PICA::Record

PICA::Record ist eine Programmbibliothek zur Verarbeitung von PICA+-Datensätzen mit der Programmiersprache Perl. PICA+ ist das interne bibliografische Datenformat der CBS und LBS-Software. Beim Import- bzw. Export von Daten in oder aus PICA-Datenbenken müssen Datensätze nach oder von PICA+ konvertiert werden - hierzu kann PICA::Record eingesetzt werden. Die Programmbibliothek wird an der VZG von Jakob Voss entwickelt und steht als Freie Software auf CPAN unter http://search.cpan.org/dist/PICA-Record/ zur Verfügung.

Zur Einführung gibt es unter Anderem das Faltblatt *Verarbeitung von PICA+ Daten mit PICA::Record* und vor allem die API-Dokumentation.

Die folgende Beschreibung bezieht sich auf Version 0.502 und folgende Versionen.

Installation

Linux

Unter Linux und anderen Unix-Varianten genügt zur Installation und Aktualisierung direkt aus CPAN ein Kommando:

```
sudo perl -MCPAN -e 'install PICA::Record'
```

Es ist auch möglich, PICA::Record ohne root-Rechte im Nutzerverzeichnis zu installieren.

Zur Installation einer Entwicklerversion, ist die entsprechende .tar.gz-Datei in ein Verzeichnis zu Entpacken, wo es dann folgendermaßen installiert wird:

```
perl Makefile.pl
make
make test
make install
```

Wobei zu beachten ist das die Kommandos in dem Verzeichnis ausgeführt werden wo zuvor die .tar.gz-Datei entpackt wurde. Des weiteren sollte man auf die Groß und Kleinschreibung der Dateinamen achten.

Mögliche Probleme

Pica::Record hat Abhänigkeiten zu anderen Bibliotheken. Wie bzw. SOAP::Lite. Normalerweise sollten diese Automatisch mit installiert werden. Ist dem nicht so muss man dies Erzwingen. Dazu muss man in der Konsole folgenden Code einfügen.

```
perl -MCPAN -e "CPAN::Shell->force(qw(install SOAP::Lite));"
```

Windows

Da Perl nicht zum Standardlieferumfang von Windows gehört, muss zunächste ActivePerl oder Strawberry Perl installiert werden. Zu empfehlen ist außerdem der Editor KomodoEdit oder ein anderer Texteditor wie zum Beispiel UltraEdit.

Über den Paketmanager lässt sich auch das Perl-Paket von PICA::Record installieren. Allerdings ist die dort angebotene Version oft nicht nicht aktuell. Um eine aktuelle Version von CPAN oder eine aktuelle Entwicklerversion zu installieren, wird zunächst das Programm NMAKE benötigt, welches hier von Microsoft kostenlos verfügbar ist. NMAKE (bzw. NMAKE.EXE und NMAKE.ERR) müssen in bin-Verzeichnis der Perl-Installation installiert werden (in der Regel C:\Perl\bin). Anschließend kann PICA::Record über CPAN oder aus einer .tar.gz Datei installiert werden:

```
perl -MCPAN -e "install PICA::Record"
```

oder

```
perl Makefile.pl
nmake test
nmake install
```

Weitere Informationen zu Perl siehe hier.

Kurzeinführung

Die PICA::Record-Programmbibliothek ist ausführlich auf Englisch dokumentiert und enthält mehrere Beispiele (in den Verzeichnissen examples und script). Die folgende Kurzeinführung soll einen schnellen Einstieg und groben Überblick geben. Weitere Informationen und Beispiele gibt es direkt in der Distribution. Einige weiteren Beispiele stehen im GBV-Wiki unter Picapm/Beispiele.

Überblick

Die Programmbibliothek besteht aus mehreren Klassen. Im Wesentlichen sind dies:

- **PICA::Record** - speichert einen PICA+ Datensatz
- **PICA::Field** - speichert ein Feld in einem PICA+ Datensatz
- **PICA::Parser** - liest PICA+ Datensätze aus Dateien oder Zeichenketten
- **PICA::Writer** - schreibt PICA+ Datensätze in verschiedenen Formaten
- **PICA::Source** - bietet Zugriff auf eine Datenbank, aus der PICA+ Datensätzen angefragt werden können (per Z39.50, SRU, unAPI...)

- **PICA::Store** - bietet Zugriff auf eine Datenbank, in der einzelne PICA+ Datensätze angelegt, abgefragt, aktualisiert und gelöscht werden können (bislang nur mittels CWS).

Zusätzlich beinhaltet die Distribution ab Version 0.45 zwei Kommandozeilenprogramme, die mit installiert werden

- **parsepica** - liest PICA+ Daten ein und gibt sie aus
- **picaimport** - trägt über CWS PICA+ Datensätze in eine Datenbank ein

Weitere Informationen und Beispiele zu diesen Programmen gibt es bei Aufruf mit dem Kommandozeilenparameter -man.

