EPrints Training: Command-Line Scripting Exercises

Getting Started

Pre-installed EPrints repositories are provided for this session. Use the following parameters to access your training repository:

Repository hostname	
Repository URL	
Repository ID	
SSH login name	
SSH login password	

Importing test data

These exercises require you to have some records in your repository. If you had to erase your repository in an earlier session, you can import some test data by running the following command:

```
eprints> /opt/eprints3/testdata/bin/import_test_data REPOSITORYID
```

Make a directory for your bin scripts

All EPrints command-line scripts are stored in the bin directory. Create a new subdirectory to hold your scripts:

```
eprints> mkdir /opt/eprints3/bin/REPOSITORYID
```

Change into your new subdirectory:

```
eprints> cd /opt/eprints3/bin/REPOSITORYID
```

Script 1: Get eprint title

Print the title of the given eprint

Enter the following into a file called `get_title.pl`

```
#!/usr/bin/perl -w -I/opt/eprints3/perl_lib
# Print the title of an eprint

use EPrints;
use strict;

# Check that we got 2 args
if( scalar @ARGV != 2 )
{
        print "Usage: $0 repositoryid eprintid\n";
        exit;
}
my $repositoryid = $ARGV[0];
my $eprintid = $ARGV[1];

# Start session
my $session = new EPrints::Session( 1, $repositoryid );
exit( 0 ) unless( defined $session );

# Try creating eprint
my $eprint = EPrints::DataObj::EPrint->new( $session, $eprintid );
if( !defined $eprint )
{
        print "$eprintid not found.\n";
}
else
{
        print $eprint->get_value( "title" )."\n";
}

# End Session
$session->terminate();
```

Run the script:

```
eprints> chmod +x get_title.pl
eprints> ./get_title.pl REPOSITORYID 1
```

Exercises

- Experiment with other eprint IDs (make sure the script fails as expected!)
- Change the code so that the abstract is also printed

Notes

- The array `@ARGV` contains the command-line arguments intended for the script, `$ARGV[0]` is the first argument (the repository ID), and `$ARGV[1]` the second argument (the eprint ID). `$0` is the command name itself (`get_title.pl`).
- Learn more about the `use strict` pragma - see `perldoc strict`

Script 2: List fields in dataset

Print a list of all the fields in a dataset

To save typing, the rest of the scripts for this session are available from:

```
http://www.eprints.org/services/training/resources/scripts/eprints3/bin/
```

Download the `iterate_fields.pl` script and examine it in a text editor (change `REPOSITORYID` to the ID you have been assigned for this session):

```
eprints> wget
http://www.eprints.org/services/training/resources/scripts/bin/iterate_fields.pl
```

Run the script:

```
eprints> chmod +x iterate_fields.pl
eprints> ./iterate_fields.pl
```

Exercises

- Change the code so that it iterates over other EPrints datasets

Script 3: Create new eprint

Create a new eprint in the repository and attach a PDF document to it

Download the `create_eprint.pl` script and examine it in a text editor (change `REPOSITORYID` to the ID you have been assigned for this session).

Download the `demo.pdf` file:

```
eprints> wget http://www.eprints.org/services/training/resources/demo.pdf
```

Run the script:

```
eprints> chmod +x create_eprint.pl
eprints> ./create_eprint.pl
```

Check that an abstract page has been created for the new eprint by opening the following URL in a browser (ID is the value returned by `create_eprint.pl`):

```
http://REPOSITORYURL/ID/
```

Exercises

- Change the code so that a Conference Item is created instead of an Article
- Add some field values appropriate to conference items (hint: look at the output of `iterate_fields.pl` for ideas)
- Change the code so that the `demo.pdf` is downloaded from a URL rather than uploaded from a file (hint: `$doc->upload_url` - this returns true if successful)
- Change the code so that rather than add `demo.pdf`, it adds the following HTML file, complete with images: `http://www.eprints.org/services/training/resources/demo.html`

Script 4: Search by date

Search the repository for eprints matching a given date

Download the `ids_by_date.pl` script and examine it in a text editor.

Run the script:

```
eprints> chmod +x ids_by_date.pl
eprints> ./ids_by_date.pl REPOSITORYID 2000
```

Exercises

- Experiment with other dates or date ranges (e.g. "2000-", "2003-2005", "-2002-06-05")
- Change the code so that it also prints the number of matches (hint: `$list->count`)
- Change the code so that rather than printing the ID of each result, it prints the title (hint: apply a function to each result using `$list->map`)
- Change the code so that the results are re-ordered by title after the search has been carried out (hint: `$list->reorder("title")`)

Script 5: Search by title

Search the repository for eprints with titles containing a given word

Download the `titles_by_word.pl` script and examine it in a text editor.

Run the script:

```
eprints> chmod +x titles_by_word.pl
eprints> ./titles_by_word.pl REPOSITORYID horse
```

Exercises

- Experiment with other words (hint: look at the titles printed by script 4a)
- Change the code so that it finds eprints with titles *or abstracts* matching the given word (hint: you can specify multiple fields in `add_field` by using an array reference - use `perldoc` to find the documentation for the `add_field` function)

Script 6: Search and modify

Search the repository for eprints and modify each result

Download the `search_and_modify.pl` script and examine it in a text editor (change `REPOSITORYID` to the ID you have been assigned for this session).

Run the script:

```
eprints> chmod +x search_and_modify.pl
eprints> ./search_and_modify.pl
```

Check that the matching eprints have been modified by opening the following URL in a browser (ID is one of the values returned by `create_eprint.pl`):

```
http://REPOSITORYURL/URL/
```

Exercises

- Change the code so that it gets the date to search for from the command line
- Change the code so that it adds the `demo.pdf` document to matching eprints