Parsing Records from TSLAC ARIS Databases into XML: Notes

(Work Area B.2- B.4)

Danielle Plumer
Mark Philips
Serhiy Polyakov

December 15, 2006
Draft 2

Table of Contents

1. Introduction

One of the goals of THDI project is enhancing interoperability through standard-based support for collections provided by IMLS Grant Partners. There are several workflows currently being investigated by the project team. These workflows include an approach when SRU/SRW gateways are installed on each Digital Asset Management System (DAMS) providing live federated search. Another approach that may be appropriate for some DAMS is aggregating metadata from those DAMS into one location (OAI-PMH but not necessarily) and providing SRU/SRW interface for that location.

The project team is also exploring possibilities of using Lucene information retrieval API to improve search functionality. Regardless of the chosen approach, simplified data workflow may be described as DAMS -> Lucene -> SRU/SRW -> Texas Heritage Online.

There are several ways to ingest data from DAMS into Lucene index. One is developing Lucene connectors for DAMS that use JDBC. Another is exporting data from DAMS into XML files and then ingesting them from XML files using XML to Lucene parsers (have to be adopted). The latter way may bring common denominator for various DAMS because anyway some DAMS are accessible only via OAI-PMH.

This document discusses the process of converting data from TSLAC ARIS database into XML files, creating XML schemas, and mapping elements to simple Dublin Core schema.

2. ARIS databases

2.1. Copying database to the test server

ARIS database (in MySQL terms) consists of three databases:

- Republic Claims
- Texas Adjutant General Service Records 1836-1935
- Confederate Pensions

Each consists of one or more tables.

ARIS database at `devlot.tsl.state.tx.us` server has been exported into slq file and uploaded at the test server at `spmachine.lis.unt.edu`. This has been done in order to get full control over the data like running php scripts, adding some test records for debugging, etc.

2.2. Republic Claims Database

2.2.1. Description of the Database

The Republic Claims series of Comptroller's records includes claims for payment, reimbursement, or restitution submitted by citizens to the Republic of Texas government from 1835 through 1846. The Republic Claims documents are stored as PDF images. Metadata is stored in MySQL database and images are web accessible. Number of records in the database is 140,653. Native search interface and information about the database can be found at: http://www2.tsl.state.tx.us/trail/RepublicSearch.jsp Summary of database fields is presented in Table 1.

Table 1. Summary of database fields

Field name in database	Column name used in public interface	Data Type in Database	Note
claimant	Claimant Name	char	required
otherinfo	Name Mentioned	char	required
type	Type	char	required

Field name in database	Column name used in public interface	Data Type in Database	Note
claimno	Claim Number	char	required
reel	Reel	int	required
first	First	int	required
last	Last	int	required
id	id	int	required
	images		Generated (see rules below)

Rules for generating URLs for images.
Domain: tslarc.tsl.state.tx.us
Directory: repclaims
Subdirectory: [1 to 3-digit reel number]
Filename: [3-digit reel number][5-digit frame number].pdf
Sample: http://tslarc.tsl.state.tx.us/repclaims/1/00100007.pdf

It should be mentioned that each republic claim record may be represented by more than one image corresponding to pages or microfilm frames. Attribute `images` is actually multivalued. The order of the images is important because each image represents a page (frame).

2.2.2. Dumping data into XML

W3C XML Schema language (XSD) has been chosen for description of the XML documents resulting from converting database records into XML. An XML schema describes a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type, beyond the basic syntax constraints imposed by XML itself.

Each record from the Republic Claims database is dumped into separate XML document using php script run on the test server (see Appendix). XML schema for these documents is presented below.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="republic_claim" type="republic_claimType"/>
    <xsd:complexType name="republic_claimType">
        <xsd:sequence>
            <xsd:element name="claimant" type="xsd:string"/>
            <xsd:element name="otherinfo" type="xsd:string"/>
            <xsd:element name="type" type="xsd:string"/>
            <xsd:element name="claimno" type="xsd:string"/>
            <xsd:element name="reel" type="xsd:int"/>
            <xsd:element name="first" type="xsd:int"/>
            <xsd:element name="last" type="xsd:int"/>
            <xsd:element name="id" type="xsd:int"/>
            <xsd:element name="image_URL" maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:simpleContent>
                    <xsd:extension base='xsd:string'>
                    <xsd:attribute name='frame' type='xsd:int' use="required"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

