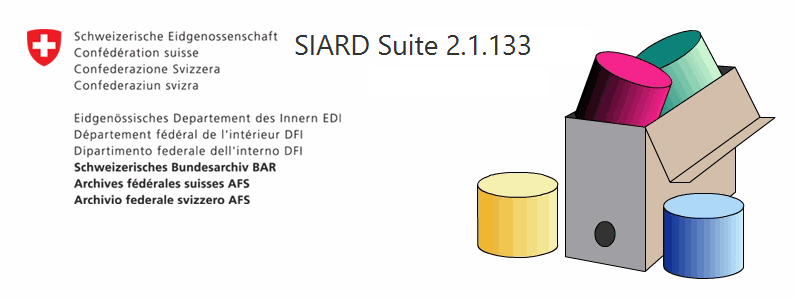
*–––* *Manual SIARD-Suite 2.1*



Date: 14.08.2020

Version: 1.0

***––– Content***

1 Summary 4

1.1 User Manual Language 4

1.2 Latest release 4

2 Introduction 5

2.1 New features of version 2.1 5

2.2 Intellectual Property Rights 6

3 SIARD Concept 9

4 Introduction to Database Archival 10

5 Prerequisites 11

5.1 JAVA 11

5.1.1 Architecture (32-bit/64-bit) 11

5.2 Databases 11

6 Installation 12

6.1 What Does Installation Mean? 12

6.2 Install 12

6.3 Uninstall 13

6.4 SIARD state properties 13

7 Components 14

7.1 SiardGui 14

7.2 SiardFromDb 14

7.3 SiardToDb 14

7.4 SiardApi 14

8 Execution 15

8.1 Initial Execution 15

8.2 Main Window 16

8.2.1 Apply and Discard 17

8.2.2 Table of Sub-Objects 17

8.3 Table of Primary Data 18

9 Menu 21

9.1 File / Download ... 22

9.2 File / Recent downloads 25

9.3 File / Upload ... 26

9.4 File / Recent uploads 30

9.5 File / Open ... 30

9.6 File / Recently opened 30

9.7 File / Save 30

9.8 File / Close 30

9.9 File / Display meta data ... 31

9.10 File / Augment meta data ... 32

9.11 File / Exit 32

9.12 Edit / Copy all 32

9.13 Edit / Copy 33

9.14 Edit / Export table ... 33

9.15 Edit / Find in meta data ... 33

9.16 Edit / Find next in meta data 34

9.17 Edit / Search in primary data... 35

9.18 Edit / Search next in primary data 36

9.19 Tools / Install ... 36

9.20 Tools / Uninstall 36

9.21 Tools / Language 36

9.22 Tools / Check integrity 37

9.23 Tools / Options ... 37

9.24 ? / Help 38

9.25 ? / Info 38

10 External LOBs 39

10.1 Download only Metadata 39

10.2 Specify External Storage Locations 40

10.3 Download LOBs to External Locations 43

11 Command Line Invocation 44

11.1 SiardFromDb 44

11.1.1 Invocation 44

11.1.2 Arguments 45

11.1.3 Notes 46

11.1.4 Archiving Database User 46

11.2 SiardToDb 47

11.2.1 Invocation 47

11.2.2 Arguments 47

11.2.3 Notes 48

12 Database Management Systems 49

12.1 JDBC URL for connecting to a database 49

12.2 Handling of proprietary data types 50

12.3 Preparation of a database for download 50

12.4 Preparation of a database for upload 50

13 Logging 51

14 Limitations 52

# Summary

This document is a technical user manual for the *SIARD Suite* *application* (Software Independent Archival of Relational Databases) of the Swiss Federal Archives.

It describes the

* technical prerequisites for deployment
* Installation
* Execution

of SIARD Suite and its components.

## User Manual Language

The user manual is also available in German, French or Italian and can be found directly in *SIARD-Suite* under *Menu > ? > Help*.

## Latest release

This version of the manual relates to the release of *SIARD Suite 2.1.133* from February 2020. Later adjustments are not described in this document. The most current descriptions can be found in the *SIARD Suite* *application* under *Menu > ? > Help.*

Author Publisher

Dr. sc. math. Hartwig Thomas Swiss Federal Archives

Enter AG Archivstrasse 24

Joweid Zentrum 1 3003 Bern

8630 Rüti ZH Switzerland

Switzerland

# Introduction

The *SIARD* format as well as the application *SIARD Suite* were developed by the Swiss Federal Archives. *SIARD* (Software-Independent Archival of Relational Databases) is used for long-term archiving of relational database content.

On behalf of the Swiss Federal Archives, Enter AG developed *SIARD Format 1.0* and *SIARD Suite 1.0* in 2007 as well as the current *SIARD Suite 2.1* in the years 2016-2018.

2015-2018 the version 2.1 of the *SIARD format* was specified by the Swiss Federal Archives in cooperation with the EU project E-ARK and the KOST. Like version 1.0, *SIARD Format 2.1* was endorsed as standard eCH-0165 by the association eCH E-Government Standards.

*SIARD Suite 2.1* is the reference implementation for archival of relational databases in the standard *SIARD Format 2.1*.

This document is the user's manual of *SIARD Suite 2.1*.

## New features of version 2.1

The main new features of *SIARD Format 2.1* different from version 1.0 are

* Conformity to SQL:2008, in particular support for advanced data types (DISTINCT, UDT, ARRAY)
* Permitting storage of large objects as external files
* Support of reversible "deflate"-Compression of the SIARD data

*SIARD* files conforming to *SIARD Format 1.0* can be read by the programs of *SIARD Suite 2.1*. However, when any changes are to be saved, they are automatically converted to *SIARD Format 2.1*.

*SIARD Suite 2.1* is the reference implementation for archiving relational databases in the standardized *SIARD Format 2.1.*

## Intellectual Property Rights

*SIARD Suite* is a development of Enter AG for the Swiss Federal Archives. The copyright owners publish *SIARD Suite* as open-source software under the CDDL-1.0 license (in the *SIARD* distribution as *doc/licenses/CDDL-1.0.txt*).

*SIARD Suite* relies on the following components of other manufacturers:

**JAVA SE 1.8 or higher**

from Oracle <http://www.oracle.com/technetwork/java/javase/downloads/>  
License: [Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX](http://www.oracle.com/technetwork/java/javase/terms/license/index.html) in the SIARD distribution as *doc/licenses/java-license.txt*

**JavaFX 8**

from Oracle as part of JAVA 8 <http://www.oracle.com/technetwork/java/javase/downloads/>  
License: [Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX](http://www.oracle.com/technetwork/java/javase/terms/license/index.html) in the SIARD distribution as *doc/licenses/java-license.txt*

**ini4j**

INI file handler for LINUX desktop links from Apache <http://ini4j.sourceforge.net/>  
License: [Apache License 2.0](https://www.apache.org/licenses/LICENSE-2.0) in the SIARD distribution as *doc/licenses/Apache-license-2.0.txt*

**mslinks**

LNK file handler for Windows desktop links from BlackOverlord666 <https://github.com/BlackOverlord666/mslinks>  
License: [WTFPL License)](http://www.wtfpl.net/) in the SIARD distribution as *doc/licenses/WTFPL.txt*

**SiardApi**

from Swiss Federal Archives  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**JavaBeans Activation Framework (Version 1.1.1)**

from Sun Microsystems Inc. <http://www.java2s.com/Code/Jar/a/Downloadactivationjar.htm> License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**Java Architecture for XML Binding (JAXB) (Version 2.3.0)**

from Oracle <http://www.java2s.com/Code/Jar/j/Downloadjaxbapi22jar.htm>  
License: [COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.11 and the GNU General Public License (GPL) Version 2 (CDDL+GPL 1.1)](https://glassfish.java.net/public/CDDL+GPL_1_1.html) in the SIARD distribution as *doc/licenses/CDDL+GPL\_1.1.txt*

**Woodstox XML processor**

An implementation of the Streaming API for XML (StAX2) for fast XML streaming while validating against an XML schema from Codehaus <https://mvnrepository.com/artifact/org.codehaus.woodstox/>  
License: [GNU Lesser Public License 2.1 (LGPLv2.1)](https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html) in the SIARD distribution as *doc/licenses/LGPL2.1.txt*

**Multi-Schema Validator (MSV)**

from SUN/Apache <https://github.com/kohsuke/msv/>  
License: [BSD license (BSD-2)](https://directory.fsf.org/wiki?title=License:FreeBSD) in the SIARD distribution as *doc/licenses/BSD-2.txt*.

**Zip64File**

from Enter AG  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in der SIARD-Distribution als *doc/licenses/CDDL-1.0.txt*.

**SqlParser**

from Enter AG  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD-Distribution als *doc/licenses/CDDL-1.0.txt*.

**ANTLR4 (Version 4.5.2)**

Parser Generator from Terence Parr <http://www.antlr.org/download.html>  
License: [BSD License (BSD-3))](http://www.antlr.org/license.html) in the SIARD distribution as *doc/licenses/BSD-3.txt*

**SiardCmd**

from Swiss Federal Archives  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**JTS Topology Suite (Version 1.14 - used by H2 and MySQL for the GEOMETRY extension)**

from Martin Davis <http://tsusiatsoftware.net/>  
License: [GNU Library General Public License (LGPLv2.0)](http://www.gnu.org/licenses/old-licenses/lgpl-2.0.txt) in the SIARD distribution as *doc/licenses/LGPLv2.0.txt*.

**JdbcBase**

from Swiss Federal Archives  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**JdbcPostgres**

from Swiss Federal Archives  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**JDBC Driver für Postgres (postresql-42.2.5.jar)**

from the PostgreSQL Global Development Group  
License: [Postgres License](https://www.postgresql.org/about/licence/) in the SIARD distribution as *doc/licenses/licensePostgres.txt*.

**JdbcOracle**

from Swiss Federal Archives  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**JDBC Driver for Oracle (ojdbc6.jar (version 12.1.0.1.0), xdb6.jar, xmlparserv2.jar)**

from Oracle  
License: [Oracle License](http://www.oracle.com/technetwork/licenses/distribution-license-152002.html) in the SIARD distribution as *doc/licenses/licenseOracle.txt*.

**JdbcMySql**

from Swiss Federal Archives  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**JDBC Driver for MySQL (Version 8.0.18)**

from Oracle <https://dev.mysql.com/downloads/connector/j/>  
License: [GNU Generial Public License (GPLv2.0)](http://www.gnu.org/licenses/old-licenses/gpl-2.0.html) in the SIARD distribution as *doc/licenses/GPLv2.0.txt*.

**JdbcMsSql**

from Swiss Federal Archives  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**JDBC Driver for SQL Server (Version 4.1)**

from Microsoft <https://msdn.microsoft.com/library/mt484311.aspx>  
License: [Microsoft License](https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774) in the SIARD distribution as *doc/licenses/license41.txt*.

**JdbcH2**

from Swiss Federal Archives  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**H2 database (Version 1.3.176)**

from Thomas Müller <http://www.h2database.com/>  
License: dual license [Eclipse Public License v1.0 (EPL1.0)](https://opensource.org/licenses/eclipse-1.0.txt) and [Mozilla Public License 2.0 (MPL2.0)](https://www.mozilla.org/media/MPL/2.0/index.815ca599c9df.txt) in the SIARD distribution as *doc/licenses/EPL1.0.txt* and *doc/licenses/MPL2.0.txt*.

**JdbcDb2**

from Swiss Federal Archives  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**JDBC Driver for DB/2 (Version 4.1)**

from IBM <http://www-01.ibm.com/support/docview.wss?uid=swg21363866>  
License: [IBM license](http://www14.software.ibm.com/cgi-bin/weblap/lap.pl?la_formnum=&li_formnum=L-LKSA-94UU9J&title=IBM+Data+Server+Driver+for+JDBC+and+SQLJ+(JCC+Driver)&l=en#LI) in the SIARD distribution as *doc/licenses/IBM JDBC 4 License.txt* and *doc/licenses/IBM jdbc4\_notices.txt*.  
This very long license essentially declares that IBM is the copyright holder of the software and makes it freely available for using, copying and redistributing. However, there are technical “licenses” which restrict its use for connecting to a DB/2 instance running on an operating system platform which is not Windows, LINUX, or UNIX. If you want to make use of *SIARD Suite* in such a context, you need to apply to the vendor of the DB/2 database instance for the appropriate technical license file from IBM and add it to the class path.

**JdbcAccess**

from Swiss Federal Archives  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD distribution as *doc/licenses/CDDL-1.0.txt*

**Jackcess (Version 2.1.6)**

from Health Market Science <http://jackcess.sourceforge.net/>  
License: [Apache License, Version 2.0](https://www.apache.org/licenses/LICENSE-2.0) in the SIARD distribution as *doc/licenses/Apache-license-2.0.txt*

**Two parts (commons-lang-2.6 6 and commons-logging-1.1.3 7 ) used by Jackcess**

from Apache Commons <http://commons.apache.org/>  
License: [Apache License, Version 2.0](https://www.apache.org/licenses/LICENSE-2.0) in the SIARD distribution as *doc/licenses/Apache-license-2.0.txt*

**EnterUtilities**

from Enter AG  
License: [CDDL-1.0 license](http://opensource.org/licenses/CDDL-1.0) in the SIARD Distribution as *doc/licenses/CDDL-1.0.txt*.

