



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement des Innern EDI

Schweizerisches Bundesarchiv BAR

Abteilung Informationsüberlieferung

Dienst Sicherung und Archivierungslösungen

Siard Suite Manual

Version 0.1.0-SNAPSHOT

Table of contents

1. Summary	1
1.1. User Manual Language	1
1.2. Latest release	1
2. Introduction	1
2.1. New features of Version 2.2	1
2.2. Intellectual Property Rights	2
3. SIARD Concept	4
4. Introduction to Database Archival	5
5. Requirements	6
5.1. Java	6
5.2. Architecture (32-bit/64-bit)	6
5.3. Databases	6
5.4. Installing and Running the Application	7
6. Using the SIARD Suite	7
6.1. SIARD Suite GUI	8
6.2. SIARD Suite CLI	10
6.3. SiardFromDb	10
6.4. SiardToDb	12
7. Tips and Tricks	14
7.1. Preparing a database for download	14
7.2. Preparing a database for upload	14
7.3. Time, Timestamps and Time Zones	14
8. Restrictions	15

1. Summary

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

It describes the

- technical requirements for use
- installation
- execution

of the SIARD Suite and its components.

1.1. User Manual Language

The extended version of the user's manual is only available in English. A short version is also available in French, German or Italian and can be found directly in SIARD-Suite.

1.2. Latest release

This version of the manual refers to the latest release of SIARD Suite 0.1.0-SNAPSHOT

2. Introduction

The SIARD format and the SIARD Suite application 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 and SIARD Suite 2.1 in 2016-2018.

In 2015-2018, version 2.1 of the SIARD format was specified by the Swiss Federal Archives in collaboration with the EU project E-ARK and KOST. Like version 1.0, SIARD format 2.1 has been approved as standard eCH-0165 by the association eCH E-Government Standards. In 2021, the DILCIS Board has developed and approved the enhanced SIARD Format 2.2. Puzzle ITC has been commissioned by the Swiss Federal Archives to develop SIARD Suite 2.2 according to the enhanced format.

SIARD Suite 2.2 is the reference implementation for the archiving of relational databases in the standard SIARD format 2.2.

This document is the SIARD Suite 2.2 user manual.

2.1. New features of Version 2.2

The main new features of SIARD format 2.2 compared to version 2.1 concern:

- Large objects (LOBs) referenced by DATALINKs and stored as external files in the database, according to the SQL standard;
- large objects (LOBs) stored outside the SIARD archive and segmented into several folders and file parts.

SIARD files conforming to SIARD formats prior to 2.2 can be read by SIARD Suite 2.2. However, by saving any changes, SIARD files are automatically converted to SIARD format 2.2.

SIARD Suite 2.2 is the reference implementation for archiving relational databases in the standardized SIARD 2.2 format.

2.2. Intellectual Property Rights

SIARD Suite has been developed by Enter AG and Puzzle ITC GmbH for the Swiss Federal Archives. The copyright holders release SIARD Suite as Open Source software under the CDDL-1.0 license (in the SIARD distribution as licenses/CDDL-1.0.txt).

SIARD Suite relies on the following third-party components:

- 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 in the SIARD distribution as 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 in the SIARD distribution as licenses/java-license.txt
- ini4j
 - INI file handler for LINUX desktop links from Apache <http://ini4j.sourceforge.net/>
 - License: Apache License 2.0 in the SIARD distribution as licenses/Apache-license-2.0.txt
- mslinks
 - LNK file handler for Windows desktop links from BlackOverlord666 <https://github.com/BlackOverlord666/mslinks>
 - License: WTFPL License in the SIARD distribution as licenses/WTFPL.txt
- SiardApi from Swiss Federal Archives
 - License: CDDL-1.0 license in the SIARD distribution as 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 in the SIARD distribution as 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) in the SIARD distribution as