Elements have the same names as fields in the database. Element `image_URL` also uses attribute `frame` that describes the order of images representing the entity (service record). Sample XML document is presented below.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<republic_claim xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="republic_claim.xsd">
    <claimant>Adams, James M.</claimant>
    <otherinfo>Logan, John D.</otherinfo>
    <type> PD</type>
    <claimno> 1393</claimno>
    <reel>131</reel>
    <first>78</first>
    <last>80</last>
    <id>10</id>
    <image_URL frame="78">http://tslarc.tsl.state.tx.us/repclaims/131/13100078.pdf</image_URL>
    <image_URL frame="79">http://tslarc.tsl.state.tx.us/repclaims/131/13100079.pdf</image_URL>
    <image_URL frame="80">http://tslarc.tsl.state.tx.us/repclaims/131/13100080.pdf</image_URL>
</republic_claim>
```

2.2.3. Mapping to Unqualified Dublin Core Element Set

Interoperable solution requires use of one schema for describing a variety of objects that will be accessible via Texas Heritage Online. OAI-PMH simple DC schema is chosen for these purposes.

Table 2. Mapping attributes to unqualified Dublin Core element set

Field name in database	**Column name used in public interface**	**DC**	**Note**
claimant	Claimant Name	title	
otherinfo	Name Mentioned	description	
type	Type	description	
claimno	Claim Number	identifier	
reel	Reel		concatenated
first	First		concatenated
last	Last		concatenated
id	id		concatenated
	images	identifier	

Resulting XML documents comply with OAI-PMH simple DC schema located at http://www.openarchives.org/OAI/2.0/oai_dc.xsd

Sample resulting XML document is presented below.

```
<?xml version="1.0" encoding="utf-8"?>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
    <dc:title>Adams, James M.</dc:title>
    <dc:description>Logan, John D.</dc:description>
    <dc:description> PD</dc:description>
    <dc:identifier> 1393</dc:identifier>
    <dc:identifier>http://tslarc.tsl.state.tx.us/repclaims/131/13100078.pdf</dc:identifier>
    <dc:identifier>http://tslarc.tsl.state.tx.us/repclaims/131/13100079.pdf</dc:identifier>
    <dc:identifier>http://tslarc.tsl.state.tx.us/repclaims/131/13100080.pdf</dc:identifier>
</oai_dc:dc>
```

There is one concern about preserving the order of images when using OAI-PMH simple DC schema because this schema does not prescribe usage of attributes for the elements.

2.3. Service Records

2.3.1. Description of the Database

The Service Records Series combines both official service record files from the Adjutant General's Office and alphabetical files created by other agencies which contain records related to an individual's service in a military unit. The documents are stored as PDF images. Metadata is stored in MySQL database and images are web accessible. Number of records in the database is 23,919. Native search interface and information about the database can be found at: http://www2.tsl.state.tx.us/trail/ServiceSearch.jsp Summary of database fields is presented in Tables 3 and 4.

Table 3. Summary of database fields: table 'servicerecord'

Field name in the database	Column name used in public interface	Data Type in Database	Note
id	-	int	required
name	Name	char	required
organization	Organization	char	required
call_no	Call Number	char	required
	Image Name		Generated (see rules below)

Table 4. Summary of database fields: table 'organization_codes'

Field name in the database	Column name used in public interface	Data Type in Database	Note
id	-	int	required
organization	-	char	required
title	-	char	required

Table 'organization_codes' is linked to table 'servicerecord' via field `organization`.

Rules for generating URLs for images.
Domain: tslarc.tsl.state.tx.us
Directory: service
Subdirectories: [2 or 3-letter organization code]
Subdirectories: [a-z] <-- first letter of last name -->
Subdirectories: [a-z][a-z] <-- first 2 letters of last name -->
Filename: [first 3 letters of last name][1-23920].pdf <-- records numbered sequentially, a-z -->
Sample: service/ARM/a/aa/aak1.pdf
Sample: service/SR/z/zw/zwe23920.pdf

2.3.2. Dumping data into XML

Each record from the Service Record database is dumped into separate XML document using php script run on the test server (see Appendix). XML schema for these documents is presented below.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
     <xsd:element name="service_record" type="service_recordType"/>
     <xsd:complexType name="service_recordType">
          <xsd:sequence>
               <xsd:element name="id" type="xsd:int"/>
               <xsd:element name="name" type="xsd:string"/>
               <xsd:element name="call_number" type="xsd:string"/>
               <xsd:element name="organization_code" type="xsd:string"/>
               <xsd:element name="organization_title" type="xsd:string"/>
               <xsd:element name="image_URL" type="xsd:string"/>
          </xsd:sequence>
     </xsd:complexType>
</xsd:schema>
```

