Catalog housekeeping scripts for Koha

housekeeping (/ˈhaʊskiːpɪŋ/)

noun

1. the management of household affairs.
2. operations such as maintenance or record-keeping which facilitate productive work in an organization.

Andreas Roussos
Systems Librarian @
Library of the Holy Monastery of Paraklitos
(Oropos, Greece)

The Monastery

The main courtyard of the Monastery with the main Church

- Located approx. 40km outside of Athens, near the seaside town of Oropos

- “Paraklitos” means “Paraclete”, i.e. the 3rd divine person (*hypostasis)* of the Holy Trinity

- Founded in 1963 and officially recognised from Church and State in 1978, today is home to 25 monks

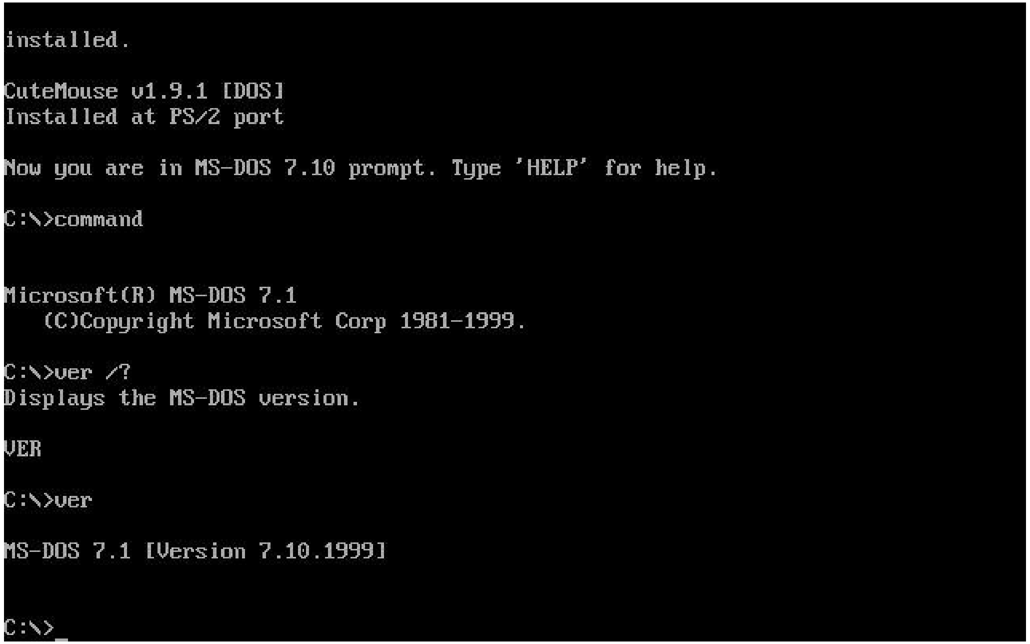
- Follows the organisation of Mount Athos monasteries

The Library

Interior of the Library – the main collection

- Contains approximately 30,000 books, a large portion of which come from donations
- Currently accommodated in two floors, but more space is planned to become available
- The main collection expands at a rate of ~500 books per year
- Focuses mostly on religion, but other disciplines in the humanities are represented as well, such as history, philosophy, psychology

The need for change



```
installed.

CuteMouse v1.9.1 [DOS]
Installed at PS/2 port

Now you are in MS-DOS 7.10 prompt. Type 'HELP' for help.

C:\>command


Microsoft(R) MS-DOS 7.1
   (C)Copyright Microsoft Corp 1981-1999.

C:\>ver /?
Displays the MS-DOS version.

VER

C:\>ver

MS-DOS 7.1 [Version 7.10.1999]


C:\>_
```

- The previous library catalog software was a DOS-based program called ABEKT running on a Windows 95 PC

- Originally installed in 2000, it served basic cataloguing needs for more than 10 years

- Offered no support for multiple user editing or OPAC

- No spine/barcode label creation and printing available

The migration

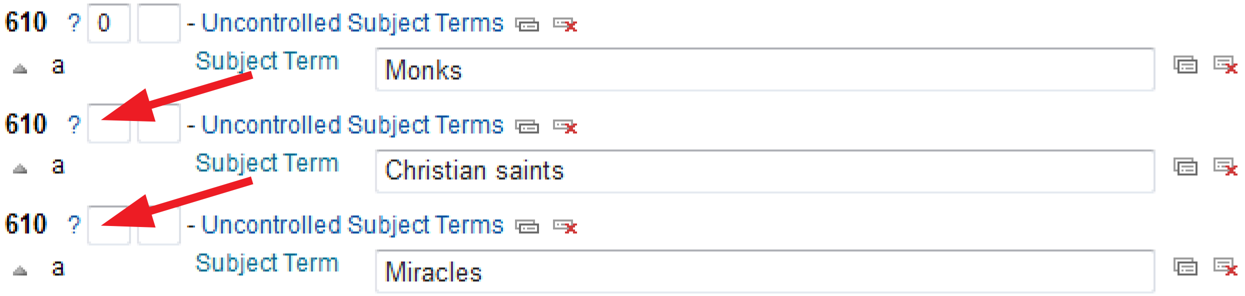
AB
EKT
Σύστημα
Αυτοματισμού
Βιβλιοθηκών ΕΚΤ
koha

- Koha was chosen among other ILSs in 2011
- Over 20,000 records exported from ABEKT 4.4 in ISO 2709 (UNIMARC) format and imported into Koha
- Originally a tarball installation (ver. 3.02) on an Ubuntu 10.04 LTS VM running on VirtualBox
- Since Aug. 2016 a package install (v. 16.05) on a Debian 8.9 VM running on an ESXi 5.0 host

A difficult inheritance

- No item or patron information, just plain biblio records
- ABEKT (previous ILS) was MARC-aware, but several fields were not correctly filled out according to the standards
- A number of challenges arose, some of which were purely bibliographic, while others purely technical
- Hence the need for re-cataloguing and developing housekeeping scripts

Bibliographic challenge #1: missing indicator 0



610 ? 0 - Uncontrolled Subject Terms

a Subject Term Monks

610 ? - Uncontrolled Subject Terms

a Subject Term Christian saints

610 ? - Uncontrolled Subject Terms

a Subject Term Miracles

- How do you detect that UNIMARC field 610 (“Uncontrolled Subject Terms”) has a missing 1st indicator?

