GBV VZG

2009

Verarbeitung von PICA+ Daten mit PICA::Record

```
002@     $0Aau
003@     $0588923168
007G     $cDNB$0991841212
010@     $ager$ceng
011@     $a2009
019@     $aXA-DE-BY
021A     $aDas @Graveyard-Buch$hNeil Gaiman. Aus dem Engl. Von
         Reinhard Tiffert
022A/01  $aThe @graveyard book <dt.>
028A     $dNeil$aGaiman$9080629490$8Gaiman, Neil *1960-*
028C     $dReinhard$aTiffert$9589449664$8Tiffert, Reinhard
032@     $a1. Aufl
033A     $pWürzburg$nArena-Verl.
101@     $a120$cPICA$dOldenburg, IBIT Universität Oldenburg <715>
101B     $015-01-09$t15:32:38.000
101D     $015-01-09$b9330$a0715
101U     $0utf8
145Z/40  $a4/Jungen, Männer/Gespenster-, Grusel- und
         Vampirgeschichten
201B/01  $015-01-09$t15:32:38.000
201D/01  $015-01-09$b9330$a0715
201U/01  $0utf8
203@/01  $0917400194
208@/01  $a15-01-09$bz
209A/01  $f108$a401-06356/09$dg$x00
```

Verarbeitung von PICA+ Daten mit PICA::Record

PICA::Record ist eine Programmbibliothek zur Verarbeitung von Daten im PICA+ Format in der Programmiersprache Perl. Sie kann unter anderem für Import- und Export von Daten in oder aus PICA-Datenbanken und für die Datenkonvertierung von und nach PICA+ eingesetzt werden. Die Programmbibliothek steht unter http://search.cpan.org/dist/PICA-Record/ im *Comprehensive Perl Archive Network (CPAN)* als freie Software zur Verfügung.

Was ist PICA+?

PICA+ ist das interne bibliografische Datenformat der CBS- und LBS-Software. Ein PICA-Datensatz besteht aus einer Liste von **Feldern** oder Kategorien, die jeweils eine Liste von **Unterfeldern** enthalten. Innerhalb einer Datenbank wird ein Datensatz durch seine **Pica-Produktionsnummer (PPN)** identifiziert, die in Feld **003@**, Unterfeld **0** steht. Unterfelder werden durch ein alphanumerisches Zeichen identifiziert und bei der Ein- und Ausgabe oft mit einem Dollarzeichen (**$**) eingeleitet, während Feldnamen mindestens aus drei Ziffern und einem Zeichen (**A-Z** oder **@**) bestehen. Beide können wiederholbar sein. Die Bedeutung der einzelnen Felder und Unterfelder ist in den jeweiligen Katalogisierungsregeln festgelegt. [1]

Zugriff auf Datensatz-Inhalte

In der Programmbibliothek werden Felder und Unterfelder als **PICA::Record** bzw. als **PICA::Field** verwaltet. Beide Objekte bieten eine Reihe von Zugriffsmethoden:

```
my $titlefield = $record->field('021A');            # das Feld 021A
my $title = $titlefield->subfield('a');            # das Unterfeld a
my $ppn = $record->subfield('003@$a');      # Unterfeld eines Feldes
my @creators = $record->field('028A','028B/..');   # mehrere Felder
my ($c1,$c2) = $record->field(2,'028B/..');      # genau zwei Felder

# Vorname (Unterfeld d) und Nachname (Unterfeld a) aller Verfasser
print join(", ", map { $_->sf('d')." ".$_->sf('a') } @creators);

# Signaturen der Exemplare aller Bibliotheken
foreach my $local ($record->local_records()) {
   print $local->sf('101@$d') . "\n";          # Name der Bibliothek
   print " " . join("\n ", $local->sf('209A/..','a')) . "\n";
}
print $record->to_string();              # Ausgabe in Standardformat
```

Die Erstellung und Bearbeitung von PICA+ Datensätzen erfolgt ebenfalls mit Methoden der Klassen **PICA::Record** und **PICA::Field**.

