



**Hartwig Thomas**, 22 December 2014

Document version 1.22

---

# SIARD Suite Manual



---

Published by:

Swiss Federal Archives  
Archivstrasse 24  
3003 Bern  
Switzerland

# Summary

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

It describes the

- technical prerequisites for deployment
- installation
- execution

of the *SIARD Suite* and its components.

## **1 Contents**

<b>1</b>	<b>CONTENTS.....</b>	<b>3</b>
<b>2</b>	<b>INTRODUCTION.....</b>	<b>4</b>
2.1	INTELLECTUAL PROPERTY RIGHTS .....	4
<b>3</b>	<b>SIARD CONCEPT .....</b>	<b>6</b>
<b>4</b>	<b>PREREQUISITES FOR USING THE <i>SIARD SUITE</i> .....</b>	<b>7</b>
4.1	JAVA.....	7
4.2	DATABASES .....	7
<b>5</b>	<b>INSTALLATION .....</b>	<b>9</b>
<b>6</b>	<b>SIARD COMPONENTS.....</b>	<b>10</b>
<b>7</b>	<b>SIARDEDIT.....</b>	<b>11</b>
7.1	INITIAL EXECUTION.....	11
7.2	MAIN WINDOW.....	12
7.3	MENU .....	15
<b>8</b>	<b>SIARDFROMDB.....</b>	<b>25</b>
8.1	INVOCATION .....	25
8.2	INVOCATION PARAMETERS.....	25
8.3	NOTES.....	26
8.4	DATABASE PREPARATION .....	26
<b>9</b>	<b>SIARDTODB .....</b>	<b>31</b>
9.1	INVOCATION .....	31
9.2	INVOCATION PARAMETERS.....	31
9.3	NOTES.....	32
9.4	DATABASE PREPARATION.....	32
<b>10</b>	<b>SUPPORTED DATABASE SYSTEMS .....</b>	<b>37</b>
<b>11</b>	<b>LOGGING .....</b>	<b>38</b>

## 2 Introduction

The Swiss Federal Archives developed the application called SIARD (Software Independent Archiving of Relational Databases) within the framework of the ARELDA (Archivierung Elektronischer DATen) project. SIARD is used for long-term archiving of relational database content.

On behalf of the Swiss Federal Archives, Enter AG gathered practical experience (2005, 2006) in archiving productive databases and, in the last quarter of 2007, developed a second version of SIARD which integrated practical experience into the product.

This document is the manual for the *SIARD Suite*.

### 2.1 Intellectual Property Rights

The *SIARD Suite* is a development of the Swiss Federal Archives. All rights rest with the Swiss Federal Archives.

The *SIARD Suite* relies on the following component of other manufacturers:

- Zip64File (from <http://sourceforge.net/projects/zip64file/>  
License: GNU Public License Version 2 (GPLv2)  
in the SIARD distribution as *doc/gpl-2.0.txt*)
- JavaHelp 2.0.05 (from <http://java.sun.com/products/javahelp>,  
License: in the SIARD distribution as *doc/JavaHelp-LICENSE.HTML*)
- SUN JDBC-ODBC bridge (deprecated, will be not be available for JAVA 8 (JDK or JRE 1.8) or later  
License: Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX<sup>1</sup>  
in the SIARD distribution as *doc/java-license.txt*)
- Oracle JDBC driver *ojdbc6.jar*  
(from <http://www.oracle.com/technetwork/topics/winsoft-085727.html>  
License: Oracle Technology Network (OTN) Development and Distribution License Terms<sup>2</sup>  
in the SIARD distribution as *doc/oracle-license.txt*)
- MySQL JDBC driver (from <http://dev.mysql.com/downloads/connector/j/>  
License: GNU Public License 2.0<sup>3</sup>  
in the SIARD distribution as *doc/gpl-2.0.txt*)
- SQL Server JDBC driver (from  
<http://msdn.microsoft.com/en-us/sqlserver/jj589698.aspx>  
License: Redistribution License For Microsoft JDBC Driver  
in the SIARD distribution as *doc/mssql-license.txt*)
- DB/2 JDBC driver (from  
[http://pic.dhe.ibm.com/infocenter/tivihelp/v29r1/index.jsp?topic=%2Fcom.ibm.tad4z.doc%2Ftdb2jdbcdriverandlicence\\_install.html](http://pic.dhe.ibm.com/infocenter/tivihelp/v29r1/index.jsp?topic=%2Fcom.ibm.tad4z.doc%2Ftdb2jdbcdriverandlicence_install.html)  
License: DB2 Personal Developers Edition: Redistributable JDBC Type 4 Driver<sup>4</sup>

---

<sup>1</sup> <http://www.oracle.com/technetwork/java/javase/terms/license/index.html>

<sup>2</sup> <http://www.oracle.com/technetwork/licenses/distribution-license-152002.html>

<sup>3</sup> <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>

<sup>4</sup>

<https://raw.githubusercontent.com/weberdls/MIB/master/konakartfull/licenses/ibm.db2.jdbc.type4.driver.redistribution.license.txt>

## SIARD Suite

in the SIARD distribution as

*doc/ibm.db2.jdbc.type4.driver.redistribution.license.txt*)

- EnterAccess 1.0 (from <http://sourceforge.net/projects/enteraccess/>, License: GNU Library or Lesser General Public License 2.0 (LGPLv2)<sup>5</sup> in the SIARD distribution as *doc/lgpl-2.0.txt*)
- Jackcess 2.0.8 (from <http://jackcess.sourceforge.net/> License: GNU Lesser General Public License (LGPL) 2.1<sup>6</sup> in the SIARD distribution as *doc/lgpl-2.1.txt*)
- H2 1.3.176 (from <http://www.h2database.com/> Mozilla Public License Version 2.0 (MPL 2.0<sup>7</sup>) and Eclipse Public License (EPL 1.0<sup>8</sup>)
- Apache POI Project 3.9 (from <http://poi.apache.org/> License: Apache License, Version 2.0<sup>9</sup> in the SIARD distribution as *doc/apache-license-2.0.txt*)
- Apache Commons Lang 2.6 (von <http://commons.apache.org> License: Apache License, Version 2.0<sup>10</sup> in the SIARD distribution as *doc/apache-license-2.0.txt*)
- Apache Commons Logging API 1.1.3 (von <http://commons.apache.org> License: Apache License, Version 2.0<sup>11</sup> in the SIARD distribution as *doc/apache-license-2.0.txt*)

To ease the installation process, these components are delivered on the SIARD CD as JAR files. Users of the *SIARD Suite* are requested to honour the license requirements of these components, which can be found in the *doc* folder.

---

<sup>5</sup> <http://www.gnu.org/licenses/lgpl.html>

<sup>6</sup> <http://www.gnu.org/licenses/lgpl.html>

<sup>7</sup> <http://www.mozilla.org/MPL/2.0>

<sup>8</sup> <http://opensource.org/licenses/eclipse-1.0.php>

<sup>9</sup> <http://www.apache.org/licenses/LICENSE-2.0>

<sup>10</sup> <http://www.apache.org/licenses/LICENSE-2.0>

<sup>11</sup> <http://www.apache.org/licenses/LICENSE-2.0>

### 3 SIARD Concept

As the Swiss Federal Archives are obliged to archive Federal Administration documents independent of the information medium<sup>12</sup>, the problem of long-term archiving of relational databases must be resolved<sup>13</sup>.

Within the framework of the ARELDA project, it was concluded that today's database systems are too proprietary in their implementations, such that no credible long term archiving is possible based on these systems.

It was decided to store the databases conforming to the standard SQL:1999<sup>14</sup> to guarantee long term availability. Data content are 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.

The database archive format is henceforth called the SIARD format.

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 be 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.

The SIARD format stores the archived database schema definition in SQL:1999 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 und Character Large Objects), referenced in the XML files, are stored in separate (binary) files.

The *SIARD Suite* currently uses only SQL:1999 "Core Features". Future versions of the *SIARD Suite* may be extended to make use of further SQL:1999 components (Packages).

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 and can be downloaded as eCH-0165<sup>15</sup>.

---

<sup>12</sup> Federal Law relating to Archiving (Archivierungsgesetz, BGA) of 26 June 1998 (update: 22 July 2003)

<sup>13</sup> Archiving of Federal Administration's electronic digital data and files in the Swiss Federal Archives (ARELDA), Project-Exposé dated 1.3.2001

<sup>14</sup> Jim Melton, Alan R. Simon: *SQL:1999 – Understanding Relational Language Components*, Morgan Kaufmann Publishers, 2002, ISBN 1-55860-456-1 and

Jim Melton: *Advanced SQL:1999 – Understanding Object-Relational and Other Advanced Features*, Morgan Kaufmann Publishers, 2003, ISBN 1-55860-677-7

<sup>15</sup> <http://www.bar.admin.ch/dienstleistungen/00823/00825/>

## **4 Prerequisites for Using the *SIARD Suite***

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

### **4.1 JAVA**

Users of the *SIARD Suite* must install JAVA in advance. The minimum technical requirement is JRE 1.6.

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). If one is not interested in downloading MS Access databases, one can also use OpenJDK (<http://openjdk.java.net/>) instead.

Unfortunately, prior to version 1.5, JAVA provided no central function for discovering the installed version. In particular, multiple versions of JAVA may be installed. For instance, Oracle Client installs its own JAVA version (usually older and faulty) and adds it to the PATH environment variable(!).

To find out whether JAVA 1.6 or higher is available under Windows, proceed as follows: In the Windows "Start / Run" menu item, enter the command *regedit*. If JAVA 1.6 or higher is installed, the key *CurrentVersion* in *HKEY\_LOCAL\_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment* contains the value "1.6" (or higher). If the registry item does not exist or *CurrentVersion* contains a value lower than "1.6", a new version of JAVA must be installed.