Bibliographic and item data storage in Koha

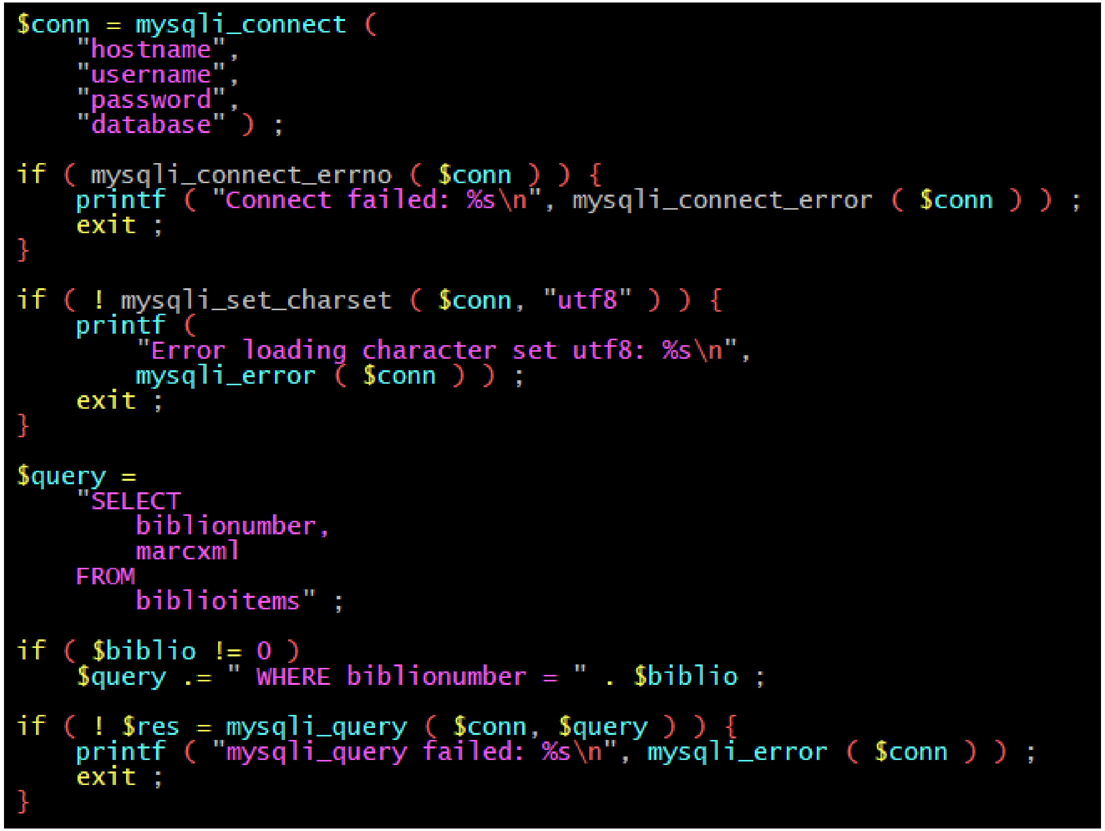
```
<datafield tag="610" ind1="0" ind2=" ">
  <subfield code="a">Monks</subfield>
</datafield>
<datafield tag="610" ind1=" " ind2=" ">
  <subfield code="a">Christian saints</subfield>
</datafield>
<datafield tag="610" ind1=" " ind2=" ">
  <subfield code="a">Miracles</subfield>
</datafield>
```

- A Koha instance stores its bibliographic and item data in the associated database tables
- MARCXML is used internally
- Prior to version 17.05, biblio data was placed in the `biblioitems.marcxml` column, in more recent versions this was changed to `biblio_metadata.metadata`

Working with (MARC)XML data

- There are many ways to process XML data, depending on the programming language
- Two common C parsers are `expat` and `libxml2`
- C++ has `xerces-c++` and `tinyxml2`
- If speed is your #1 priority, there's even one coded in Assembly: `AsmXml`
- Perl has `XML::LibXML` and `MARC::XML`
- PHP has `SimpleXML` (built-in), or one can use the (external) `File_MARC` package from PEAR (PHP Extension and Application Repository)

A common block of code

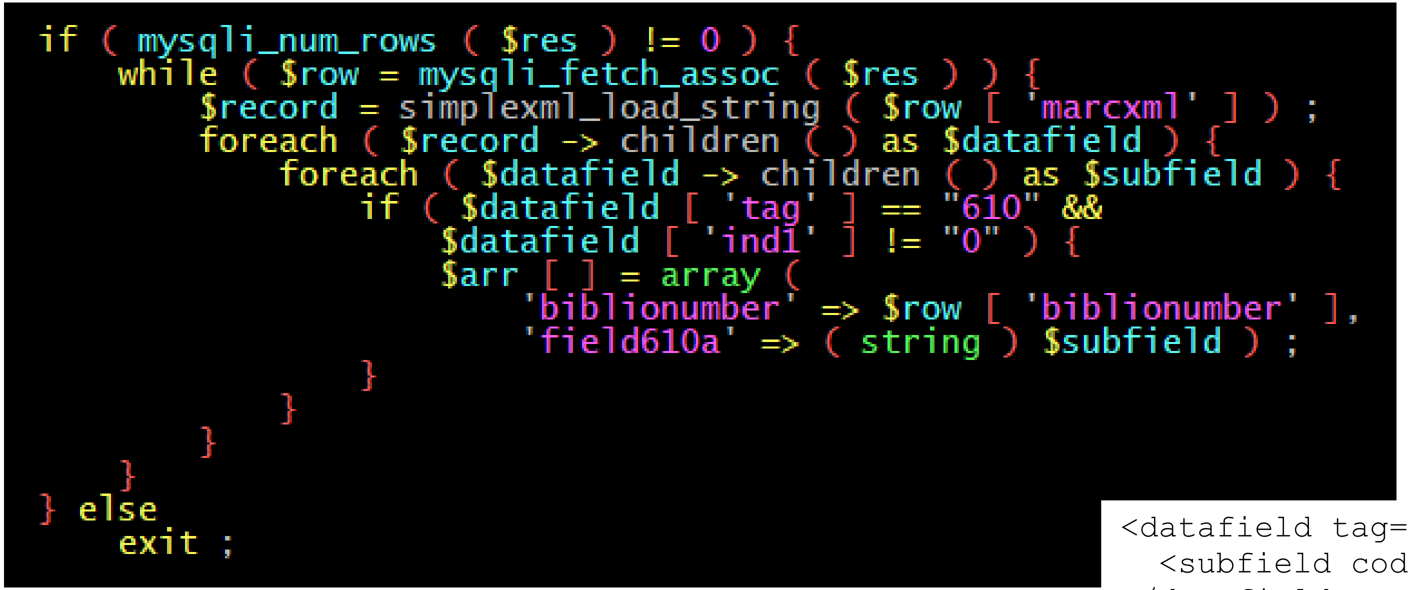


```
$conn = mysqli_connect (
    "hostname",
    "username",
    "password",
    "database" ) ;

if ( mysqli_connect_errno ( $conn ) ) {
    printf ( "Connect failed: %s\n", mysqli_connect_error ( $conn ) ) ;
    exit ;
}

if ( ! mysqli_set_charset ( $conn, "utf8" ) ) {
    printf (
        "Error loading character set utf8: %s\n",
        mysqli_error ( $conn ) ) ;
    exit ;
}

$query =
    "SELECT
        biblionumber,
        marcxml
    FROM
        biblioitems" ;

if ( $biblio != 0 )
    $query .= " WHERE biblionumber = " . $biblio ;

if ( ! $res = mysqli_query ( $conn, $query ) ) {
    printf ( "mysqli_query failed: %s\n", mysqli_error ( $conn ) ) ;
    exit ;
}
```

This block of code:

- sets the SQL connection parameters and establishes a connection to the Koha database
- sets the connection character set
- runs a simple `SELECT` query that fetches the biblio number and associated MARCXML data for all books in the DB (or for a specific biblio)
- performs some basic error checking