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) in the SIARD distribution as licenses/LGPL2.1.txt
- Multi-Schema Validator (MSV) from SUN/Apache <https://github.com/kohsuke/msv/>
 - License: BSD license (BSD-2) in the SIARD distribution as licenses/BSD-2.txt.
- Zip64File from Enter AG
 - License: CDDL-1.0 license in der SIARD-Distribution als licenses/CDDL-1.0.txt.
- SqlParser from Enter AG
 - License: CDDL-1.0 license in the SIARD-Distribution als 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)) in the SIARD distribution as licenses/BSD-3.txt
- SiardCmd from Swiss Federal Archives
 - License: CDDL-1.0 license in the SIARD distribution as licenses/CDDL-1.0.txt
- JTS Topology Suite (Version 1.14 - used by MySQL for the GEOMETRY extension) from Martin Davis <http://tsusiatsoftware.net/>
 - License: GNU Library General Public License (LGPLv2.0) in the SIARD distribution as licenses/LGPLv2.0.txt.
- JdbcBase from Swiss Federal Archives
 - License: CDDL-1.0 license in the SIARD distribution as licenses/CDDL-1.0.txt
- JdbcPostgres from Swiss Federal Archives
 - License: CDDL-1.0 license in the SIARD distribution as licenses/CDDL-1.0.txt
- JDBC Driver for Postgres (postgresql-42.2.5.jar) from the PostgreSQL Global Development Group
 - License: Postgres License in the SIARD distribution as licenses/licensePostgres.txt.
- JdbcOracle from Swiss Federal Archives
 - License: CDDL-1.0 license in the SIARD distribution as 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 in the SIARD distribution as licenses/licenseOracle.txt.
- JdbcMySQL from Swiss Federal Archives
 - License: CDDL-1.0 license in the SIARD distribution as licenses/CDDL-1.0.txt
- JDBC Driver for MySQL (Version 8.0.18) from Oracle <https://dev.mysql.com/downloads/connector/j/>

- License: GNU General Public License (GPLv2.0) in the SIARD distribution as licenses/GPLv2.0.txt.
- JdbcMsSql from Swiss Federal Archives
 - License: CDDL-1.0 license in the SIARD distribution as 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 in the SIARD distribution as licenses/license41.txt.
- JdbcDb2 from Swiss Federal Archives
 - License: CDDL-1.0 license in the SIARD distribution as 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 in the SIARD distribution as licenses/IBM JDBC 4 License.txt and 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 in the SIARD distribution as licenses/CDDL-1.0.txt
- Jackcess (Version 2.1.6) from Health Market Science <http://jackcess.sourceforge.net/>
 - License: Apache License, Version 2.0 in the SIARD distribution as 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 in the SIARD distribution as licenses/Apache-license-2.0.txt
- EnterUtilities from Enter AG
 - License: CDDL-1.0 license in the SIARD Distribution as licenses/CDDL-1.0.txt.

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

3. SIARD Concept

The Swiss Federal Archives are obliged to archive the documents of the federal administration regardless of the information medium. Therefore, the problem of long-term archiving of relational databases has to be solved.

Since 2007, the SIARD format has been used by the Swiss Federal Archives and many other archives

around the world as a normalization format for the long-term preservation of relational databases.

With the SIARD format 2.2, the databases are stored according to the SQL:2008 standard in order to guarantee long-term availability. The data content is stored as a collection of XML files. Since the resulting archive format is based on these two ISO standards, it is assumed that the data will be readable in the long term.

An important requirement of the data content archived in the SIARD format is that it should have a "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 business 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?" by Hartwig Thomas.

The SIARD format stores the archived database schema definition in SQL:2008-compliant XML files, while the table and field documentation and the actual data are also stored in XML files. To avoid excessive XML file size inflation, Binary Large Objects (BLOBs) and Character Large Objects (CLOBs) referenced in the XML files are stored in separate (binary) files.

This document does not further explain the SIARD format and structure, which are described in a separate document delivered with the SIARD Suite. In 2013, the SIARD format was approved as an eCH standard. In 2018, version 2.1 of the SIARD format was made available as standard eCH-0165. In 2021, the DILCIS Board developed and approved the extended SIARD Format 2.2.

4. Introduction to Database Archival

This is a brief introduction to archiving databases with SIARD Suite. It also deals with organizational issues that should be taken into account.