Elements have the same names as fields in the database except '`organization`' is named '`organization_code'` and `'title'` is named `'organization_title`'. Sample XML document is presented below.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<service_record xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="service_record.xsd">
     <id>1</id>
     <name>Aake, Charles</name>
     <call_number>401-1</call_number>
     <organization_code>ARM</organization_code>
     <organization_title>Army of the Republic</organization_title>
     <image_URL>http://tslarc.tsl.state.tx.us/service/ARM/a/aa/aak1.pdf</image_URL>
</service_record>
```

2.3.3. Mapping to Unqualified Dublin Core Element Set

Table 5. Mapping attributes to unqualified Dublin Core element set

Field name in database	Column name used in public interface	DC	Note
id	-		concatenated
name	Name	title	
organization	Organization	description	
call_no	Call Number	identifier	
	Image Name	identifier	
id	-		concatenated
organization	-		concatenated
title	-	description	

Resulting XML documents comply with OAI-PMH simple DC schema located at http://www.openarchives.org/OAI/2.0/oai_dc.xsd

Sample resulting XML document is presented below.

```
<?xml version="1.0" encoding="utf-8"?>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
     <dc:title>Aake, Charles</dc:title>
     <dc:identifier>401-1</dc:identifier>
     <dc:description>ARM</dc:description>
     <dc:description>Army of the Republic</dc:description>
     <dc:identifier>http://tslarc.tsl.state.tx.us/service/ARM/a/aa/aak1.pdf</dc:identifier>
</oai_dc:dc>
```

2.4. Confederate Pensions

2.4.1. Description of the Database

The Index to Confederate Pension Applications provides the names, county of residence, and pension number of some 54,634 approved, rejected, and home pensions issued by the Texas government between 1899 and 1975.

The documents are supposed to be stored as PDF images but they are not yet accessible online. Metadata is stored in MySQL database. Number of records in the database is 54,571. Native search interface and information about the database can be found at:
http://www.tsl.state.tx.us/arc/pensions/index.html
Summary of database fields is presented in Tables 6.

Table 6. Summary of database fields: table 'servicerecord'

Field name in the database	Column name used in public interface	Data Type in Database	Note

Field name in the database	Column name used in public interface	Data Type in Database	Note
name	Claimant Name	char	required
number	Application Number	char	required
county	County	char	optional
husband	Husband	char	optional
husband_number	Husband's Application Number	char	optional
id	-	int	required
	Image Name		Generated (see rules below)

Rules for generating URLs for images.
Domain: tslarc.tsl.state.tx.us <-- only a few images exist; not yet web-accessible -->
Directory: pensions
Subdirectories: [a-z] <-- first letter of last name -->
Subdirectories: [a-z][a-z] <-- first 2 letters of last name -->
Filename: [first 3 letters of last name][6-digit unique id].pdf

Sample: pensions/a/aa/aar006249.pdf

2.4.2. Dumping data into XML

Each record from the Service Record database is dumped into separate XML document using php script run on the test server (see Appendix). XML schema for these documents is presented below.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
     <xsd:element name="pension" type="pensionType"/>
     <xsd:complexType name="pensionType">
          <xsd:sequence>
               <xsd:element name="id" type="xsd:int"/>
               <xsd:element name="name" type="xsd:string"/>
               <xsd:element name="number" type="xsd:string"/>
               <xsd:element name="county" type="xsd:string" minOccurs="0" />
               <xsd:element name="husband" type="xsd:string"  minOccurs="0" />
               <xsd:element name="husband_number" type="xsd:string" minOccurs="0" />
               <xsd:element name="image_URL" type="xsd:string"/>
          </xsd:sequence>
     </xsd:complexType>
</xsd:schema>
```

