



**Hartwig Thomas**, 22. December 2014

Dokumentversion 1.22

---

# SIARD Suite

## Handbuch



---

Herausgegeben von:

Schweizerisches Bundesarchiv  
Archivstrasse 24  
3003 Bern  
Schweiz

# Zusammenfassung

Dieses Dokument ist ein technisches Benutzerhandbuch für die Anwendung *SIARD Suite* (Software-Independent Archival of Relational Databases) des Schweizerischen Bundesarchivs.

Es beschreibt

- die technischen Voraussetzungen für den Einsatz
- die Installation
- das Ausführen

der *SIARD Suite* und seiner Komponenten.

## **1 Inhalt**

<b>1</b>	<b>INHALT .....</b>	<b>3</b>
<b>2</b>	<b>EINLEITUNG .....</b>	<b>4</b>
2.1	NEUERUNGEN .....	4
2.2	URHEBERRECHT .....	4
<b>3</b>	<b>KONZEPT SIARD .....</b>	<b>6</b>
<b>4</b>	<b>VORAUSSETZUNGEN FÜR DIE BENUTZUNG DER <i>SIARD SUITE</i> .....</b>	<b>7</b>
4.1	JAVA .....	7
4.2	DATENBANKEN .....	7
<b>5</b>	<b>INSTALLATION .....</b>	<b>9</b>
<b>6</b>	<b>SIARD-KOMPONENTEN .....</b>	<b>10</b>
<b>7</b>	<b>SIARDEDIT .....</b>	<b>11</b>
7.1	ERSTAUFRUF .....	11
7.2	HAUPTFENSTER .....	12
7.3	MENÜ .....	15
<b>8</b>	<b>SIARDFROMDB .....</b>	<b>26</b>
8.1	AUFRUF .....	26
8.2	AUFRUFPARAMETER .....	26
8.3	BEMERKUNGEN .....	27
8.4	VORBEREITUNG DATENBANK .....	27
<b>9</b>	<b>SIARDTODB .....</b>	<b>33</b>
9.1	AUFRUF .....	33
9.2	AUFRUFPARAMETER .....	33
9.3	BEMERKUNGEN .....	34
9.4	VORBEREITUNGEN DATENBANK .....	34
<b>10</b>	<b>UNTERSTÜTZTE DATENBANKSYSTEME .....</b>	<b>39</b>
<b>11</b>	<b>PROTOKOLLIERUNG .....</b>	<b>40</b>
<b>12</b>	<b>BESCHRÄNKUNGEN .....</b>	<b>41</b>

## 2 Einleitung

Das Schweizerische Bundesarchiv hat im Rahmen des Projekts ARELDA (ARchivierung ELEktronischer DATen) das SIARD (Software-Independent Archival of Relational Databases) Datenformat und die Anwendung *SIARD Suite* entwickelt, welche dazu dient, Datenbankinhalte langfristig zu archivieren.

Die Enter AG hat im Auftrag des Bundesarchivs praktische Erfahrung mit der Archivierung produktiver Datenbanken gesammelt (2005, 2006) und gegen Ende 2007 die *SIARD Suite* entwickelt, in welcher die Erfahrungen aus der Praxis zur Verbesserung des Produkts umgesetzt sind.

Das vorliegende Dokument ist das Handbuch dieser *SIARD Suite*.

### 2.1 Neuerungen

Mit der Version 1.71 von SIARD Suite kann man neu MS Access Datenbanken mit Hilfe der „pure JAVA“ JDBC-Schnittstelle *EnterAccess* herunterladen. Diese Alternative zum ODBC-Zugang wird diesen nach einer Übergangsphase, wo beide zur Verfügung stehen, gänzlich ablösen. Damit entfallen Probleme mit der Einrichtung von ODBC, mit der OS-Architektur und mit der Einschränkung, dass MS Access-Dateien nur unter Windows archiviert werden können. Unter JAVA 1.8 steht die JDBC-ODBC-Brücke ohnehin nicht mehr zur Verfügung.

### 2.2 Urheberrecht

Die *SIARD Suite* ist eine Entwicklung des Schweizerischen Bundesarchivs. Alle Rechte liegen beim Schweizerischen Bundesarchiv.

Die *SIARD Suite* stützt sich auf folgende Komponente anderer Hersteller:

- Zip64File (von <http://sourceforge.net/projects/zip64file/>  
Lizenz: GNU Public License Version 2 (GPLv2)  
in der SIARD-Auslieferung unter *doc/gpl-2.0.txt*)
- JavaHelp 2.0.05 (von <http://java.sun.com/products/javahelp>,  
Lizenz: in der SIARD-Auslieferung unter *doc/JavaHelp-LICENSE.HTML*)
- SUN JDBC-ODBC Bridge (deprecated, wird mit JAVA 8 ausser Betrieb genommen  
Lizenz: Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX<sup>1</sup>  
in der SIARD-Auslieferung unter *doc/java-license.txt*)
- Oracle JDBC-Driver *ojdbc6.jar*  
(von <http://www.oracle.com/technetwork/topics/winsoft-085727.html>  
Lizenz: Oracle Technology Network (OTN) Development and Distribution License Terms<sup>2</sup>  
in der SIARD-Auslieferung unter *doc/oracle-license.txt*)
- MySQL JDBC-Driver (von <http://dev.mysql.com/downloads/connector/j/>  
Lizenz: GNU Public License 2.0<sup>3</sup>  
in der SIARD-Auslieferung unter *doc/gpl-2.0.txt*)
- SQL Server JDBC Driver (von  
<http://msdn.microsoft.com/en-us/sqlserver/jj589698.aspx>  
Lizenz: Redistribution License For Microsoft JDBC Driver

---

<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>

## SIARD Suite

- in der SIARD-Auslieferung unter *doc/mssql-license.txt*)
- DB/2 JDBC-Driver (von [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)  
Lizenz: DB2 Personal Developers Edition: Redistributable JDBC Type 4 Driver<sup>4</sup>  
in der SIARD-Auslieferung unter *doc/ibm.db2.jdbc.type4.driver.redistribution.license.txt*)
- EnterAccess 1.0 (von <http://sourceforge.net/projects/enteraccess/>,  
Lizenz: GNU Library or Lesser General Public License 2.0 (LGPLv2)<sup>5</sup>  
in der SIARD-Auslieferung unter *doc/lgpl-2.0.txt*)
- Jackcess 2.0.8 (von <http://jackcess.sourceforge.net/>  
Lizenz: GNU Lesser General Public License (LGPL) 2.1<sup>6</sup>  
in der SIARD-Auslieferung unter *doc/lgpl-2.1.txt*)
- H2 1.3.176 (von <http://www.h2database.com/>  
Mozilla Public License Version 2.0 (MPL 2.0<sup>7</sup>) und Eclipse Public License (EPL 1.0<sup>8</sup>)
- Apache POI Project 3.9 (von <http://poi.apache.org/>  
Lizenz: Apache License, Version 2.0<sup>9</sup>  
in der SIARD-Auslieferung unter *doc/apache-license-2.0.txt*)
- Apache Commons Lang 2.6 (von <http://commons.apache.org>  
Lizenz: Apache License, Version 2.0<sup>10</sup>  
in der SIARD-Auslieferung unter *doc/apache-license-2.0.txt*)
- Apache Commons Logging API 1.1.3 (von <http://commons.apache.org>  
Lizenz: Apache License, Version 2.0<sup>11</sup>  
in der SIARD-Auslieferung unter *doc/apache-license-2.0.txt*)

Diese Komponenten werden mit der SIARD-Auslieferung als JAR-Dateien ausgeliefert, um den Benutzern die Installation zu erleichtern. Die Benutzer der *SIARD Suite* sind gehalten, die Lizenzbestimmungen dieser Komponenten zu befolgen, welche im Ordner */doc* vorliegen.

---

<sup>4</sup>

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

<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 Konzept SIARD

Da das Schweizerische Bundesarchiv verpflichtet ist, Unterlagen der Bundesverwaltung unabhängig vom Informationsträger<sup>12</sup> zu archivieren, musste die Frage der Langzeit-Archivierung relationaler Datenbanken gelöst werden<sup>13</sup>.

Im Rahmen des Projekts ARELDA kam man zum Schluss, dass die heutigen Datenbanksysteme alle zu proprietär implementiert sind, so dass auf ihrer Basis allein keine glaubwürdige Langzeit-Archivierung möglich ist.