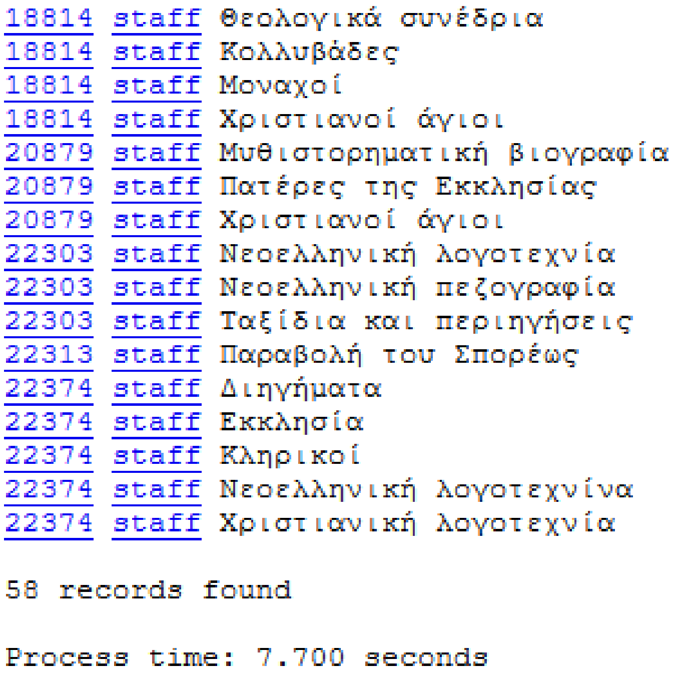
Loading MARCXML data with SimpleXML



```
if ( mysqli_num_rows ( $res ) != 0 ) {
    while ( $row = mysqli_fetch_assoc ( $res ) ) {
        $record = simplexml_load_string ( $row [ 'marcxml' ] ) ;
        foreach ( $record -> children ( ) as $datafield ) {
            foreach ( $datafield -> children ( ) as $subfield ) {
                if ( $datafield [ 'tag' ] == "610" &&
                    $datafield [ 'ind1' ] != "0" ) {
                    $arr [ ] = array (
                        'biblionumber' => $row [ 'biblionumber' ],
                        'field610a' => ( string ) $subfield ) ;
                }
            }
        }
    }
} else
    exit ;
```

```
<datafield tag="610" ind1="0" ind2=" ">
  <subfield code="a">Monks</subfield>
</datafield>
<datafield tag="610" ind1=" " ind2=" ">
  <subfield code="a">Christian saints</subfield>
</datafield>
<datafield tag="610" ind1=" " ind2=" ">
  <subfield code="a">Miracles</subfield>
</datafield>
```

Displaying the results in the web browser



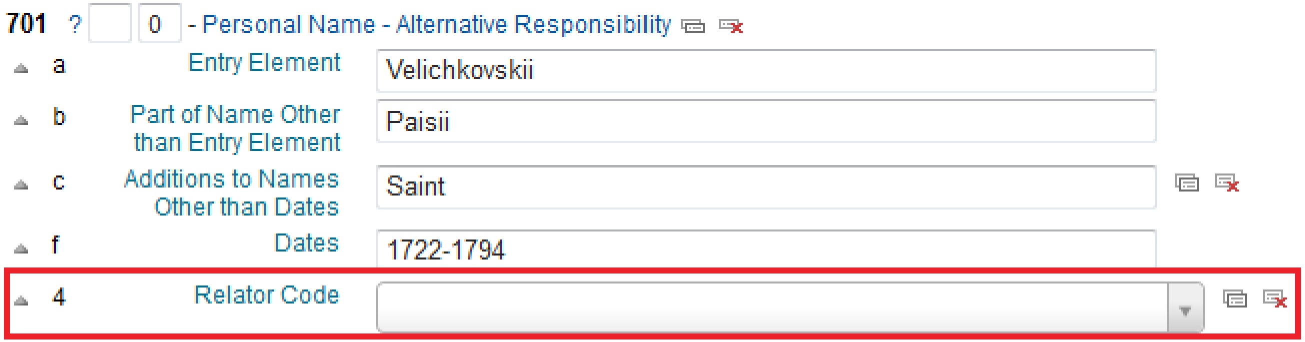
```
18814 staff Θεολογικά συνέδρια
18814 staff Κολλυβάδες
18814 staff Μοναχοί
18814 staff Χριστιανοί άγιοι
20879 staff Μυθιστορηματική βιογραφία
20879 staff Πατέρες της Εκκλησίας
20879 staff Χριστιανοί άγιοι
22303 staff Νεοελληνική λογοτεχνία
22303 staff Νεοελληνική πεζογραφία
22303 staff Ταξίδια και περιηγήσεις
22313 staff Παραβολή του Σπορέως
22374 staff Διηγήματα
22374 staff Εκκλησία
22374 staff Κληρικοί
22374 staff Νεοελληνική λογοτεχνίνα
22374 staff Χριστιανική λογοτεχνία

58 records found

Process time: 7.700 seconds
```

- The results are sorted by biblio number, then by subject term
- The first hyperlink allows you to see all subject terms for a particular biblio number
- The second hyperlink directly takes you to the Staff interface view for that biblio for direct editing
- The third field is the actual subject term that is missing a `0` for the 1st indicator

The ‘File_MARC’ PHP package

- `File_MARC` is a PHP package that allows you to manipulate MARC/MARCXML records
- Currently at version 1.3.0, with extensive documentation at https://pear.php.net
- Methods for retrieving information: `getLeader()`, `getField()`, `getFields()`, `getTag()`, `getCode()`, `getData()`, `getPosition()`, `getIndicator()`, `getContents()`, `getSubfield()`, `getSubfields()`
- Methods for working with fields and subfields: `appendField()`, `prependField()`, `insertField()`, `deleteFields()`, `appendSubfield()`, `prependSubfield()`, `insertSubfield()`, `deleteSubfield()`
- Methods for manipulating leader/field/subfield data: `setLeader()`, `setTag()`, `setCode()`, `setData()`, `setPosition()`, `setIndicator()`