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

With *JAVA 1.6* or later it is possible, to install a 64-bit version of JAVA on a 64-bit operating system platform (e.g. Windows 7, Ubuntu, MacIntosh). This is generally recommended but not mandatory. The JAVA code of *SIARD Suite* – or any JAVA program – is the same, whether it runs on the 32-bit or the 64-bit version of the JAVA runtime. The only dependency on the architecture of the operating system platform is the JDBC-ODBC bridge used for downloading MS Access databases<sup>16</sup>. If 64-bit JAVA is used, then 64-bit MS Access (i.e. the corresponding ACE ODBC driver) must be installed. (N.B.: The MS Office setup installs the 32-bit version of MS Access by default even on a 64-bit operating system platform!) It is impossible to access 32-bit ODBC data sources from 64-bit JAVA. It is equally impossible to access 64-bit ODBC data sources from 32-bit JAVA. Unfortunately the 64-bit ODBC driver from Microsoft is still very buggy. This prevents successful conversion of some MS Access files to the SIARD format.

If both the 32-bit as well as the 64-bit version of JAVA are installed on a system, *SIARD Suite* will start using the 64-bit version by default. However, it is possible to start the SIARD scripts manually with an option to use the 32-bit version of JAVA.

In one of the next SIARD versions the ODBC access will be completely removed, as it is replaced by the „pure JAVA“ JDBC driver EnterAccess. If the ODBC driver is not used, no problems with the architecture of the operating system is experienced.

### **4.2 Databases**

At the time of first delivery, *SIARD Suite* supports these database systems:

---

<sup>16</sup> <http://www.enterag.ch/enterag/downloads/platform.xhtml>

## SIARD Suite

- MS Access
- Oracle
- SQL Server
- MySQL (or MariaDB)
- DB/2

Further database systems may be integrated at a later date.

The *SIARD Suite* can access the following database systems using these JDBC drivers:

EnterAccess-1.0.jar	For accessing MS Access databases (MDB, ACCDB 2000 - 2012) using the open source software Jackcess, which is independent of the current operating system
ojdbc6-fixed.jar	Only for accessing Oracle databases <i>Oracle 8i</i> to <i>Oracle 12c</i> (errors in the Oracle-Distribution, which prevent the creation of correct XSD files, were fixed, by keeping only oracle.-Packages)
sqljdbc4.jar	Only for accessing SQL Server databases <i>SQL Server 2000</i> to <i>SQL Server 2012</i>
mysql-connector-java-5.1.34-bin.jar	Only for accessing MySQL databases 5.5 or higher or MariaDB databases.
db2jcc.jar und db2jcc_license_cu.jar	Only for accessing DB/2 databases version 8 or higher
SUN JDBC-ODBC bridge (in JAVA Runtime)	for ODBC access of MS Access databases – only available under Windows and only with JAVA versions less than 8 (JDK/JRE 1.8). For a short transition period it is still available side by side with EnterAccess.

These drivers are delivered and installed along with the *SIARD Suite*. It is probable that they will be compatible with future versions of these database systems for some years to come.



### 5 Installation

As long as a correctly installed JAVA (1.6 or higher) is available, the *SIARD Suite* can be executed at any time directly from a USB Stick, CD-ROM or network drive. Consequently, it is executable under Windows, Linux, MacIntosh, etc.

User specific settings are stored under `<home>/java/siard suite.properties`, where `<home>` is the platform specific user "Home Folder" (usually `C:\Documents and Settings\<userid>` under Windows XP; `C:\Users\<userid>` under Windows 7).

The *SIARD Suite* is delivered without a specific installation program. Instead, the *SiardEdit* main program detects whether, where and in which version, the *SIARD Suite* was installed by the user.

Thus, one simply starts *SiardEdit* in order to install the *SIARD Suite*, either with the platform specific script to be found in the top level folder of the delivery medium (*siardedit.cmd* or *siardedit.sh*) or as a JAVA application by executing `java -jar bin/SiardEdit.jar`.

If *SiardEdit* discovers that no user specific settings exist, it offers to install the *SIARD Suite* on the user's machine. `<home>/applications/siard suite` is the preset standard folder for the installation.

Using the items under the *Tools* menu in *SiardEdit*, the user can either deinstall the software or reinstall it at a different location at any time without administrator rights.

Even if an old version of the *SIARD Suite* is installed on the user's machine, a new version can be started alongside it from a USB Stick, CD-ROM or network drive without changing the old version.

If the *SIARD Suite* is installed under Windows, *SiardEdit* automatically inserts links on the Desktop and under *Start / Programs* using the *desktop.cmd* script.



# 6 SIARD Components

The *SIARD Suite* is made up of the following components:

- SiardEdit
- SiardFromDb
- SiardToDb

*SiardEdit* features an interactive graphical user interface. It facilitates processing the metadata in a SIARD archive.

*SiardEdit* is the central instrument with which SIARD formatted data are processed. Primary data cannot be changed. *SiardEdit* is not suitable for complex research. For such, it is recommended that the SIARD archive be loaded into a database system and database techniques be used.

*SiardFromDb* is a command line application for extracting and storing a database in a SIARD file. This application's functionality is identical with the function available in *SiardEdit* under the *New from database ...* menu item. Especially when downloading large databases, using the command line version is more comfortable. Further, long lasting downloads can be better documented using *stdout* and *stderr* redirection (see "Using command redirection operators" in Windows *Help* under *Start or online*<sup>17</sup>).

*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 *SiardEdit* under the *Load into database ...* menu item. 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" in Windows *Help* under *Start or online*<sup>18</sup>).

---

<sup>17</sup> [http://technet.microsoft.com/de-de/library/cc772622\(WS.10\).aspx](http://technet.microsoft.com/de-de/library/cc772622(WS.10).aspx)

<sup>18</sup> [http://technet.microsoft.com/de-de/library/cc772622\(WS.10\).aspx](http://technet.microsoft.com/de-de/library/cc772622(WS.10).aspx)

### 7 SiardEdit

The *SiardEdit* program features an interactive graphical user interface. It facilitates processing the metadata in a SIARD archive.

Using *SiardEdit*, one can:

- create an "empty" SIARD archive (without primary data), to get a first overview of the archiving process,
- synchronise (take over all element descriptions which exist in the new and the template archive) the metadata of a candidate SIARD archive with that of a template archive (whether empty or not),
- 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.

*SiardEdit* is the central instrument with which SIARD formatted data are processed. Primary data cannot be changed. *SiardEdit* is not suitable for complex research. For such, 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 file. If you would prefer to interpret the times in the database unchanged as UTC times, you must start *SiardEdit* with the option

```
-Duser.timezone=GMT
```

It is possible to call *SiardEdit* with the name of a SIARD file to be opened as single argument.

#### 7.1 Initial Execution

The *SIARD Suite* is delivered as a ZIP file to be downloaded and must first be unpacked. The file *SiardEdit.jar* is situated in the *bin* 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 *siardedit.cmd* (Windows) or *siardedit.sh* (LINUX).

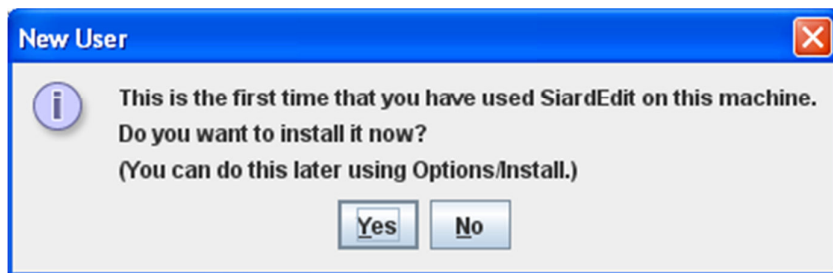
If this doesn't work or one is using a different operating system, *SiardEdit* can also be started from the command line in the *SIARD Suite*'s *bin* folder as follows:

```
javaw -jar SiardEdit.jar (Windows)
```

```
java -jar SiardEdit.jar (all platforms)
```

For this to succeed, Java's *bin* folder must have been added to the *PATH* variable. Otherwise, one must write out the full path name of the executable *java* program.

Upon initial execution of *SiardEdit*, the following message appears.



## SIARD Suite

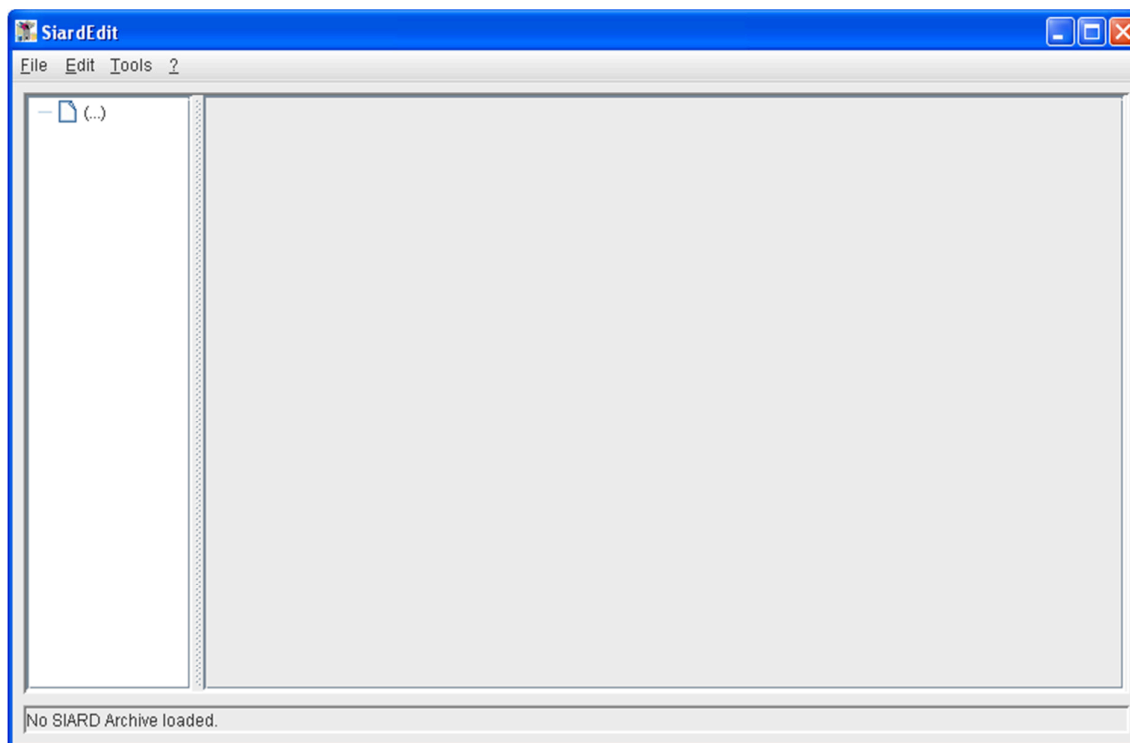
(As *SiardEdit* doesn't know the user's language at this point, the language of this message depends on the operating system language and the language chosen when JAVA was installed.)

