Updating Items via Simple Archive Format

Item Update Tool

ItemUpdate is a batch-mode command-line tool for altering the metadata and bitstream content of existing items in a DSpace instance. It is a companion tool to ItemImport and uses the DSpace simple archive format to specify changes in metadata and bitstream contents. Those familiar with generating the source trees for ItemImporter will find a similar environment in the use of this batch processing tool.

For metadata, ItemUpdate can perform 'add' and 'delete' actions on specified metadata elements. For bitstreams, 'add' and 'delete' are similarly available. All these actions can be combined in a single batch run.

ItemUpdate supports an undo feature for all actions except bitstream deletion. There is also a test mode, as with ItemImport. However, unlike ItemImport, there is no resume feature for incomplete processing. There is more extensive logging with a summary statement at the end with counts of successful and unsuccessful items processed.

One probable scenario for using this tool is where there is an external primary data source for which the DSpace instance is a secondary or down-stream system. Metadata and/or bitstream content changes in the primary system can be exported to the simple archive format to be used by ItemUpdate to synchronize the changes.

A note on terminology: **item** refers to a DSpace item. **metadata element** refers generally to a qualified or unqualified element in a schema in the form `[schema].[element].[qualifier]` or `[schema].[element]` and occasionally in a more specific way to the second part of that form. **metadata field** refers to a specific instance pairing a metadata element to a value.

DSpace Simple Archive Format

As with ItemImporter, the idea behind the DSpace's simple archive format is to create an archive directory with a subdirectory per item. There are a few additional features added to this format specifically for ItemUpdate. Note that in the simple archive format, the item directories are merely local references and only used by ItemUpdate in the log output.

The user is referred to the previous section DSpace Simple Archive Format.

Additionally, the use of a **delete_contents** is now available. This file lists the bitstreams to be deleted, one bitstream ID per line. Currently, no other identifiers for bitstreams are usable for this function. This file is an addition to the Archive format specifically for ItemUpdate.

The optional suppress_undo file is a flag to indicate that the 'undo archive' should not be written to disk. This file is usually written by the application in an undo archive to prevent a recursive undo. This file is an addition to the Archive format specifically for ItemUpdate.

ItemUpdate Commands

Command used:	`[dspace]/bin/dspace itemupdate`
Java class:	*org.dspace.app.itemupdate.ItemUpdate*
Arguments short and (long) forms:	Description
`-a` or `--addmetadata [metadata element]`	Repeatable for multiple elements. The metadata element should be in the form dc.x or dc.x.y. The mandatory argument indicates the metadata fields in the dublin_core.xml file to be added unless already present (multiple fields should be separated by a semicolon ';'). However, duplicate fields will not be added to the item metadata without warning or error.
`-d` or `--deletemetadata [metadata element]`	Repeatable for multiple elements. All metadata fields matching the element will be deleted.
`-A` or `--addbitstreams`	Adds bitstreams listed in the contents file with the bitstream metadata cited there.
`-D` or `--deletebitstreams [filter plug classname or alias]`	Not repeatable. With no argument, this operation deletes bitstreams listed in the `deletes_contents` file. Only bitstream IDs are recognized identifiers for this operation. The optional filter argument is the classname of an implementation of `org.dspace.app.itemdupate.BitstreamFilter` class to identify files for deletion or one of the aliases (e.g. ORIGINAL, ORIGINAL_AND_DERIVATIVES, TEXT, THUMBNAIL) which reference existing filters based on membership in a bundle of that name. In this case, the `delete_contents` file is not required for any item. The filter properties file will contains properties pertinent to the particular filer used. Multiple filters are not allowed.
`-h` or `--help`	Displays brief command line help.

`-e` or `--eperson`	Email address of the person or the user's database ID **(Required)**
`-s` or `--source`	Directory archive to process **(Required)**
`-i` or `--itemfield`	Specifies the metadata field that contains the item's identifier; Default value is "dc.identifier.uri" (Optional)
`-t` or `--test`	Runs the process in test mode with logging. But no changes applied to the DSpace instance. (Optional)
`-P` or `--provenance`	Prevents any changes to the provenance field to represent changes in the bitstream content resulting from an Add or Delete. In other words, when this flag is specified, no new provenance information is added to the DSpace Item when adding/deleting a bitstream. No provenance statements are written for thumbnails or text derivative bitstreams, in keeping with the practice of MediaFilterManager. (Optional)
`-F` or `--filter-properties`	The filter properties files to be used by the delete bitstreams action (Optional)
`-v` or `--verbose`	Turn on verbose logging.

CLI Examples

Adding Metadata:

```
[dspace]/bin/dspace itemupdate -e joe@user.com -s [path/to/archive] -a dc.description
```

This will update all DSpace Items listed in your archive directory, adding a new `dc.description` metadata field. Items will be located in DSpace based on the handle found in 'dc.identifier.uri' (since the `-i` argument wasn't used, the default metadata field, dc.identifier.uri, from the dublin_core.xml file in the archive folder, is used).