Elements have the same names as fields in the database. Sample XML document is presented below.

```
<?xml version="1.0" encoding="utf-8"?>
<pension xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="pension.xsd">
     <id>3</id>
     <name>Bell, M. E. (Mrs)</name>
     <number>home</number>
     <county>Travis</county>
     <husband>Bell, Joseph Athel</husband>
     <husband_number>10584</husband_number>
     <image_URL>http://tslarc.tsl.state.tx.us/pensions/b/be/bel3.pdf</image_URL>
</pension>
```

2.4.3. Mapping to Unqualified Dublin Core Element Set

Table 7 Mapping attributes to unqualified Dublin Core element set

Field name in database	Column name used in public interface	DC	Note

Field name in database	**Column name used in public interface**	**DC**	**Note**
name	Claimant Name	title	
number	Application Number	description	
county	County	description	optional
husband	Husband	description	optional
husband_number	Husband's Application Number	description	optional
id	-	-	concatenated
	Image Name	identifier	

Resulting XML documents comply with OAI-PMH simple DC schema located at http://www.openarchives.org/OAI/2.0/oai_dc.xsd

Sample resulting XML document is presented below. Note: some elements are optional.

```
<?xml version="1.0" encoding="utf-8"?>
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
    <dc:title>Bell, M. E. (Mrs)</dc:title>
    <dc:description>home</dc:description>
    <dc:description>Travis</dc:description>
    <dc:description>Bell, Joseph Athel</dc:description>
    <dc:description>10584</dc:description>
    <dc:identifier>http://tslarc.tsl.state.tx.us/pensions/b/be/bel3.pdf</dc:identifier>
</oai_dc:dc>
```

Appendix: Script to dump ARIS database records into original XML and simple OAI-DC XML.

Republic Claims

```
<?php

$conn = mysql_connect("localhost", "[login_here]", "[password_here]");
mysql_select_db("aris", $conn);

//remove limit 0, 1000 later

$sql="select concat('republic_claim', LPAD(id, 6, '0'), '.xml') as XML_File, claimant, otherinfo,
type, claimno, reel, first, last, id from rclaims limit 1000";

$result=mysql_query($sql, $conn) or die(mysql_error());

while ($rowArr = mysql_fetch_array($result))
{
     // Write in original DB schema:

     $fh = fopen('/var/www/html/thdi/aris/republic_claims/'.$rowArr['XML_File'], "w") or
die("Error!!");

     fwrite($fh, '<?xml version="1.0" encoding="utf-8"?>'."\r\n");

     fwrite($fh, '<republic_claim xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="republic_claim.xsd">'."\r\n");

     fwrite($fh, '<claimant>'.$rowArr['claimant'].'</claimant>'."\r\n");
     fwrite($fh, '<otherinfo>'.$rowArr['otherinfo'].'</otherinfo>'."\r\n");
     fwrite($fh, '<type>'.$rowArr['type'].'</type>'."\r\n");
     fwrite($fh, '<claimno>'.$rowArr['claimno'].'</claimno>'."\r\n");
     fwrite($fh, '<reel>'.$rowArr['reel'].'</reel>'."\r\n");
     fwrite($fh, '<first>'.$rowArr['first'].'</first>'."\r\n");
     fwrite($fh, '<last>'.$rowArr['last'].'</last>'."\r\n");
     fwrite($fh, '<id>'.$rowArr['id'].'</id>'."\r\n");

     for ($i = $rowArr['first']; $i <= $rowArr['last']; $i = $i+1)
     {
          fwrite($fh, '<image_URL frame="'.$i.'">');

          $nnn =
'http://tslarc.tsl.state.tx.us/repclaims/'.$rowArr['reel'].'/'.str_pad($rowArr['reel'], 3, "0",
STR_PAD_LEFT).str_pad($i, 5, "0", STR_PAD_LEFT).'.pdf';

          fwrite($fh, $nnn);

          fwrite($fh, '</image_URL>'."\r\n");
     }

     fwrite($fh, '</republic_claim>'."\r\n");

     fclose($fh);


     // OAI simple DC schema:

     $fh = fopen('/var/www/html/thdi/aris/republic_claims_oai_dc/'.$rowArr['XML_File'], "w") or
die("Error!!");

     fwrite($fh, '<?xml version="1.0" encoding="utf-8"?>'."\r\n");

     fwrite($fh, '<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">'."\r\n");

     fwrite($fh, '<dc:title>'.$rowArr['claimant'].'</dc:title>'."\r\n");
     fwrite($fh, '<dc:description>'.$rowArr['otherinfo'].'</dc:description>'."\r\n");
     fwrite($fh, '<dc:description>'.$rowArr['type'].'</dc:description>'."\r\n");
     fwrite($fh, '<dc:identifier>'.$rowArr['claimno'].'</dc:identifier>'."\r\n");

     for ($i = $rowArr['first']; $i <= $rowArr['last']; $i = $i+1)
     {
          fwrite($fh, '<dc:identifier>');

          $nnn =
'http://tslarc.tsl.state.tx.us/repclaims/'.$rowArr['reel'].'/'.str_pad($rowArr['reel'], 3, "0",
STR_PAD_LEFT).str_pad($i, 5, "0", STR_PAD_LEFT).'.pdf';

          fwrite($fh, $nnn);

          fwrite($fh, '</dc:identifier>'."\r\n");
     }

     fwrite($fh, '</oai_dc:dc>'."\r\n");

     fclose($fh);

}

?>
```