Bibliographic challenge #2: 7xx missing role code

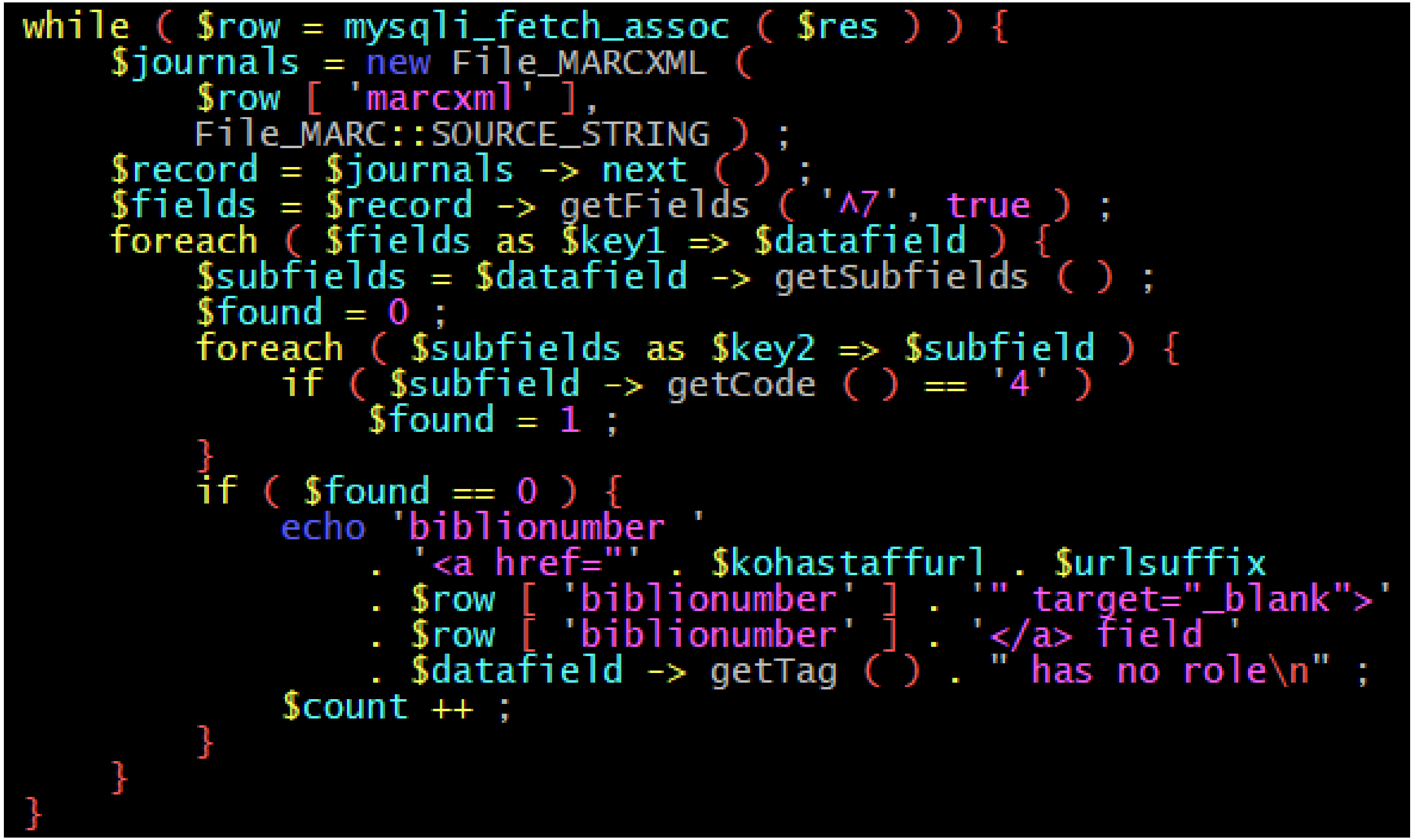


701 ? □ 0 - Personal Name - Alternative Responsibility

a	Entry Element	Velichkovskii
b	Part of Name Other than Entry Element	Paisii
c	Additions to Names Other than Dates	Saint
f	Dates	1722-1794
4	Relator Code	

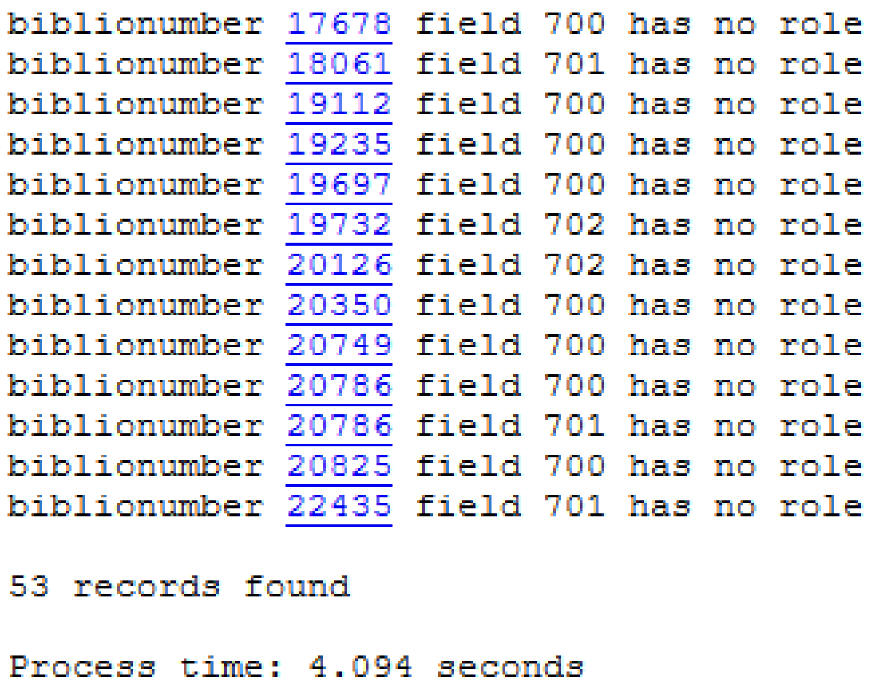
- Here the creator’s role code (author, translator, photographer, etc.) is missing (marked with the red rectangle)

Loading MARCXML data with File_MARC



```
while ( $row = mysqli_fetch_assoc ( $res ) ) {
    $journals = new File_MARCXML (
        $row [ 'marcxml' ],
        File_MARC::SOURCE_STRING ) ;
    $record = $journals -> next ( ) ;
    $fields = $record -> getFields ( '^7', true ) ;
    foreach ( $fields as $key1 => $datafield ) {
        $subfields = $datafield -> getSubfields ( ) ;
        $found = 0 ;
        foreach ( $subfields as $key2 => $subfield ) {
            if ( $subfield -> getCode ( ) == '4' )
                $found = 1 ;
        }
        if ( $found == 0 ) {
            echo 'biblionumber '
                . '<a href="' . $kohastaffurl . $urlsuffix
                . $row [ 'biblionumber' ] . '" target="_blank">'
                . $row [ 'biblionumber' ] . '</a> field '
                . $datafield -> getTag ( ) . " has no role\n" ;
            $count ++ ;
        }
    }
}
```