A copy of all licenses can be found in the *doc/licenses* folder of the distribution ZIP file. A copy of all third party binaries used by *SIARD Suite* can be found in the *lib* folder of the distribution.

# SIARD Concept

The Swiss Federal Archives are obliged to archive Federal administration documents [independent of the information medium](https://www.admin.ch/opc/en/classified-compilation/19994756/index.html). Therefore, the problem of long-term archiving of relational databases must be resolved.

The *SIARD Format* has been used since 2007 by the Swiss Federal Archives and by many other archives around the world as a normalization format for the long-term preservation of relational databases.

With *SIARD Format 2.1* the databases are stored conforming to the standard [SQL:2008](https://www.iso.org/standard/38640.html) in order to guarantee long-term availability. Data content is stored as a collection of XML files. Because the resulting archive format is based on these two ISO standards, it is believed that lasting data comprehensibility is assured.

An important requirement of data content archived in *SIARD format* is that it should have "documentary character", i.e. the content of the archived tables should be comprehensible independently of any front-end processing applications and should represent the enterprise information of the institutions operating the subject databases. Neither executable code nor objects are archived by the *SIARD Suite* but only enterprise information from database tables. This is explained in more detail in the report[*"Long-term Preservation of Relational Databases, What needs to be preserved how?"*](http://www.enterag.ch/hartwig/SIARD_Criterion.pdf) by Hartwig Thomas.

The *SIARD format* stores the archived database schema definition in SQL:2008 conformant XML files while documentation in respect of the tables and fields, as well as the actual data, is also stored in XML files. In order to avoid excessive XML file size inflation, BLOBs and CLOBs (Binary Large OBjects and Character Large OBjects), referenced in the XML files, are stored in separate (binary) files.

This document does not further explain the *SIARD format* and structure as they are described in a separate document, which was delivered together with the *SIARD Suite*. In 2013 the *SIARD format* was recognized as an eCH-Standard. In 2018 the version 2.1 of the *SIARD format* has been made available as standard [eCH-0165](https://www.ech.ch/vechweb/page?p=categoryList&site=/documents/Alle/No).

# Introduction to Database Archival

This is a quick introduction on how to archive databases with *SIARD Suite*. It also covers organizational issues that should be considered.

1. Make sure you know which parts of the database need to be archived. If needed, get in touch with the responsible personnel, e.g. someone from the archive responsible for appraisal.
2. Prepare the database for archival: create a new user on the database system that only has read permissions to objects that need to be archived. If needed, create a copy of your database (or certain tables/parts of it) or create views. The database may not be changed during the archival process; otherwise, extraction with *SIARD Suite* will fail. Never archive from a live system.
3. Download the database using *SIARD Suite*.
4. Quality control: check the *SIARD file* to make sure that everything needed is included, spot check some entries to ensure everything went well.
5. Advanced quality control: load *SIARD file* into a database system again. Run some defined queries on the original database as well as on the archived one and compare results.
6. Supplement *SIARD file* with metadata.
7. Define which external documentation needs to be archived together with the *SIARD file* to ensure comprehensibility of the data (e.g. code tables, system documentation, Entity-Relationship-Diagram, …).

# Prerequisites

A JAVA installation is a prerequisite for using *SIARD Suite*. A suitable database system infrastructure is a prerequisite for loading or storing database content.

## JAVA

Users of the *SIARD Suite* must install JAVA in advance. The minimum technical requirement is JRE 1.8.

JAVA is freely available from <http://www.java.com/> (JRE – JAVA Runtime Environment) or <http://www.oracle.com/technetwork/java/javase/downloads/index.html> (JDK - JAVA Development Kit). *SIARD Suite* makes use of features of JavaFX, which are part of the JAVA SE distribution but not yet integrated in OpenJDK. Therefore, one cannot use OpenJDK (<http://openjdk.java.net/>) instead.

To find out whether JAVA 1.8 or higher is available under Windows, proceed as follows: in the Windows "Start" menu item, type the command "cmd" and enter "java -version" in the command window.

### Architecture (32-bit/64-bit)

In former versions of *SIARD Suite* a dependency on ODBC necessitated the use of 32-bit JAVA for accessing MS Access databases. *SIARD Suite 2.1* does not use ODBC for accessing MS Access databases anymore. Therefore those databases can be accessed on any platform (e.g. LINUX) and *SIARD Suite 2.1* is compatible with 32-bit JAVA as well as 64-bit JAVA. It is recommended to choose the JAVA architecture according to the architecture of your operating system.

## Databases

*SIARD Suite 2.1* currently supports the following database systems:

* MS Access 2007 or higher
* DB/2 8 or higher
* H2 database 1.4 or higher
* MySQL (or MariaDB) 5.5 or higher
* Oracle 10 or higher
* PostgreSQL 11 or higher
* SQL Server 2012 or higher

Further database systems may be integrated at a later date. The JDBC drivers of the database vendors usually do not conform to SQL:2008. Most of them even fail to conform to the JDBC 4 standard with respect to metadata or advanced data types. Therefore a JDBC wrapper needs to be developed for each database system, which conforms to the standards at least to the extent required by *SIARD Suite*.

# Installation

The following menu items are concerned with installation issues:

* [Install](about:blank#install): installing the currently running copy of *SIARD Suite*,
* [Uninstall](about:blank#uninstall): uninstalling the currently installed copy of *SIARD Suite*.

## What Does Installation Mean?

*SIARD Suite* can be deployed on any platform where JAVA can be installed. Therefore installation is not implemented by a separate platform-specific installation program (e.g. setup.exe) but inside the *SIARD Suite* itself.

In fact installation really means the copying of the program files e.g. from a removable medium or another temporary location to some fixed location (network or fixed disk) and the saving of some properties in the user's personal "home" directory which contains the state based on past executions (e.g. *C:\Users\<User>\.java\siard\_suite-2.1.properties*).

Thus, it is possible to run *SIARD Suite* without "installing" it. It will then just be executed directly from the temporary location without changing another installed version linked to the personal state properties.

In particular, a new version can be executed alongside an old version that is already installed.

## Install

The menu item *Tools/Install* permits installation of a new version of *SIARD Suite* and removes a previously installed version if there is one.

The menu item *Tools/Install* will be disabled if the version number of the currently running instance of *SIARD Suite* is less or equal to the version of the installed instance.

Otherwise executing *Tools/Install* will:

* copy the files of the currently running instance to a folder to be chosen by the user,
* create a personal state properties file, containing the version and location of the installed instance, and
* attempt to create a shortcut for running *SIARD Suite* on the desktop. This will not always work on all operating systems, because LINUX distributions use a wide variety of ever-changing desktops.

## Uninstall

The menu item *Tools/Unistall* removes a previously installed version of *SIARD Suite* if there is one.

The menu item *Tools/Uninstall* will be disabled, if no personal state properties file can be found.

Otherwise executing *Tools/Uninstall* will remove:

* desktop shortcut for running *SIARD Suite*, if one can be found,
* the folder with the program files unless the currently running instance is the installed instance. This folder must then be removed manually, and
* the personal state properties file, if its removal was requested.

## SIARD state properties

*SIARD Suite* reads and writes its state properties from a file *siard\_suite\_2.1.properties,* which is located in the folder *.java* in the user's home directory.

The user's home directory is identified as the JAVA system property *user.home*. Unless it is redirected manually using a JVM -D argument, this system property has the same value as the environment variable *%USERPROFILE%* on Windows platforms or the environment variable *$HOME* (synonymous with "~") on LINUX/UNIX platforms.

The personal properties file is therefore usually:

* *C:\Users\<User>\.java\siard\_suite\_2.1.properties* on Windows,
* */home/<User>/.java/siard\_suite\_2.1.properties* on most LINUX distributions.

# Components

The SIARD Suite is made up of the following components:

* SiardGui
* SiardFromDb
* SiardToDb
* SiardApi

## SiardGui

*SiardGui* implements an interactive graphical user interface facilitating processing of the data in a *SIARD archive*. Meta data can be edited, primary data cannot be changed. *SiardGui* is not suitable for complex research. For complex research, it is recommended that the *SIARD archive* be loaded into a database system and database techniques be used.

## SiardFromDb

*SiardFromDb* is a command-line application for extracting and storing a database in a *SIARD file*. This application's functionality is identical to the download function available in *SiardGui*. The command-line version is especially more comfortable when downloading large databases or downloading a number of databases in a batch. Further, long lasting downloads can be better documented using *stdout* and *stderr* redirection (see [*"Using command redirection operators"*](https://technet.microsoft.com/en-us/library/bb490982.aspx)).

## SiardToDb

*SiardToDb* is a command-line application for uploading a database from a *SIARD file*. This application's functionality is identical with the function available in *SiardGui*. Especially when uploading large databases, using the command-line version is more comfortable. Further, long lasting uploads can be better documented using *stdout* and *stderr* redirection (see [*"Using command redirection operators"*](https://technet.microsoft.com/en-us/library/bb490982.aspx)).

## SiardApi

*SiardApi* is a JAVA API for reading and writing *SIARD archives*. Its Javadoc documentation is available in the folder *doc/siard-api* of the *SIARD Suite* distribution.

The *SiardApi* is implemented in the *siardapi.jar* in the *lib* folder of the *SIARD Suite* distribution. In addition, the following JAR files are required for its execution:

* jaxb-api.jar
* jaxb-core.jar
* jaxb-impl.jar
* msv-core-2010.2.jar
* stax2-api-3.1.1.jar
* woodstox-core-lgpl-4.1.2.jar
* woodstox-msv-rng-datatype-20020414.jar
* xsdlib-2010.1.jar
* Zip64File.jar

# Execution

The *SiardGui* program features an interactive graphical user interface (GUI). Using *SiardGui*, one can:

* download a database and store it in a *SIARD archive*,
* display, variously sort and browse, manually add and change a *SIARD archive's* metadata as long as the primary data are not affected,
* display, variously sort and browse the primary data in a *SIARD archive*,
* upload a *SIARD archive* into a database for research purposes,
* download the meta data for a *SIARD archive* (without primary data), from a database in order to get a first overview of the archiving process,
* import a template of meta data with existing descriptions for a

*SiardGui* is the central instrument with which *SIARD* formatted data are processed. Primary data cannot be changed. *SiardGui* is not suitable for complex research. For complex research, it is recommended to load a *SIARD archive* into a database system and use database techniques.

The conversion of the database fields of type TIME and TIMESTAMP depends on the local time zone. If the time 15:30 is stored on a machine in Zurich, then it will be stored as the UTC time 14:30 (in winter!) in the XML metadata. If you would prefer to interpret the times in the database unchanged as UTC times, you must start *SiardGui* with the option:

It is possible to call *SiardGui* with the name of a SIARD file to be opened as single argument. This permits setting *siardgui.cmd* as the default application for opening files with a *.siard* extension.

## Initial Execution

The *SIARD Suite* is delivered as a ZIP file and must first be unpacked. The file *SiardGui.jar* is situated in the *lib* folder of the distribution. If JAVA is installed correctly, one can execute the program under Windows by double-clicking on it. One can also execute the platform-specific script *siardgui.cmd* (Windows) or *siardgui.sh* (LINUX).

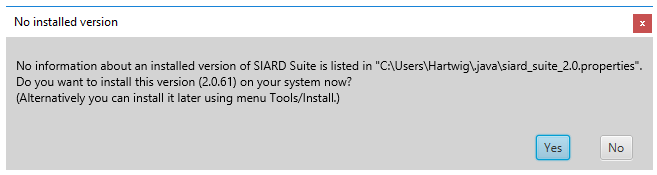
If this does not work or one is using a different operating system, *SiardGui* can also be started from the command line in the *SIARD Suite*'s *lib* folder as follows:



For this to succeed, JAVA's *bin* folder must have been added to the PATH variable. Normally that was already done by the installation process of JAVA. Otherwise, one must write out the full path name of the executable java program (e.g. including the quotation marks):



Upon initial execution of *SiardGui*, this type of message appears:



As *SiardGui* doesn't know the user's language at this point, the language of this message depends on the operating system language and the language chosen when JAVA was installed.

If this message is answered with *Yes*, one is given the possibility to enter a new or empty folder name where a copy of the SIARD distribution should be installed. After the successful installation, *SiardGui* can, in future, be started from the chosen folder or from the installed desktop icon.