1. Make sure that you know which parts of the database need to be archived. If necessary, contact the person in charge, for example, the person in charge of the archives.
2. Prepare the database for archiving: Create a new user on the database system with read-only privileges for the objects to be archived. If necessary, create a copy of your database (or specific tables/parts of it) or create views. The database must not be modified during the archiving process, otherwise the extraction with SIARD Suite will fail. Do not archive from a live system.
3. Download the database with SIARD Suite.
4. Quality control: check the SIARD file to make sure it contains everything you need, test some entries to make sure everything went well.
5. Advanced quality control: reload the SIARD file on a database system. Execute some defined queries on the original database as well as on the archived one and compare the results.
6. Add metadata to the SIARD file.
7. Define which external documentation must be archived with the SIARD file to ensure the comprehensibility of the data (e.g. code tables, system documentation, entity relationship diagram, ...).

5. Requirements

5.1. Java

The SIARD Suite application is a Java FX application. As such, it can run on any system where a Java Runtime Environment 8 (JRE) with the JavaFX extension is installed.

You can download a JRE 8 with the JavaFX extension from various sites:

- <https://www.azul.com/downloads/?version=java-8-lts&package=jre-fx#zulu>
- <https://www.oracle.com/java/technologies/downloads/#jre8>

However, there are distributions of the SIARD Suite that already contain the required Java runtime and there are native Installers for Windows, Linux and Mac that you can use.

Each distribution is listed below. All distributions can be downloaded from <https://github.com/sfa-siard/siard-suite/releases>.

Distribution Name	Description
SIARD-Suite-v0.1.0-SNAPSHOT	Distribution without runtime, requires JRE with JavaFX installed on the system
SIARD-Suite-v0.1.0-SNAPSHOT-Linux	Distribution with included runtime for Linux
SIARD-Suite-v0.1.0-SNAPSHOT-Windows	Distribution with included runtime for Windows
SIARD-Suite-v0.1.0-SNAPSHOT-macOS	Distribution with included runtime for MacOS
SIARD-Suite-v0.1.0-SNAPSHOT-Linux-deb	Native installer for Linux
SIARD-Suite-v0.1.0-SNAPSHOT-win-exe	Native installer for Windows (exe)
SIARD-Suite-v0.1.0-SNAPSHOT-win-msi	Native installer for Windows (msi)
SIARD-Suite-v0.1.0-SNAPSHOT-macOS-dmg	

Note: When using the native installer, the Siard CLI (Command Line Interface) is not available. To use the CLI, a distribution with or without an included JRE should be used.

5.2. Architecture (32-bit/64-bit)

In previous versions of SIARD Suite, the dependence on ODBC required the use of 32-bit JAVA to access MS Access databases. SIARD Suite 2.2 no longer uses ODBC to access MS Access databases. Therefore, these databases can be accessed on any platform (e.g. LINUX) and SIARD Suite 2.2 is compatible with both 32-bit and 64-bit JAVA. It is recommended to choose the JAVA architecture according to the architecture of your operating system.

5.3. Databases

SIARD Suite 2.2 currently supports the following database systems:

- MS Access 2007 or higher
- DB/2 8 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.

5.4. Installing and Running the Application

When using the native installers, the installation is simply done through the system. If necessary, administrator privileges are used for this. The installers also create the appropriate shortcuts to start the application.

The distributions with and without JRE are delivered as a ZIP archive. This can be unpacked anywhere. After that the application can be started by one of the start scripts:

script name	description
siard-suite (.bat)	Start the graphical user interface for Unix or Windows systems
siard-from-db (.bat)	Archiving a database (CLI)
siard-to-db (.bat)	Upload a SIARD archive to a database (CLI)

If the installation of SIARD-Suite is prevented due to security barriers on your computer, please follow the instructions in the additional document “SIARD-Suite – getting started”