Displaying the results in the web browser



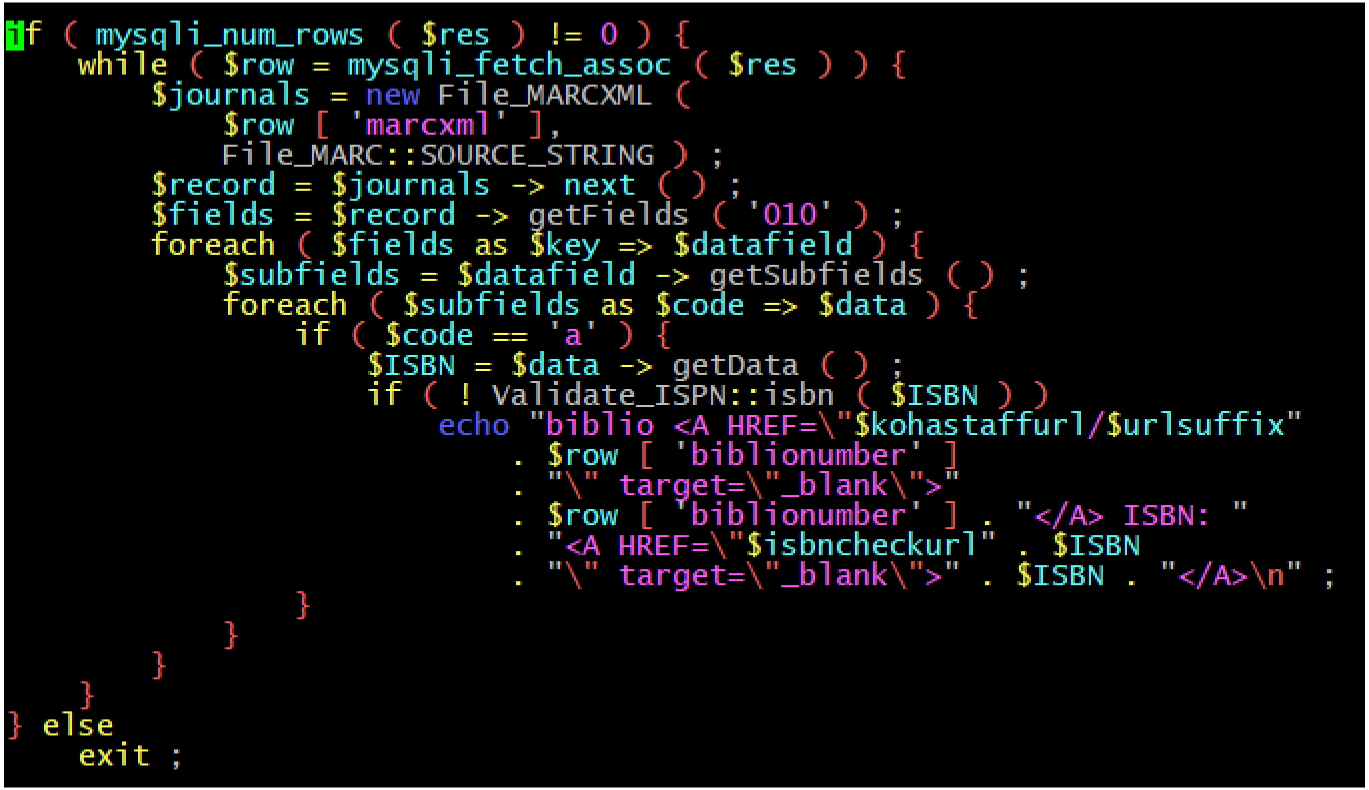
```
biblionumber 17678 field 700 has no role
biblionumber 18061 field 701 has no role
biblionumber 19112 field 700 has no role
biblionumber 19235 field 700 has no role
biblionumber 19697 field 700 has no role
biblionumber 19732 field 702 has no role
biblionumber 20126 field 702 has no role
biblionumber 20350 field 700 has no role
biblionumber 20749 field 700 has no role
biblionumber 20786 field 700 has no role
biblionumber 20786 field 701 has no role
biblionumber 20825 field 700 has no role
biblionumber 22435 field 701 has no role

53 records found

Process time: 4.094 seconds
```

- The hyperlink points to the Staff interface view for the particular biblio number

Technical challenge #1: ISBN validator

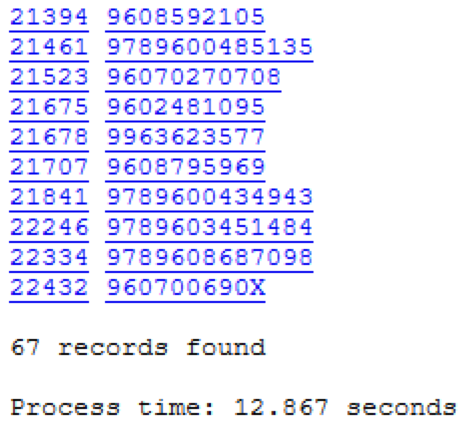
- In UNIMARC flavour, the ISBN is stored in field `010$a`
- A small typo when entering the ISBN can make it invalid
- Thankfully, a PHP package (`Validate_ISPN`) exists, that can check ISBNs for correctness
- Coupled with a lookup on http://www.isbn-check.de, the user can easily spot trivial mistakes

Code: detecting invalid ISBNs



```
if ( mysqli_num_rows ( $res ) != 0 ) {
    while ( $row = mysqli_fetch_assoc ( $res ) ) {
        $journals = new File_MARCXML (
            $row [ 'marcxml' ],
            File_MARC::SOURCE_STRING ) ;
        $record = $journals -> next ( ) ;
        $fields = $record -> getFields ( '010' ) ;
        foreach ( $fields as $key => $datafield ) {
            $subfields = $datafield -> getSubfields ( ) ;
            foreach ( $subfields as $code => $data ) {
                if ( $code == 'a' ) {
                    $ISBN = $data -> getData ( ) ;
                    if ( ! Validate_ISPN::isbn ( $ISBN ) )
                        echo "biblio <A HREF=\"$kohastaffurl/$urlsuffix"
                            . $row [ 'biblionumber' ]
                            . "\" target=\"_blank\">"
                            . $row [ 'biblionumber' ] . "</A> ISBN: "
                            . "<A HREF=\"$isbncheckurl" . $ISBN
                            . "\" target=\"_blank\">" . $ISBN . "</A>\n" ;
                }
            }
        }
    }
} else
    exit ;
```

Displaying the results in the web browser

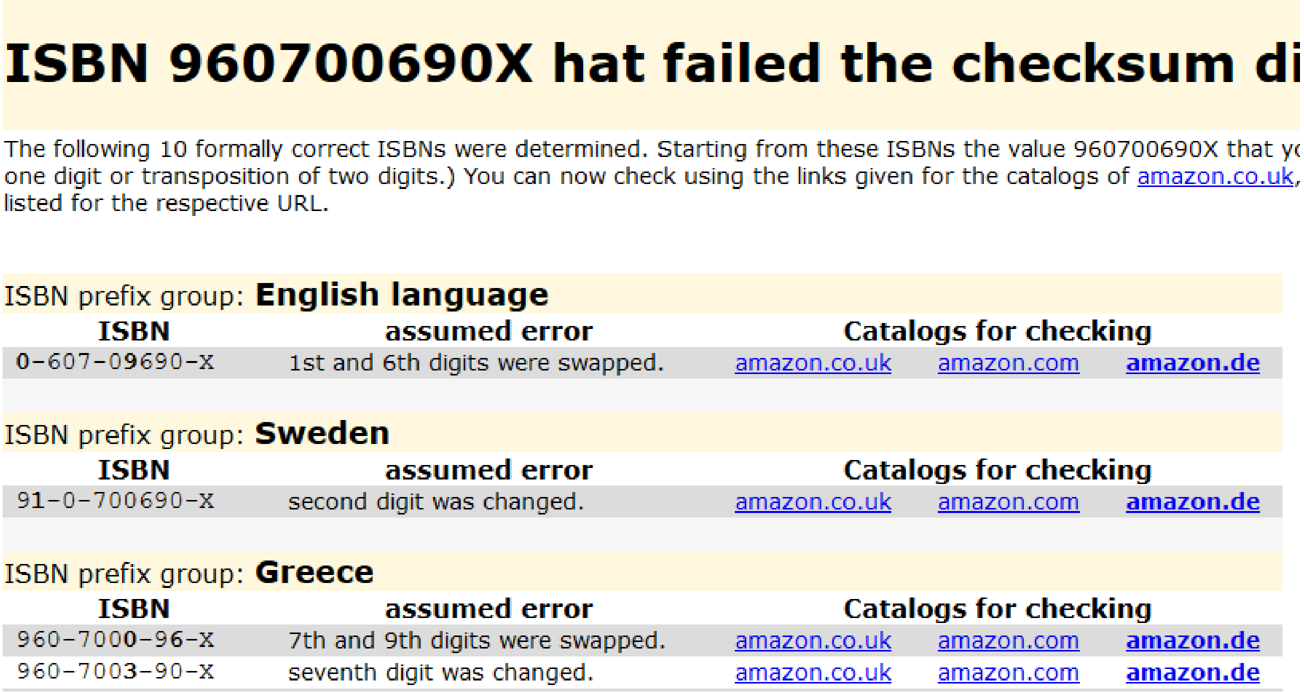


```
21394 9608592105
21461 9789600485135
21523 96070270708
21675 9602481095
21678 9963623577
21707 9608795969
21841 9789600434943
22246 9789603451484
22334 9789608687098
22432 960700690X

67 records found

Process time: 12.867 seconds
```