Service Records

```
<?php

$conn = mysql_connect("localhost", "[login_here]", "[password_here]");
mysql_select_db("aris", $conn);

## output file and image path and name are generated here in sql (can do in php)

$sql="select concat('service_record', LPAD(servicerecord.id, 5, '0'), '.xml') as XML_File,
servicerecord.id as id, servicerecord.name as name, servicerecord.call_no as call_number,
organization_codes.organization as organization_code, organization_codes.title as
organization_title, concat('http://tslarc.tsl.state.tx.us/service/',
organization_codes.organization, '/', lower(substring(servicerecord.name, 1, 1)), '/',
lower(substring(servicerecord.name, 1, 2)), '/', lower(substring(servicerecord.name, 1, 3)),
servicerecord.id, '.pdf') as image_URL from servicerecord left outer join organization_codes ON
servicerecord.organization = organization_codes.organization limit 1000";


$result=mysql_query($sql, $conn) or die(mysql_error());

while ($rowArr = mysql_fetch_array($result))
{

     // Original DB schema


     $fh = fopen('/var/www/html/thdi/aris/service_records/'.$rowArr['XML_File'], "w") or
die("Error!!");

     fwrite($fh, '<?xml version="1.0" encoding="utf-8"?>'."\r\n");

     fwrite($fh, '<service_record xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="service_record.xsd">'."\r\n");


     fwrite($fh, '<id>'.$rowArr['id'].'</id>'."\r\n");
     fwrite($fh, '<name>'.$rowArr['name'].'</name>'."\r\n");
     fwrite($fh, '<call_number>'.$rowArr['call_number'].'</call_number>'."\r\n");
     fwrite($fh,
'<organization_code>'.$rowArr['organization_code'].'</organization_code>'."\r\n");
     fwrite($fh,
'<organization_title>'.$rowArr['organization_title'].'</organization_title>'."\r\n");
     fwrite($fh, '<image_URL>'.$rowArr['image_URL'].'</image_URL>'."\r\n");


     fwrite($fh, '</service_record>'."\r\n");

     fclose($fh);



     // OAI simple DC schema

     $fh = fopen('/var/www/html/thdi/aris/service_records_oai_dc/'.$rowArr['XML_File'], "w") or
die("Error!!");

     fwrite($fh, '<?xml version="1.0" encoding="utf-8"?>'."\r\n");

     fwrite($fh, '<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">'."\r\n");

     fwrite($fh, '<dc:title>'.$rowArr['name'].'</dc:title>'."\r\n");
     fwrite($fh, '<dc:identifier>'.$rowArr['call_number'].'</dc:identifier>'."\r\n");
     fwrite($fh, '<dc:description>'.$rowArr['organization_code'].'</dc:description>'."\r\n");
     fwrite($fh, '<dc:description>'.$rowArr['organization_title'].'</dc:description>'."\r\n");
     fwrite($fh, '<dc:identifier>'.$rowArr['image_URL'].'</dc:identifier>'."\r\n");

     fwrite($fh, '</oai_dc:dc>'."\r\n");

     fclose($fh);


}


?>
```