Um die Verfügbarkeit über lange Zeiträume zu gewährleisten, entschied man sich, Datenbanken SQL:1999<sup>14</sup>-konform zu speichern. Die Dateninhalte wurden in einer Kollektion von XML-Dateien gespeichert. Weil das resultierende Archivformat für Datenbanken auf diesen beiden ISO-Standards beruht, wird ihm eine länger dauernde Verstehbarkeit zugetraut.

Im Folgenden wird das Archivformat für Datenbanken SIARD-Format genannt.

Eine wesentliche inhaltliche Anforderung an im SIARD-Format archivierte Daten ist, dass der Datenbank-Inhalt „Unterlagen-Charakter“ hat. Das heisst, dass die Inhalte der archivierten Tabellen Geschäftsdaten der Institution enthalten, welche die Datenbank betreibt, die unabhängig von Frontend-Anwendungen verstehbar sind. Mit der *SIARD Suite* werden weder ausführbarer Programmcode noch Objekte archiviert, sondern nur die Geschäftsdaten in Datenbanktabellen.

Im SIARD-Format ist die Schemadefinition einer archivierten Datenbank in SQL:1999-konformen XML-Dateien abgespeichert, während die inhaltliche Dokumentation der Tabellen und Felder sowie die eigentlichen Daten in XML-Dateien abgelegt sind. Damit diese nicht übermässig aufgebläht werden, sind BLOB- und CLOB-Daten (Binary Large Objects und Character Large Objects) in separaten (binären) Dateien gespeichert, die in den XML-Dateien referenziert sind.

In der *SIARD Suite* werden zur Zeit nur die Kernfunktionen, die eigentlichen „Core Features“ von SQL:1999 benutzt. In zukünftigen Versionen der *SIARD Suite* kann die Unterstützung auf weitere Bestandteile, sogenannte „Packages“ von SQL:1999 erweitert werden.

Hier wird auf die Struktur des SIARD-Formats nicht weiter eingegangen, da diese in einem separaten Dokument beschrieben ist, welches zusammen mit der *SIARD Suite* ausgeliefert wird. Im Jahr 2013 wurde das SIARD Format als eCH-Standard anerkannt und kann als eCH-0165<sup>15</sup> heruntergeladen werden.

---

<sup>12</sup> Bundesgesetz über die Archivierung (Archivierungsgesetz, BGA) vom 26. Juni 1998 (Stand am 22. Juli 2003)

<sup>13</sup> Archivierung von elektronischen digitalen Daten und Akten der Bundesverwaltung im Schweizerischen Bundesarchiv (ARELDA), Projekt-Exposé vom 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 und

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 Voraussetzungen für die Benutzung der *SIARD Suite*

Voraussetzung für die Benutzung der *SIARD Suite* ist eine vorhandene Installation von JAVA. Voraussetzung zum Laden aus oder zu einem Datenbank-System ist eine entsprechende Infrastruktur.

## 4.1 JAVA

Benutzer der *SIARD Suite* müssen vorgängig JAVA installieren. Die minimale technische Voraussetzung ist JRE 1.6.

JAVA ist frei erhältlich auf der Seite <http://www.java.com/de/> (JRE – JAVA Runtime Environment) oder <http://www.oracle.com/technetwork/java/javase/downloads/index.html> (JDK - JAVA Development Kit). Falls man nicht daran interessiert ist, MS Access Datenbanken über ODBC herunterzuladen, kann man stattdessen auch OpenJDK (<http://openjdk.java.net/>) benutzen.

Leider hat JAVA vor der Version 1.5 keine zentrale Möglichkeit eingeräumt, um die installierte JAVA-Version zu eruieren. Insbesondere können mehrere JAVA-Versionen installiert sein. So installiert der Oracle Client etwa seine eigene JAVA-Version (meistens älter und fehlerbehaftet) und fügt sie zur Umgebungsvariablen PATH hinzu(!).

Falls JAVA 1.6 oder höher unter Windows verfügbar ist, kann man dies folgendermassen prüfen: Unter *Start / Ausführen* den Befehl *regedit* eintippen. Wenn JAVA 1.6 oder höher installiert ist, hat in *HKEY\_LOCAL\_MACHINE\ SOFTWARE\JavaSoft\Java Runtime Environment* der Schlüssel *CurrentVersion* den Wert „1.6“ oder höher. Falls der Registry-Ast nicht existiert, oder eine tiefere *CurrentVersion* eingetragen ist, muss eine neue JAVA-Version installiert werden.

### 4.1.1 Architektur (32-bit/64-bit)

Ab *JAVA 1.6* hat man die Möglichkeit, auf einem 64-bit Betriebssystem (z.B.: Windows 7, Ubuntu, MacIntosh) eine 64-bit-Version von JAVA zu installieren. Dies ist generell empfehlenswert. Der JAVA-Code von *SIARD Suite* ist unabhängig davon, ob er auf der 32-bit- oder der 64-bit-Version der JAVA Runtime ausgeführt wird. Es besteht aber eine Abhängigkeit zur Architektur des Betriebssystems bei der JDBC-ODBC-Brücke, welche noch für MS Access Datenbanken verwendet werden kann<sup>16</sup>. Wenn 64-bit-JAVA verwendet wird, muss auch 64-bit MS Access (bzw. der zugehörige ACE ODBC-Treiber) installiert sein. Mit 64-bit-JAVA kann man nicht auf 32-bit ODBC-Datenquellen zugreifen. Mit 32-bit JAVA kann man nicht auf 64-bit ODBC-Datenquellen zugreifen. Leider hat der 64-bit ODBC-Treiber von Microsoft noch viele Bugs. Dies verunmöglicht die Konversion mancher MS Access Dateien.

Wenn sowohl 32-bit-JAVA als auch 64-bit JAVA installiert ist, startet *SIARD Suite* standardmässig mit der 64-bit-Runtime. Es kann aber manuell auch mit der 32-bit-Runtime aufgerufen werden.

In einer der nächsten SIARD-Versionen wird der ODBC-Zugang ganz wegfallen, da er durch die „pure JAVA“-Schnittstelle EnterAccess ersetzt wird. Wer diese verwendet, hat keine Probleme mit der Architektur des Betriebssystems.

## 4.2 Datenbanken

Die *SIARD Suite* unterstützt zum Zeitpunkt dieser Auslieferung die folgenden Datenbanksysteme:

---

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

## SIARD Suite

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

Zu einem späteren Zeitpunkt können weitere Datenbanksysteme eingebunden werden.

Die *SIARD Suite* kann mit folgenden JDBC-Treibern auf diese Datenbanken zugreifen:

EnterAccess-1.0.jar	für direkten „pure JAVA“ Zugriff auf MS Access Datenbanken (MDB, ACCDB 2000-2012) mit Hilfe von Jackcess – auch unter anderen Betriebssystemen als Windows
ojdbc6-fixed.jar	nur für Zugriff auf Oracle Datenbanken Oracle 8i bis Oracle 12c (Fehler im Oracle-Treiber, welche die Erzeugung von XSD-Dateien verhinderten, wurden korrigiert, indem nur die oracle.-Packages beibehalten wurden)
sqljdbc4.jar	nur für Zugriff auf SQL Server Datenbanken SQL Server 2000 bis SQL Server 2012
mysql-connector-java-5.1.34-bin.jar	nur für Zugriff auf MySQL Datenbanken Version 5.5 oder höher oder MariaDB Datenbanken.
db2jcc.jar und db2jcc_license_cu.jar	für Zugriff auf DB/2 Datenbanken Version 8 oder höher.
SUN JDBC-ODBC Brücke (in JAVA Run- time)	für ODBC Zugriff auf MS Access Datenbanken – nur unter Windows und nur mit JAVA Versionen kleiner als 1.8 verfügbar. Wird während einer Übergangsphase parallel zu EnterAccess zur Verfügung gestellt.

Diese Treiber werden mit der *SIARD Suite* zusammen mitgeliefert und installiert. Es ist wahrscheinlich, dass sie auch noch einige Jahre mit zukünftigen Versionen der Datenbanksysteme kompatibel sind.



### 5 Installation

Die *SIARD Suite* kann jederzeit ab USB Stick, CD-ROM oder Netzwerklaufwerk gestartet werden, sofern ein korrekt installiertes JAVA 1.6 oder höher zur Verfügung steht. Damit ist es unter Windows, LINUX, MacIntosh etc. lauffähig.