- The first hyperlink points to the Staff interface view for the particular biblio number

- The second link points to the www.isbn-check.de website for suspected errors

www.isbn-check.de



ISBN 960700690X hat failed the checksum di

The following 10 formally correct ISBNs were determined. Starting from these ISBNs the value 960700690X that yo
one digit or transposition of two digits.) You can now check using the links given for the catalogs of amazon.co.uk,
listed for the respective URL.

ISBN prefix group: **English language**

ISBN	assumed error	Catalogs for checking		
0-607-09690-X	1st and 6th digits were swapped.	amazon.co.uk	amazon.com	**amazon.de**

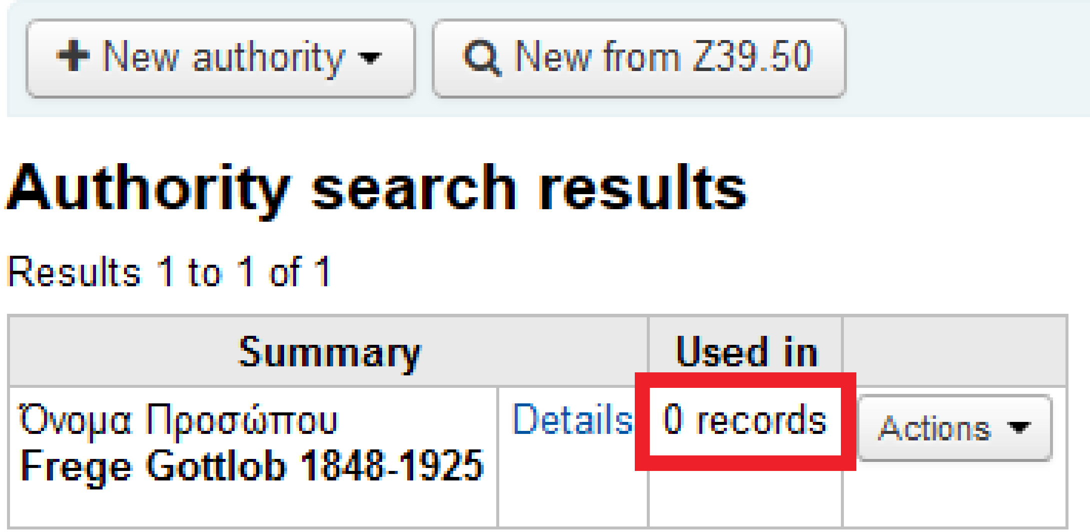
ISBN prefix group: **Sweden**

ISBN	assumed error	Catalogs for checking		
91-0-700690-X	second digit was changed.	amazon.co.uk	amazon.com	**amazon.de**

ISBN prefix group: **Greece**

ISBN	assumed error	Catalogs for checking		
960-7000-96-X	7th and 9th digits were swapped.	amazon.co.uk	amazon.com	**amazon.de**
960-7003-90-X	seventh digit was changed.	amazon.co.uk	amazon.com	**amazon.de**

Technical challenge #2: unused authority records



Authority search results

Results 1 to 1 of 1

Summary		Used in	
Όνομα Προσώπου **Frege Gottlob 1848-1925**	Details	0 records	Actions

Querying Zebra using the YAZ toolkit

- For this, we concluded that is faster to query Zebra to get information from our catalogue

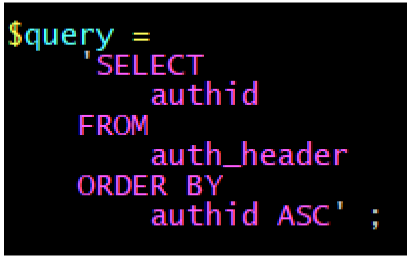
- With a few small changes in `/etc/koha/sites/<INSTANCE>/koha-conf.xml` you can set up your own Z39.50 server listening on localhost

- `yaz` is another PHP package from PECL (PHP Extension Community Library), implementing a Z39.50 client

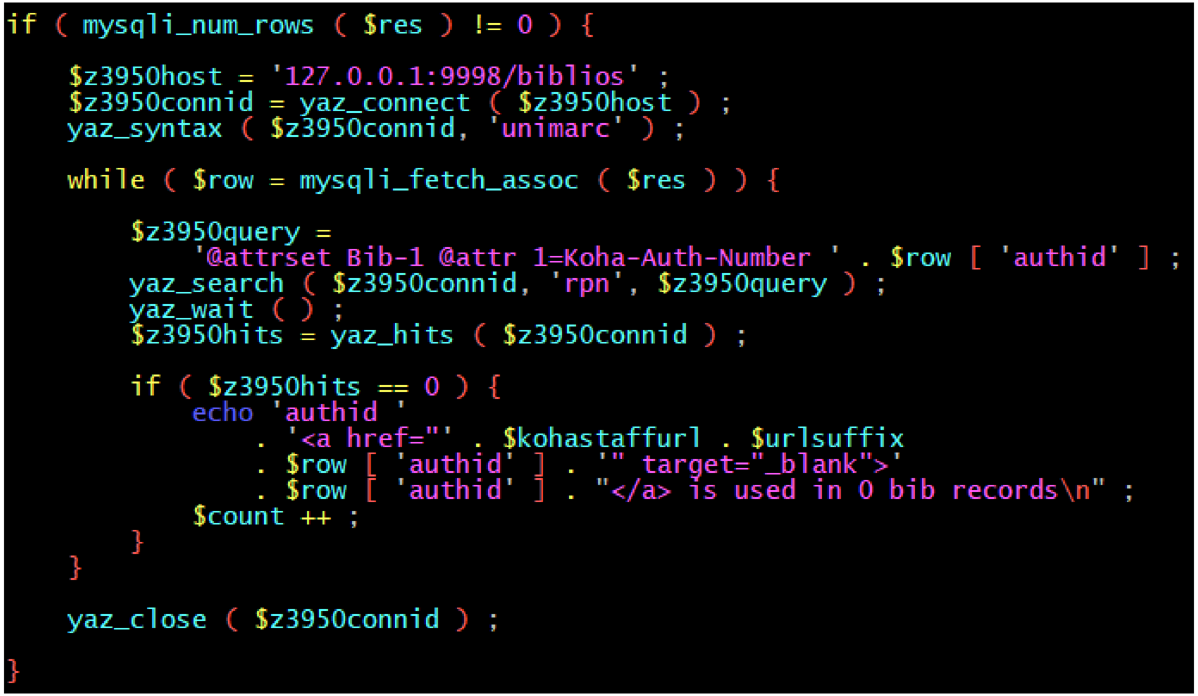
- The query we will be issuing is:

  ```
  @attrset Bib-1 @attr 1=Koha-Auth-Number $AUTHORITY_ID
  ```