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, *SiardEdit* can, in future, be started from the chosen folder.

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

### 7.2 Main Window

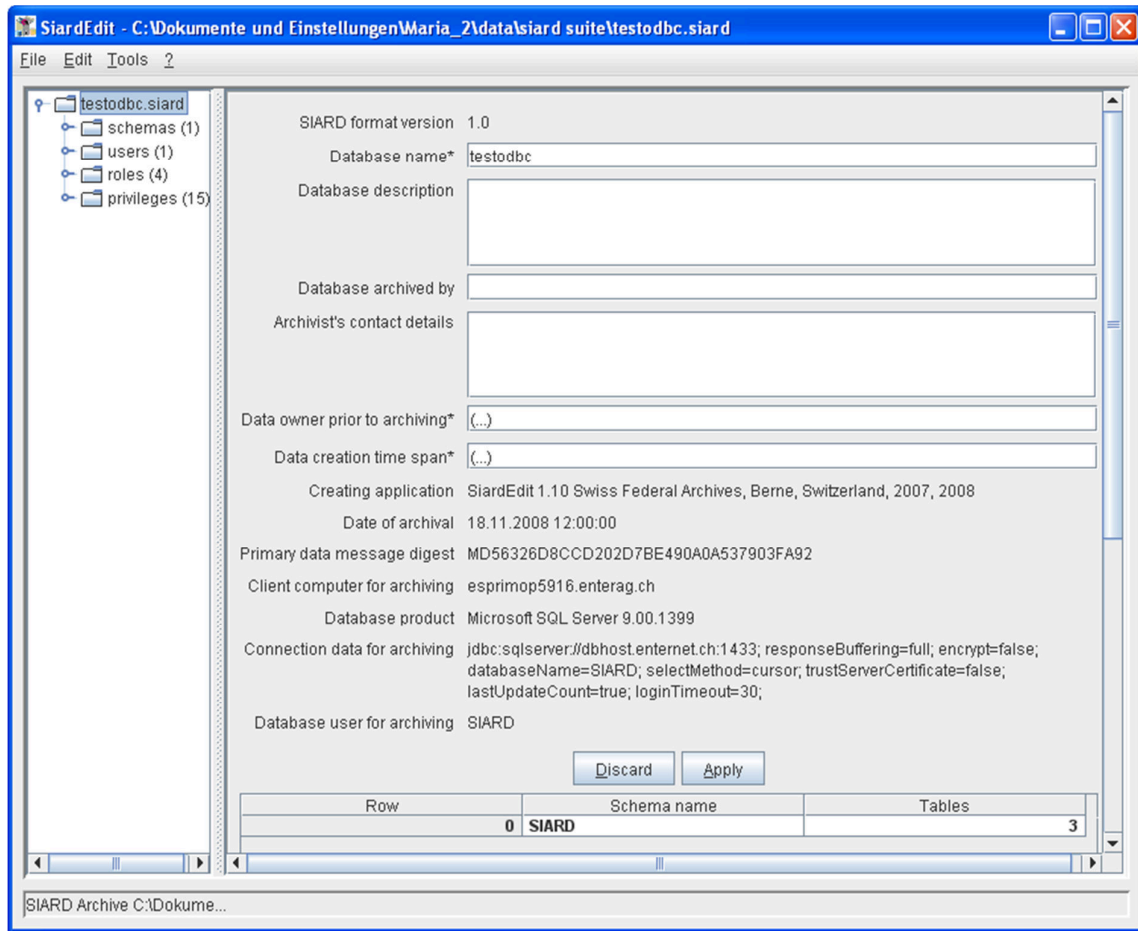
The main window consists of a menu (top), navigation tree (left), content (right) and a status line (bottom).



The border between navigation and content can be freely adjusted. The size of the whole window can also be freely adjusted (but not below a defined minimum).

When a SIARD file is loaded into *SiardEdit*, the main window appears as follows:

## SIARD Suite



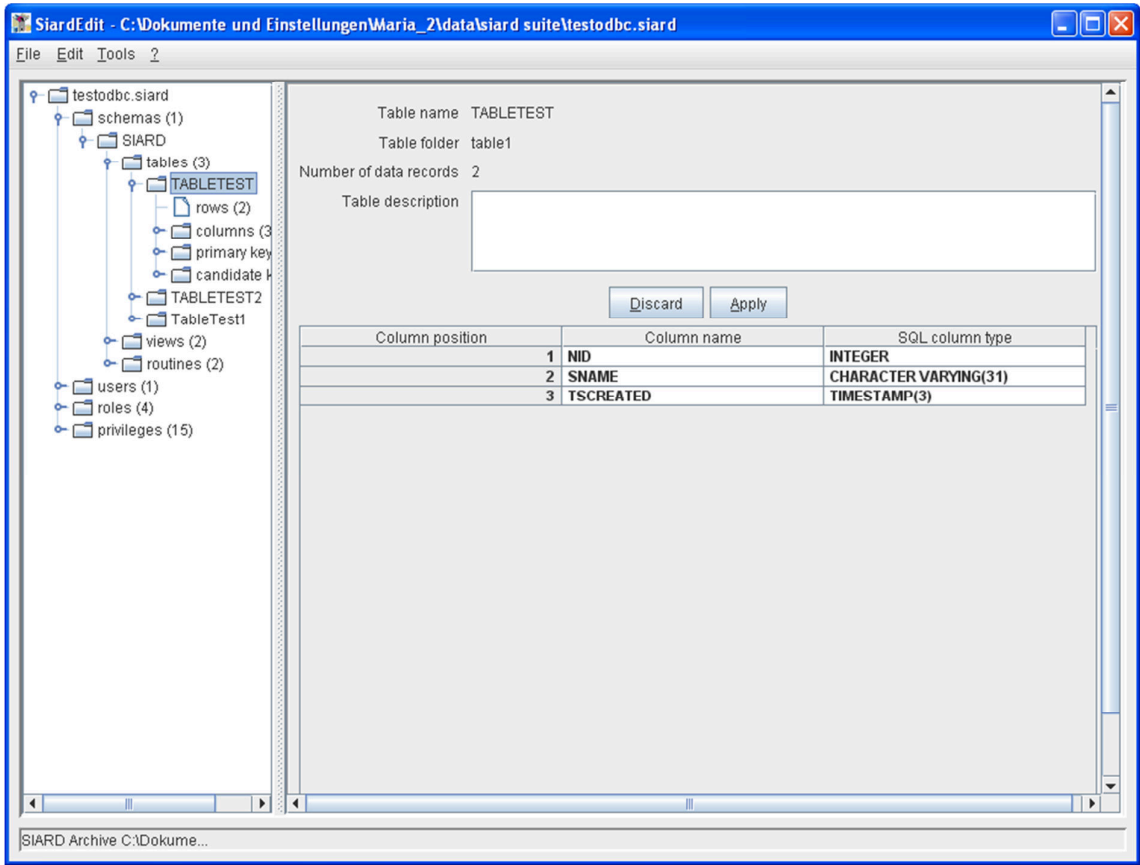
The left pane is used to navigate in the metadata tree. In the upper region of the right pane one can enter or change alterable metadata which belong to the database object selected in the left pane.

The *Apply* button makes 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.

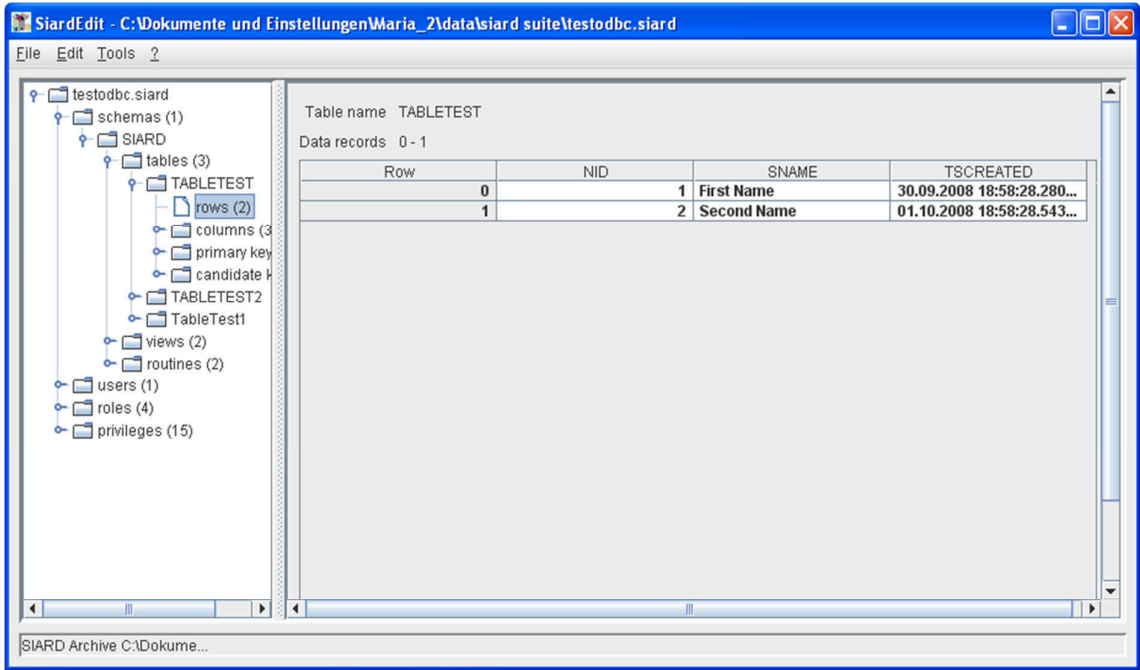
A table of the most important sub-objects is shown under the metadata. Clicking a column title sorts the table on this column. Double-clicking on a row selects the relevant sub-object.