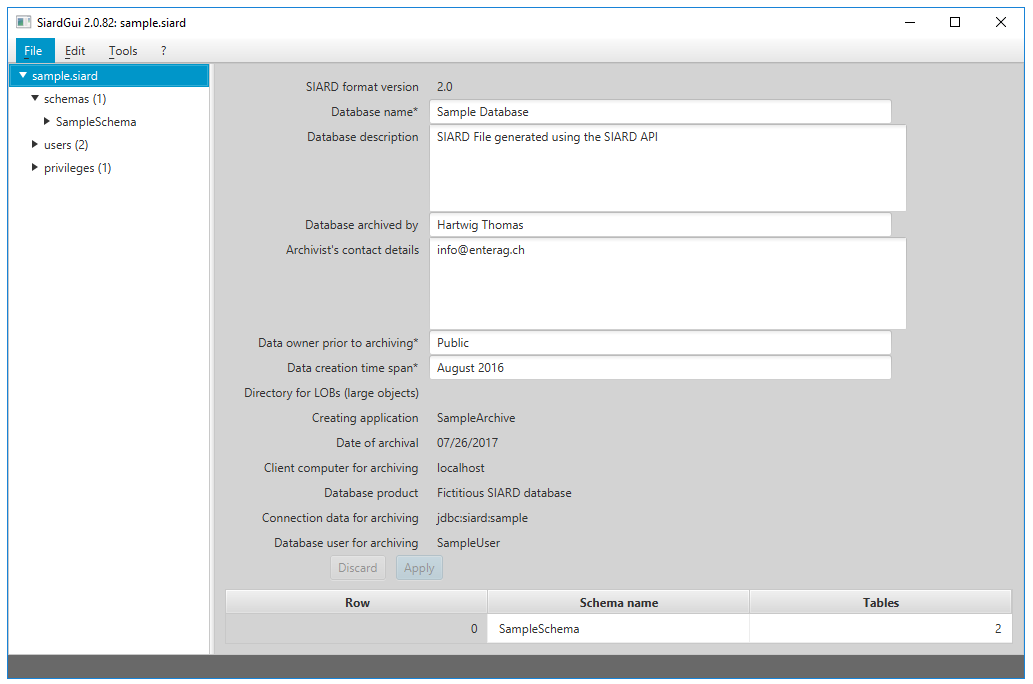
Irrespective of whether *SiardGui* is started only from USB-Stick or CD-ROM or whether it is installed on the user's PC, the following main window appears.

## Main Window

The main window consists of a menu (top), navigation tree (left), content (right) and a status line (bottom).



The border between navigation and content can freely be adjusted. The size of the whole window can also freely be adjusted (but not below a defined minimum). When a *SIARD file* is loaded into *SiardGui*, the main window appears as follows:



The left pane is used to navigate in the metadata tree. In the upper region of the right pane, one can enter or change alterable metadata, which belong to the database object selected in the left pane.

### Apply and Discard

The *Apply* button applies the changes to the metadata in the currently open *SIARD file*. Clicking on the *Discard* button undoes all changes made since the last *Apply* action.

### Table of Sub-Objects

A table of the most important sub-objects is shown under the metadata. Clicking a column title sorts the table on this column. As tables in schemata and columns in tables have no natural ordering in the metadata and *SiardGui* normally displays in alphabetical order, this sort function is useful when finding one's way about in large database schemata.

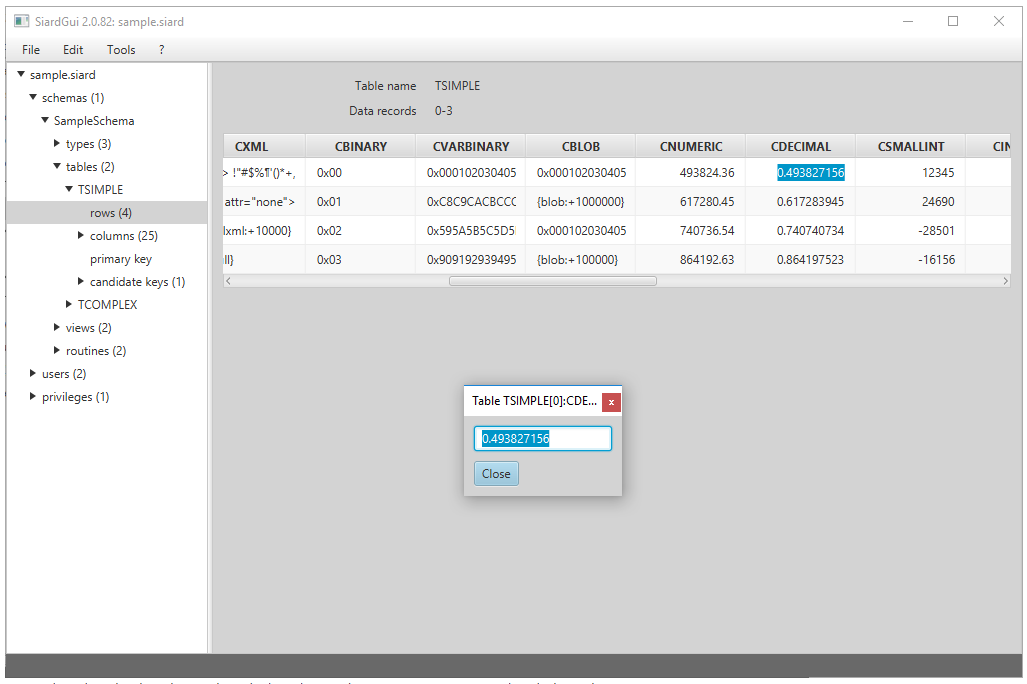
## Table of Primary Data

Under *rows* of each table its primary data are displayed. As tables can grow very large, it is impossible and not very useful to load and display all records at once. Instead an overview of at most 50 records distributed over the table is shown, when *rows* is selected. Then one can choose which branch to display in more detail until the level is reached, where each record is shown.

When a column header of a primary data display of a table is clicked, the whole table is sorted (in a temporary XML file, which is deleted when the program is closed). This may take a while but is very useful for navigating to a particular value of a column.

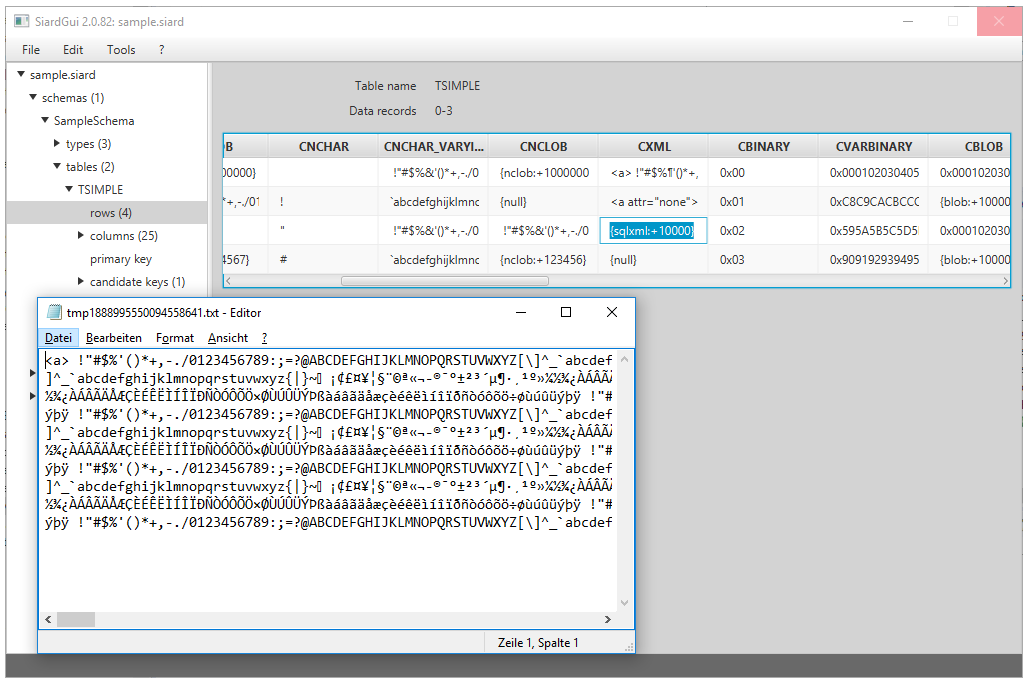
The column widths of the display of primary data can be changed by dragging the separator between column headers.

The value display in the table is only useful for short values. More explicit value display is available when a cell is double-clicked.

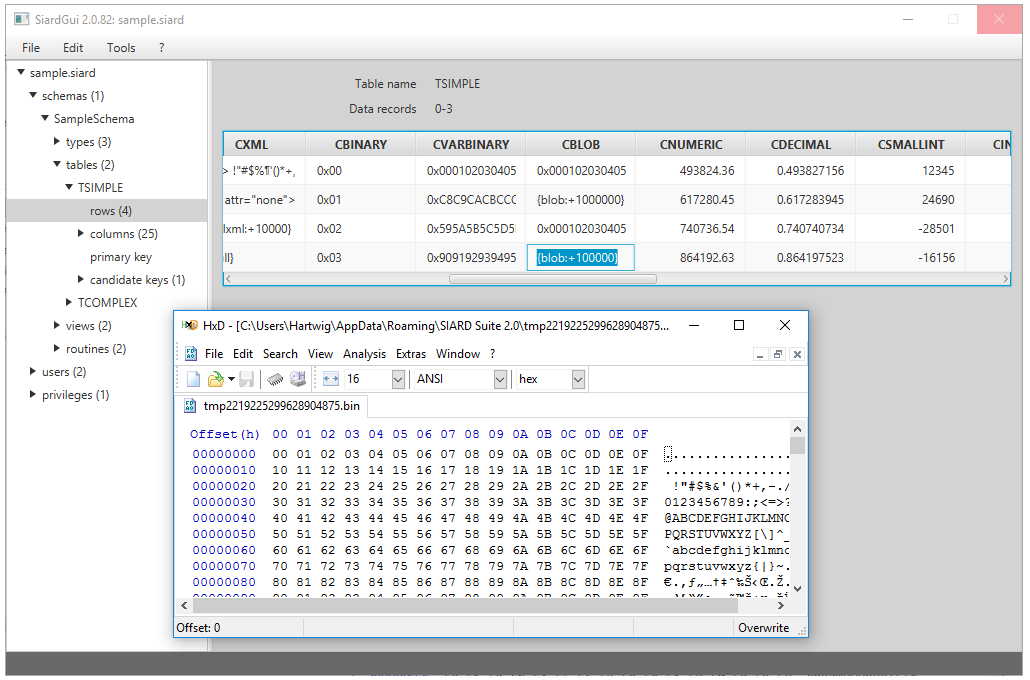


A simple value display shows the whole value and permits copying it to the clipboard.

Long text values (e.g. VARCHAR, CLOB or XML values) are displayed in the external text editor application, which can be configured under the menu item *Tools / Options*. Under Windows, the default for this application is *Notepad*.

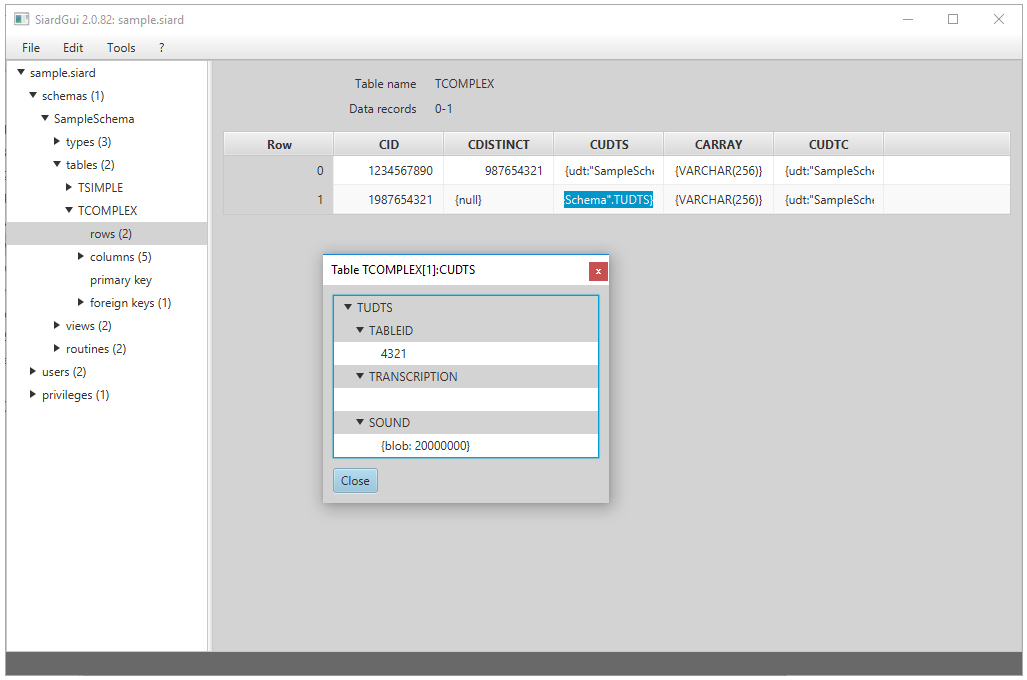


Long binary values (e.g. VARBINARY or BLOB values) are display in the external binary editor application, which can be configured under menu item *Tools / Options*. Under Windows, the default for this application is the freeware program *HxD*, which is packaged with the *SIARD* distribution for convenience.

