View on GitHub Download v2-beta6

Database Preservation

Relational databases are one of the most important technologies supporting today's information management activities. They are designed to store, organize and explore digital records that not only support but also document day-to-day business operations. Very often, these records are irreplaceable or prohibitively expensive to reacquire by other means rendering the preservation of databases a serious concern.

This page focus on workflows, tools and standards to allow information managers to extract, archive and preserve records of information currently managed by relational databases.

The most relevant initiatives in this context are the Database Preservation Toolkit, the Database Visualization Toolkit and the SIARD 2.0 preservation format.

The following screencast aims to illustrate how all these tools fit together in a full-cycle archiving and preservation workflow for relational databases.

More detailed information about these tools and standards can be found on the following sections.

RODA – A digital repository made for preservation

RODA is a complete digital repository solution that delivers functionality for all the main functional units of the OAIS reference model. RODA is capable of ingesting, managing and providing access to the various types of digital content produced by large corporations or public bodies. RODA is based on open-source technologies and is supported by existing standards such as the Open Archival Information System (OAIS), Metadata Encoding and Transmission Standard (METS), Encoded Archival Description (EAD), Dublin Core (DC) and PREMIS (Preservation Metadata).

For more information please visit https://github.com/keeps/roda

RODA-in – The ultimate SIP creation tool

RODA-in is a tool specially designed for producers and archivists to create Submission Information Packages (SIP) ready to be submitted to an Open Archival Information System (OAIS). The tool creates SIPs from files and folders available on the local file system.

In version 2 we revolutionized the way SIPs are created to satisfy the need for mass processing of data. In this version you can create thousands of valid SIPs with just a few clicks, complete with data and metadata.

The tool includes features such as:

- Create, load and edit classification schemas
- Automatic association of files/folders to SIP
- Automatic association of metadata to SIP
- Definition of metadata templates
- Support for various metadata formats (EAD, DC, etc.)
- Creation of SIP of unlimited size
- Creation of SIP in various formats: BagIt and E-ARK

For more information please visit https://github.com/keeps/roda-in

Database Preservation Toolkit

The Database Preservation Toolkit allows conversion between Database formats, including connection to live systems, for purposes of digitally preserving databases. The toolkit allows conversion of live or backed-up databases into preservation formats such as **SIARD**, a XML-based format created for the purpose of database preservation. The toolkit also allows conversion of the preservation formats back into live systems to allow the full functionality of databases.

This toolkit was part of the RODA project and now has been released as a project by its own due to the increasing interest on this particular feature. It is now being further developed in the EARK project together with a new version of the SIARD preservation format.

The toolkit is created as a platform that uses input and output modules. Each module supports read and/or write to a particular database format or live system. New modules can easily be added by implementation of a new interface and adding of new drivers.

EARK and SIARD 2.0

A new version of the this tool, together with a new version of the SIARD preservation format, is currently being designed and developed on the EARK project. Meanwhile, if you'd like to know more and even send us use cases and requirements, contact us.

Database Visualization Toolkit

The Database Visualization Toolkit is a lightweight web viewer for relational databases, specially if preserved in SIARD 2, that uses SOLR as a backend, and allows browsing, search, and export. It uses the Database Preservation Toolkit to process new relational databases that are in the SIARD2 format or on the original live DBMS.

For more information please visit http://visualization.database-preservation.com

How to use

To use the program, open a command-line and try out the following command (replace x.y.z accordingly to the version of the binary in use):

```
$ java -jar dbptk-app-X.Y.Z.jar
```

Using this command you will be presented with the application usage, describing all supported modules and their parameters. This information is also available in the application usage page.

To use the application an input and an output module must be selected and some configuration parameters must be provided.

Supported Database Management Systems and Preservation formats

The Database Preservation Toolkit supports the following Database Management Systems:

- MySQL/MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server
- Microsoft Access
- And other databases (using JDBC)

Database Preservation Toolkit can convert any of the above DBMS to the following preservation formats:

- SIARD 1
- SIARD 2
- SIARD DK

The Database Preservation Toolkit is also capable of loading preserved databases into any of the above DBMS.