Code: querying Zebra

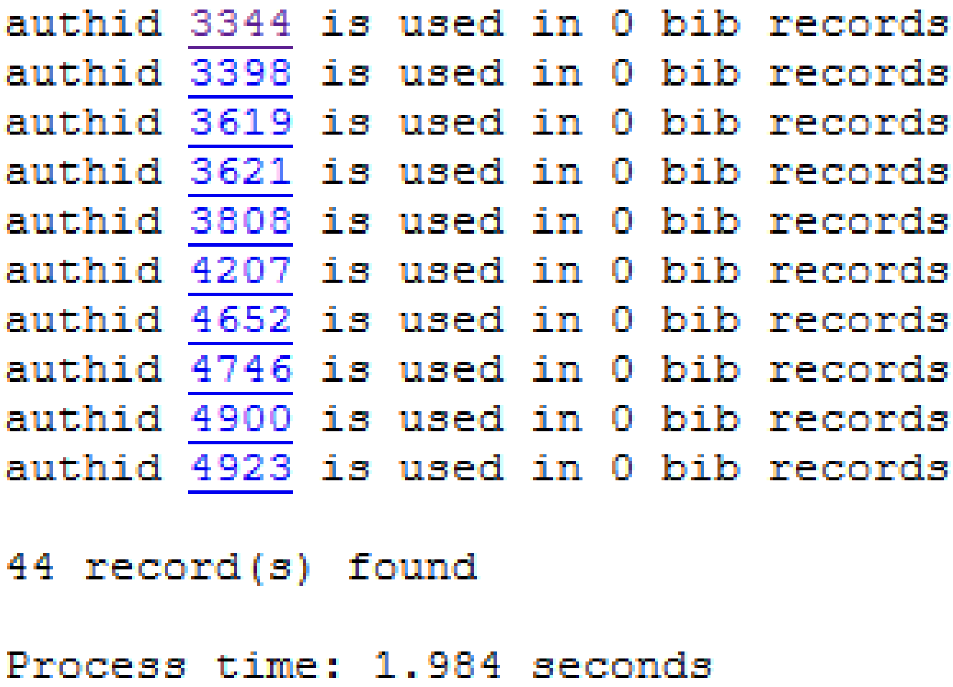


```
$query =
    'SELECT
        authid
    FROM
        auth_header
    ORDER BY
        authid ASC' ;
```



```
if ( mysqli_num_rows ( $res ) != 0 ) {

    $z3950host = '127.0.0.1:9998/biblios' ;
    $z3950connid = yaz_connect ( $z3950host ) ;
    yaz_syntax ( $z3950connid, 'unimarc' ) ;

    while ( $row = mysqli_fetch_assoc ( $res ) ) {

        $z3950query =
            '@attrset Bib-1 @attr 1=Koha-Auth-Number ' . $row [ 'authid' ] ;
        yaz_search ( $z3950connid, 'rpn', $z3950query ) ;
        yaz_wait ( ) ;
        $z3950hits = yaz_hits ( $z3950connid ) ;

        if ( $z3950hits == 0 ) {
            echo 'authid '
                . '<a href="' . $kohastaffurl . $urlsuffix
                . $row [ 'authid' ] . '" target="_blank">'
                . $row [ 'authid' ] . "</a> is used in 0 bib records\n" ;
            $count ++ ;
        }
    }

    yaz_close ( $z3950connid ) ;

}
```

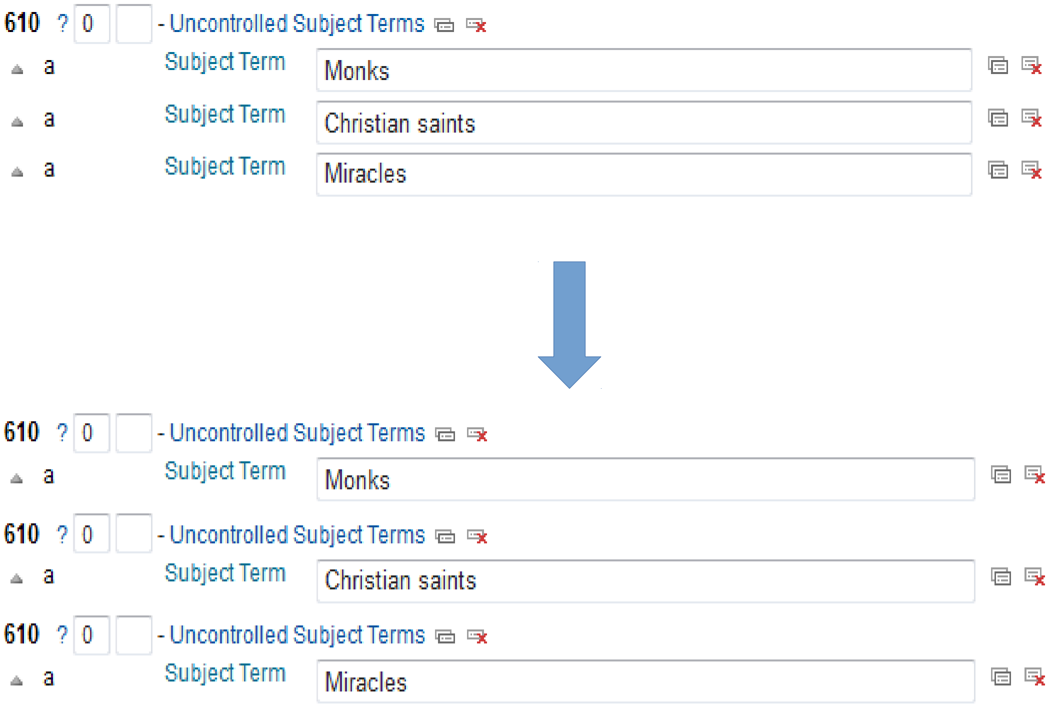
Displaying the results in the web browser



```
authid 3344 is used in 0 bib records
authid 3398 is used in 0 bib records
authid 3619 is used in 0 bib records
authid 3621 is used in 0 bib records
authid 3808 is used in 0 bib records
authid 4207 is used in 0 bib records
authid 4652 is used in 0 bib records
authid 4746 is used in 0 bib records
authid 4900 is used in 0 bib records
authid 4923 is used in 0 bib records

44 record(s) found

Process time: 1.984 seconds
```