1 Siehe http://www.gbv.de/du/katricht/inhalt.shtml für die Regeln im GBV.

Datensätze verändern und erstellen

```
$record = PICA::Record->new(                          # neuer Datensatz
  '028A', 'd' => 'Vorname', 'a' =>'Nachname',
  '021A', 'a' => 'Titel'
);
$record->append('037A', 'a' => 'Anmerkung...');   # Feld hinzufügen
$record->replace('037A', 'a' => 'Kommentar...');    # Feld ersetzen

$field = PICA::Field->new( '011@', 'a' => 1999 );       # neues Feld
$field->update( 'a' => 2000 );                     # Feldinhalt ändern
$record->append( $field );                          # Feld hinzufügen
$record->delete_fields( '037A' );                    # Feld entfernen
```

Lesen und Schreiben von PICA+

Die Ein- und Ausgabe von PICA+ Datensätzen ist in verschiedenen Formaten von normalisiertem PICA+ bis zu PICA-XML möglich, sofern Unicode in UTF8 verwendet wird. Mit den Klassen **PICA::Parser** und **PICA::Writer** lassen sich auch große Mengen von Datensätzen verarbeiten. Den lesenden Zugriff über Schnittstellen wie Z39.50, SRU und unAPI erleichtert die Klasse **PICA::Source**. Die Änderung von Datensätzen in einer Datenbank ist mit der Klasse **PICA::Store** möglich. [2]

```
$record = PICA::Record->new( $picadata );  # liest aus Zeichenkette
$record = PICA::Record->new( \*FILE );     # liest aus Datei-Handle
                                          # liest mehrere Datensätze
@records = PICA::Parser->parsefile( "picadata.xml" )->records();
@records = PICA::Parser->parsefile( \*STDIN )->records();
print $record->to_xml( header => 1 );    # Ausgabe in XML mit Header
PICA::Writer->new( \*STDOUT, format => 'xml' )->write( $record );
                                                  # Abfrage per SRU/CQL
$gso = PICA::Source->new( SRU => "http://gso.gbv.de/sru/DB=2.1/" );
$result = $gso->cqlQuery( "pica.tit=Schnupperkurs", Limit => 10 );
$gso = PICA::Source->new( Z3950 => "z3950.gbv.de:20012/GVK" );
$result = $gso->z3950Query('@attr 1=4 microformats'); # per Z39.50
$record = $gso->getPPN( "1234567890" );            # Abfrage mit PPN
%result = $server->get( $id );          # Zugriff auf PICA-Datenbank
%result = $server->create( $record );
%result = $server->update( $id, $record, $version );
%result = $server->delete( $id );
```

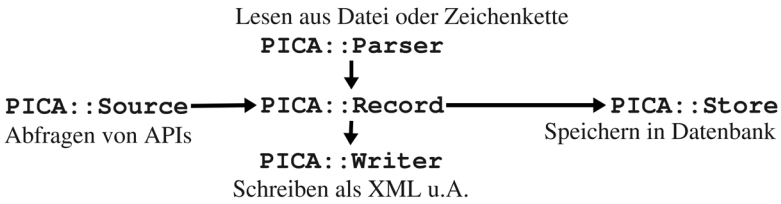
2 Im GBV über die Collection Webservices (CWS), siehe http://www.gbv.de/wikis/cls/CWS.

Kommandozeilenprogramme

Für viele Verarbeitungsschritte reichen bereits die Kommandozeilenprogramme **parsepica** und **picawebcat** aus, die in der Programmbibliothek enthalten sind. **parsepica** liest PICA+ Datensätze aus einer Datei oder über eine Schnittstelle und gibt sie gezielt aus, während mit **picawebcat** die Collection Webservices angesprochen werden.

```
parsepica record.pica record.xml            # umwandeln nach PICAXML
parsepica record.pica -s 021A\$a -qo - -q        # gibt nur Titel aus
                                          # Abfragen per SRU/Z39.50
parsepica http://gso.gbv.de/sru/DB=2.1/ pica.isb=3-423-31039-1
parsepica z3950.gbv.de:20012/GVK '@attr 1=4 microformats'
                       # Datensatz per CWS einspielen bzw. verändern
picawebcat -conf webcat.conf create record.pica
picawebcat -conf webcat.conf update 123456789 record.pica
```

Übersicht der Basisklassen



Weitere Informationen

Die Programmbibliothek **PICA::Record** ist ausführlich auf CPAN dokumentiert und enthält eine Reihe von Beispielen (größtenteils in Englisch). Im GBV-Wiki gibt es zusätzlich eine eigene Seite mit einer Kurzeinführung auf Deutsch.

- CPAN: http://search.cpan.org/dist/PICA-Record/
- GBV-Wiki: http://www.gbv.de/wikis/cls/PICA::Record

Ansprechpartner:

Für Fragen steht Ihnen in der VZG
Jakob Voß (jakob.voss@vzg.de) zur Verfügung.

Impressum
Verbundzentrale des Gemeinsamen Bibliotheksverbundes (VZG)
Platz der Göttinger Sieben 1 • 37073 Göttingen
Tel.: 0551/39-5207 • Fax: 0551/39-22408 • E-Mail: gbv@gbv.de
Sandholzer (Mai 2009)