Confederate Pensions

```
<?php

$conn = mysql_connect("localhost", "[login_here]", "[password_here]");
mysql_select_db("aris", $conn);

## output file and image path and name are generated here in sql (can do in php)

$sql="select concat('pension', LPAD(pensions.id, 5, '0'), '.xml') as XML_File, pensions.name as
name, pensions.number as number, pensions.county as county, pensions.husband as husband,
pensions.husband_number as husband_number, pensions.id as id,
concat('http://tslarc.tsl.state.tx.us/pensions/', lower(substring(pensions.name, 1, 1)), '/',
lower(substring(pensions.name, 1, 2)), '/', lower(substring(pensions.name, 1, 3)), pensions.id,
'.pdf') as image_URL from pensions limit 1000";


$result=mysql_query($sql, $conn) or die(mysql_error());

while ($rowArr = mysql_fetch_array($result))
{

     // Original DB schema


     $fh = fopen('/var/www/html/thdi/aris/pensions/'.$rowArr['XML_File'], "w") or die("Error!!");

     fwrite($fh, '<?xml version="1.0" encoding="utf-8"?>'."\r\n");
```

```
    fwrite($fh, '<pension xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="pension.xsd">'."\r\n");


    fwrite($fh, '<id>'.$rowArr['id'].'</id>'."\r\n");
    fwrite($fh, '<name>'.$rowArr['name'].'</name>'."\r\n");
    fwrite($fh, '<number>'.$rowArr['number'].'</number>'."\r\n");
    if ($rowArr['county'] != '')
    {
        fwrite($fh, '<county>'.$rowArr['county'].'</county>'."\r\n");
    }
    if ($rowArr['husband'] != '')
    {
        fwrite($fh, '<husband>'.$rowArr['husband'].'</husband>'."\r\n");
    }
    if ($rowArr['husband_number'] != '')
    {
        fwrite($fh, '<husband_number>'.$rowArr['husband_number'].'</husband_number>'."\r\n");
    }
    fwrite($fh, '<image_URL>'.$rowArr['image_URL'].'</image_URL>'."\r\n");


    fwrite($fh, '</pension>'."\r\n");

    fclose($fh);



    // OAI simple DC schema

    $fh = fopen('/var/www/html/thdi/aris/pensions_oai_dc/'.$rowArr['XML_File'], "w") or
die("Error!!");

    fwrite($fh, '<?xml version="1.0" encoding="utf-8"?>'."\r\n");

    fwrite($fh, '<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">'."\r\n");

    fwrite($fh, '<dc:title>'.$rowArr['name'].'</dc:title>'."\r\n");
    fwrite($fh, '<dc:description>'.$rowArr['number'].'</dc:description>'."\r\n");
    if ($rowArr['county'] != '')
    {
        fwrite($fh, '<dc:description>'.$rowArr['county'].'</dc:description>'."\r\n");
    }
    if ($rowArr['husband'] != '')
    {
        fwrite($fh, '<dc:description>'.$rowArr['husband'].'</dc:description>'."\r\n");
    }
    if ($rowArr['husband_number'] != '')
    {
        fwrite($fh, '<dc:description>'.$rowArr['husband_number'].'</dc:description>'."\r\n");
    }
    fwrite($fh, '<dc:identifier>'.$rowArr['image_URL'].'</dc:identifier>'."\r\n");

    fwrite($fh, '</oai_dc:dc>'."\r\n");

    fclose($fh);

}


?>
```