

File_MARC under PHP

Dan Scott
Laurentian University
#code4lib tribe

http://pear.php.net/File_MARC

Zzzz...

- File_MARC is a PEAR library (currently in alpha)
 - Builds on emilda.org's php_marc module
 - Core parsing routines borrowed from File::MARC
- Why?
 - PEAR is the de facto repository for PHP libraries
 - PEAR enforces coding standards
 - OO interface and unit tests
 - PHP does not have to be ugly!

Basic features: reading

- Read MARC 21 and MARCXML from any streamable source, including compressed files or network resources

```
<?php
require 'File/MARC.php';

// Retrieve a set of MARC
records from a file
$journals = new
File_MARC('journals.mrc');

require 'File/MARCXML.php';
// Retrieve a set of MARCXML
records from a string
$journals = new
File_MARCXML($xml_data,
SOURCE_STRING);
```

Basic features : searching

- Retrieval of fields and subfields matching exact strings or regular expressions

```
// Iterate through the records
while ($rec = $bibs->next()) {
// Retrieve an array of all of the
650 fields
    $subs = $rec->getFields('650');
    if ($subs) {
        // Get the first 24* field
        print $rec->getField('24.',
true);

        // Print the retrieved subjects
        foreach ($subs as $field) {
            print $field;
        }
    }
}
```

Adding fields and subfields

- You can insert fields and subfields at any point in the record

```
// Create the new field
```

```
$subfields[] = new File_MARC_Data_Subfield('a', 'Scott, Daniel .');
```

```
$new_field = new File_MARC_Data_Field('100', $subfields, 0, null);
```

```
// Retrieve the target field for our insertion point
```

```
$sub = $rec->getField('650');
```

```
// Insert the new field
```

```
$rec->insertField($new_field, $sub, true);
```

Iteration

- **File_MARC** implements the **ArrayIterator** interface to enable you to iterate through fields and subfields using **foreach()**

```
foreach ($marc_record-  
>getFields() as $tag=>$value) {  
    print "$tag: ";  
    if ($value instanceof  
File_MARC_Control_Field) {  
        print $value->getData();  
    } else {  
        foreach ($value-  
>getSubfields() as  
$code=>$subdata) {  
            print "_$code";  
        }  
    }  
}
```

Output: raw MARC or MARCXML

- toRaw() returns a MARC 21 representation of the File_MARC_Record object
- toXML() returns a MARCXML representation**

** PHP knows nothing about character sets, so this is currently effectively limited to UTF-8 MARC data

Try it out

- File_MARC is pure PHP, so it doesn't have to be compiled or rely on any external libraries
- File_MARC is still an alpha package, so you need to specify the alpha version of the package for now
- `$ pear install File_MARC-alpha`
- Browse through the unit tests for examples
- Learn more about it at http://pear.php.net/File_MARC

Next steps

- Defy PEAR coding standards by offering warnings instead of exceptions in some cases
- Pray for MARC8 / ANSEL encoding support to be added to ICU (I18N engine underpinning PHP 6)
- Improve documentation
- Get feedback on API and iterate correspondingly...
- Get a stable 1.0 release out