As tables in schemata and columns in tables have no natural ordering in the metadata and *SiardEdit* normally displays in alphabetical order, this sort function is useful when finding one's way about in large database schemata.

SIARD Suite



If one is at the table level in the tree, one can navigate to the primary data of a table by choosing the *rows* object under the table.



## SIARD Suite

While displaying primary data, *SiardEdit* alters its behaviour slightly. If one now clicks on a column title, the whole table will be sorted by this column. With large tables, this may take a considerable amount of time.

To properly deal with such very large tables, *SiardEdit* chooses to display only every *n*th record (*n* a power of 50) to avoid having a scroll window containing huge numbers of records. If one double-clicks on one of the line numbers shown, more records between this line and the next displayed are shown. One can repeat this down to the lowest level, where all records are consecutively displayed.

If one double-clicks on a text, numeric or date field, the content is displayed in a text editor. If one double-clicks on a binary field, the content is displayed in a binary editor. One can configure which editors are used under *Tools / Options*. One can also move the column boundaries to properly view the entries in numeric or date columns.

If one selects a line in the table by clicking on it, it changes colour. If one chooses the menu item *Edit / Copy* (*CTRL-C*), one can copy the line into the clipboard. If one chooses the menu item *Edit / Copy all* (*CTRL-A*), one can copy the whole displayed table into the clipboard. One can, for example, subsequently copy the data into a spreadsheet program for further processing.

### 7.3 Menu

The individual menu items are described in the following subsections.

#### 7.3.1 File

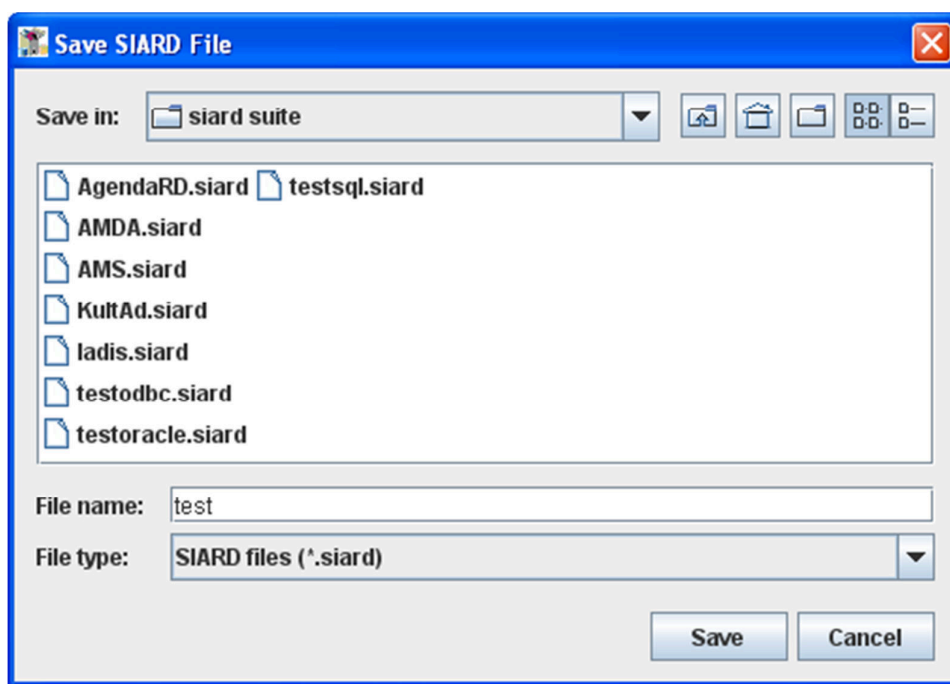
*New from Database ...*

This command downloads a database, stores it as a SIARD file and then displays the file in *SiardEdit*.

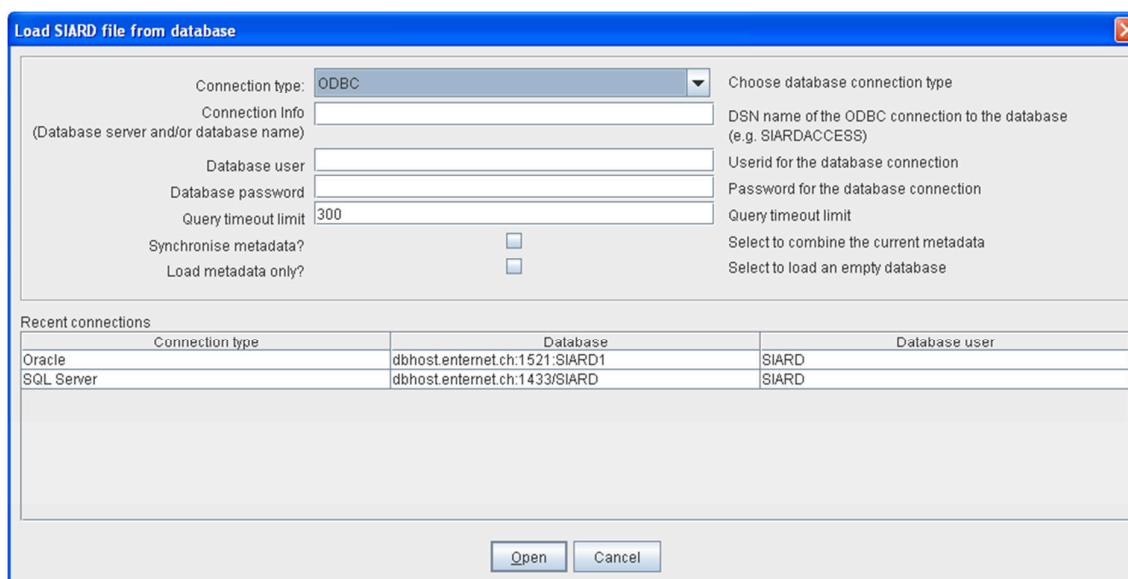
This action is functionally identical to the command line program *SiardFromDb*. The program description contains important advice about how to prepare various database systems for this operation.

The first step is to give the name under which the SIARD file will be stored:

## SIARD Suite



One is requested to enter the connection data for the database to be downloaded in the dialog "Load SIARD file from database":



### Connection type

Choose the connection type from

- *Access* for “pure JAVA” access to MS Access databases,
- *ODBC* for MS Access databases (deprecated),
- *Oracle* for Oracle databases,



## SIARD Suite

- *MySQL for MySQL databases*, or
- *SQL Server* for SQL Server databases.
- *DB/2 for DB/2 databases*

### Connection data

The connection data vary depending on the Connection type.

#### Access

In this case, the file name of the MS Access database is to be entered in the connection data.

#### ODBC (deprecated)

To use the ODBC connection type, one must have previously set up an ODBC connection for the chosen database. The relevant procedure is documented in the operating system's help documentation.

In this case, the ODBC connection's DSN (Data Source Name) is to be entered in the connection data.

#### Oracle

Enter the character string made up of the database server's DNS name (DNS = Domain Name System), the listener port and the database's SID (System ID), separated by colons. An example of Oracle connection data is: *athlon700b.enterag.ch:1521:SIARD*.

*athlon700b.enterag.ch* is the server's DNS-Name, *1521* is the database's listener port (this is the standard value for Oracle databases, if not changed by the database administrator) and *SIARD* is the database's SID.

#### SQL Server

Enter the character string made up of the database server's DNS name (DNS = Domain Name System), the listener port and the database name. Server and port are separated by a colon. The database name is separated by a slash. Omitting the database name results in a connection to the database defined as the login default in the SQL server for this user.

An example of SQL server connection data is: *athlon800.enterag.ch:1433/siard*.

*athlon800.enterag.ch* is the server's DNS-Name, *1433* is the database's listener port (this is the standard value for SQL Server databases, if not changed by the database administrator) and *siard* is the database name.

Omitting the database name results in a connection to the database defined as the login default in the SQL server for the calling database user.

#### MySQL

Enter the character string made up of the database server's DNS name (DNS = Domain Name System), the listener port and the database name (= schema name in MySQL). Server and port are separated by a colon. The database name (i.e. schema name) is separated by slash.

An example of MySQL server connection data is: *asusp5b.enterag.ch:3306/SIARD*.

*asusp5b.enterag.ch* is the server's DNS name, *3306* is the database's listener port (this is the standard value for MySQL databases, if not changed by the database administrator) and *SIARD* is the database (i.e. schema) name.

## SIARD Suite

DB/2

Enter the character string made up of the database server's DNS name (DNS = Domain Name System), the listener port and the database name. Server and port are separated by a colon. The database name is separated by slash.

An example of DB/2 server connection data is: *vmu64.enterag.ch:50000/SIARD*.

*vmu64.enterag.ch* is the server's DNS name, 50000 is the database's listener port (this is the standard value for DB/2 databases, if not changed by the database administrator) and *SIARD* is the database name.

### Database user

Enter the database user whose database view should be archived. Warning: some database systems ignore character case, some don't!

If only parts of the database are to be archived, set up an archive user whose read rights in the database cover only those objects to be archived.

### Database password

Database user and password must be created by the database administrator. If you don't know them, ask the database administrator. Warning: some database systems ignore character case, some don't!

Normally, in MS Access databases, the user is *Admin* and the password is not set (empty string).

### Load only metadata?

If this checkbox is chosen, a SIARD file containing no primary data is produced. This is useful in getting an overview of the expected database size. Such an "empty" SIARD file can be used to add explanations to the metadata and to merge them into the full SIARD file by using *Synchronise metadata* on a later download of the full database.

### Synchronise metadata?

This checkbox is only displayed when the main window already contains a loaded SIARD file. Selecting this results in the metadata of the previously loaded file being, as much as possible, integrated into the metadata of the newly downloaded SIARD file. (I.e. those entries for tables, columns, etc., which are the same in both database schemata are taken over.)

### Recent Connections

This table lists the user's latest successful connections. They are stored in the file *siard.properties* in the folder *java* in the user's home directory.