Die benutzerspezifischen Einstellungen werden unter `<home>/java/siard suite.properties` gespeichert. Hier bezeichnet `<home>` das plattformabhängige Home-Verzeichnis des Benutzers. (Unter Windows XP ist dies normalerweise `C:\Dokumente und Einstellungen\<userid>`, unter Windows 7, ist es `C:\Benutzer\<userid>`.)

Die *SIARD Suite* wird ohne spezielles Installationsprogramm ausgeliefert. Statt dessen detektiert das Hauptprogramm *SiardEdit*, ob, wo und in welcher Version die *SIARD Suite* schon einmal beim Benutzer installiert war.

Um also die *SIARD Suite* zu installieren startet man einfach *SiardEdit* – entweder mit einem plattformspezifischen Skript im Hauptverzeichnis der Auslieferung (*siardedit.cmd* bzw. *siardedit.sh*) oder als JAVA-Anwendung mittels Ausführen von `java -jar bin/SiardEdit.jar`.

Wenn *SiardEdit* feststellt, dass noch keine benutzerspezifischen Einstellungen vorhanden sind, offeriert es die Installation auf das System des Benutzers. Der vorgegebene Standardordner für die Installation ist `<home>/applications/siard suite`.

Zu jeder Zeit kann man die *SIARD Suite* ohne Administratorrechte in *SiardEdit* unter dem Menüpunkt *Extras* deinstallieren bzw. neu an einem anderen Ort installieren.

Auch wenn schon eine alte Version der *SIARD Suite* auf dem System des Benutzers installiert ist, kann man daneben eine neue Version ab USB Stick, CD-ROM oder Netzwerklaufwerk starten, ohne die alte Version zu ändern.

Wenn die *SIARD Suite* unter Windows installiert wird, fügt *SiardEdit* mit Hilfe des Befehlskripts *desktop.cmd* automatisch je eine Verknüpfung auf dem Arbeitsplatz und unter *Start/Programme* ein.



# 6 SIARD-Komponenten

Die *SIARD Suite* besteht aus folgenden Komponenten:

- SiardEdit
- SiardFromDb
- SiardToDb

Das Programm *SiardEdit* besitzt eine interaktive graphische Benutzeroberfläche. Es ermöglicht das Bearbeiten der Metadaten in einem SIARD-Archiv.

Das Programm *SiardEdit* ist das zentrale Instrument, mit dem Daten im SIARD-Format bearbeitet werden. Primärdaten können nicht verändert werden. Für komplexe Recherchen ist *SiardEdit* nicht geeignet. Für diese empfiehlt es sich, das SIARD-Archiv in eine Datenbank zu laden und mit Hilfe der Datenbank-Techniken zu recherchieren.

Die Befehlszeilenanwendung *SiardFromDb* dient dem Herunterladen einer Datenbank und deren Speicherung in einer SIARD-Datei. Die Funktionalität dieser Anwendung ist identisch mit der in *SiardEdit* unter dem Menüpunkt *Neu von Datenbank ...* verfügbaren Funktion. Besonders beim Herunterladen grosser Datenbanken ist die Benutzung der Befehlszeilenversion bequemer. Mittels Umleitung von *stdout* und *stderr* kann das lange dauernde Herunterladen auch besser dokumentiert werden (s. „Verwenden von Befehlsumleitungsoperatoren“ in der Windows-Hilfe unter *Start* oder online<sup>17</sup>).

Die Befehlszeilenanwendung *SiardToDb* dient dem Hinaufladen einer Datenbank aus einer SIARD-Datei. Die Funktionalität dieser Anwendung ist identisch mit der in *SiardEdit* unter dem Menüpunkt *In Datenbank laden ...* verfügbaren Funktion. Besonders beim Hinaufladen grosser Datenbanken ist die Benutzung der Befehlszeilenversion bequemer. Mittels Umleitung von *stdout* und *stderr* kann das lange dauernde Hinaufladen auch besser dokumentiert werden (s. „Verwenden von Befehlsumleitungsoperatoren“ in der Windows-Hilfe unter *Start* oder online<sup>18</sup>).

---

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

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

### 7 SiardEdit

Das Programm *SiardEdit* besitzt eine interaktive graphische Benutzeroberfläche. Es ermöglicht das Bearbeiten der Metadaten in einem SIARD-Archiv.

Mittels *SiardEdit* kann man:

- ein „leeres“ SIARD-Archiv (ohne Primärdaten) herstellen, um sich eine erste Übersicht über den Archivierungsprozess zu verschaffen,
- die Metadaten eines neu zu erzeugenden SIARD-Archivs anhand der Metadaten eines anderen (leeren oder nicht-leeren) Vorlagen-Archivs abgleichen (alle Beschreibungen von Elementen übernehmen, die im neuen und im Vorlagen-Archiv vorkommen),
- die Metadaten eines SIARD-Archivs anzeigen, verschieden sortieren und durchmustern, manuell ändern und ergänzen, soweit dies die im SIARD-Archiv gespeicherten Primärdaten nicht tangiert,
- die Primärdaten eines SIARD-Archivs anzeigen, verschieden sortieren und durchmustern.

Das Programm *SiardEdit* ist das zentrale Instrument, mit dem Daten im SIARD-Format bearbeitet werden. Primärdaten können nicht verändert werden. Für komplexe Recherchen ist *SiardEdit* nicht geeignet. Für diese empfiehlt es sich, das SIARD-Archiv in eine Datenbank zu laden und mit Hilfe der Datenbank-Techniken zu recherchieren.

Die Umwandlung der TIMES und TIMESTAMPS in der Datenbank ist abhängig von der lokalen Zeitzone. Wenn man etwa die Zeit 15:30 in Zürich abspeichert, wird in der XML-Datei die UTC-Zeit 14:30 (im Winter!) abgespeichert. Sind die Zeiten der Datenbank unverändert als UTC-Zeiten zu interpretieren, muss man *SiardEdit* mit der Option

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

Man kann *SiardEdit* mit einer zu öffnenden SIARD-Datei als einzigem Argument aufrufen.

#### 7.1 Erstaufwurf

Die *SIARD Suite* wird als ZIP-Datei zum Herunterladen zur Verfügung gestellt. Diese muss als erstes entpackt werden. Im Verzeichnis *bin* dieser Auslieferung befindet sich die Datei *SiardEdit.jar*. Wenn JAVA korrekt installiert wurde, führt man das Programm unter Windows aus, indem man auf dieser Datei doppelklickt. Man kann auch das plattformspezifische Skript *siardedit.cmd* (für Windows) oder *siardedit.sh* (für LINUX) ausführen.

Sollte dies nicht funktionieren oder arbeitet man unter einem anderen Betriebssystem, kann man *SiardEdit* auch von der Befehlszeile im Verzeichnis *bin* der *SIARD Suite* wie folgt aufrufen:

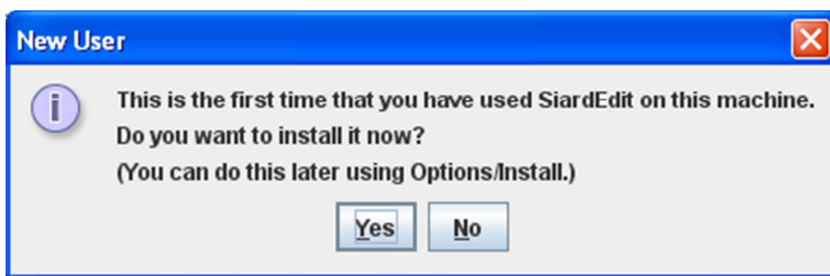
```
javaw -jar SiardEdit.jar (Windows)
```

```
java -jar SiardEdit.jar (alle Plattformen)
```

Hierbei ist vorausgesetzt, dass das Verzeichnis *bin* der JAVA-Installation im *PATH* vorhanden ist. Sonst muss man den vollen Pfad des ausführbaren Programms *java* ausschreiben.