Examples

If you want to connect to a live MySQL database and export its content to SIARD 2.0 format, you can use the following command.

```
$ java -jar dbptk-app-x.y.z.jar \
--import mysql --import-hostname=localhost --import-database="example_db" --import-username=username --import-password="p4ssw0rd" \
--export siard-2 --export-file=example.siard
```

Or using the equivalent short version of the parameters:

```
$ java -jar dbptk-app-x.y.z.jar \
-i mysql -ih localhost -idb "example_db" -iu username -ip "p4ssw0rd" \
-e siard-2 -ef example.siard
```

More examples containing only required parameters:

Oracle to SIARD 2

```
$ java -jar dbptk-app-x.y.z.jar \
--import oracle --import-server-name=127.0.0.1 --import-database="example_db" --import-username=username --import-password="p4ssw0rd" --import
--export siard-2 --export-file=example.siard
```

MySQL to SIARD 2

```
$ java -jar dbptk-app-x.y.z.jar \
--import mysql --import-hostname=localhost --import-database="example_db" --import-username=username --import-password="p4ssw0rd" \
--export siard-2 --export-file=example.siard
```

PostgreSQL to SIARD 2

```
$ java -jar dbptk-app-x.y.z.jar \
--import postgresql --import-hostname=localhost --import-database="example_db" --import-username=username --import-password="p4ssw0rd" \
--export siard-2 --export-file=example.siard
```

Microsoft SQL Server to SIARD 2

```
$ java -jar dbptk-app-x.y.z.jar \
--import microsoft-sql-server --import-server-name=localhost --import-database="example_db" --import-username=username --import-password="p4ss
--export siard-2 --export-file=example.siard
```

The conversion in the opposite direction is also possible, check the complete application usage to know more about the supported modules and respective configurations.

How to use JDBC import and export modules

To use Database Preservation Toolkit with an unsupported database, one can connect by providing the name of the JDBC driver class (and adding the JDBC driver to the classpath) and the JDBC connection string. The steps to run Database Preservation Toolkit this way are as follows:

1. Obtain the JDBC driver for the database you want to use (this is typically a file with `jar` extension). For Oracle12C this file can be downloaded from http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html;
2. Identify the driver class. For Oracle 12C this would be something like `oracle.jdbc.driver.OracleDriver`;
3. Prepare the connection string. For Oracle 12C this could be something like `jdbc:oracle:thin:username/password@serverName:port/database`;
4. Run Database Preservation Toolkit by providing files to add to the classpath and the main entry point.

Please be aware that using this method the conversion quality cannot be assured, as it depends on the used driver. Furthermore, non-tested drivers are more prone to possible errors during the conversion. A specialized module for the database, if available, would always be preferable to this generic JDBC module.

Example to convert from Oracle to SIARD2:

Using the method described above, the Windows command to extract a database from an Oracle database to SIARD 2 is as the following:

```
java -cp "C:\path\to\dbptk-app-x.y.z.jar;C:\path\to\jdbc_driver.jar" com.databasepreservation.Main \
  --import=jdbc --import-driver=oracle.jdbc.driver.OracleDriver \
    --import-connection="jdbc:oracle:thin:username/password@serverName:port/database" \
  -e siard-2 -ef C:\path\to\output.siard
```

And on Linux the equivalent command would be (note that the jarfile separator is `:` instead of `;`):

```
java -cp "/path/to/dbptk-app-x.y.z.jar:/path/to/jdbc_driver.jar" com.databasepreservation.Main \
  --import=jdbc --import-driver=oracle.jdbc.driver.OracleDriver \
    --import-connection="jdbc:oracle:thin:username/password@serverName:port/database" \
  -e siard-2 -ef /path/to/output.siard
```

How to build from source

1. Download the latest stable release.
2. Unzip and open the folder on a command-line terminal
3. Build with Maven `mvn clean package`

Binaries will be on the `target` folder

Related publications & presentations

- Presentation "Database migration: CLI" by José Ramalho at "A Pratical Approach to Database Archiving", Danish National Archives, Copenhagen, Denmark, 2012-02-07.
- Presentation "RODA: a service-oriented digital repository: database archiving" by José Ramalho at "A Pratical Approach to Database Archiving", Danish National Archives, Copenhagen, Denmark, 2012-02-07.
- Presentation "RODA - Repository of Authentic Digital Objects" by Luis Faria at the International Workshop on Database Preservation, Edinburgh, 2007.
- José Carlos Ramalho, Relational database preservation through XML modelling, in proceedings of the International Workshop on Markup of Overlapping Structures (Extreme Markup 2007), Montréal, Canada, 2007.
- Marta Jacinto, Bidirectional conversion between XML documents and relational data bases, in proceedings of the International Conference on CSCW in Design, Rio de Janeiro, 2002.
- Ricardo Freitas, Significant properties in the preservation of relational databases, Springer, 2010.

