and 2000-01-01T00:00:00.

To check a package to make sure it is complete and intact:

```
downloader -c example.zip
```

Profiler

Profile generator. Create resource profiles for SPASE resource descriptions.
Profiles all have a common schema which can be used in a solr search engine.

Usage:
    java org.spase.tools.Profiler [options] [file...]

Options:

-f,--file <arg>	File. File containing a list of file names to scan.
-h,--help	Display this text
-i,--id <arg>	ID. The registry ID to set for each resource
-l,--lookup <arg>	Lookup. THe URL to the resource lookup service to resolve resource IDs. Default: http://www.spase-group.org/registry/
-o,--output <arg>	Output. Output generated profiles to {file}. Default: System.out.
-r,--recurse	Recurse. Process all items in the current folder. Recurse into sub-folders.
-v,--verbose	Verbose. Show status at each step.
-x,--extension <arg>	Extension. The file name extension for files to process (default: .xml)

Acknowledgements:
    Development funded by NASA's VMO project at UCLA.

Example
To create profiles for all resources in the current directory and below use the command:

```
profiler -o /temp/vmo.xml -r *
```

The profiles will be written to the file "/temp/vmo.xml". The profiles can then be posted the appropriate solr search engine.

The expected schema for the solr index is:

Field Name	Type	Indexed	Stored	Required	Multi-valued
resourceid	string	true	true	true	
registryid	string	true	true	true	
resourcetype	string	true	true		
resourcename	text	true	true		
measurementtype	text	true	true		
phenomenontype	text	true	true		
observedregion	text	true	true		
observatoryid	string	true	true		
observatoryname	text	true	true		
observatorytype	text	true	true		
observatorygroup	text	true	true	true	
instrumentid	string	true	true		
instrumentname	text	true	true		
instrumenttype	text	true	true		
releasedate	date	true	true		
startdate	date	true	true		
stopdate	date	true	true"		
cadence	string	true	true"		
latitude	string	true	true		
longitude	string	true	true"		
description	text	true	true		
authority	text	true	true		
association	text	true	true		true
word	text	true	false		true

Validator

SPASE Resource Description grammar checker.

Checks a resource description for compliance to a specified version of the SPASE data model.

Usage:
    validator [options] file
            -or-
    java org.spase.tools.Validator [options] file

Options:

-h,--help	Dispay this text
-n,--version <arg>	Version of standard schema to use available from www.spase-group.org (default: 1.2.1)
-r,--recurse	Recursively process all files starting at path.
-s,--schema	Path to the XML schema document (XSD) to use for checking files.
-v,--verbose	Verbose. Show status at each step
-x,--ext <arg>	File name extension for filtering files when processing folders (default: .xml)

Acknowledgements
    Development funded by NASA's VMO project at UCLA.

Example
To validate the SPASE description in the file "example.xml" to version 2.0.0 of the SPASE schema use the command:

    validator -n 2.0.0 example.xml

The schema will be loaded from the web site www.spase-group.org. If you are not one a network you can use a local XML Schema document with the command:

    validator -s spase-2_0_0.xsd example.xml

RefCheck

SPASE Resource Description reference checker.

Can check both resource identifiers and URLs for referential integrity.
Can also produce lists of identifiers and URLs.

Usage:
        refcheck [options] file
        -or-
        java org.spase.tools.RefCheck [options] file

Options:

-h,--help	Dispay this text
-i,--identifier	Check each identifier in the resource description.
-l,--list	List all identifiers and URLs. Do not perform referential checks.
-r,--recurse	Recursively process all files starting at path.
-s,--service <arg>	The URL to the registry service to look-up resource identifiers (default: http://www.spase-group.org/registry/lookup).
-u,--urlcheck	Check each all URLs in the resource description.
-v,--verbose	Verbose. Show status at each step
-x,--ext <arg>	File name extension for filtering files when processing folders (default: .xml)

Acknowledgements
        Development funded by NASA's VMO project at UCLA.

Example
To check that all Resource IDs in all files can be resolved:

        refcheck -i -r *

To check all Resource IDs and URLs use the command:

        refcheck -i -r *

To see detailed information while checking add the verbose flag (-v):

        refcheck -v -i -r *

XMLGrep

XML Parser, XPath generator and search tool.

Parses an XML file and flattens the document content.
Values can be retrieved using an XPath that can contain regular expressions.
All values in the XML file can be listed with a corresponding XPath.

Usage:
>   java org.spase.tools.XMLGrep [options] file

Options:

-e,--extract <arg>	Extract. Extract all nodes with a given XPath
-f,--find <arg>	Find. Locate the value associated with an XPath
-h,--help	Display this text
-n,--nodes <arg>	Nodes. List all nodes at the given XPath
-v,--verbose	Verbose. Show status at each step

Acknowledgements:
>   Development funded by NASA's VMO project at UCLA.

Example
To find all items in an XML file that have an XPath ending in "ResourceID" use the command:

```
xmlgrep -f ".*/ResourceID" example.xml
```

will output something like:

```
/Spase/NumericalData/ResourceID: spase://VMO/NumericalData/IMP8/MAG/PT15.36S
```

To list the XPath tagged list of all content in an XML file use the command:

```
xmlgrep example.xml
```

will generate a list like:

```
/Spase/Version: 2.0.0
/Spase/NumericalData/ResourceID: spase://VMO/NumericalData/IMP8/MAG/PT15.36S
/Spase/NumericalData/ResourceHeader/ResourceName: IMP 8 Magnetic Field
/Spase/NumericalData/ResourceHeader/ReleaseDate: 2008-06-02T18:53:44Z
... and so on ...
```

Appendix A: Release Notes

All Tools written by Todd King

Releases

Release 1.0.0: Initial release.
Release 2.0.0: Aligned with current SPASE registry services.