Wenn man *SiardEdit* zum ersten Mal ausführt, erscheint hierauf folgende Meldung:

## SIARD Suite



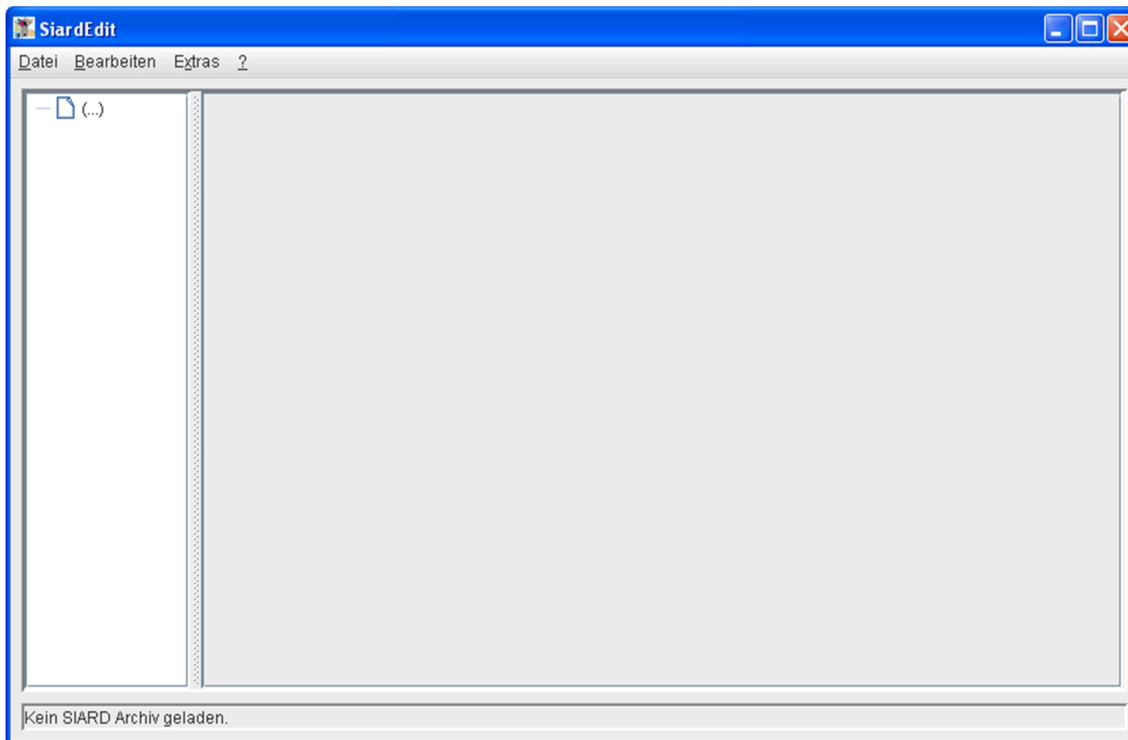
(Da *SiardEdit* zu diesem Zeitpunkt die Benutzersprache noch nicht kennt, hängt die Sprache dieser Meldung von der auf dem Betriebssystem und bei der Installation von JAVA gewählten Sprache ab.)

Wenn hier mit *Ja* geantwortet wird, erhält man die Möglichkeit, ein leeres oder neues Verzeichnis anzugeben, wo eine Kopie der SIARD Auslieferung angelegt werden soll. Nach der erfolgreichen Installation kann *SiardEdit* in Zukunft vom gewählten Verzeichnis aus gestartet werden.

Unabhängig davon, ob *SiardEdit* nur einmalig oder ab USB-Stick oder CD-ROM ausgeführt werden soll, oder auf dem PC des Benutzers installiert wurde, erscheint darauf das Hauptfenster von *SiardEdit*.

### 7.2 Hauptfenster

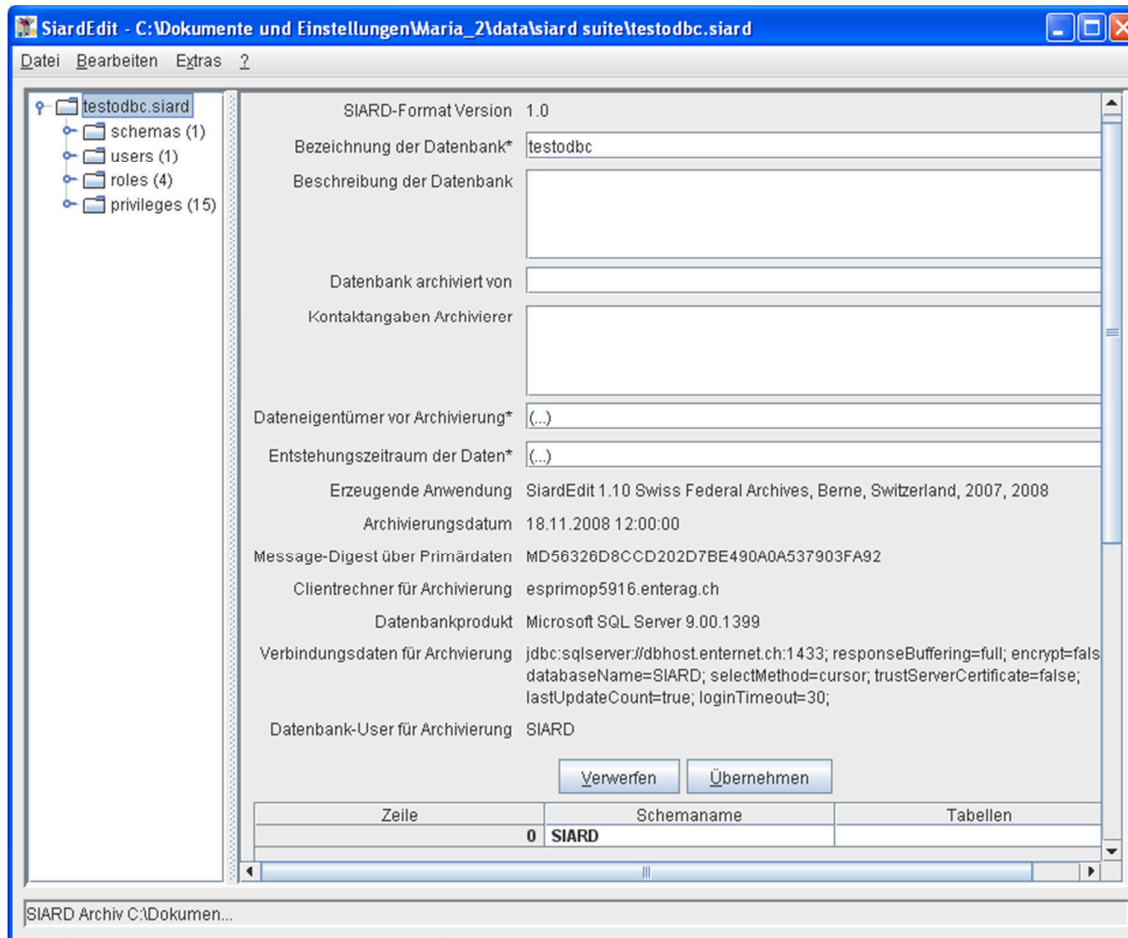
Das Hauptfenster besteht aus Menü (oben), Navigationsbaum (links), Inhalt (rechts) und Statuszeile (unten).



## SIARD Suite

Man kann die Grenze zwischen Navigation und Inhalt flexibel verschieben. Auch die Grösse des gesamten Fensters kann beliebig verändert werden bis auf eine vorgegebene Minimalgrösse.

Wenn eine SIARD-Datei in *SiardEdit* geladen ist, sieht das Hauptfenster folgendermassen aus:



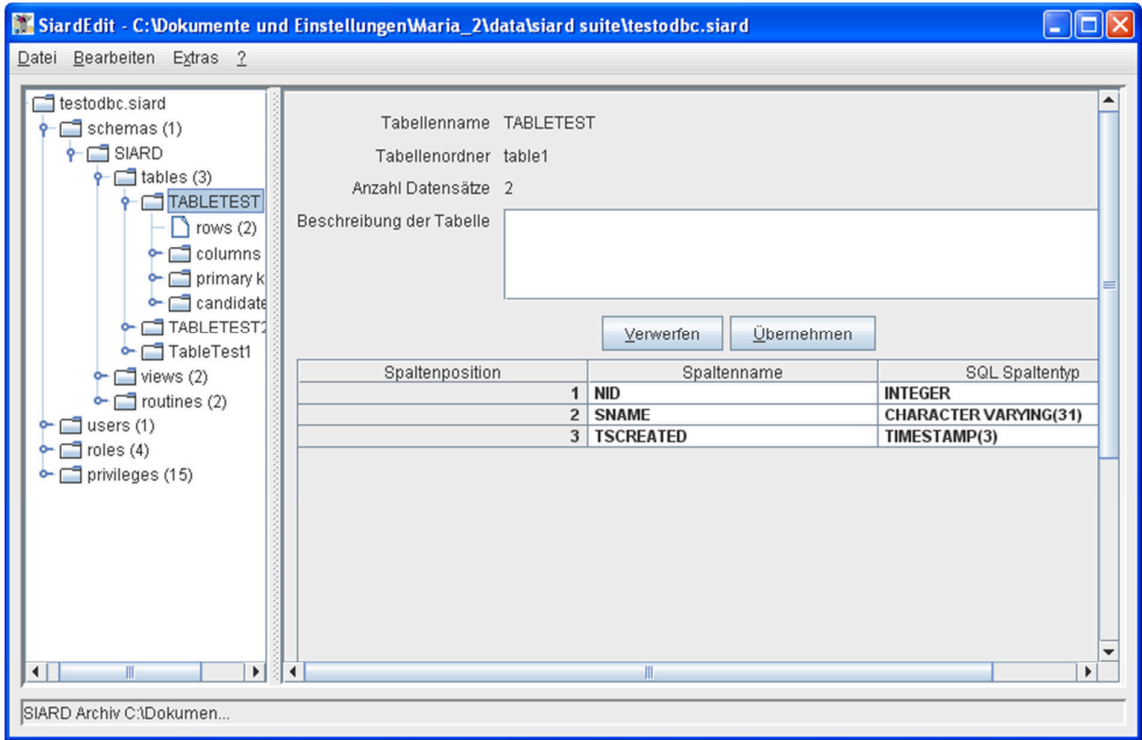
In der linken Hälfte kann man im Metadatenbaum navigieren. In der rechten Hälfte kann man oben veränderbare Metadaten eingeben bzw. ändern, die zum links ausgewählten Datenbankobjekt gehören.

Die Schaltfläche *Übernehmen* dient der Übertragung der Eingaben in die Metadaten der momentan geöffneten SIARD-Datei. Wenn man *Verwerfen* betätigt, werden alle Änderungen seit der letzten Übernahme rückgängig gemacht.

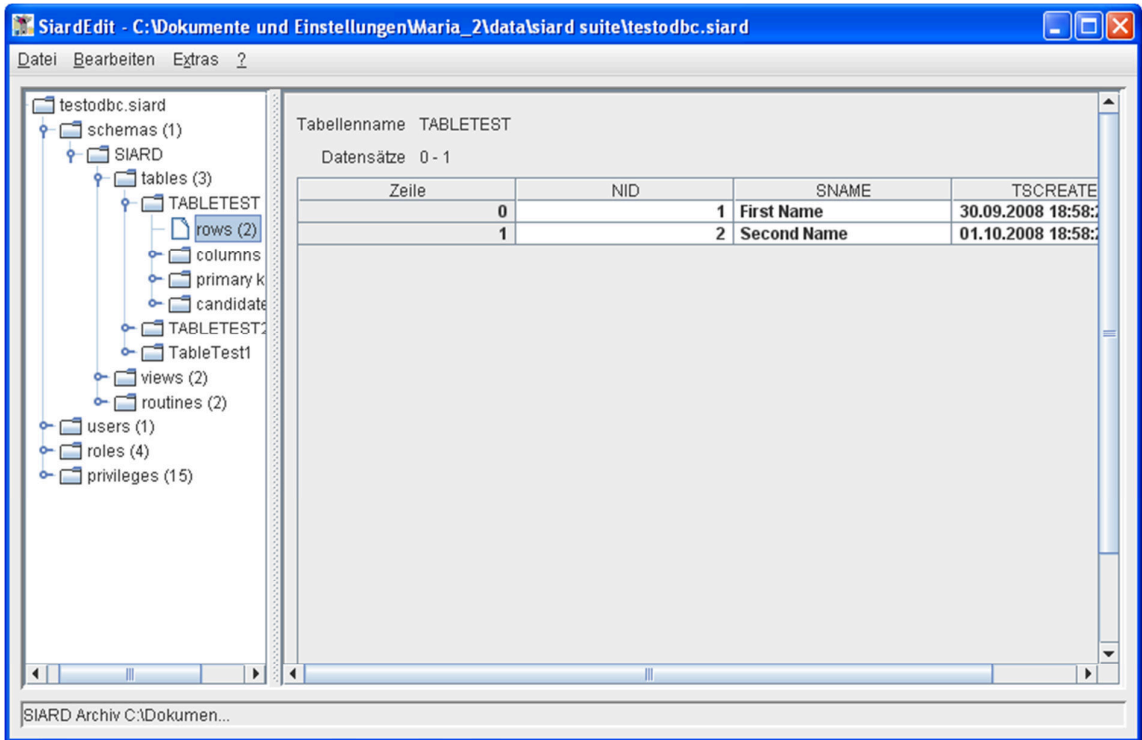
Unter diesen Metadaten erscheint eine Tabelle der wichtigsten Unterobjekte. Diese Tabelle kann man mittels Klick auf einen Spaltentitel nach dieser Spalte sortieren. Wenn man auf einer Zeile doppelklickt, wird das entsprechende Unterobjekt ausgewählt.

Da die Tabellen in einem Schema oder die Spalten in einer Tabelle naturgemäss in den Metadaten keine spezielle Reihenfolge haben und von *SiardEdit* normalerweise alphabetisch sortiert angezeigt werden, ist diese Sortierfunktion hilfreich, um sich in grossen Datenbankschemas zurechtzufinden.

SIARD Suite



Wenn man in diesem Baum auf Ebene Tabelle navigiert, kann man zu den Primärdaten einer Tabelle navigieren, indem man das Objekt *rows* unter der Tabelle auswählt.



## SIARD Suite

Während der Anzeige der Primärdaten verändert *SiardEdit* sein Verhalten geringfügig. Klickt man nun auf einen Spaltentitel, wird die ganze Tabelle nach dieser Spalte sortiert. Bei grossen Tabellen kann dies lange dauern!

Um auch die Arbeit mit sehr grossen Tabellen zu ermöglichen, zeigt *SiardEdit* nur jeden  $n$ -ten ( $n$  eine Potenz von 50) Datensatz an, um zu vermeiden, dass ein Fenster mit einer unmöglich grossen Anzahl von Zeilen angezeigt werden muss. Wenn man auf einer Zeile die Zeilennummer doppelklickt, werden die zwischen dieser Zeile und der nächsten liegenden Zeile liegenden Datensätze angezeigt bis die unterste Stufe erreicht ist, wo alle Datensätze fortlaufend angezeigt werden.

Wenn man auf einem Text-, Zahlen- oder Datumsfeld doppelklickt, wird der Dateninhalt in einem Text-Editor angezeigt. Wenn man auf einem binären Feld doppelklickt, wird der Inhalt in einem Binär-Editor geöffnet. Welche Anwendungen zur Anzeige der Felder benutzt werden, kann man unter *Extras / Optionen* einstellen. Man kann auch die Spaltenbegrenzungen verschieben, um die Einträge in Zahlen- oder Datumsspalten vollständig zu sehen.

Wenn man eine Zeile in der Tabelle mittels Anklicken anwählt, ändert diese die Farbe. Wenn man den Menüpunkt *Bearbeiten / Kopieren (Ctrl-C)* anwählt, kann man die Zeile in der Zwischenablage speichern. Wenn man den Menüpunkt *Bearbeiten / Alles Kopieren (Ctrl-A)* anwählt, kann man die ganze angezeigte Tabelle in die Zwischenablage kopieren. Die Zwischenablage fügt man danach z.B. in einem Tabellenkalkulationsprogramm ein, um sie weiter zu bearbeiten.

### 7.3 Menü

In den folgenden Unterkapiteln werden die Menüpunkte einzeln beschrieben.