Clicking on a row in this table copies the corresponding fields into the connection data dialog fields. The cursor is placed in the password field, which remains empty.

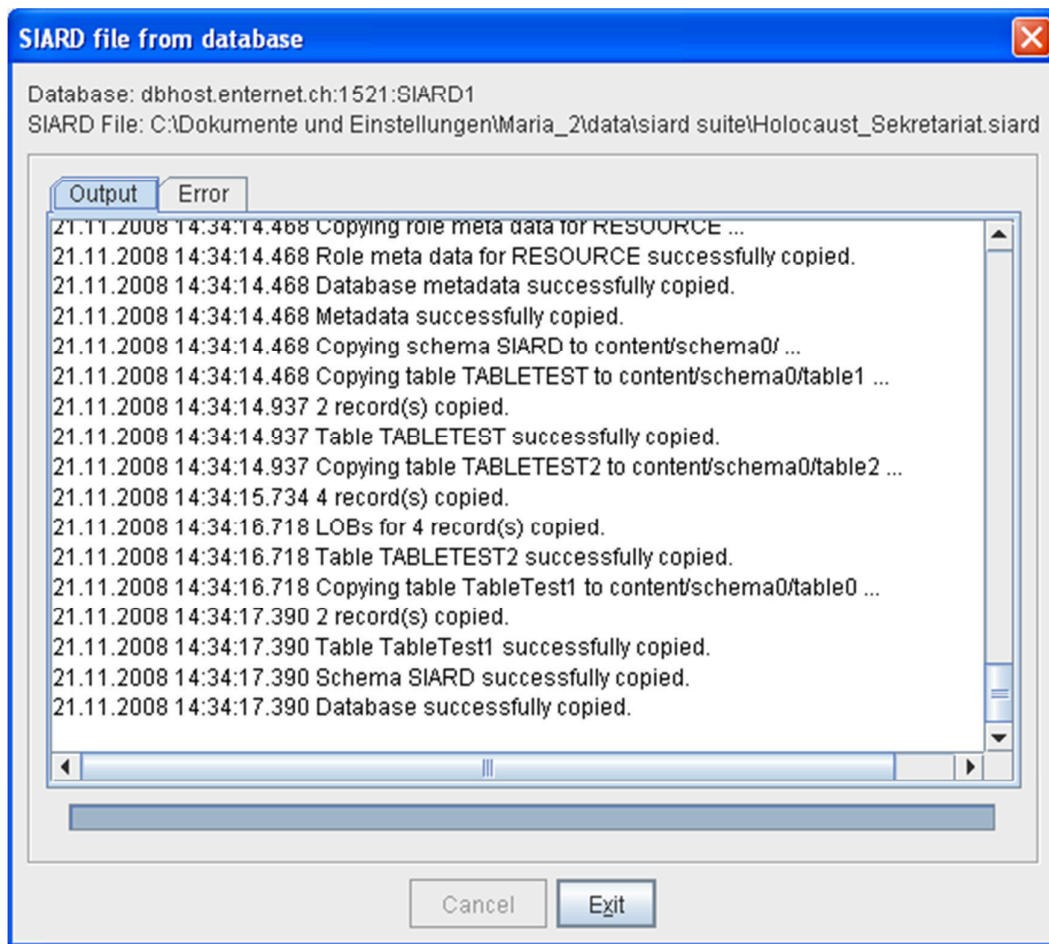
### Open

This button loads the database into the chosen SIARD file. A log, which is identical to that produced by *SiardFromDb*, is displayed.

### Cancel

Use this button to cancel the current action.

## SIARD Suite



At the end of the load operation, one can mark/copy/paste the log information into a text editor for printing. The time information shown at each step can be used to assess the total load time of larger databases.

The newly loaded file is opened in the main window after one clicks on the *Exit* button.

*Open ...*

This menu item can be used to open (previously downloaded) SIARD files.

The SIARD archive's metadata appear in the main window and can be saved with any changes.

*Save ... (Ctrl-S)*

This menu item saves the changed metadata in the opened SIARD file.

*Load into Database ...*

Use this menu item to load the contents of an open SIARD file into a database.

This action is functionally identical to the command line program *SiardToDb*. The *SiardToDb* documentation provides important information about preparing the various database systems for this operation. Especially, a database user with the necessary rights must be created. For uploading into MS Access, an empty database must be created. If the ODBC driver is to be used, it must also be made accessible via ODBC.

## SIARD Suite

On choosing this menu item, in the *Load SIARD file to database* dialog, one is requested to provide the database connection data for the database into which the data is to be loaded.

Connection type	Database	Database user
SQL Server	dbhost.enternet.ch:1433/SIARD	SIARD
Oracle	dbhost.enternet.ch:1521/SIARD1	SIARD

After the connection to the upload database has been established a dialog is displayed, where each schema name in the SIARD archive can be associated with a schema name in the upload database.

Uploading database content is subject to the limitations of the database system to which the data is to be loaded. For example, MS Access does not support database schemata. Thus, all tables are loaded into this database system without Schemata separation.

The advantage in uploading the database content is that one has the complete expressiveness of SQL available to formulate queries.

### *Export Metadata ...*

This command exports the SIARD metadata to an XML file.

Theoretically, one could simply open the SIARD file with *PKZIP* and export the metadata from the *header* folder. However, not all users have a *PKZIP* licence. A licence is necessary because other ZIP programs, such as Info-ZIP, Microsoft Windows XP ZIP folder etc., do not (yet?) support PKZIP's 64 bit standard. Since SIARD files can be very large, it was necessary to use the 64 bit standard to be able to store files larger than 4GB and containing more than 66'000 files. Instead of PKZIP one can also make use of the open source application *Zip64File*<sup>19</sup> for unpacking ZIP64 files.

With the help of simple XSLT control files, the exported *metadata.xml* file can be transformed into a SIARD file database metadata report. If one copies the *metadata.xsl* file from the SIARD installation folder *doc* into the same folder where the *metadata.xml* was copied and then opens it with one of the more modern browsers, the metadata will be displayed in such a formatted report.

The SIARD file format and metadata structures are described in *SIARD Format\_en.pdf*, their formal definition in the XML Schema definition *metadata.xsd* in folder *doc* in the SIARD installation location.

---

<sup>19</sup> <http://sourceforge.net/projects/zip64file/>

## SIARD Suite

### *Most Recently Used Files*

This menu gives fast access to previously opened files.

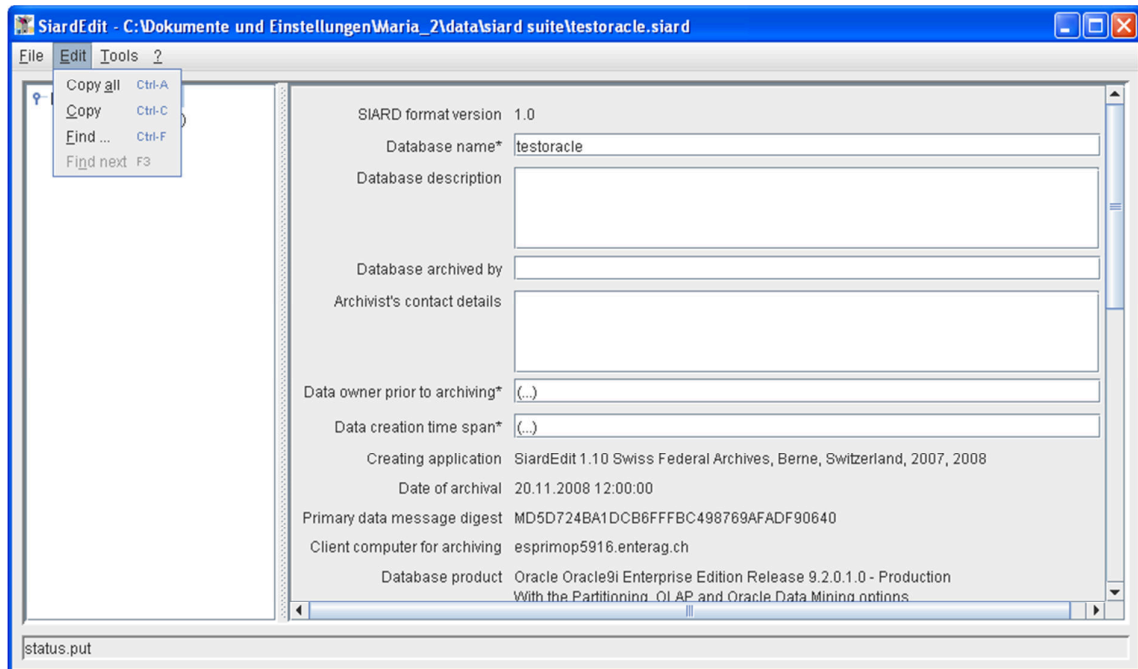
### *Exit*

There is an *Exit* item at the bottom of the *File* menu – it closes the application.

### 7.3.2 *Edit*

#### *Copy all (Ctrl-A)*

The *Copy all (Ctrl-A)* menu item is situated in the *Edit* menu.



With its help one can save the displayed table (lower right pane) as tab separated text in the clipboard. It is useful to paste this into a spreadsheet program for further processing.

#### *Copy (Ctrl-C)*

This command can be used to copy the highlighted part of the table into the clipboard for later pasting into other applications.

#### *Find ... (Ctrl-F)*

Use this command to search for an arbitrary string in the metadata.

#### *Find next (F3)*

Use this command to successively find the search string.

### 7.3.3 *Tools*

#### *Install*

The *Tools* menu contains the item *Install*, which installs SIARD onto the user's PC.

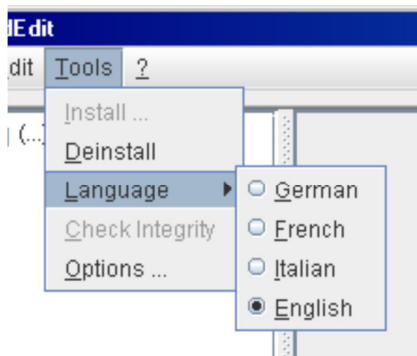
#### *Deinstall*

Use this *Tools* menu item to remove the locally installed copy of SIARD.