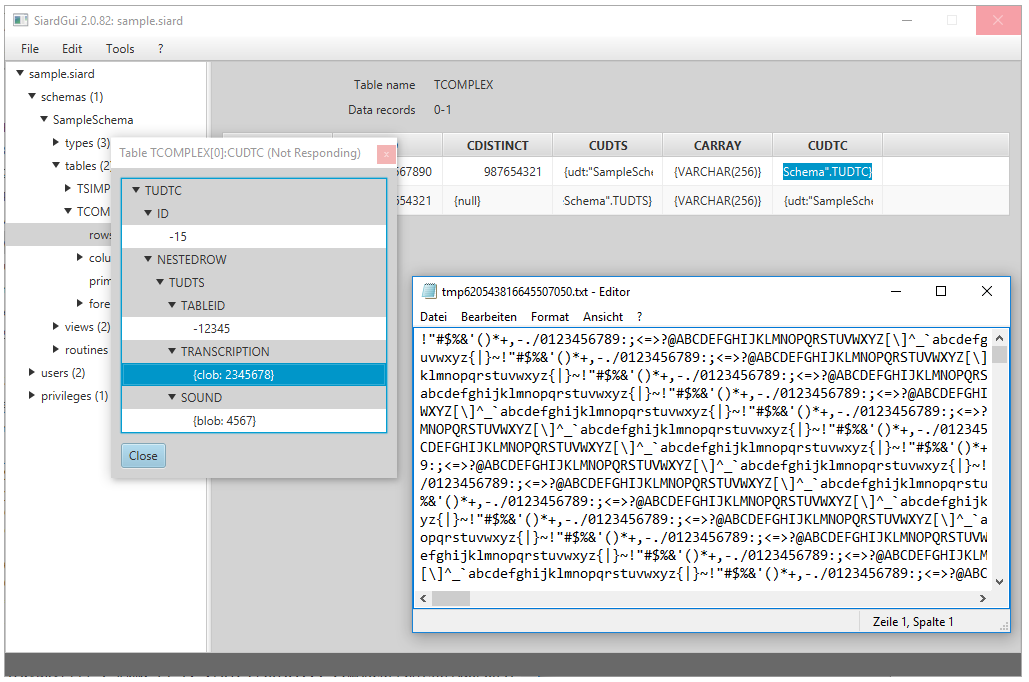


If one knows that a BLOB column holds values of a very specific type, e.g. images or PDF data, then one can choose a binary editor instead which is able to open this type of data.

User-defined data types (UDTs) are displayed hierarchically with the attribute names in gray and the values in white.



Each of these values can be double-clicked again in order to display it in detail.



# Menu

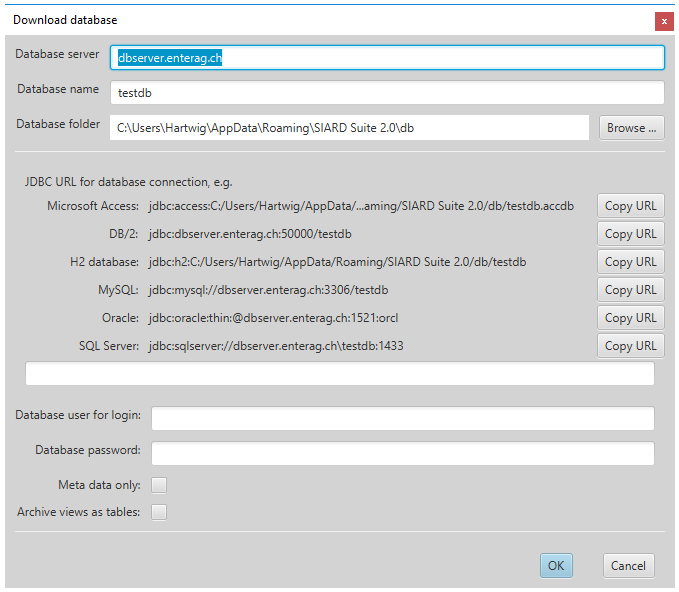
The following menu items are available in *SiardGui*:

* File / Download ...
* File / Recent downloads
* File / Upload ...
* File / Recent uploads
* File / Open ...
* File / Recently opened
* File / Save
* File / Close
* File / Display meta data ...
* File / Augment meta data ...
* File / Exit
* Edit / Copy all
* Edit / Copy
* Edit / Export table ...
* Edit / Find in meta data ...
* Edit / Find next in meta data
* Edit / Search in primary data ...
* Edit / Search next in primary data
* Tools / Install ...
* Tools / Uninstall
* Tools / Language
* Tools / Check integrity ...
* Tools / Options ...
* ? / Help
* ? / Info

The menu items are disabled when they are not applicable to the current situation. Thus, initially, only *Download* ... and *Open* ... are available.

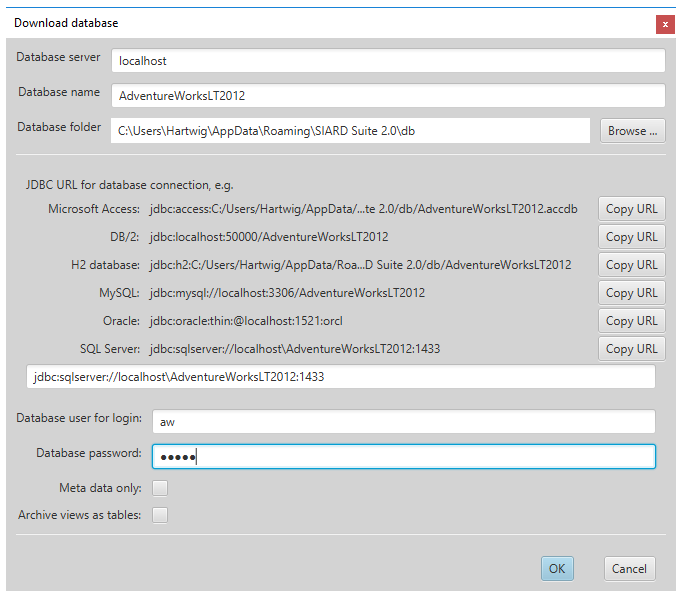
## File / Download ...

When this menu item is chosen, a dialog is displayed where the connection data for the database can be entered.



The long text entry field in the middle must be filled with a JDBC URL and the database user for archival with password should be given. If only the metadata are to be downloaded (e.g. for a preliminary examination of the extent of the database) the box *Meta data only* must be checked. If views are to be archived as tables, the box *Archive views as tables* must be checked.

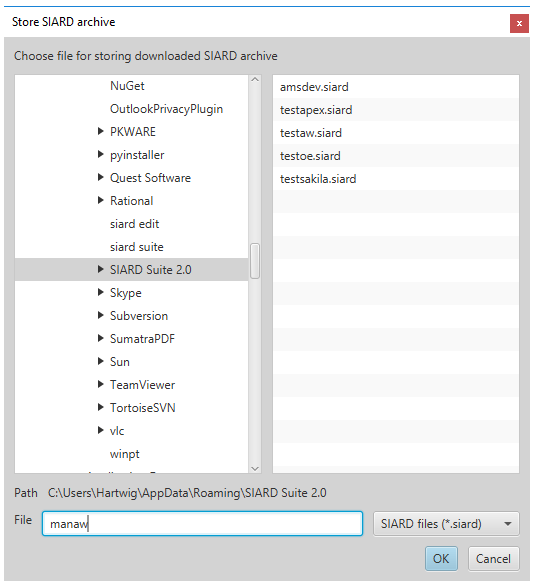
The server name, database name and database folder above only serve to help construct a correct URL for the *target database management system* (DBMS). Changing them changes the sample URLs displayed for each DBMS. Clicking on the *copy URL* Button next to the sample URL copies it to the input field for the JDBC URL.



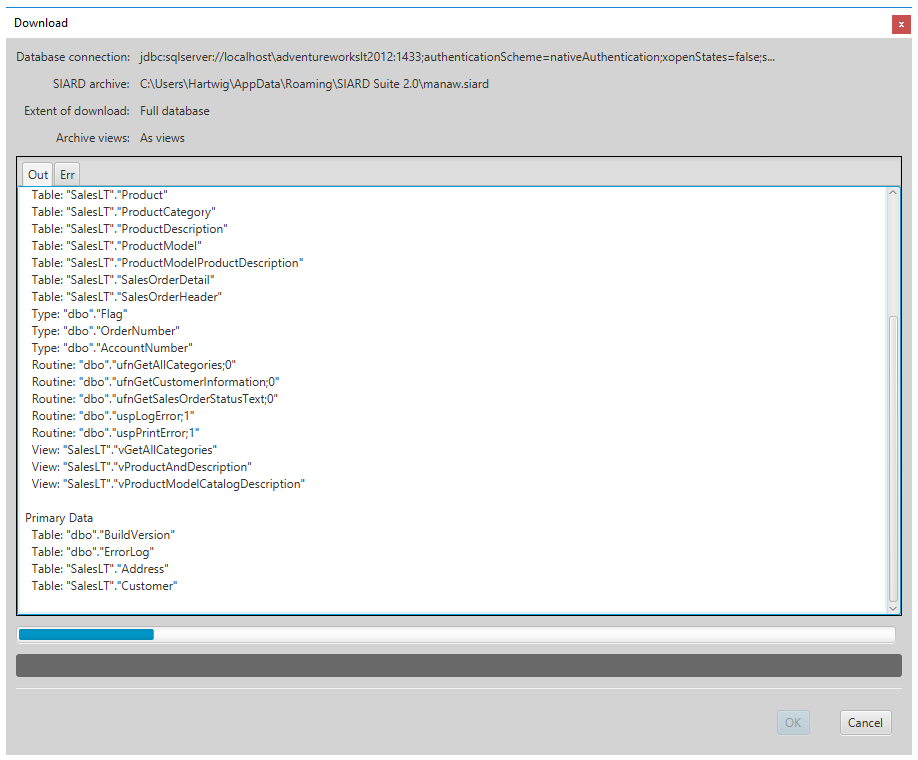
However, any string can be entered as JDBC URL manually. This allows for site-specific security configurations such as Windows login or Kerberos. The vendor-specific definitions of JDBC URLs must be consulted if the simple standard presented here is insufficient (v. [Database Management Systems](about:blankdbms.html#jdbcurl)).

It is generally inadvisable to use the database administrator user (DBA, root, dbo, SYSTEM, sa, dbadmin, ...) for downloading a *SIARD archive*. The extent of the *SIARD archive* is defined by the objects to which the archiving database user has read-access. The global database administrator usually has read access to all databases on the system as well as numerous system tables that should not be archived. Therefore, it is important to [prepare the download](about:blankdbms.html#prepdownload) by choosing or creating a suitable archiving user.

If the connection cannot be established, the dialog is redisplayed until a valid JDBC URL has been entered or *Cancel* was pressed. If *Meta data only* was checked, a temporary *SIARD file* is created automatically which will be deleted when the program terminates. (However, the downloaded metadata can be edited, displayed, and exported before closing the file.) Otherwise, the name and location of the *SIARD archive* to be created must be chosen.



Then the download starts.



If the download was successful, the dialog can be closed pressing *OK* and the data downloaded are shown in the main window. There additional metadata should be specified giving at least a name for the database, the data owner prior to archiving and the time span during which the data was created.

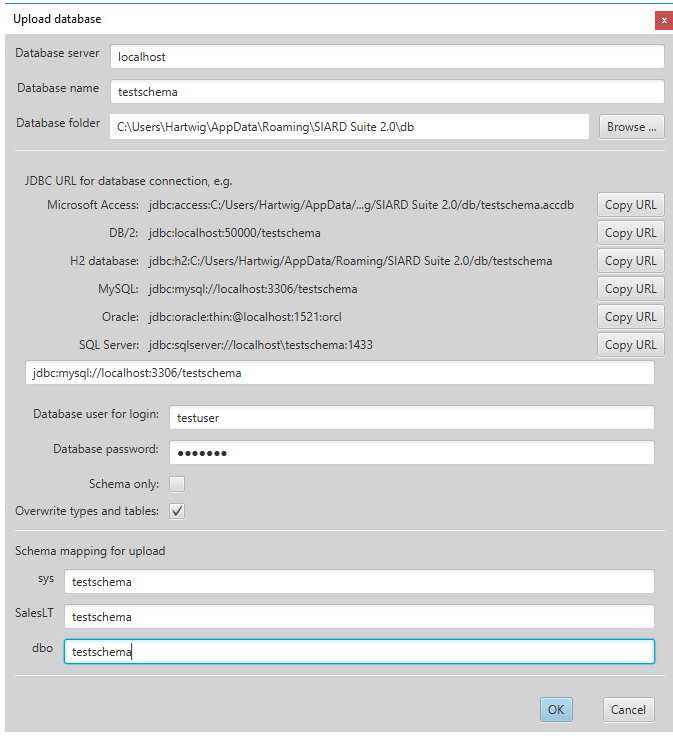
Also, if the connection could be established successfully, the JDBC URL used is entered in a list of most recently used connection strings, which is available under the next menu item.

## File / Recent downloads

The most recently used connection data for download is available using this menu item. Choosing one of them opens the connection dialog with the JDBC URL and the database user filled in. Only the password must still be entered.

## File / Upload ...

When this menu item is chosen, a dialog is displayed, where the connection data for the database can be entered.