#### 7.3.1 Datei

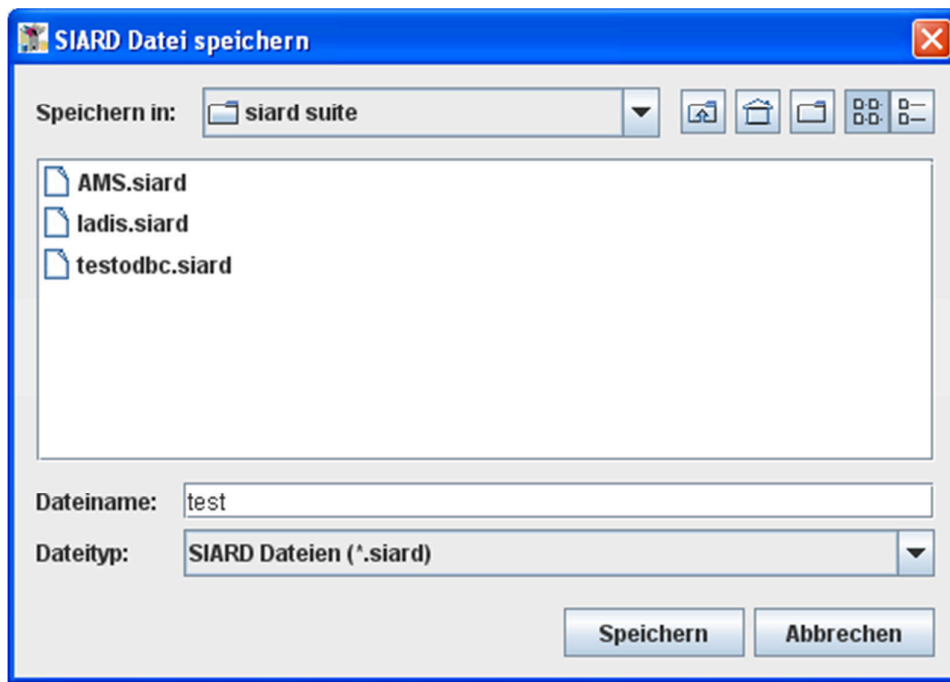
*Neu von Datenbank ...*

Mit diesem Befehl kann man eine Datenbank herunterladen und als SIARD-Datei abspeichern. Die gespeicherte SIARD-Datei wird anschliessend in *SiardEdit* angezeigt.

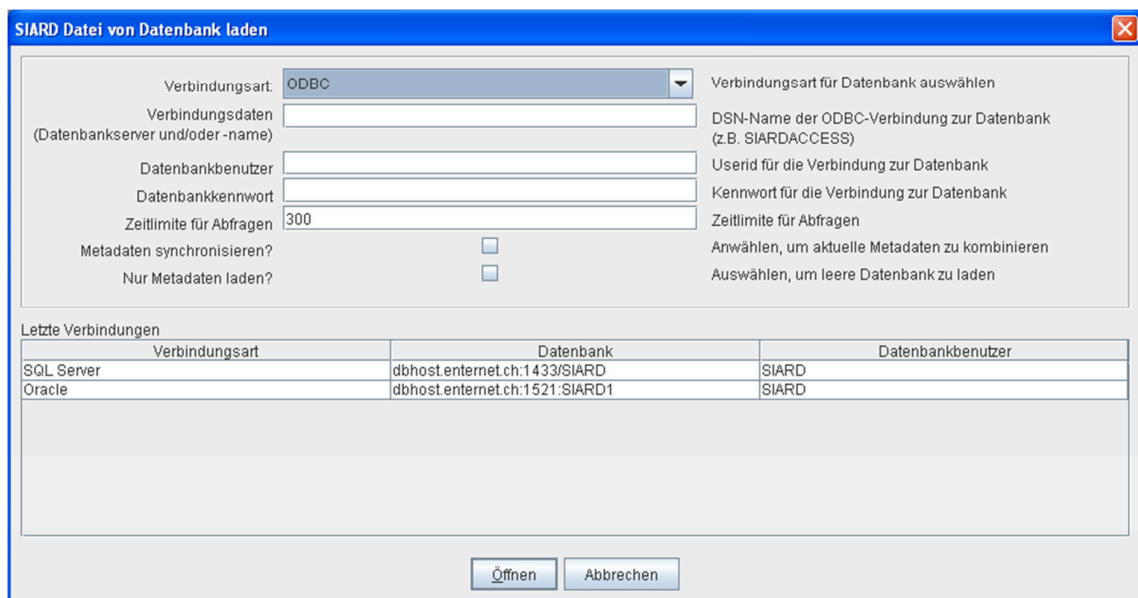
Diese Aktion ist funktional identisch mit der Befehlszeilenanwendung *SiardFromDb*. In der Beschreibung dieses Programms findet man wichtige Hinweise darauf, wie die verschiedenen Datenbanksysteme für diese Operation vorzubereiten sind.

Wenn man diesen Menüpunkt wählt, muss man als erstes angeben, unter welchem Namen die neu zu erzeugende SIARD-Datei gespeichert werden soll:

## SIARD Suite



Als nächstes wird man im Dialog „SIARD Datei von Datenbank laden“ aufgefordert, die Verbindungsdaten zur Datenbank einzugeben, die heruntergeladen werden soll:



### Verbindungsart

Hier ist unter Verbindungsart

- *Access* für „pure JAVA“-Zugriff auf MS-Access-Datenbanken,
- *ODBC* für ODBC-Zugriff auf MS-Access-Datenbanken (abgekündigt),



## SIARD Suite

- *Oracle* für Oracle-Datenbanken,
- *MySQL* für MySQL-Datenbanken,
- *SQL Server* für SQL-Server-Datenbanken oder
- *DB/2* für DB/2-Datenbanken

auszuwählen.

### Verbindungsdaten

Die Verbindungsdaten sind je nach Verbindungsart verschieden strukturiert.

### Access

Hier ist der Dateiname der MS Access Datei einzugeben.

### ODBC (abgekindigt)

Voraussetzung für die Benutzung der Verbindungsart *ODBC* ist, dass man vorgängig eine ODBC-Verbindung für die gewünschte Datenbank eingerichtet hat. Wie man dabei vorgeht, findet man in der Hilfedokumentation des Betriebssystems.

Unter Verbindungsdaten ist in diesem Fall der DSN (Datasource Name) der ODBC-Verbindung einzugeben.

### Oracle

Hier ist eine Zeichenkette einzugeben, die aus DNS-Namen (DNS = Domain Name System) des Datenbankservers, dem Listener-Port und der SID (System ID) der Datenbank besteht. Diese werden mit Doppelpunkten getrennt.

Beispiel für Oracle-Verbindungsdaten: *athlon700b.enterag.ch:1521:SIARD*.

Hier ist *athlon700b.enterag.ch* der DNS-Name des Servers, *1521* der Listener-Port der Datenbank (Standardwert für Oracle-Datenbanken, sofern ihre Konfiguration nicht vom Datenbank-Administrator geändert wurde) und *SIARD* die SID der Datenbank.

### SQL Server

Hier ist eine Zeichenkette einzugeben, die aus DNS-Namen (DNS = Domain Name System) des Datenbankservers, dem Listener-Port und dem Datenbanknamen besteht. Server und Port werden mit Doppelpunkt getrennt. Der Datenbankname wird mit einem Schrägstrich abgetrennt. Wird der Datenbankname weggelassen, wird die Verbindung mit derjenigen Datenbank hergestellt, die dem Datenbankbenutzer im SQL-Server als Standardzugang zugeordnet ist.

Beispiel für SQL-Server-Verbindungsdaten: *athlon800.enterag.ch:1433/siard*.

Hier ist „athlon800.enterag.ch“ der DNS-Name des Server, 1433 der Listener-Port der Datenbank (Standardwert für SQL-Server-Datenbanken, sofern ihre Konfiguration nicht vom Datenbank-Administrator geändert wurde) und „siard“ der Datenbankname der Datenbank.

### MySQL

Hier ist eine Zeichenkette einzugeben, die aus DNS-Namen (DNS = Domain Name System) des Datenbankservers, dem Listener-Port und dem Datenbanknamen (=SchemaNamen) besteht. Server und Port werden mit Doppelpunkt getrennt. Der Datenbankname bzw. Schemaname wird mit einem Schrägstrich abgetrennt.

Beispiel für MySQL -Verbindungsdaten: *asusp5b.enterag.ch:3306/SIARD*.

## SIARD Suite

Hier ist „asusp5b.enterag.ch“ der DNS-Name des Server, 3306 der Listener-Port der Datenbank (Standardwert für MySQL-Datenbanken, sofern ihre Konfiguration nicht vom Datenbank-Administrator geändert wurde) und SIARD der Datenbankname der Datenbank.

### DB/2

Hier ist eine Zeichenkette einzugeben, die aus DNS-Namen (DNS = Domain Name System) des Datenbankservers, dem Listener-Port und dem Datenbanknamen besteht. Server und Port werden mit Doppelpunkt getrennt. Der Datenbankname wird mit einem Schrägstrich abgetrennt.