## SIARD Suite

### Language

The user can choose the desired language in the menu item *Tools / Language*.



The *SiardEdit* main window adjusts immediately to the chosen language. At program termination, the user language is stored in the file `<home>/java/siard suite.properties`. `<home>` represents `$HOME` under Linux and `C:\Documents and Settings\<userid>` under Windows XP and `C:\Users\<userid>` under Windows 7 (`<userid>` is the name under which the user logged in under Windows.)

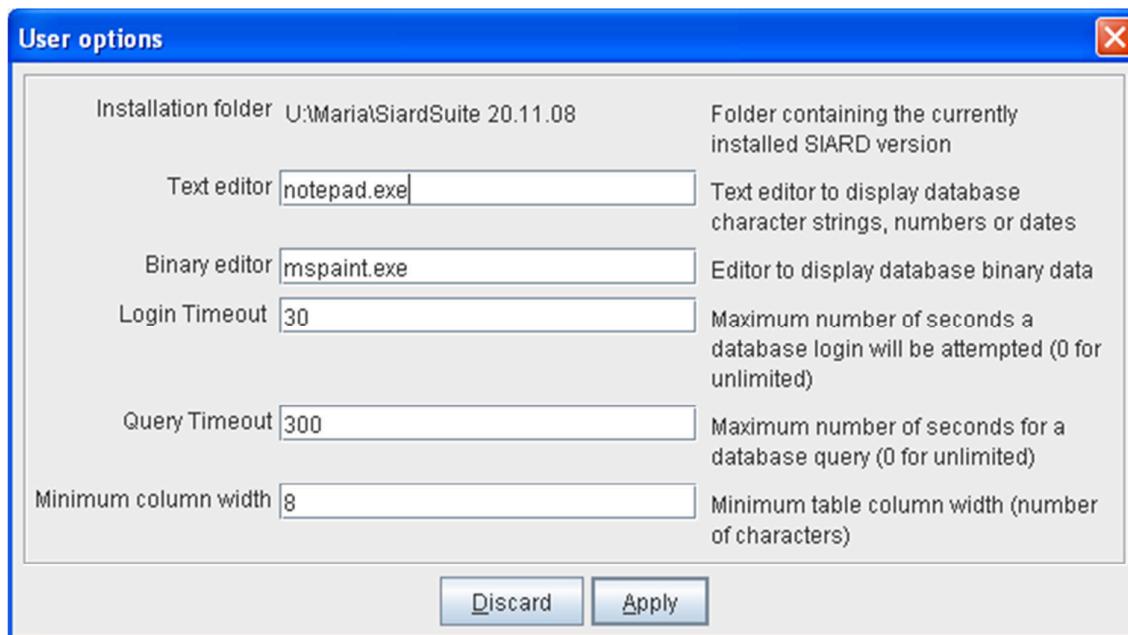
Expert users can force a different directory by adding the option “-Duser.dir=<other directory>” to the command line.

### Check Integrity

This menu is used to verify, that the primary data have not been changed since their “message digest” (MD5 or SHA-1) has been computed and saved in the meta data. *SIARD Suite* seals the primary data with a message digest only once after loading them from the database.

### Options

The *Options* menu is under the *Tools* menu. Some user specific settings can be entered:



## SIARD Suite

The *Texteditor* and *Binaryeditor* entries are used when one double-clicks on a primary data cell of type character string or binary data. The cell content is stored into a temporary file and the defined application is invoked. The *Binaryeditor*, in particular, should be set appropriately for the BLOB binary data being examined.

The *Maximum login time* controls after how many seconds a database login attempt will be abandoned. It should only be changed if one has reason to assume that the network or database react slowly.

The *Maximum query duration* should be set to nought for downloading very large databases as, in such cases, the query will last longer than the default setting of 5 minutes.

The *Minimum column width* controls the depiction of the tables in the lower right window area.

### 7.3.4 ?

#### Help

Invoking this menu item displays the contents of the help system.

#### Info

The ? menu also contains *SiardEdit*'s copyright notice:



In addition the architecture and version of the operating system and the JAVA runtime are displayed on this dialog.

### *7.3.5 Context-sensitive Help*

Basically, context-sensitive help is available to the user everywhere by pressing F1. The help shown is related to the element or window which, at that moment, receives keyboard or mouse input.

This explanation is provided when the user chooses the *Help* item under the ? menu.

One can call up individual pages from the contents list in the help window or one can search for keywords. The contents list is synchronised with the currently displayed page.



## 8 SiardFromDb

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

- a "full" SIARD archive (metadata *and* primary data) based on the database (option "full"),
- an "empty" SIARD archive (metadata *only*) containing a Schema definition based on the database (option "empty").

### 8.1 Invocation

```
java -cp <siardpath>/bin/SiardEdit.jar ch.admin.bar.siard.SiardFromDb [-h] | [-e]
    [-c <config file>] -d <data source> -n <database name> -u <database user>
    -p <database password> -s <siard file> [-q <query timeout>] [-t <template file>]
```

Specify *<siardpath>* as the folder where SIARD is installed. The file *SiardEdit.jar* is in the *bin* subfolder with its class *ch.admin.bar.siard.SiardFromDb*, 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 given on the command line.

As the call can have quite a lot of parameters, one can simplify it using a configuration file which contains all the parameters except the password. If one is available, the call reduces to:

```
java -cp <siardpath>/SiardEdit.jar ch.admin.bar.siard.SiardFromDb -c <config file>
    -p <database password>
```

If all parameters and, additionally, a configuration file are given, the parameters on the command line supplement those given in the configuration file, which simplifies repetitive SIARD downloads.

With the *-e (empty)* option, only the metadata is archived.

### 8.2 Invocation parameters

Parameter	Meaning
<data source>	Access, ODBC, Oracle, "SQL Server", MySQL, or "DB/2"
<database name>	Database name, structured depending on the connection: Access: file name of MS Access database ODBC (deprecated): ODBC connection's configured Datasourcename (DSN) Oracle: e.g. dbhost.enternet.ch:1521:SIARD1 SQL Server: e.g. dbhost.enternet.ch:1433/SIARD MySQL: e.g. dbhost.enternet.ch:3306/SIARD DB/2: e.g. dbhost.enternet.ch:50000/SIARD
<database user>	UserId for the download session

## SIARD Suite

<i>Parameter</i>	<i>Meaning</i>
<database password>	Password for the download session
<siard file>	File name for the SIARD archive being written
<query timeout>	Timeout in seconds for SQL queries (o for unlimited)
<template file>	SIARD template file to be used for filling manually entered fields.

### 8.3 Notes

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

The archiving process either wholly succeeds or wholly fails.

For large tables, it is recommended to download an empty SIARD file 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 o* 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

```
-Duser.timezone=GMT
```

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

### 8.4 Database Preparation

Various preparations are necessary, depending on the database system used. It is important to realize that the choice of the parameter database user determines the extent of the “database” which will be downloaded. The resulting SIARD file will contain all database objects visible to the database user. Therefore it is usually not a good idea, to use the database administrator (SYS or SYSTEM for Oracle, *dbo* for SQL Server, *mysql* or *information\_schema* for MySQL, db2inst1 or SYSIBM for DB/2) as the database user for archiving. A large number of system objects are visible to these users which normally should not be archived. For MS Access databases, on the other hand, it is usually desired, that all content be archived. Therefore using the *Admin* user in this case is recommended.

The fact that the database user's privileges determines the extent of the database archived can be used to precisely control what will end up in the SIARD file: One just needs to create a new archiving user and grant it reading rights for all objects that are to be archived.

#### 8.4.1 Preparation of “pure JAVA” Access to MS Access Databases

No preparatory steps are necessary. The MS Access file must be readable in the file system.

Most MS Access databases can be opened with the UserId *Admin* and the password "" (empty string). One must consult the developer's documentation to gain access to better protected databases.

#### 8.4.2 Preparation of ODBC Access to MS Access Databases (deprecated)

##### *Accdb and Mdb Files*

MS Access 2007 supports two file formats: The “old” format of files with extension *.mdb* and the new one with extension *.accdb*. The *accdb* files support new data types – particularly

## SIARD Suite

multi-valued fields and attachment fields (also multi-valued). They also prohibit access to the system tables and thus make less metadata available through the ODBC (and therefore the JDBC) interface. The *mdb* files permitted access to the system tables *MSysObjects*, *MSysRelationships* and *MSysQueries*. These tables made it possible for *SIARD Suite* to access additional metadata information not available through the *DatabaseMetaData* class of the JDBC driver.

*SIARD Suite* handles this new situation as follows: If the *MSys* tables are accessible, their additional information is saved on download. If they are not accessible, neither foreign key constraints nor views can be stored. For the purposes of archiving this is probably a minor problem, as no primary data are lost.

More problematic is the new attachment data type of *accdb* files. Microsoft has decided to make these data completely inaccessible through the ODBC (and therefore the JDBC) interface. The current version of the SIARD format would also be unable to accommodate a multi-valued BLOB field. Thus *SIARD Suite* – like Microsoft’s own migration tools – archives such fields as text fields containing a list of file names referencing the attachments. But the content of the attachments is lost! Unfortunately Microsoft prevents *SIARD Suite* – and all other ODBC or JDBC clients – detecting this loss. Thus it rests with the archiver to take measures to archive attachment content separately.

If *mdb* files are to be archived, the archiver has the choice, whether to archive only the metadata that would be archived if it were an *accdb* file – without foreign keys and views – or to prepare the *mdb* by making the system tables accessible to the *Admin* user.

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

If *SIARD Suite* was started with a 32-bit JAVA Runtime, only 32-bit ODBC drivers can be accessed. If 64-bit MS Access was installed on 64-bit Windows, the default ODBC administrator (C:\Windows\system32\odbcad32.exe) only permits editing 64-bit ODBC connections. These can be used by *SIARD Suite* only, if it was started with a 64-bit JAVA Runtime. In order to create and configure a 32-bit ODBC connection, one must start the 32-bit ODBC administrator (C:\Windows\SysWOW64\odbcad32.exe). These 32-bit ODBC connections are visible but not editable in the 64-bit ODBC administrator. *SIARD Suite* can only access them, if it was started with a 32-bit JAVA Runtime.