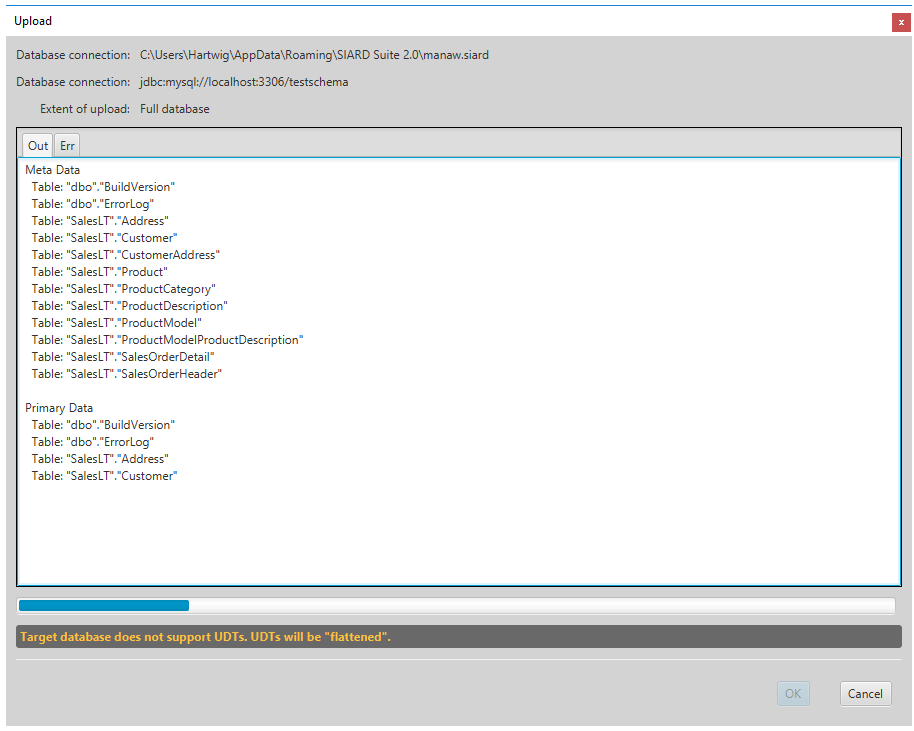
The JDBC URL can be combined using database host, database name and database folder and then copied in the same way as for the download connection dialog. As the content of the *SIARD file* is independent of the DBMS from which it was downloaded, the data can be uploaded to a different DBMS.

In addition, one can check the option that database tables and types with the same name may be overwritten. This is dangerous if one connects using a database administrator account with very many privileges. On the other hand, it is useful if a previous upload is to be repeated.

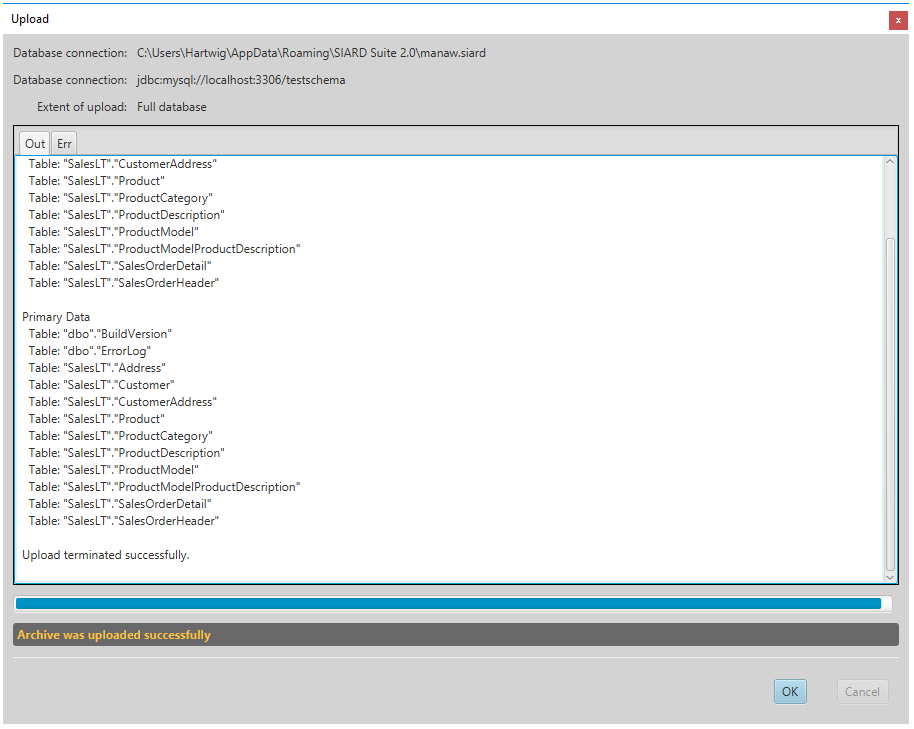
If *Schema only* is checked, only the schemas, types and empty tables are created without uploading any primary data.

On the bottom of the dialog, a list of all schemas in the *SIARD file* is displayed. Here the names of the schemas of the DBMS can be entered that shall receive the data of the schemas of the *SIARD archive*. These or schemas must have been created [prior to the upload](about:blankdbms.html#prepupload). The database user entered in this dialog must have the privilege to create types and tables in these schemas. The database user entered must have the privilege of creating types and tables in the schemas entered here. It is therefore often easiest to choose the root user for the upload.

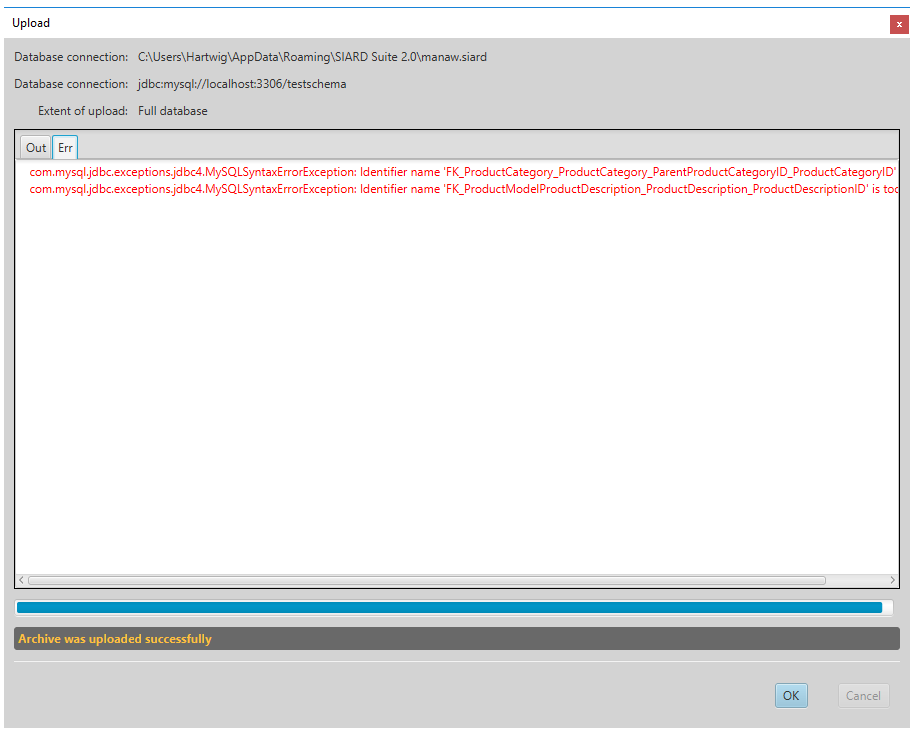
If the target DBMS does not support UDTs or ARRAYs, the data will be "flattened" on upload, i.e. each UDT or ARRAY is uploaded by creating a separate column for each component.



When the creation and upload of types and tables was successful, the upload is considered as successful. Some types or tables may have been renamed to accommodate length limitations etc. of the target DBMS. In that case the long suffix is replaced by a number.



An attempt is only made at the end to enable the constraints. This may fail because one DBMS may have more strict rules than the other. Such a failure is displayed in the *Err* of the upload dialog.



If the upload was successful, the JDBC URL used is entered in a list of most recently used connection strings, which is available under the next menu item.

## File / Recent uploads

The most recently used connection data for upload is available using this menu item. Choosing one of them opens the connection dialog with the JDBC URL and the database user filled in. Only the password and the schema mapping must still be entered.

## File / Open ...

Choosing this menu item opens a file selector where an existing SIARD file can be chosen. After it is opened, it is displayed in the main window where metadata can be amended and the primary data can be browsed.

If a SIARD file could be opened or downloaded successfully, its name is added to a list of most recently used files, which is available under the next menu item.

## File / Recently opened

Choosing one of the most recently used files opens it immediately in the main window.

## File / Save

If the metadata of the *SIARD file* has been changed, the changes are only written to the disk when they are explicitly saved.

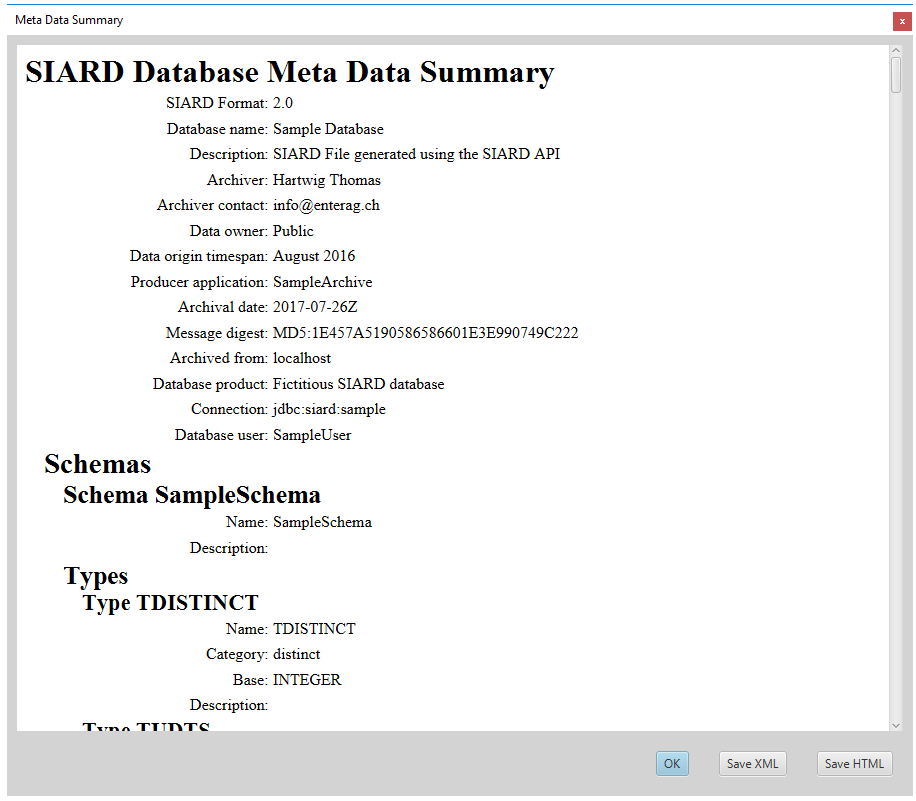
Temporary *SIARD files* created by downloading with option *Meta data only* cannot be saved. However, their metadata can be edited, displayed and exported before closing the file.

## File / Close

Closing a SIARD archive makes it possible to download or open another one.

## File / Display meta data ...

The metadata of the SIARD archive displayed in the main window can be examined as a human-readable document, when this menu item is chosen.



An HTML version of the metadata XML is displayed which was generated using the currently selected metadata XSL (XML Stylesheet Language) transformation to HTML. By default, a simple transformation is found in *etc/metadata.xsl*. But other more extensively designed ones can be given under *Tools / Options*.

The original metadata XML can be saved to an external file by pressing the button *Save XML* below. If the button *Save HTML* is pressed instead, the HTML version is saved instead which is the result of the XSL transformation of the original XML.

## File / Augment meta data ...

Externally saved metadata can be very useful when the "same" database is archived at a later date. Then it is not necessary to enter all descriptions of all tables and columns manually again. Instead, one can augment the *SIARD archive* by externally saved metadata where these descriptions have been entered before.

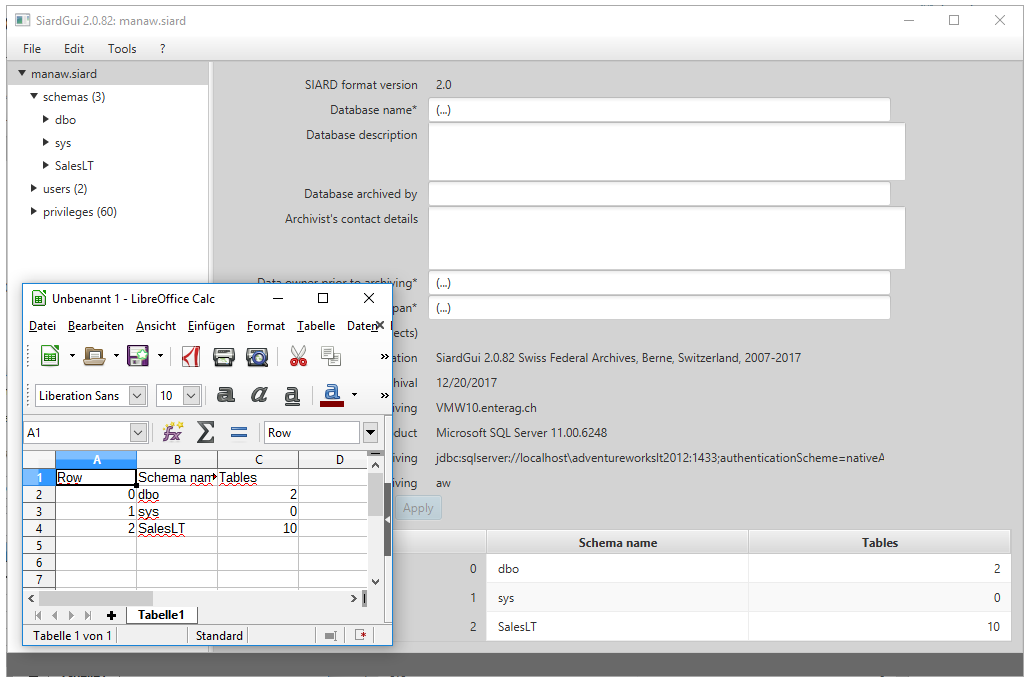
After choosing a meta data XML file for augmenting the current *SIARD archive*, all descriptions are copied from the external XML to the open *SIARD archive* where the names of database objects (schema, table, column, ...) match. Accordingly, even if the current database is slightly different from the database documented in the imported metadata, most of the descriptions will be copied.