PICA-Datensätze bearbeiten

Ein PICA+ Datensatz besteht aus einer Liste von Feldern ("Kategorien"), die jeweils aus einer Feldnummer (*tag*) und einer Liste von Unterfeldern (*subfields*) bestehen. PICA::Record und PICA::Field bieten eine Vielzahl von Möglichkeiten, um einzelne PICA+ Datensätze bzw. -Felder zu erzeugen, zu verändern und um Inhalte auszulesen.

```
$record = PICA::Record (
  "021A", "a" => "Titel",
  "028A", "d" => "Vorname", "a" =>"Nachname"
);

$field = PICA::Field->new( "011@", "a" => 1999 );

$record->append( $field );

print $record->to_string() . "\n";

$nachname = $record->sf('028A$a'); # 'sf' oder 'subfield'

$local = $record->local_records();

$linkfields = $record->f("009P/03"); #'f' oder 'field'
```

PICA+ Daten einlesen

Mit PICA::Parser können PICA+ Daten in verschiedenen PICA+ Varianten und in PICA XML eingelesen werden:

```
@records = PICA::Parser->parsefile( "picadata.xml" )->records();

@records = PICA::Parser->parsefile( \*STDIN, Limit => 10 )->records();

$parser = PICA::Parser->new( Proceed => 1 );
$parser->parsedata( $picadata );
print $parser->counter() . " Datensätze gelesen.\n";
```

Beim Einlesen können die PICA+ Daten durch einen Filter laufen oder direkt verarbeitet werden, was vor allem bei großen Datenmengen von Vorteil ist:

```
sub levelfilter {
  my $field = shift;
  return $field if $field->level() == 0; # nur Ebene Null
}
sub record_handler {
  my $record = shift;
  return $record if $record->field("021A"); # nur Datensätze mit Titel
}
my @records = PICA::Parser->parsefile( "dumpfile.gz",
  Record => \&record_handler,
  Field => \&levelfilter
);

PICA::Parser->parsefile( \*SDTIN,
  Record => sub { # Datensätze direkt ausgeben ohne zu speichern
  my $record = shift;
  print $record->to_string() . "\n";
  }
);
```

Für Einfache Verarbeitungen reicht zur Ein- und Ausgabe das Kommandozeilenprogramm parsepica.

PICA+ Daten ausgeben

Zur Ausgabe stellt PICA::Record die Methoden to_string und to_xml bereit. Um mehrere Datensätze auszugeben, gibt es die Klasse PICA::Writer und davon abgeleitete Klassen.

PICA+ Daten holen

Zum Einlesen von PICA-Daten gibt es die Klasse PICA::Parser (siehe oben) und um Datensätze von einem Server über eine Schnittstelle wie Z39.50, SRU oder unAPI abzurufen die Klasse PICA::Source. Die Wesentlichen Methoden sind ebenfalls in parsepica umgesetzt.

Das folgende Skript läd per SRU Datensätze aus dem GSO-Katalog Einträge mit dem Titel "Märchen" als PICA+ in eine Datei.

```
use PICA::Source;

sub handler {
  my $record = shift;
  print FILE $record->to_string();
}

open FILE, ">", "märchen.pica";
my $gso = PICA::Source->new( SRU => "http://gso.gbv.de/sru/DB=2.1/" );
my $parser = $gso->cqlQuery( "pica.tit=märchen", Record => \&handler );
```

Der folgende Aufruf von parsepica erfüllt den gleichen Zweck:

```
parsepica -o märchen.pica http://gso.gbv.de/sru/DB=2.1/ pica.tit=märchen
```

Das Script z3950.pl im Verzeichnis examples enthält ein Beispiel zum holen von Daten per Z39.50.

Daten holen über WinIBW

Neben der Abfrage von PICA+ Datensätzen über standardisierte Schnittstellen (SRU, Z39.50, unAPI, CWS) kann über einige Umwege auch die WinIBW-Software verwendet werden. Beim Download-Kommando der WinIBW können folgende Formate angegeben werden:

- **P** - PICA+ (nur Ebene 0)
- **PA** - PICA+ mit Lokaldaten aller Bibliotheken
- **PNr** - PICA+ mit Lokaldaten der Bibliothek Nr

Die Datensätze werden an die Datei download.txt im Download-Ordner (siehe Menü Datei, Menübefehl Download) angehängt. Allerdings ist das Download-Format *kein valides PICA+* sondern muss zur Weiterverarbeitung erst mit dem Skrip winibw2pica bereinigt werden!

PICA+ Daten speichern

Zum Speichern von PICA+ Datensätzen in einer Datenbank gibt es die Klasse PICA::Store. Damit kann über die Collection Webservices (CWS) auch auf CBS-Datenbanken zugegriffen werden. Im script-Verzeichnis befindete sich der Kommandozeilenclient picaimport. Siehe dazu PICA+ Daten einspielen mit picaimport.

Das folgende Skript liest den ersten PICA+ Datensatz aus einer Datei und, trägt ihn über webcatws in eine Datenbank ein und gibt bei Erfolg die PPN aus:

```
use PICA::Store;
use PICA::Record;
use IO::Handle;

my $filename = "record.pica";
my $record = PICA::Record->new( IO::Handle->new("< $filename") );

if ($record) {
  my $server = PICA::Store->new( SOAP => $baseurl, dbsid => $dbsid
  userkey => $userkey, password => $password );
  my %result = $server->create( $record );
  print "PPN: " . $result{id} . "\n" if $result{id};
}
```

Anmerkungen

Kommentare und Fehlerberichte für die Weiterentwicklung sind herzlich willkommen! Beispiele können in diesem Wiki unter Picapm/Beispiele eingetragen werden. Etwas offtopic: es gibt auch einen Spielfilm namens Perl oder Pica.

In der Version 0.47 umfasst PICA::Record 4153 Zeilen Quellcode, 951 Zeilen Tests und 752 Zeilen für die Kommandozeilenprogramme parsepica und picawebcat.