The batch commands *siardedit.cmd*, *siardfromdb.cmd* and *siardtodb.cmd* (as well as the shortcut on the desktop created by the installation) use the 64-bit JAVA Runtime, if one is available. If both, a 32-bit and a 64-bit JAVA Runtime is installed, *SIARD Suite* can be forced manually to use the 32-bit one, by calling the batch scripts with the optional argument /32 on the command line. Unfortunately the 64-bit ODBC driver from Microsoft is still very buggy. So it is safer, to convert MS Access files using the 32-bit ODBC Jet Engine.

### *Making system tables of mdb files accessible to the Admin user*

1. Make the *MSysObjects*, *MSysRelationships* and *MSysQueries* system tables accessible to the archiving user (mostly *Admin*).

Ms Access before 2007:

Use *Tools / Options / View* to make system objects visible and, after that, use *Tools / Security / User and Group Permissions* to make the tables *MSysObjects*, *MSysRelationships* and *MSysQueries* readable for the user *Admin*.

MS Access 2007:

*Office Button / Access Options / Current Database / Navigation Options... / Show System Objects* to make system objects visible and, after that, *Database Tools / Administer / Users and Permissions / User and Group Permissions* to make the tables

## SIARD Suite

*MSysObjects*, *MSysRelationships* and *MSysQueries* readable for the user *Admin*. (The whole group *Database Tools / Administer / Users and Permissions* is missing if the open database file is an *accdb* file.)

MS Access 2010 or later:

Same as MS Access 2007 but use *File* ribbon instead of *Office Button*.

2. Set up an ODBC connection which points at the MDB file (see results for keyword ODBC in Windows Help under *Start / Help* for instructions on how to set up an ODBC connection via *Start / Settings / Administration Tools / Data Sources (ODBC)*) and use its extended properties under *Configure / Advanced* to set *ExtendedAnsiSQL=1*. The ODBC connection can be set up as a *User DSN* by non-administrators.

### *Automatic configuration and conversion*

The open-source *accesstosiard.js* command script<sup>20</sup>, which has been developed as an addition to the *SIARD Suite*, relieves the user of all this configuration work and simply converts an MDB file to a SIARD file.

### 8.4.3 Oracle Preparation

The database administrator configures an Oracle database for export most easily by creating a new archiving user and granting this user read rights on all database objects to be exported. Using this login name, one can use sample queries to check whether all relevant references are visible. It makes sense to comment and document the sample queries for future archive users.

#### *Troubleshooting Oracle*

For *SIARD Suite* to be able to successfully connect to the desired Oracle database, the permission, firewalls etc must be configured such, that the following Oracle command can connect to the database successfully:

```
sqlplus      <database>      user>/<database>      password>@(description=(address=(protocol=TCP)
(host=<hostname>) (port=<port>)) (connect_data=(service_name=<service>)))
```

Then the database name *<hostname>:<port>:<service>* is to be used in *SIARD Suite*.

### 8.4.4 SQL Server Preparation

The database administrator most easily configures an SQL Server database for export by creating a new archiving user and granting this user read rights on the database to be exported.

If this archiving user has no owner privileges for the schemas to be archived, it cannot access the existing privileges and the resulting information about users and roles in the SIARD file may be incomplete.

If the archiving user has SELECT privileges on the whole database, the schema „dbo“ is automatically archived, which is usually not desirable. Thus the best settings for the archiving user are:

Under database permissions only grant the CONNECT permission. Under the user's properties, define the archiving user as owner of all schemas to be archived (N.B.: This removes previous ownerships. This setting may have to be restored to its previous value after the download of the database.)

---

<sup>20</sup> <http://sourceforge.net/projects/accesstosiard/>

## SIARD Suite

### Troubleshooting SQL Server

For *SIARD Suite* to be able to successfully connect to the desired SQL Server database, the permission, firewalls etc must be configured such, that the following SQL Server command can connect to the database successfully:

```
sqlcmd -U <database user> -P <database password> -S tcp:<hostname>,<port> -d <database>
```

Then the database name *<hostname>:<port>/<database>* is to be used in *SIARD Suite*.

### 8.4.5 MySQL Preparation

It is recommended to configure a MySQL database for export by using the database administrator (DBA) to create an archiving user with all read rights on the database objects to be exported.

```
CREATE USER '<USER>' IDENTIFIED BY '<PASSWORD>';
```

```
GRANT SELECT ON <SCHEMA>.* TO '<USER>';
```

(MySQL treats database and schema as the same concept. So for *<SCHEMA>* the database name is to be entered.)

With the help of this new userid one can enter a number of sample queries to check, if all relevant references are visible to it. It makes sense to document and comment the sample queries for future users of the archive.

### Troubleshooting MySQL

For *SIARD Suite* to be able to successfully connect to the desired MySQL database, the permission, firewalls etc must be configured such, that the following MySQL command can connect to the database successfully:

```
mysql -h<hostname> -P<port> -D<database> -u<database user> -p<database password>
```

Then the database name *<hostname>:<port>/<database>* is to be used in *SIARD Suite*. (N.B: No space between options and values, e.g. between *-p* and *<password>*!)

If you want to check the access privileges of a specific user of MySQL you must examine four privileges tables in INFORMATION\_SCHEMA: USER\_PRIVILEGES (global access), SCHEMA\_PRIVILEGES (schema access), TABLE\_PRIVILEGES (table access), and COLUMN\_PRIVILEGES (column access).

### 8.4.6 DB/2 Preparation

DB/2 database users are users of the operating system. In order to create an archiving user, one must create it as an operating system user on the machine, where the DB/2 database is running and then grant it access rights to a DB/2 database with the GRANT ON DATABASE command.

N.B.: On LINUX the name of a DB/2 user must be all lower-case. Also a DB/2 userid must not be longer than eight characters!

### Troubleshooting DB/2

On the workstation, where *SIARD Suite* is run, a DB/2 client must be installed. Under Windows the command *db2cmd* must then be issued to access the *db2* command-line environment. Under LINUX this environment is directly available.

Using the command-line processor *db2* a TCP/IP node must be created using the CATALOG command and then a connection can be established using CONNECT. In the DB/2 command window or the *db2* command-line processor enter the following:

#### a) LIST NODE DIRECTORY

This will list the nodes registered on the client. If no node for the desired DB/2 instance is available, one must create one, using the command:

## SIARD Suite

CATALOG TCPIP NODE <nodename> REMOTE <dnsname> SERVER <port>

Here one can use e.g. the first portion of the DNS name for the <nodename>. For <dnsname> the full DNS name of the host running the database must be entered. As <port> the port number of the DB/2 listener is to be specified (default: 50000). If one enters LIST NODE DIRECTORY again now, the new node name will appear.

b) ATTACH TO <nodename> USER <siarduser>

Here the <nodename> is to be used, which represents the remote database instance on this client. As <siarduser> enter the userid of the archiving user of the database on this instance. On executing the ATTACH command, you will be asked for the password of this user.

c) LIST DATABASE DIRECTORY

This will display a list of all databases on the DB/2 instance, that one is attached to.

d) CONNECT TO <database> USER <siarduser>

Again, enter the userid of the archiving user of this database for <siarduser>. On executing the CONNECT command, you will be prompted for the password of this user. All further commands (SQL statements) apply to this database.

If the connection to the database was successful, one can download the database using SIARD Suite entering <dnsname>:<port>/<database> as the database name and <siarduser> and the corresponding password.

## 9 SiardToDb

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

### 9.1 Invocation

```
java -cp <siardpath>/bin/SiardEdit.jar ch.admin.bar.siard.SiardToDb [-h] |
  -c <config file>] -d <data source> -n <database name> -u <database user>
  -p <database password> -s <siard file> [-q <query timeout>] [<schema> <mappedschema>]*
```

Specify *<siardpath>* as the folder where SIARD is installed. The file *SiardEdit.jar* is in the *bin* subfolder with its class *ch.admin.bar.siard.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 given on the command line.

As the call can have quite a lot of parameters, one can simplify it using a configuration file which contains all the parameters except the password. If one is available, the call reduces to:

```
java -cp <siardpath>/SiardEdit.jar ch.admin.bar.siard.SiardToDb -c <config file>
  -p <database password>
```

If all parameters and, additionally, a configuration file are given, the parameters on the command line supplement those given in the configuration file, which simplifies repeating SIARD uploads.

### 9.2 Invocation parameters

Parameter	Meaning
<data source>	Access, ODBC, Oracle, "SQL Server", MySQL, or "DB/2"
<database name>	Database name, structured depending on the connection: Access: file name of MS Access database ODBC (deprecated): ODBC connection's configured Datasourcename (DSN) Oracle: e.g. dbhost.enternet.ch:1521:SIARD1 SQL Server: e.g. dbhost.enternet.ch:1433/SIARD MySQL: e.g. dbhost.enternet.ch:3306/SIARD DB/2: z.B. dbhost.enternet.ch:50000/SIARD
<database user>	UserId for the upload session
<database password>	Password for the upload session
<siard file>	File name for the SIARD archive being uploaded
<query timeout>	Timeout in seconds for SQL queries (0 for unlimited)
<schema>	Schema name in SIARD archive (can be repeated any number of times each time followed by <mappedschema>)

## SIARD Suite

<i>Parameter</i>	<i>Meaning</i>
<mappedschema>	Schema name to be used for the upload. For all unmapped schemas their own name is used for the upload.

### 9.3 Notes

As older databases are not SQL:1999 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 SQL Server 2000, Oracle 8 and Oracle 9.

Uploading only creates tables. All other database objects are not created. Furthermore, certain sacrifices are made. In MS Access, all tables land in the same MDB. 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 *SiardEdit* 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

```
-Duser.timezone=GMT
```

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

### 9.4 Database preparation

Various preparations for the upload are necessary, depending on the database system used.

#### 9.4.1 Preparation of “pure JAVA” Access to MS Access Databases

An empty target database must be created preparatory for loading data into it.