## File / Exit

Choosing this menu item closes any open file and exits the program.

## Edit / Copy all

Choosing this menu item copies the table displayed in the right area to the clipboard. This may be a list of sub-objects or an extract of primary data.



The content of the clipboard can be pasted into any other application, which can accept text or tabular data. The table cells are separated by tabs. Therefore pasting into MS Excel or LibreOffice Calc will create an accessible tabular copy.

## Edit / Copy

This menu item is activated if a cell in the table is clicked. Choosing it then copies the single record, which contains the selected cell.

## Edit / Export table ...

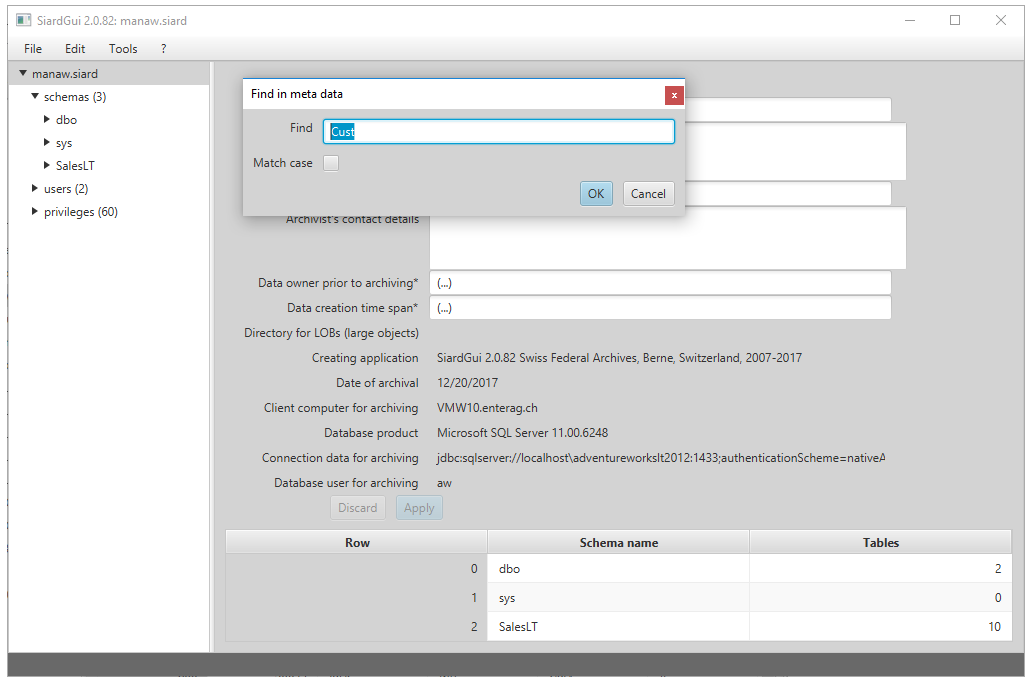
Sometimes it is necessary to work on the whole table in a different application. For this purpose, any table can be exported as an HTML file which essentially only contains a table.

The HTML format was chosen because it can be opened in MS Excel or LibreOffice Calc just like a CVS file. On the other hand, it does not have some of the weaknesses of a CVS file and it permits tables in tables for UDT values and links to external files for large object values (CLOB, BLOB, XML, ...).

When this menu item is chosen, target HTML file must be specified using a file selector dialog. The large object files, however, are stored in a special LOBs folder which can be modified under *Tools / Options ...*

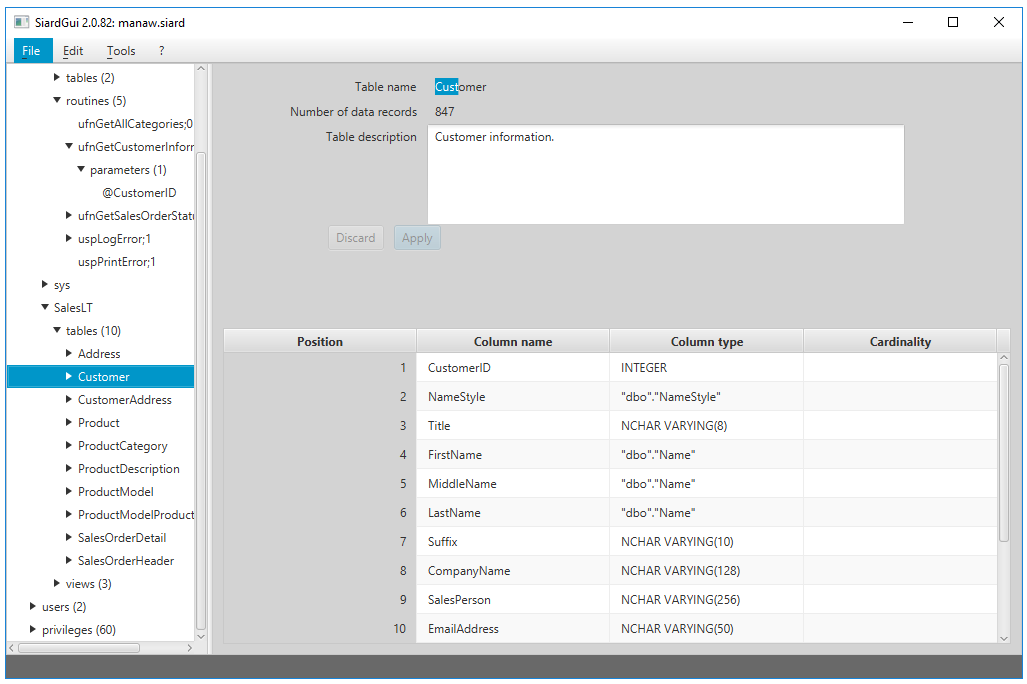
## Edit / Find in meta data ...

If there are many tables and columns, it is often difficult to find a particular piece of metadata again. With the help of the function *Find in metadata ...* all metadata can be found that contain a piece of text.



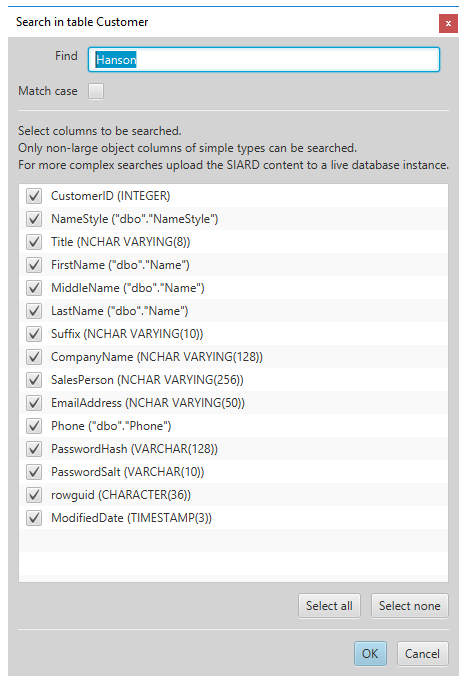
## Edit / Find next in meta data

Using *Find next in metadata* or *Shift-F3* all occurrences of the find string can be visited.



## Edit / Search in primary data...

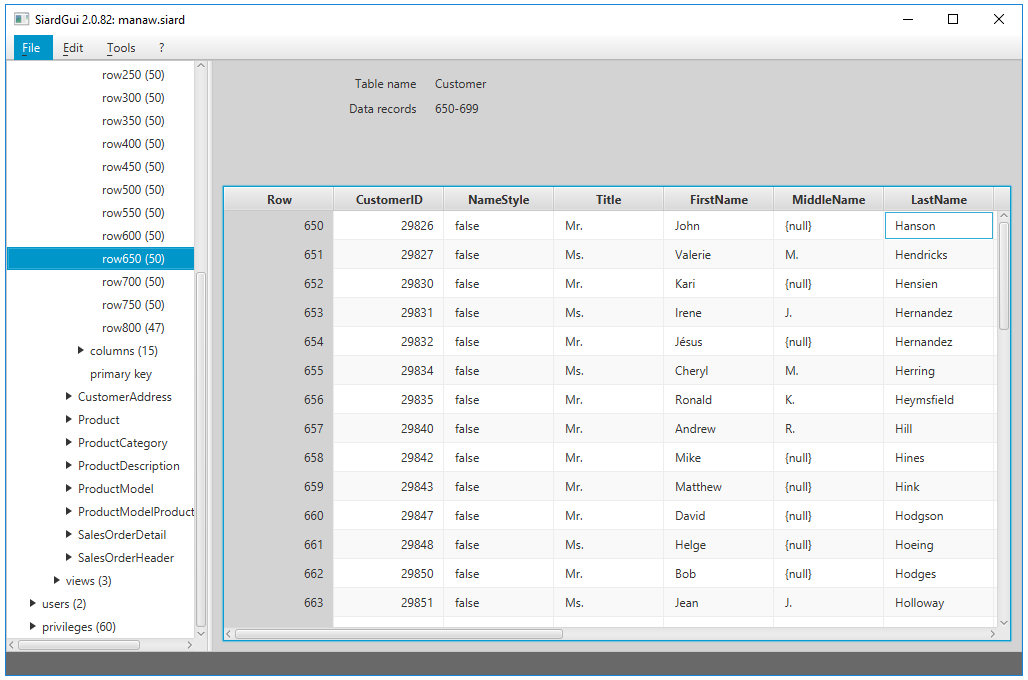
Similarly, it is sometimes desirable to search primary data tables for a particular string.



The dialog for entering the string is more complex. The search is limited to simple columns. A subset of simple columns to be searched can be selected. The search executed is a simple text search (numbers and dates are treated like the texts that are displayed in the table). Also the search proceeds sequentially and may take some time for a large table. In order to search faster or for data in objects of large or complex types (CLOB, BLOB, XML, UDT, ARRAY, ...), it is preferable to upload the database to a DBMS and use SQL for the search.

## Edit / Search next in primary data

Choosing *Search next in primary data* or pressing F3 locates the next occurrence of the entered string.



## Tools / Install ...

As has been mentioned in the chapter [Installation](about:blankinstallation.html) *SIARD Suite* can be installed anytime provided no installed version exists or the installed version has a lower version number.

## Tools / Uninstall

An installation of *SIARD Suite* can be removed by choosing this menu item. Before proceeding the user is asked whether the personal preferences should also be removed or be kept for later installations of *SIARD Suite*.

## Tools / Language

Any of the supported user interface languages can be chosen here.

## Tools / Check integrity

If the *SIARD archive* contains a message digest over the primary data, this is recomputed and compared to the value stored.

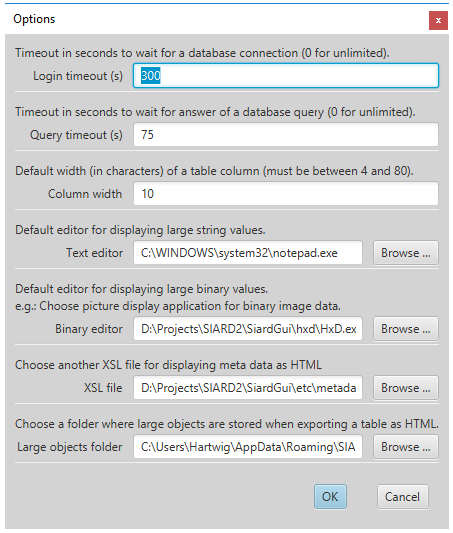
*SIARD Suite* computes and stores the message digest in the metadata immediately after the first download. If the *SIARD file* was unzipped and some data was changed and the data then re-zipped the integrity check will fail.

However, it is quite easy to compute a message digest over the primary data and stick it the metadata. Thus, the integrity check at best proves that changes were not made *manually* but rather using some program.

A better approach is to store all message digests generated at the time of download in a separate tightly managed database. Then the message digest in the metadata is first compared to the message digest in the external database. Only if it is unchanged, the integrity check can here be considered proof, that the primary data was not changed after the download.

## Tools / Options ...

The options dialog permits changing some values, which will be stored as personal configuration values. Only if these values are changed in an installed instance of *SIARD Suite* will they be stored to the personal preferences when the program terminates. Otherwise, changes are only valid until the end of the session.

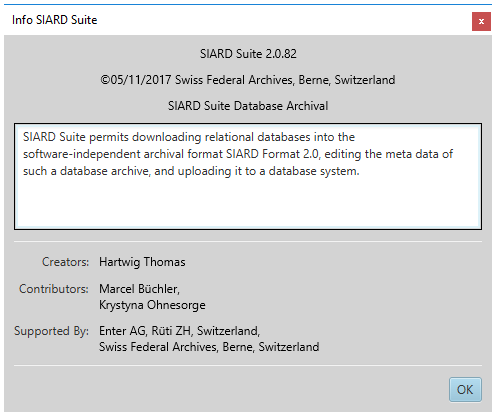


## ? / Help

This menu item displays this manual.