Beispiel für DB/2 -Verbindungsdaten: *vmu64.enterag.ch:50000/SIARD*.

Hier ist „vmu64.enterag.ch“ der DNS-Name des Server, 50000 der Listener-Port der Datenbank (Standardwert für DB/2-Datenbanken vom Typ Express-C, sofern ihre Konfiguration nicht vom Datenbank-Administrator geändert wurde) und SIARD der Datenbankname der Datenbank.

### Datenbankbenutzer

Hier ist der Datenbankbenutzer einzugeben, dessen Sicht der Datenbank archiviert werden soll. Achtung: Manche Datenbanksysteme ignorieren Klein-/Grossschreibung, manche nicht!

Wenn nur Teile einer Datenbank archiviert werden sollen, richtet man einen Archivbenutzer ein, und erteilt ihm nur Leserechte auf diejenigen Datenbankobjekte, die archiviert werden sollen.

### Datenbankpasswort

Sowohl Datenbankbenutzer und Datenbankpasswort müssen vom Datenbankadministrator erzeugt werden. Sie können von diesem erfragt werden, wenn man sie nicht kennt. Achtung: Ältere Datenbanksysteme ignorieren Klein-/Grossschreibung, neuere nicht!

Bei MS Access-Datenbanken heisst der Benutzer normalerweise *Admin* und das Passwort ist nicht gesetzt (leere Zeichenkette).

### Nur Metadaten laden?

Wenn man diese Checkbox anwählt, wird eine SIARD-Datei ohne Primärdaten erzeugt. Dies ist dann von Nutzen, wenn man sich zuerst eine Übersicht über die zu erwartende Grösse der Datenbank verschaffen will. In einer solchen leeren SIARD-Datei kann man schon Erklärungen der Metadaten einfügen und diese später beim Herunterladen der vollständigen Datenbank mittels “Synchronisieren” dieser hinzufügen.

### Synchronisieren?

Diese Checkbox erscheint nur, wenn im Hauptfenster schon eine andere SIARD-Datei geladen war. Wird diese Option angewählt, dann werden die Metadaten der vorher geladenen SIARD-Datei so gut wie möglich in die Metadaten der neu heruntergeladenen SIARD-Datei kopiert. (D.h. diejenigen Einträge zu Tabellen, Spalten etc. die in beiden Datenbank-Schemas gleich sind, werden übernommen.)

### Letzte Verbindungen

In dieser Tabelle werden die letzten vom Benutzer erfolgreich etablierten Verbindungen aufgeführt. Diese sind in der Datei *siard.properties* im Ordner *java* im Home-Verzeichnis des Benutzers gespeichert.

Wenn man eine Zeile dieser Tabelle anklickt, werden die entsprechenden Verbindungsdaten im Dialog ausgefüllt. Nur das Passwort bleibt leer und der Cursor springt auf das Passwort-Feld.

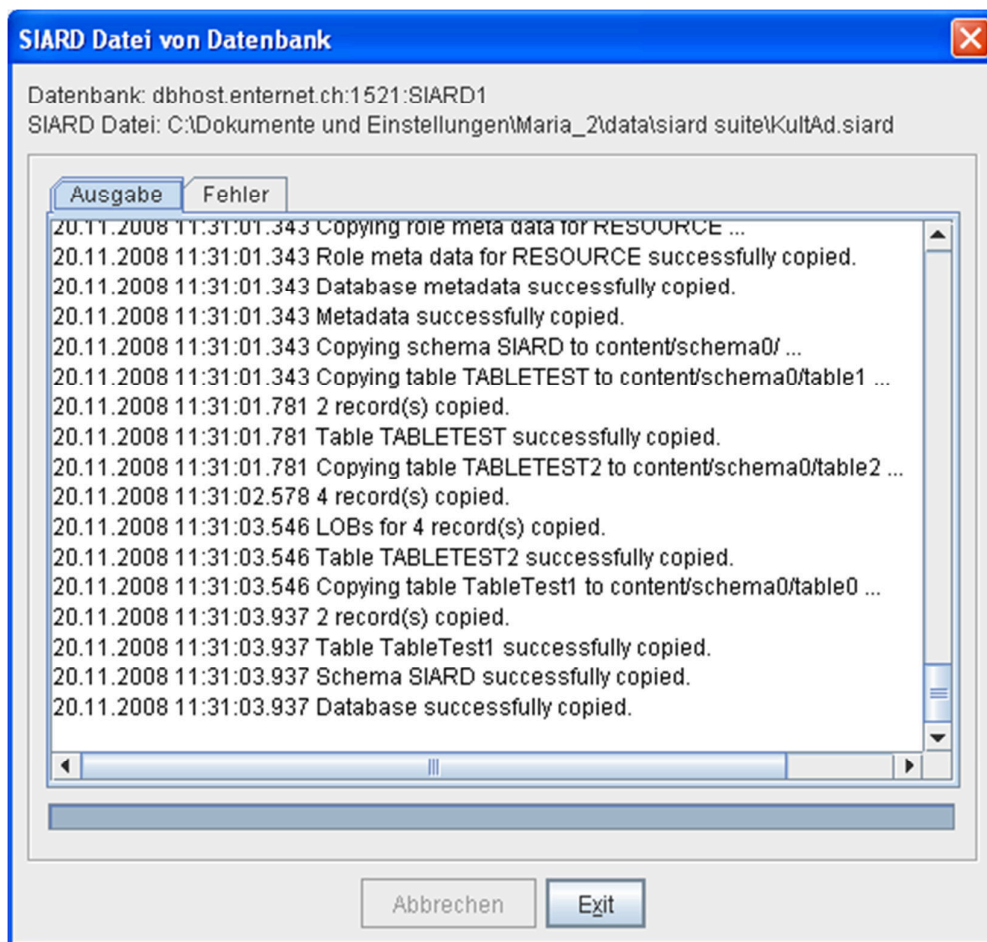
## SIARD Suite

### Öffnen

Mit dieser Schaltfläche wird die Datenbank in die ausgewählte SIARD-Datei geladen. Es erscheint ein Protokoll, das identisch mit demjenigen ist, welches von *SiardFromDb* ausgegeben wird.

### Abbrechen

Mit dieser Schaltfläche kann man die Aktion abbrechen.



Am Ende des Ladevorgangs kann man das Protokoll noch markieren, kopieren und mit Hilfe eines Textprogramms ausdrucken. Die Zeitangaben neben den einzelnen Schritten helfen, die Gesamtladezeit grösserer Datenbanken abzuschätzen.

Wenn man hier die Schaltfläche *Exit* wählt, erscheint die neu geladene Datei im Hauptfenster geöffnet.

### Öffnen ...

Wenn man diesen Menüpunkt betätigt, kann man (früher heruntergeladene) SIARD-Datei öffnen.

Ihre Metadaten des SIARD-Archivs erscheinen dann im Hauptfenster und können ergänzt und verändert abgespeichert werden.

## SIARD Suite

### Speichern ... (Ctrl-S)

Dieser Menüpunkt speichert die veränderten Metadaten in der geöffneten SIARD-Datei.

### In Datenbank laden ...

Mit diesem Menüpunkt kann man den Inhalt einer offenen SIARD-Datei in eine Datenbank laden.

Diese Aktion ist funktional identisch mit der Befehlszeilenanwendung *SiardToDb*. In der Beschreibung von *SiardToDb* findet man wichtige Hinweise darauf, wie die verschiedenen Datenbanksysteme für diese Operation vorzubereiten sind. Insbesondere muss vorgängig ein geeigneter Datenbankbenutzer mit den notwendigen Rechten erzeugt werden. Für das Hochladen in MS Access ist vorgängig eine leere Datenbank anzulegen und allenfalls über ODBC zugänglich zu machen.

Wenn man diesen Menüpunkt anwählt, wird man im Dialog *SIARD Datei in Datenbank laden* aufgefordert, die Verbindungsdaten zur Datenbank einzugeben, in welche die Daten zu laden sind.