- The hyperlink points to the authority details view in the Staff interface

Bibliographic challenge #3: repeatable ‘a’ subfields



610 ? 0 - Uncontrolled Subject Terms
a Subject Term Monks
a Subject Term Christian saints
a Subject Term Miracles

610 ? 0 - Uncontrolled Subject Terms
a Subject Term Monks
610 ? 0 - Uncontrolled Subject Terms
a Subject Term Christian saints
610 ? 0 - Uncontrolled Subject Terms
a Subject Term Miracles

- Following the migration from the old cataloguing software, single keyword subjects were inherited as repeatable 610 ‘a’ subfields
- Their hyperlinks returned results for all keywords as a string (heading), instead of the desired results for each keyword
- Very time-consuming and error-prone to fix by hand since it affected many biblios
- There was a need to globally correct the offending records

Bibliographic challenge #3: repeatable ‘a’ subfields

A

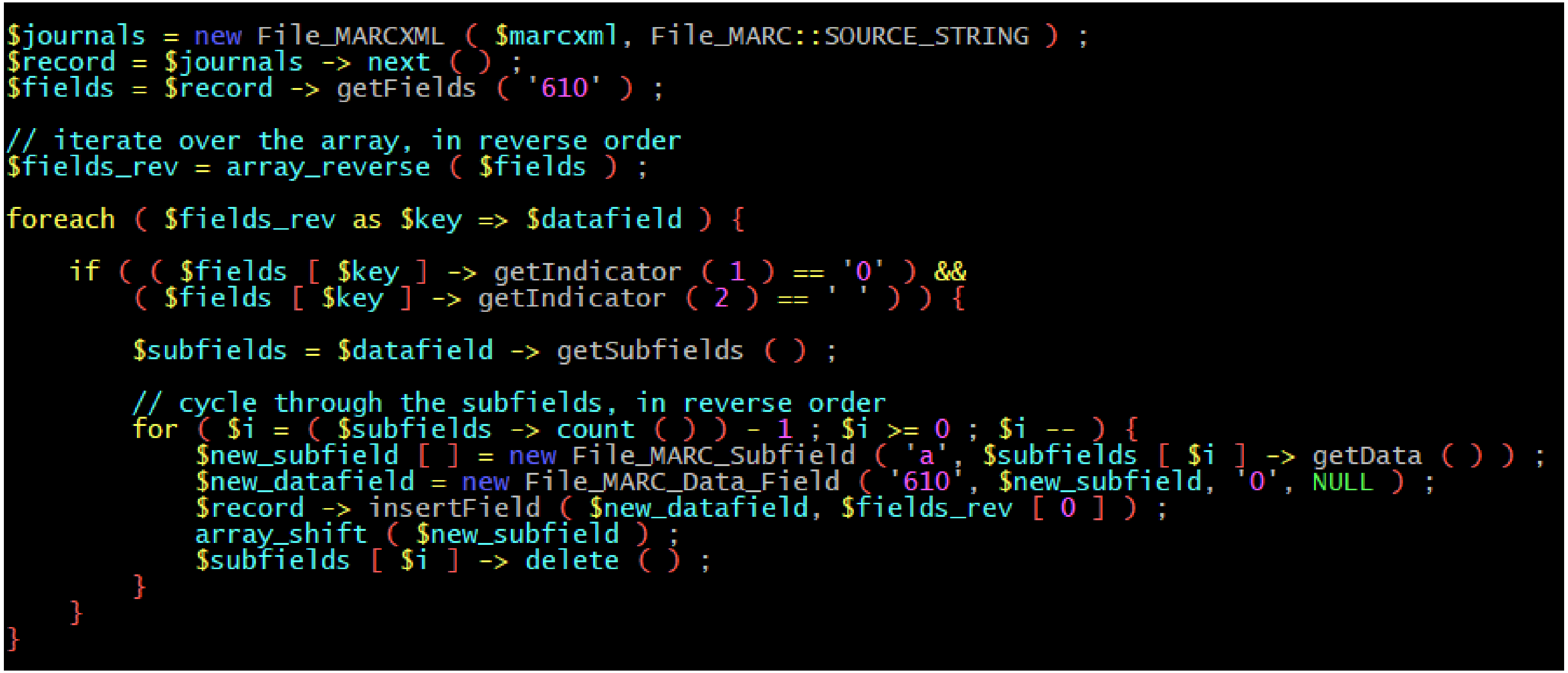
```
<datafield tag="610" ind1="0" ind2=" ">
  <subfield code="a">Monks</subfield>
  <subfield code="a">Christian saints</subfield>
  <subfield code="a">Miracles</subfield>
</datafield>
```

- How do you get from **A** to **B**?

B

```
<datafield tag="610" ind1="0" ind2=" ">
  <subfield code="a">Monks</subfield>
</datafield>
<datafield tag="610" ind1="0" ind2=" ">
  <subfield code="a">Christian saints</subfield>
</datafield>
<datafield tag="610" ind1="0" ind2=" ">
  <subfield code="a">Miracles</subfield>
</datafield>
```

Code: fixing multiple 610$a subfields



```
$journals = new File_MARCXML ( $marcxml, File_MARC::SOURCE_STRING ) ;
$record = $journals -> next ( ) ;
$fields = $record -> getFields ( '610' ) ;

// iterate over the array, in reverse order
$fields_rev = array_reverse ( $fields ) ;

foreach ( $fields_rev as $key => $datafield ) {

    if ( ( $fields [ $key ] -> getIndicator ( 1 ) == '0' ) &&
        ( $fields [ $key ] -> getIndicator ( 2 ) == ' ' ) ) {

        $subfields = $datafield -> getSubfields ( ) ;

        // cycle through the subfields, in reverse order
        for ( $i = ( $subfields -> count ( ) ) - 1 ; $i >= 0 ; $i -- ) {
            $new_subfield [ ] = new File_MARC_Subfield ( 'a', $subfields [ $i ] -> getData ( ) ) ;
            $new_datafield = new File_MARC_Data_Field ( '610', $new_subfield, '0', NULL ) ;
            $record -> insertField ( $new_datafield, $fields_rev [ 0 ] ) ;
            array_shift ( $new_subfield ) ;
            $subfields [ $i ] -> delete ( ) ;
        }
    }
}
```

The future

- Develop more scripts ;-) Current ideas include:
  - Detection of English characters ABEHIKMNOPTXYZ in otherwise Greek words
  - Auto-fill indicators `0 2` for `CORPO_NAME` type authorities

- Place repeated code (such as the MySQL connection parameters, the URLs pointing to Koha’s Staff interface, etc.) into a file (e.g. `common.php`) and have the scripts include it

- Include screenshots in GitHub’s `README.md` displaying the output of the scripts

- Attempt to re-write and package one of the smallest scripts as a Koha plugin

GitHub repository

- Most of the PHP scripts shown today are available at:

  https://github.com/a-roussos/php-koha

- More will be added in due course