6. Using the SIARD Suite

The SIARD Suite is composed of the following components

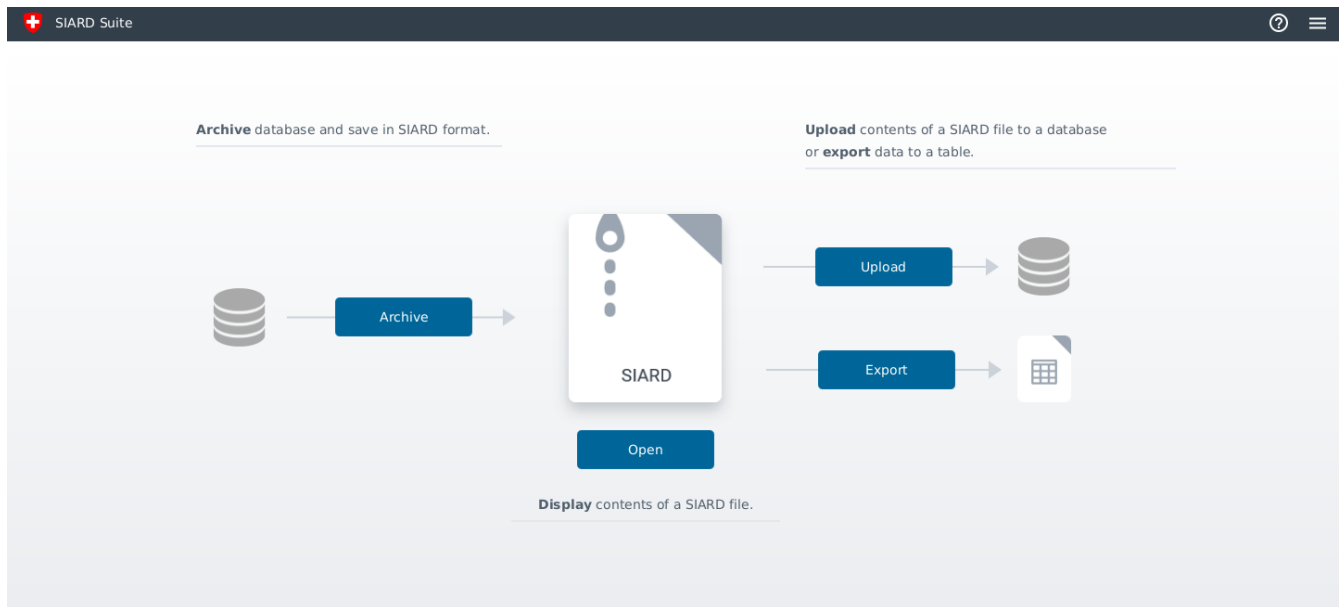
- Graphical user interface (GUI)
- SiardFromDb (CLI)
- SiardToDB (CLI)

The purpose and the use of these components are explained below.

6.1. SIARD Suite GUI

The GUI is the central application to create SIARD archives and to view them later without modifying the primary data of the archive. For complex analysis of the archived data, the archive should be reloaded into a database.

The start screen is the entry point to the application. The four main functionalities of the application can be initiated from here:



- Archive database
- Open SIARD archive
- Upload a SIARD archive to a database
- Export data from a SIARD database

The application guides the user through the different operations.

6.1.1. Archiving Databases with Binary Data

The SIARD Format 2.2 standard specifies that large objects (LOBs) of a database can be stored in the external file system.

The location of the data can be recorded during archiving in the "metadata" step.

Meta Data

Edit meta data for your archive

What will be archived?

Database name

testdb

Database description (optional)

Delivering office: Data databaseOwner before archiving

Time of origin

Who has archived?

Your name (optional)

Your contact details (e.g. e-mail, phone number) ⓘ

Folder to export large objects (optional)

../lobs/

☐ Export views as tables

Back

Cancel

Next

Save only metadata

The path can be absolute as a file URI (**file:/**) or relative, e.g. **../lobs/** - in either case the path must end with a slash (/). It is recommended to use relative paths so that the archive can be moved along with the external LOB data - this is not possible if absolute paths are used. The structure within the specified folder is then automatically generated and follows the structure of the database: **schema/table/column**. SIARD Suite automatically detects the MIME type and the file extensions are automatically set correctly. This also allows the binary files to be opened directly from SIARD Suite in the correct application installed on the system.

6.1.2. Settings

Under the menu item "Options" you can configure the application.

Options

X

Login Timeout (s)

Query Timeout (s)

Cancel

Save

The following settings are available:

Option	Description
Login Timeout (s)	Maximum time to wait for a database connection (in seconds). 0 for unlimited
Query Timeout (s)	Maximum time to wait for a database query (0 for unlimited).