Other related publications:

- Neal Fitzgerald, "Using data archiving tools to preserve archival records in business systems – a case study", in proceedings of iPRES 2013, Lisbon, 2013.

Troubleshooting

Getting exception "java.net.ConnectException: Connection refused"

Most databases are not configured by default to allow TCP/IP connections. Check your database configuration if it accepts TCP/IP connection and if your IP address is allowed to connect. Also, ensure that the user has permissions to access the database from your IP address.

Problems importing from Microsoft Access

To import from Microsoft Access you need to be on a Windows machine with Microsoft Access installed. This is because the current Microsoft Access import module is implemented using ODBC connection. Therefore, you need Windows installed to be able to use ODBC. Also, you need Microsoft Access installed so its ODBC driver is installed on your system.

Furthermore, in order to extract DB structures we need to have access to the internal database table `Msysrelationships`. You need to perform some hacking over the DBMS and this is version dependent. Please follow the instructions described on Microsoft's white paper, which explains how to do this for all Microsoft Access versions: "Preparing a Microsoft Access Database for Migration".

Got error "java.lang.OutOfMemoryError: Java heap space"

The toolkit might need more memory than it is available by default (normally 64MB). To increase the available memory use the `-Xmx` option. For example, the following command will increase the heap size to 3 GB.

```
$ java -Xmx3g -jar dbptk-app-x.y.z.jar ...
```

The toolkit needs enough memory to put the table structure definition in memory (not the data) and to load each data row or row set, which might include having some BLOBs completely in memory, but this depends on the database driver implementation.

Main hard drive gets full due to temporary files

Due to the structure of some export modules (e.g. SIARD) and because we only want to pass throught the database once with minimum amount of used memory, all BLOBs and CLOBs of a database table must be kept on temporary files during the export of a table. This can cause your main disk to get full and the execution to fail. To select a diferent folder for the temporary files, e.g. on a bigger hard drive, use the option `-Djava.io.tmpdir=/path/to/tmpdir`. For example, the following command will use the folder `/media/BIGHD/tmp` as the temporary folder:

```
$ java -Djava.io.tmpdir=/media/BIGHD/tmp -jar dbptk-app-x.y.z.jar ...
```

Information & Commercial support

For more information or commercial support, contact KEEP SOLUTIONS.

Development build passing

To develop we recommend the use of Maven and Eclipse (or Intellij with Eclipse Code Formatter plugin).

The following plugins should be installed in Eclipse:

- ANSI Escape in Console to have coloured output in tests

And the following environment variables should be set:

- **DPT_MYSQL_USER** - MySQL user that must be able to create new users and give them permissions (uses 'root' if not defined)
- **DPT_MYSQL_PASS** - MySQL user's password (uses blank password if not defined)
- **DPT_POSTGRESQL_USER** - PostgreSQL user that must be able to create new users and give them permissions (uses 'postgres' if not defined)
- **DPT_POSTGRESQL_PASS** - PostgreSQL user's password (uses blank password if not defined)

To run PostgreSQL tests, a local PostgreSQL database is required and *postgres* user or another user with permission to create new databases and users can be used. This user must be accessible by IP connection on localhost. The access can be tested with `psql -U username -h 127.0.0.1 -d postgres -W`.

To run MySQL tests, a local MySQL (or MariaDB) database is required and 'root' user or another user with permission to create new databases and users can be used. This user must be accessible by IP connection on localhost. The access can be tested with `mysql --user="username" -p --database="mysql" --host="127.0.0.1"`.

Building common parts that may be used by other projects

Use `mvn clean install -Pcommon` to locally install the common artifacts so they can be used by other projects. Note that this is not necessary unless you do not have access to KEEPS Artifactory or you want to make changes to the common artifacts to use in other projects.

Changing XML Schema files

After changing SIARD XML Schema files, maven must be used to compile a new artifact from the XML Schema (using JAXB). To do this, run `mvn clean install -Pdbptk-bindings` from project root folder. This will install the artifacts locally and they will be used instead of the ones in KEEPS Artifactory.

License

Database Preservation Toolkit licence is LGPLv3

```
                   GNU LESSER GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.


  This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

  0. Additional Definitions.

  As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

  "The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

  An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

  A "Combined Work" is a work produced by combining or linking an
Application with the Library.  The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

  The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

  The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

  1. Exception to Section 3 of the GNU GPL.

  You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

  2. Conveying Modified Versions.

  If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
```

that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

> 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
>
> 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

db-preservation-toolkit is maintained by keeps. This page was generated by GitHub Pages using the Cayman theme by Jason Long.