## ? / Info

This menu item displays the copyright notice for *SIARD Suite.*



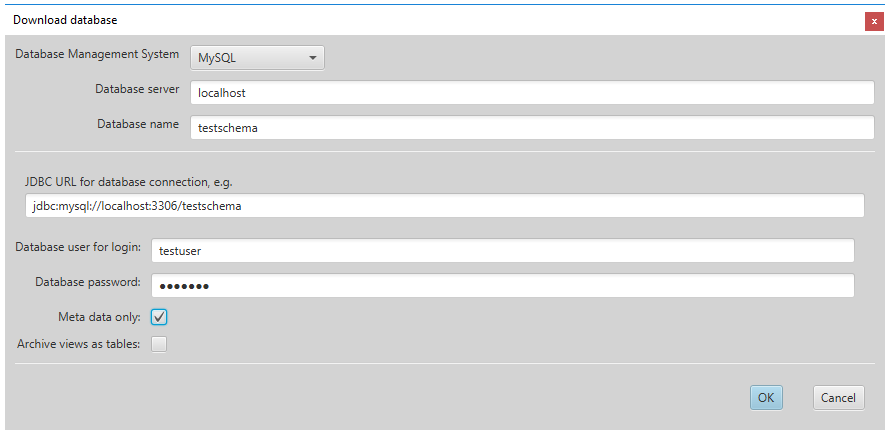
# External LOBs

The standard of the *SIARD Format 2.1* states, that large objects (LOBs) of a database may be stored in the external file system instead of inside the *SIARD archive*. The storage location must be specified in the metadata of the *SIARD archive*.

If some LOBs are to be stored externally, the corresponding LOB columns must first be associated with suitable external storage locations. Afterwards the database can be downloaded.

## Download only Metadata

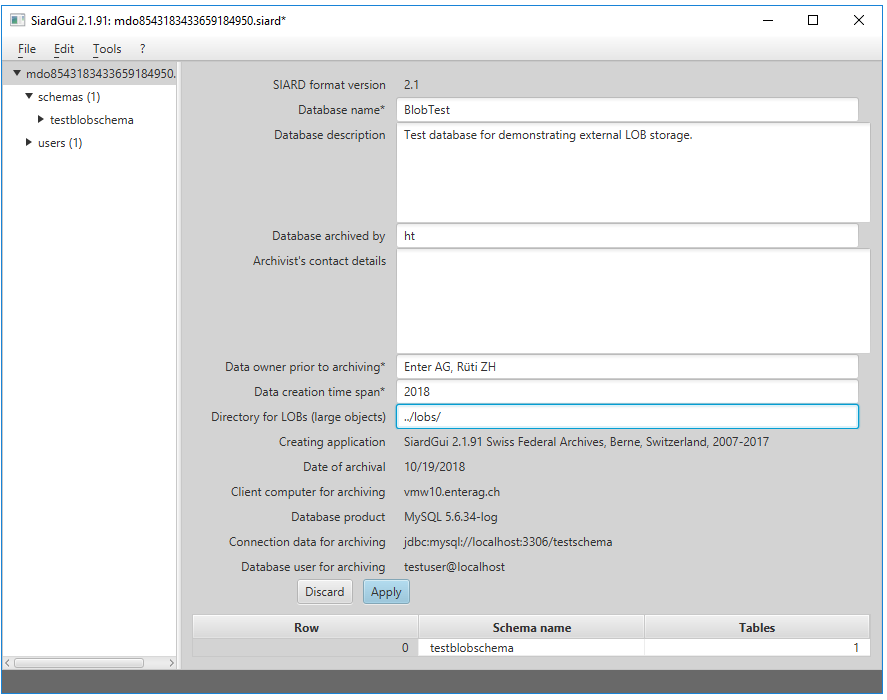
In order to associate storage locations with database columns, the metadata of a database must be downloaded first.



## Specify External Storage Locations

The metadata fields "LOB Folder" and "MIME Type" can now be entered.

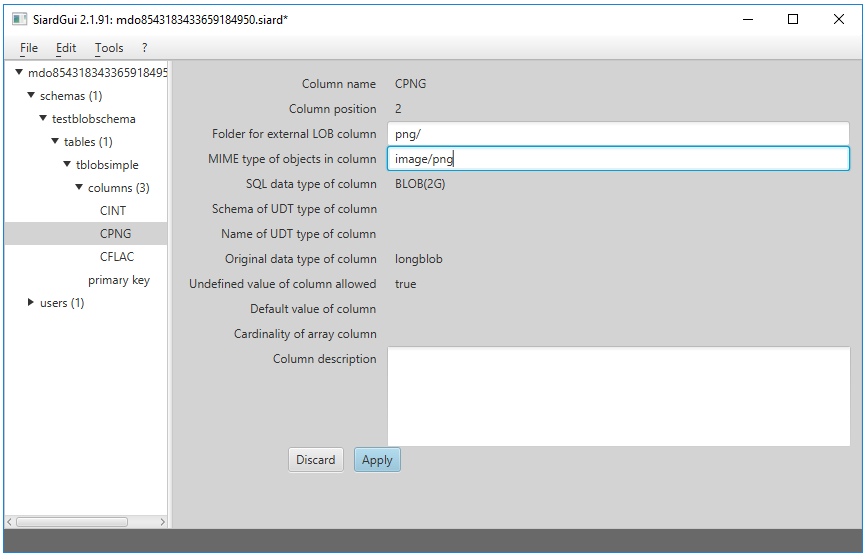
The storage location of a LOB column may be specified as an absolute *file:*-URI. However, it is recommended, to specify all LOB storage locations relative to a global URI in the global section of the *SIARD* metadata. In addition, it makes sense to specify the global metadata for database name etc. at the same time:



*N.B.: All LOB Folder locations must end with a slash indicating, that they refer to existing directories in the file system.*

The global external storage location may be specified as an absolute *file:*-URI. However, that would prevent moving the *SIARD archive* together with its external LOBs to a different location. Therefore it is recommended, to specify the external storage location relative to the directory where the *SIARD archive* resides, which is indicated by "../". In the example the global storage location is given as "../lobs/". Therefore, all external LOBs will be stored in locations relative to the folder *lobs* in the directory, where the *SIARD archive* is stored.

The storage location of a specific LOB column is then specified relative to the global external storage location:

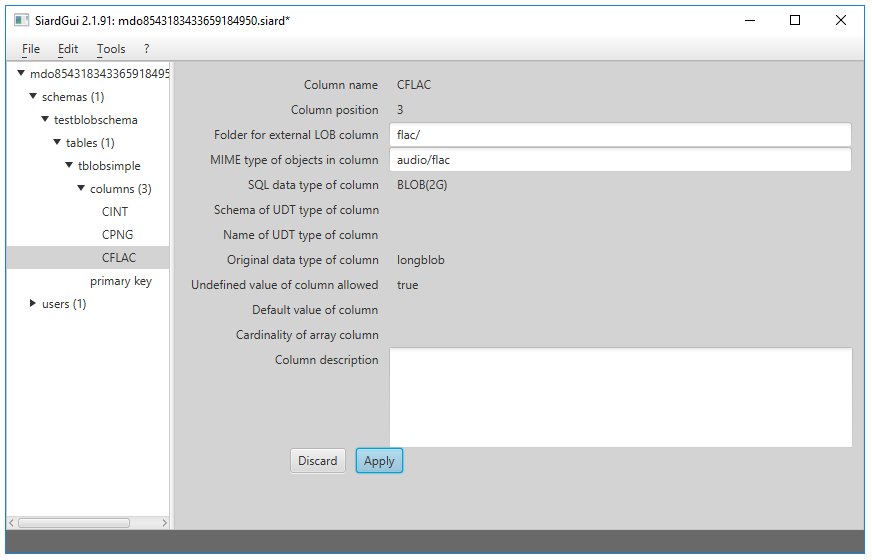


The value "png/" in this example directs *SIARD Suite* to store and load externally stored LOBs of the *CPNG* column of the table *tblobsimple* in the existing external folder *lobs/png/* relative to the location where the *SIARD file* is stored.

N.B.: If a maximum number of LOBs per folder is specified in the options dialog, the individual LOB files will not be stored in *lobs/png* directly, but rather in numbered subfolders of *lobs/png/*, which contain at most the configured maximum number of LOBs.

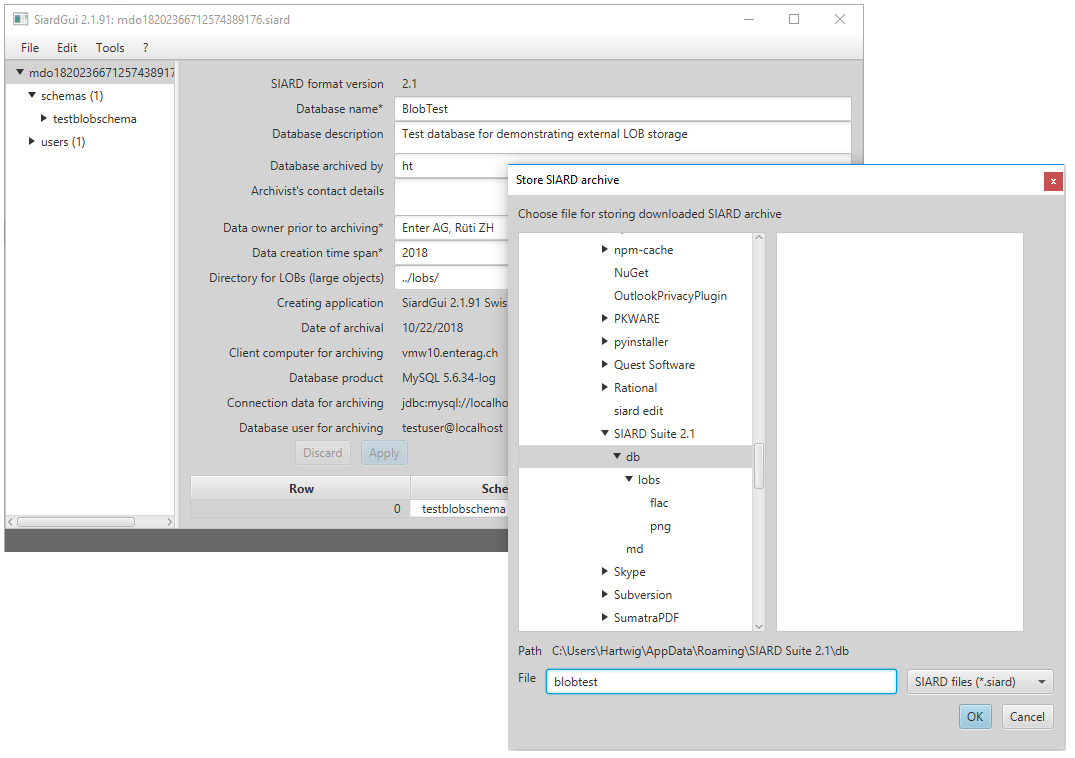
For externally stored LOB columns, a MIME type ("image/png" in the example) may be specified. This MIME type will be used by *SIARD Suite* to determine a suitable file name extension for the large objects. (E.g. *.png* for MIME type *image/png*.)

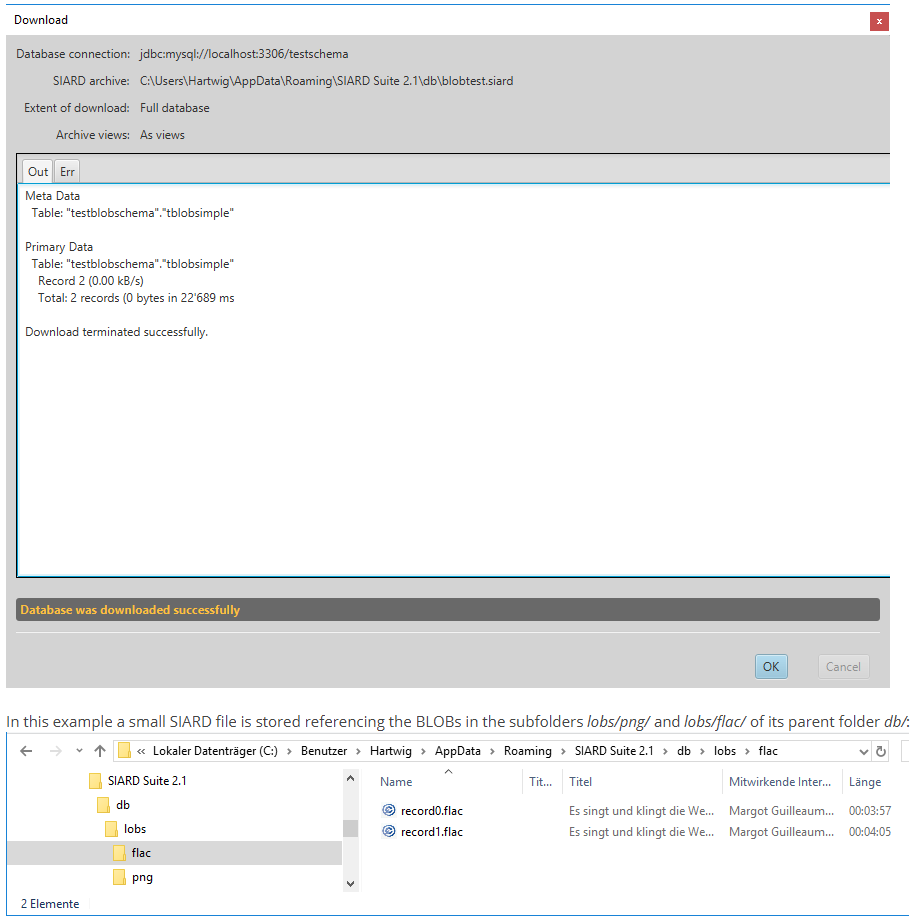
In a database, more than one LOB column can be stored externally:



After these preparations, it is advisable to display and store the metadata thus modified as an XML file.