#### 9.4.2 Preparation of ODBC Access to MS Access Databases (deprecated)

To load a SIARD file into an MS Access database, proceed as follows:

1. Use MS Access to create an empty Mdb.
2. In case the *MSysObjects*, *MSysRelationships* and *MSysQueries* are not readable, make them readable to user *Admin*.

MS Access prior to 2007:

Use *Tools / Options / View* to make System Objects visible and then use *Tools / Security / User and Group permissions* to make the tables *MSysObjects*, *MSysRelationships* and *MSysQueries* readable for the user *Admin*.

MS Access 2007:

*Office Button / Access Options / Current Database / Navigation Options... / Show System Objects* to make system objects visible and, after that, *Database Tools / Administer / Users and Permissions / User and Group Permissions* to make the tables *MSysObjects*, *MSysRelationships* and *MSysQueries* readable for the user *Admin*.



## SIARD Suite

(The whole group *Database Tools / Administer / Users and Permissions* is missing if the open database file is an *accdb* file.)

MS Access 2010 or later:

Same as MS Access 2007 but use *File* ribbon instead of *Office Button*.

3. Set up an ODBC connection to the just created, empty Upload MDB (see results for keyword ODBC in Windows Help under *Start / Help* for instructions on how to set up an ODBC connection via *Start / Settings / Administration Tools / Data Sources (ODBC)*) and their extended properties under *Configure / Advanced* to set *ExtendedAnsiSQL=1*. The ODBC connection can be set up as a *User DSN* by non-administrators.

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

If *SIARD Suite* was started with a 32-bit JAVA runtime, only the 32-bit ODBC driver is accessible to it.. If 64-bit MS Access is installed on a 64-bit Windows, the standard ODBC administrator program (*C:\Windows\system32\odbcad32.exe*) only creates 64-bit ODBC connections. These are only accessible to *SIARD Suite* if it was started with a 64-bit JAVA runtime. If you want to configure a 32-bit ODBC connection on 64-bit Windows, you must use the 32-bit ODBC administrator program (*C:\Windows\SysWOW64\odbcad32.exe*). The 32-bit ODBC connections created with this ODBC administrator are only accessible to *SIARD Suite*, if it was started with a 32-bit JAVA runtime.

The batch scripts *siardedit.cmd*, *siardfromdb.cmd*, and *siardtodb.cmd* (as well as the shortcut link on the desktop created by installation) use the 64-bit JAVA runtime by default, if it is available. If a 32-bit JAVA runtime is available on the same machine, then *SIARD Suite* can be started manually with it, by calling the batch scripts with the optional argument */32* on the command line. If problems are encountered on upload of an MS Access file with a 64-bit ODBC connection, it is worth trying the same upload with a 32-bit ODBC connection. At the time of writing the 64-bit ODBC driver is still very buggy.

### 9.4.3 Oracle Preparation

To load a SIARD file into an Oracle database, proceed as follows:

1. For each Schema in the SIARD archive, use the CREATE USER command to set up a schema user (UPPER CASE only; Oracle support for user names with lower case characters is deficient).
2. Grant CONNECT and RESOURCE privileges to an archive user (e.g. one of the Schema users).
3. Grant all create and write rights to all schemata to the archive user chosen in 2 above.

SIARD archives contain more than one schema only in exceptional circumstances. The instructions above mostly reduce to these lines:

```
CREATE USER <SCHEMA> IDENTIFIED BY <PASSWORD>
GRANT CONNECT, RESOURCE TO <SCHEMA>
```

Potentially, uploading will convert long character strings (e.g. VARCHAR(4000) in UTF-16 databases which accept a maximum of VARCHAR(2000)) into CLOBs. Furthermore, for the upload, the session parameter NLS\_LENGTH\_SEMANTICS will be set to CHAR (Standard: BYTE).

### 9.4.4 SQL Server Preparation

Proceed as follows to load a SIARD archive into an SQL Server database:

## SIARD Suite

1. As Administrator, create an empty upload database. (In Microsoft SQL Server Management Studio via *New Database ...* under *Databases* or by using the CREATE DATABASE command.)
2. Create a *Login* for the archive user. No server privileges are to be given this "archive user".
3. SQL Server 2000 only: for each Schema in the upload database, create a user which is associated with the *Login* of the archive user.
4. The uploaded database grants the archive user at least the following privileges:
  - a. Alter any role
  - b. Alter any schema
  - c. Alter any user
  - d. Alter
  - e. Connect
  - f. Create function
  - g. Create procedure
  - h. Create role
  - i. Create rule
  - j. Create schema
  - k. Create table
  - l. Create type
  - m. Create view
  - n. Delete
  - o. Execute
  - p. Insert
  - q. References
  - r. Select
  - s. Update
  - t. View database state
  - u. View definition

Users and Schemata were identical in SQL Server 2000. For each Schema in SQL Server 2005/2010/2012, one carries out the command

```
CREATE SCHEMA <SCHEMA> AUTHORIZATION <SCHEMAUSER>.
```

In this case it is sufficient to grant the archive user just the Create privileges mentioned above.

### 9.4.5 MySQL Preparation

Proceed as follows to load a SIARD archive into a MySQL database:

1. As database administrator create a database (schema) for each schema in the SIARD file. using the CREATE SCHEMA command.
2. Grant all create and write privileges for all schemas to the archiving user.

## SIARD Suite

Only rarely does a SIARD archive contain more than one schema. In that case just execute the following commands:

```
CREATE SCHEMA <SCHEMA>;  
CREATE USER '<USER>' IDENTIFIED BY '<PASSWORD>';  
GRANT ALL ON <SCHEMA>.* TO '<USER>';
```

### 9.4.6 DB/2 Preparation

In order to upload a SIARD archive to a DB/2 database, one proceeds as follows:

Create an archiving user (e.g. *siardusr*) on the operating system of the machine, where the DB/2 instance is running.

N.B.: If the database runs on a LINUX system, the userid must be all lower-case (or digits) and not consist of more than eight characters!

As database administrator create a new empty database as a container for the data in the SIARD archive to be uploaded.

Using the command-line processor *db2* a TCP/IP node must be created using the CATALOG command and then a connection can be established using CONNECT. In the DB/2 command window or the *db2* command-line processor enter the following:

a) LIST NODE DIRECTORY

This will list the nodes registered on the client. If no node for the desired DB/2 instance is available, one must create one, using the command:

```
CATALOG TCPIP NODE <nodename> REMOTE <dnsname> SERVER <port>
```

Here one can use e.g. the first portion of the DNS name for the <nodename>. For <dnsname> the full DNS name of the host running the database must be entered. As <port> the port number of the DB/2 listener is to be specified (default: 50000). If one enters LIST NODE DIRECTORY again now, the new node name will appear.

b) ATTACH TO <nodename> USER <dbadm>

Here the <nodename> is to be used, which represents the remote database instance on this client. As <dbadm> enter the user id of a database administrator on this instance. On executing the ATTACH command, you will be asked for the password of this user.

c) LIST DATABASE DIRECTORY

This will display a list of all databases on the DB/2 instance, that one is attached to.

d) CREATE DATABASE <newdatabase> USING CODESET UTF-8 TERRITORY DE

The name <newdatabase> for the new database on the DB/2 instance can be freely chosen. It must not already be in use. Also it must not be longer than eight characters. Upper-/lower-case are ignored.

Use of Codeset UTF-8 is recommended, because strings with special characters may not fit into fixed-length fields otherwise, which are always measured in byte lengths in DB/2. When the codeset is specified, entry of a territory is mandatory. One can choose other ISO country codes (e.g. CH, US, ...)

e) CONNECT TO <newdatabase> USER <dbadm>

## SIARD Suite

As <dbadm> enter the user id of a database administrator of this DB/2 instance. On execution of this CONNECT request, you will be prompted for the password of the database administrator. All further commands will apply to the newly created database.

- f) CREATE SCHEMA <schema> AUTHORIZATION <siarduser>

for each schema in the SIARD archive a schema must be created. For <siarduser> enter the user id of the archiving user created above, who will be uploading the SIARD data.

- g) For testing purposes connect yourself as <siarduser> to the new database:

CONNECT TO <newdatabase> USER <siarduser>

This will prompt for the password of this user.

All these preparatory actions obviously require administrator access to the machine where the DB/2 instance runs. The rest of the upload can be executed using the archiving user only.

## **10 Supported Database Systems**

The *SIARD Suite* supports "MS Access", "Oracle", "SQL Server", "MySQL" and "DB/2" database systems. The JDBC/ODBC Bridge from SUN is used for MS Access. Database specific JDBC drivers are used for accessing the other database systems.

Due to their shortcomings, the use of JDBC and ODBC drivers for accessing databases other than MS Access is not advised.

Different database systems support different sets of data types. SIARD Suite's mapping of these product-specific data types to SQL:1999 data types is described in a separate document for each supported database system.

### 11 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 is redirected to the console with level INFO. In order to redirect it to a log file, remove the comment # in front of

```
handlers= java.util.logging.FileHandler, java.util.logging.ConsoleHandler.
```

In addition one must change the file name after the equal sign in the line

```
java.util.logging.FileHandler.pattern =logs/siard%u.log
```

by the absolute path of the log file to be written. In properties files backslashes (\) must be duplicated. Under Windows one can always use slashes (/) instead of backslashes (\) in file names. E.g.:

```
java.util.logging.FileHandler.pattern = C:/Projects/Bundesarchiv/SIARD/logs/siard%u.log.
```

In this configuration file one can also set the log level (.level). Acceptable values for the log level are: NONE, SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST, ALL. ALL produces the largest output which is useful only for developers. SEVERE produces only information about major program failures. The INFO level is a useful everyday compromise.

## **12      Limitations**

The SIARD-Format implies the following limitations:

1. The size of a SIARD file cannot be larger than 18'446'744'073'709'551'615 Bytes (ca. 18 ExaBytes) (ZIP64 limitation).
2. The number of (table and lob) files cannot be larger than 4'294'967'295 (ca. 4 billion) (ZIP64 limitation).
3. All of the metadata of the database must fit into JAVA memory (heap).

The first two limitations are probably irrelevant because real databases will not reach such sizes for quite some time.

The third 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`.