6.2. SIARD Suite CLI

In addition to the graphical user interface, databases can also be archived or an archive can be read back into a database via the command line.

For this purpose the scripts `siard-from-db` and `siard-to-db` are available in the ZIP distribution.

6.3. SiardFromDb

With `siard-from-db` a database can be archived without using the GUI.

6.3.1. Usage

In a terminal, change to the directory where the ZIP archive was unpacked, open the directory `bin` there and execute it with

```
siard-from-db [-h] | [-o] [-v] [-l=<login timeout>] [-q=<query timeout>] [-i=<import meta data>] [-x=<external lob folder>] [-m=<mime type>] -j=<JDBC URL> -u=<database user> -p=<database password> -s=<siard file> -e=<export meta data>"
```

6.3.2. Arguments

argument	description
-o	Overwrite output file(s) if available
-v	archive views as tables
<login timeout>	login timeout in seconds (0 for unlimited)
<query timeout>	query timeout in seconds (0 for unlimited)
<import metadata>	name of the metadata XML file to use as template
<external LOB folder>	Folder where the data of the largest LOB (large object) column of the database should be stored externally (contents with the same name will be overwritten!)
<mime type>	MIME type of the data of the largest LOB column of the database (influences the choice of the name extension of the externally stored LOBs)
<JDBC URL>	JDBC URL of the database to be downloaded. Examples: MS Access: jdbc:access:<path>/<filename>.mdb DB/2: jdbc:<host>:50000/<db-name> MySQL: jdbc:mysql://<host>:3306/<db-name> Oracle: jdbc:oracle:thin:@<host>:1521:orcl postgres: jdbc:postgresql://<host>:5432/<db-name> SQL server: jdbc:sqlserver://<host>/<db-name>:1433
<database user>	database user
<database password>	database password
<siard file>	name of the .siard file to create
<export metadata>	name of the .xml metadata file to export

6.3.3. Notes

At least one of the two parameters **siard-file** or **export metadata** must be set (it is also possible to use both parameters).

Note: The contents of the database should not be changed during the archiving process - use a snapshot of the database if necessary.

The option `Archive views as tables` usually leads to redundant storage of the same data and is therefore not recommended. It is useful if the database user used for archiving can see the contents of the views but not the base tables.

The archiving operation is either completely successful or fails.

For large databases, it is recommended to download the metadata XML first. This contains all the metadata and table sizes and helps to estimate the execution time required for the download. Also, for large tables, the `-q 0` option should be used, as it is not possible to estimate how many seconds a size query will take.

6.3.4. Database user

It is generally not recommended to use the database administrator (DBA, root, dbo, ...) with all rights to download a SIARD archive. The scope of the data stored in the SIARD archive is defined by the objects to which the archive user has read access. The global DBA can usually read all databases on the system, plus many system tables that should not be archived. For this reason, it is important to use a suitable archive user for the download, which may need to be created for this purpose.

6.4. SiardToDb

SiardToDb is a command line program that loads a SIARD archive into a database instance for retrieval.

6.4.1. Usage

```
siard-to-db [-h] [-o] [-l=<login timeout>] [-q=<query timeout>] -s=<siard-file>  
-j=<JDBC-URL> -u=<database user> -p=<database password [<schema> <mappedschema>]
```

6.4.2. arguments

argument	description
<code>-o</code>	overwrite identically named types and/or tables in the database, if any
<code><login timeout></code>	timeout in seconds for login (0 for unlimited)
<code><query timeout></code>	query timeout in seconds (0 for unlimited)
<code><siard file></code>	name of the .siard file to upload

argument	description
<JDBC URL>	<p>JDBC URL of the target database Examples:</p> <p>MS Access: jdbc:access:<path>/<file name>.mdb</p> <p>DB/2: jdbc:<host>:50000/<db-name></p> <p>MySQL: jdbc:mysql://dbserver.enterag.ch:3306/<db-name></p> <p>Oracle: jdbc:oracle:thin:@<host>:1521:orcl</p> <p>postgres: jdbc:postgres://<host>:5432/<db-name></p> <p>SQL server: jdbc:sqlserver://<host>\<db-name>:1433</p>
<database user>	database user
<database password>	database password
<schema>	name of the schema in the SIARD archive
<mappedschema>	name of the schema to be used in the database

6.4.3. Notes

Since older databases are not SQL:2008 compliant, some manual preparation for upload is inevitable. There are no schema objects in MS Access. In Oracle, users and schemas are inseparable. In MySQL, schema objects and databases are one and the same. Therefore, all schemas must be created manually before uploading. In addition, the database user must have the right to create tables and types in these schemas. Since this is not always easy, the database schemas are mapped to the schema names according to the mapping list on the command line.