## Download LOBs to External Locations

When such prepared metadata are available (e.g. imported from an external XML or downloaded as "meta data only") when a database is downloaded, they are used as "template" metadata. I.e. all entries for global metadata, descriptions, LOB folders etc. of the template are copied for the download of the primary data:



N.B.: The Windows Explorer shows embedded metadata of the FLAC files from the BLOBs, because the file name extension *.flac* is known to it.

# Command Line Invocation

The download and upload functionality of *SIARD Suite* can be called from the command line:

* [*SiardFromDb*](about:blank#siardfromdb): download of a *SIARD archive*,
* [*SiardToDb*](about:blank#siardtodb): upload of a *SIARD archive*.

## SiardFromDb

*SiardFromDb* is a command-line program, which extracts a database to a *SIARD archive*. One can use *SiardFromDb* to create:

* a *SIARD archive* (metadata and primary data) based on the database (option *-s*), and/or
* *SIARD* metadata XML, containing a definition of the database schema (option *-e*).

### Invocation



Specify <siardpath> as the folder where *SIARD Suite* is installed. The file *siardcmd.jar* is in the *lib* subfolder with its class *ch.admin.bar.siard2.cmd.SiardFromDb*, whose *main()* is invoked with *java* (it is better to use *javaw* under Windows).

The call syntax is displayed if the *-h (help)* option is entered on the command line.

### Arguments

|  |  |
| --- | --- |
| **Argument** | **Meaning** |
| -o | overwrite output file(s) if they exist |
| -v | archive views as tables |
| <login timeout> | timeout in seconds for login (0 for unlimited) |
| <query timeout> | timeout in seconds for query (0 for unlimited) |
| <import meta data> | name of meta data XML file to be used as a template |
| <external LOB folder> | folder for storing the data of the largest LOB column of database externally (contents will be deleted if they exist!) |
| <mime type> | MIME type of data in the largest LOB column of database (influences file extension of externally stored LOBs) |
| <JDBC URL> | JDBC URL of database to be downloaded e.g.  **for *MS Access***  *jdbc:access:D:\Projekte\SIARD2\JdbcAccess\testfiles\testdb.mdb*  **for *DB/2***  jdbc:dbserver.enterag.ch:50000/testdb  **for *H2 database***  *jdbc:h2:D:/Projekte/SIARD2/JdbcH2/data/testdb*  **for *MySQL***  *jdbc:mysql://dbserver.enterag.ch:3306/testdb*  **for *Oracle***  *jdbc:oracle:thin:@dbserver.enterag.ch:1521:orcl*  **for *Postgres***  *jdbc:postgresql://dbserver.enterag.ch:5432/testdb*  **for SQL Server**  *jdbc:sqlserver://dbserver.enterag.ch\testdb:1433*  (in bash shell the latter must be quoted with duplicated backslash:  *"jdbc:sqlserver://dbserver.enterag.ch\\testdb:1433"*) |
| <database user> | database user |
| <database password> | database password |
| <siard file> | name of .siard file to be written |
| <export meta data> | name of meta data .xml file to be exported |

### Notes

Either the *SIARD file* or the export metadata file or both must be given.

The *SiardFromDb* program should be used against a database snapshot, which doesn't change during the archiving process.

The option *archive views as tables* usually leads to data duplication and is therefore not recommended. However, it is useful when the archival user has read access to the views but not to the base tables.

The archiving process either wholly succeeds or wholly fails.

For large databases, it is recommended to download just the metadata XML beforehand. This gives insight into all the metadata and table sizes, which helps to estimate the download time needed. Furthermore, one should use the *-q 0* option for large tables, as it is not possible to estimate how many seconds a size query will take.

The conversion of TIMEs and TIMESTAMPs in the database depends on the local time zone. If the time 15:30 is stored in Zurich, the UTC time value 14:30 will be stored in the *SIARD file* – in winter. To suppress this conversion one must start *SiardFromDb* with the option



which tells SIARD to interpret all database times as UTC times.

### Archiving Database User

It is generally inadvisable to use the database administrator user (DBA, root, dbo, ...) for downloading a *SIARD archive*. The extent of the *SIARD archive* is defined by the objects to which the archiving database user has read-access. The global DBA usually has read access to all databases on the system as well as numerous system tables that should not be archived. Therefore, it is important that a suitable archiving user be created for the download if one does not exist.

## SiardToDb

*SiardToDb* is a command-line program, which loads a *SIARD archive* into a database for research purposes.

### Invocation



Specify <siardpath> as the folder where *SIARD Suite* is installed. The file *siardcmd.jar* is in the *lib* subfolder with its class *ch.admin.bar.siard2.cmd.SiardToDb*, whose *main()* should be invoked with *java* (it is better to use *javaw* under Windows).

The call syntax is displayed if the *-h (help)* option is entered on the command line.

### Arguments

|  |  |
| --- | --- |
| **Argument** | **Meaning** |
| -o | overwrite types and/or tables in the database if they exist |
| <login timeout> | timeout in seconds for login (0 for unlimited) |
| <query timeout> | timeout in seconds for query (0 for unlimited) |
| <siard file> | name of .siard file to be uploaded |
| <JDBC URL> | JDBC URL of the target database e.g.  **for *MS Access***  *jdbc:access:D:\Projekte\SIARD2\JdbcAccess\testfiles\testdb.mdb*  **for *DB/2***  *jdbc:dbserver.enterag.ch:50000/testdb*  **for *H2 database***  *jdbc:h2:D:/Projekte/SIARD2/JdbcH2/data/testdb*  **for *MySQL***  *jdbc:mysql://dbserver.enterag.ch:3306/testdb*  **for *Oracle***  *jdbc:oracle:thin:@dbserver.enterag.ch:1521:orcl*  **for *Postgres***  *jdbc:postgres://dbserver.enterag.ch:5432/testdb*  **for *SQL Server***  *jdbc:sqlserver://dbserver.enterag.ch\testdb:1433*  (in bash shell the latter must be quoted with duplicated backslash:  *"jdbc:sqlserver://dbserver.enterag.ch\\testdb:1433"*) |
| <database user> | database user |
| <database password> | database password |
| <schema> | schema name in *SIARD file* |
| <mappedschema> | schema name to be used in database |

### Notes

As older databases are not SQL:2008 compliant, a considerable amount of manual configuration effort is unavoidable in preparation for the upload. There are no Schema objects in MS Access. User and Schema objects are not separate in Oracle. Schemas and databases are not distinct in MySQL. Therefore, target schemas must be created before upload. Also the database user must have the right to create tables and types in those schemas. Because this is not always easily possible, *SIARD* schemas are mapped to database schemas according to the list of schema mappings on the command line.

Uploading only creates tables and types and attempts to enable unique and foreign key constraints. No other database objects are created. If the constraints cannot be enabled, a warning is issued but the upload is nevertheless considered to be successful. Even without constraints SQL SELECT queries can be issued against the database.

Furthermore, certain sacrifices are made. In MS Access, all tables end up in the same MDB/ACCDB. In Oracle, all names longer than 30 characters are abbreviated. To avoid collisions, table and column names are extended by a counter. (E.g. "A far too long a table name for Oracle" becomes "A far too long a table name01".)

Where the maximum precision and the maximum number of decimals (for instance MS Access) are smaller than required, the values are uploaded with less precision. *SIARD* helps as much as is possible in the target database system. Consulting the database metadata via *SiardGui* allows the correct assignment of designations and values.

The conversion of TIMEs and TIMESTAMPs in the database depends on the local time zone. The UTC time 14:30 in the *SIARD* file is uploaded in Zurich as the local time 15:30 to the database – in winter. To suppress this conversion one must start *SiardToDb* with the option:



which tells *SIARD* to interpret all database times as UTC times.

# Database Management Systems

There are some areas where the various database management systems (DBMS) are treated differently in *SIARD Suite*:

* JDBC URL for connecting to a database,
* Handling of proprietary data types,
* Preparation of a database for download,
* Preparation of a database for upload.

## JDBC URL for connecting to a database

*SIARD Suite* documents the standard JDBC URL for connecting to a supported database system. However, there are many variations how database management systems embed platform (e.g. Windows login) or network security (e.g. Kerberos) into their access control. It is not possible here to document every specialty of every DBMS. However, as long as an acceptable JDBC URL is used, a connection to a database can be established using *SIARD Suite*. For details about an acceptable JDBC URL for a DBMS its native documentation must be consulted.

**MS Access**

The JDBC implementation for MS Access only permits a single type of JDBC URL: jdbc:access:<path to mdb/accdb>

**DB/2**

<https://www.ibm.com/support/knowledgecenter/en/SSEPGG_9.7.0/com.ibm.db2.luw.apdv.java.doc/src/tpc/imjcc_r0052342.html>

**H2 database**

<http://www.h2database.com/html/features.html#database_url>

**MySQL**

<https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-reference-jdbc-url-format.html>

**Oracle**

<http://docs.oracle.com/cd/B28359_01/java.111/b31224/jdbcthin.htm>

**PostgreSQL**

<https://jdbc.postgresql.org/documentation/head/connect.html>

**SQL Server**

<https://docs.microsoft.com/en-us/sql/connect/jdbc/building-the-connection-url>

When connecting to a database using *SIARD Suite* fails, try a native connection not involving *SIARD Suite* first. When that is successful but the JDBC URL still fails to connect to the database, try the same JDBC URL using [SQuirreL](http://squirrel-sql.sourceforge.net/" \t "_blank).

## Handling of proprietary data types

The proprietary data types are mapped to SQL:2008 data types by the JDBC wrapper for each DBMS. The mapping is documented in the LibreOffice tables*Jdbc<DBMS>-TypeInfo.ods* in the folder *doc/datatypes*.

## Preparation of a database for download

*SIARD Suite* will download everything that is readable by the database user used for the connection. Choosing a suitable database user for the download determines the extent of the database being archived. Often a suitable "technical user" of a database application associated with the database has exactly the access rights needed for the archival of the database.

However, if no such user is available, one should *not* use the master database user (database administrator, DBA, dbo, root, SYSTEM, sa, ...) for the download because this master database user can read many system tables that should not be archived with the database. Instead, it is recommended, to create a new database user for the purpose of archiving a database. This *archival user* then should be granted read access to all schemas, tables, views, and types needed for archival. The documentation of the DBMS in question must be consulted to learn how to create such a user and grant the necessary rights.

After a suitable archival user has been determined or created, the download of the database can proceed using the credentials of the archival user.

## Preparation of a database for upload

For uploading a database to a DBMS using *SIARD Suite*, suitable database schemas must be available on the target DBMS. Those schemas can then be used in the schema mapping part of the upload dialog (or in the schema mapping part of the arguments of the command-line application *SiardToDb*).

Ideally, the target schemas are empty. However, due to security constraints one does not always have the choice to choose or create schemas freely. If the target schemas are not empty, *SIARD Suite* will only upload the database if either no name collisions of tables and types prevail, or else the "overwrite" option has been checked.

The database user chosen for the upload of a database must have the privilege to create types and tables and insert data into them. The DBMS documentation must be consulted for information as to how the database schemas can be created and how the database user can be granted the privilege to create types and tables in them and insert data into them. If one has access to the master database user (database administrator, DBA, dbo, root, ...) it may be convenient to use it for upload of a database. However, in that case the "overwrite" option should not be selected. Otherwise, there is too great a risk that vital tables or types are overwritten.

# Logging

When problems occur with using the *SIARD Suite*, it is useful to save an execution log. Such a log (or – for security reasons – fragments of it) should accompany error reports to maintenance.

To create such a log, proceed as follows:

The file *logging.properties* is found in the *etc* folder in the *SIARD* distribution. In this file, logging with level WARNING is redirected to the console and all levels are directed to a log file *%t/siard2%u.log*. Here the %t indicates the temporary folder (value of the TEMP or TMP environment variable) and the %u is replaced by a number. Under Windows this will result in something like *C:\Users\<user>\AppData\Local\Temp\siard20.log*).

The global level is initially set to INFO. More logging can be enabled by changing the line  
***.level=INFO***  
to  
***.level=ALL****.*

This will slow down execution.

# Limitations

The SIARD Format implies the following limitations:

* The size of a *SIARD file* cannot be larger than 18’446’744’073’709’551’615 Bytes (ca. 18 ExaBytes) (ZIP64 limitation).
* The number of (table and lob) files cannot be larger than 4'294'967’295 (ca. 4 billion) (ZIP64 limitation).

These limitiations are probably irrelevant because real databases will not reach such sizes for quite some time.

The *SIARD Suite* is further limited by this condition:

* All of the metadata of the database must fit into JAVA memory (heap).

This limitation, however, can be reached if the CPU storage is small or the database is very complex. One can circumvent the problem by running *SIARD* on a machine with enough main storage space (e.g. 4 GB) and manually increase the JAVA heap using the command-line option –Xmx2000m.