The upload only creates tables and types and tries to activate primary and foreign keys. No other objects are created in the target database. If the keys cannot be activated, a warning is issued, but the upload is still considered successful. SQL SELECT queries on the database can be executed even without these keys.

There may also be limitations when uploading. With MS Access, all tables end up in the same MDB/ACCDB. With Oracle, all names longer than 30 characters are truncated. To avoid collisions, table and column names are given a counter. (For example, "A table name that is way too long for Oracle" becomes "A table name that is Table01").

If the maximum precision and the maximum number of decimal places (e.g. in MS Access) are less than required, the values are uploaded with less precision. SIARD helps itself as much as possible in the target database system. If the metadata of the database is consulted with SiardGui, the correct labels and values can be assigned.

7. Tips and Tricks

7.1. Preparing a database for download

SIARD Suite downloads all the data visible to the database user used for the connection. The choice of a suitable database user therefore determines the size of the archived database. Often a suitable "technical database user" of a database application is available, which is used to manage the database. This user usually has exactly the access rights needed to archive the database.

However, if such a database user is not available, the database administrator (DBA, dbo, root, SYSTEM, sa, ...) should not be used for the download. This is because this user has read access to many system tables and tables from other databases that are not to be archived. Instead, it is recommended that a database user be created for archiving. This archiving user must have read access to all schemas, tables, views and types belonging to the database to be archived. Refer to your DBMS documentation for information on how to create a user and grant it the necessary privileges.

Once a suitable user has been found or created, the database download can be performed using the user's credentials.

7.2. Preparing a database for upload

To upload a database to an instance of a DBMS with SIARD Suite, suitable database schemas must be available in the target database. These schemas can then be mapped to the schemas in the SIARD archive in the schema mapping of the upload dialog (or in the schema mapping of the SiardToDb command line).

Ideally, the target schemas should be empty. However, depending on the DBMS security settings, it is not always possible to create or select schemas at will. If the target schemas are not empty, the archived data will only be uploaded if either there are no name collisions for the types and tables, or the Overwrite option has been explicitly selected.

The database user whose credentials are used for the upload must have the right to create types and tables in the target schemas and to insert data into these tables. Refer to your DBMS documentation for information on how to create schemas and give the database user the necessary rights. The easiest way, if possible, is to use the credentials of the database administrator. This is because the database administrator usually has all the necessary privileges. In this case, however, the Overwrite option should not be selected. Otherwise, the risk of overwriting important tables or types in the system is too great.

7.3. Time, Timestamps and Time Zones.

The conversion of **TIMES** and **TIMESTAMPS** in the database depends on the local time zone. The UTC time 14:30 in the SIARD archive is loaded into the database as local time 15:30 when uploading in Zurich - in winter. To suppress this conversion, SiardToDb must be started with the option **-Duser.timezone=GMT**, which causes SIARD to interpret all database times as UTC times.

8. Restrictions

The SIARD format has the following limitations

- The size of a SIARD file must not exceed 18'446'744'073'709'551'615 bytes (about 18 ExaBytes) (ZIP64 limitation).
- The number of files (tables and LOB) must not exceed 4'294'967'295 (about 4 billion) (ZIP64 limitation).

The SIARD Suite is also limited by

- All the metadata of the database must fit in the JAVA memory (heap). This limitation can occur with an unfavorable constellation of available main memory and complexity of the database metadata. It can be circumvented by starting the SIARD applications on a machine with sufficient main memory and, if necessary, enlarging the JAVA heap with the JAVA command line parameter `-Xmx2000m` or a similar parameter on the JAVA command line.