**SIARD Datei in Datenbank laden**

Verbindungsart: ODBC (Dropdown) | Verbindungsart für Datenbank auswählen

Verbindungsdaten (Datenbankserver und/oder -name): | DSN-Name der ODBC-Verbindung zur Datenbank (z.B. SIARDACCESS)

Datenbankbenutzer: | Userid für die Verbindung zur Datenbank

Datenbankkennwort: | Kennwort für die Verbindung zur Datenbank

Zeitlimite für Abfragen: 300 | Zeitlimite für Abfragen

**Letzte Verbindungen**

Verbindungsart	Datenbank	Datenbankbenutzer
SQL Server	dbhost.enternet.ch:1433/SIARD	SIARD
Oracle	dbhost.enternet.ch:1521:SIARD1	SIARD

Buttons: Öffnen, Abbrechen

Nach der Herstellung der Verbindung mit der Upload-Datenbank, zeigt *SiardEdit* einen Dialog an, in welchem man den Schemas im SIARD-Archiv Schemanamen in der Upload-Datenbank zuordnen kann.

Das Hochladen von Datenbankinhalten unterliegt den Einschränkungen des Datenbanksystems, in welches die Daten geladen werden. So unterstützt etwa MS Access keine Datenbankschemas. Deshalb werden in dieses Datenbanksystem alle Tabellen ohne Schema-Trennung in die Datenbank geladen.

Der Vorteil des Hochladens von Datenbank-Inhalten besteht darin, dass man in einem Datenbanksystem die ganze Ausdruckskraft von SQL zur Verfügung hat, um Abfragen zu formulieren.

### Metadaten exportieren ...

Mit diesem Befehl kann man die SIARD-Metadaten als XML-Datei exportieren.

Theoretisch kann man die SIARD-Datei auch einfach mit *PKZIP* öffnen und die Metadaten aus dem Verzeichnis *header* exportieren. Aber nicht alle Benutzer haben eine *PKZIP*-Lizenz.

## SIARD Suite

Eine solche würde aber benötigt, weil andere ZIP-Programme (Info-ZIP, Microsoft Windows XP ZIP-Ordner etc.) den 64-bit-Standard des ZIP-Dateiformats (noch?) nicht unterstützen. Da SIARD-Dateien naturgemäss sehr gross werden können, war es notwendig, diesen 64-bit-Standard zu verwenden, welcher ermöglicht, ZIP-Dateien zu speichern, die grösser als 4 GB sind und mehr als 66'000 Dateien enthalten. An Stelle von PKZIP kann man auch die Open Source Anwendung *Zip64File*<sup>19</sup> verwenden, um ZIP64-Dateien zu entpacken.

Die exportierte Datei *metadata.xml* kann mit Hilfe einfacher XSLT-Steuerdateien in einen Bericht über die Datenbank-Metadaten der SIARD-Datei verwandelt werden. Wenn man die Datei *metadata.xsl* im Ordner *doc* der SIARD-Installation in dasselbe Verzeichnis wie die exportierte XML-Datei *metadata.xml* kopiert, und dann diese etwa mit einem der moderneren Browser öffnet, werden die Metadaten in einem solchen Berichtformat präsentiert.

Die Beschreibung der Metadatenstruktur und des SIARD-Dateiformats findet man im Dokument *SIARD Format\_de.pdf*, ihre formale Definition in der XML Schemadefinition *metadata.xsd* im Verzeichnis *doc* der SIARD-Installation.

### *Zuletzt benutzte Dateien*

Hier findet man die zuletzt geöffneten Dateien für den schnellen Zugriff. Diese sind in der Datei *siard.properties* im Ordner *java* im Home-Verzeichnis des Benutzers gespeichert.

### *Exit*

Unter dem Menü *Datei* findet man zuunterst den Menüpunkt *Exit*. Dieser ermöglicht das Beenden der Anwendung.

### 7.3.2 *Bearbeiten*

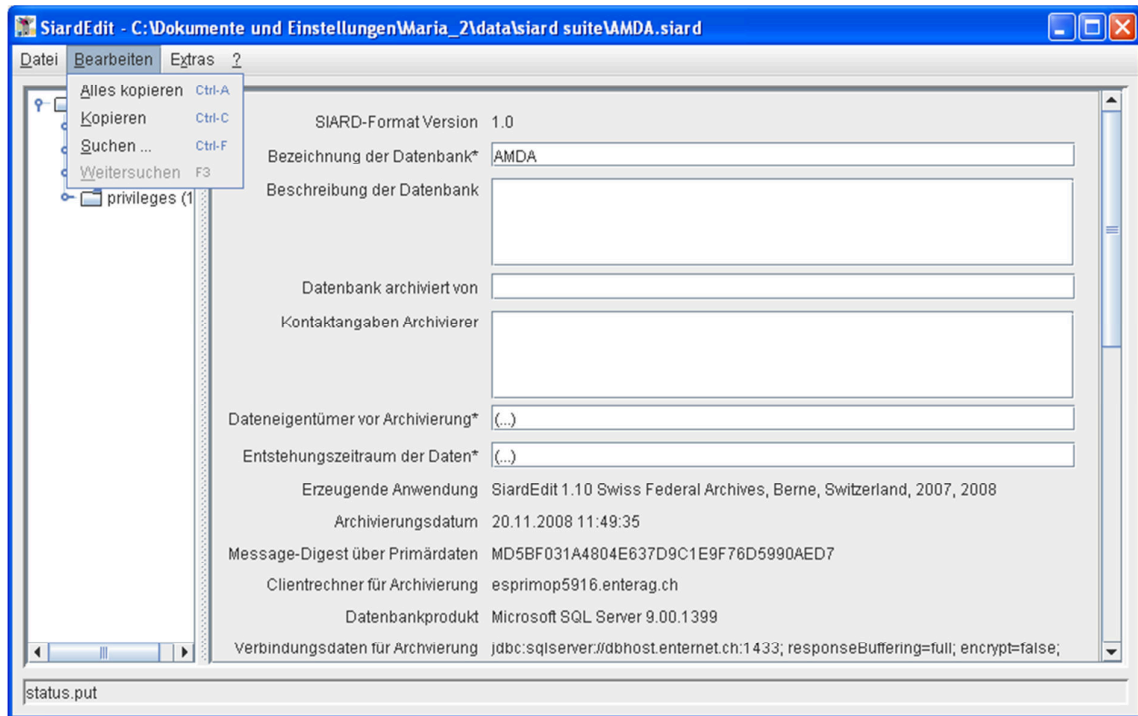
#### *Alles kopieren (Ctrl-A)*

Unter dem Menü *Bearbeiten* findet man den Menüpunkt *Alles kopieren (Ctrl-A)*.

---

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

## SIARD Suite



Mit dessen Hilfe kann man die Tabelle rechts unten als tabulator-getrennten Text in die Zwischenablage speichern. Diese fügt man mit Vorteil in einer Tabellenkalkulations-Software für die weitere Bearbeitung ein.

### *Kopieren (Ctrl-C)*

Mit diesem Befehl kann man die rosa markierte aktuelle Zeile der Tabelle rechts unten tabulator-getrennt in die Zwischenablage übernehmen, um diese in einer anderen Anwendung einfügen zu können.

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

Mit diesem Befehl kann man nach einer beliebigen Zeichenkette in den Metadaten suchen.

### *Weitersuchen (F3)*

Mit diesem Befehl kann nach weiteren Vorkommnissen der gesuchten Zeichenkette in den Metadaten gesucht werden.

## 7.3.3 Extras

### *Installieren*

Im Menü *Extras* findet man auch den Menüpunkt *Installieren*, der das nachträgliche Installieren von SIARD auf dem PC des Benutzers ermöglicht.

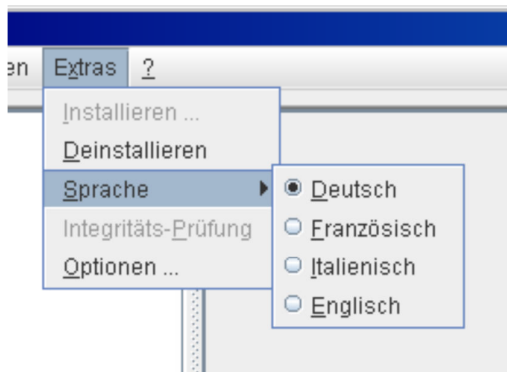
### *Deinstallieren*

Im Menü *Extras* kann man mit diesem Menüpunkt die lokale Installation von SIARD entfernen.

### *Sprache*

Unter dem Menü *Extras* mit dem Menüpunkt *Sprache* kann der Benutzer die gewünschte Benutzersprache auswählen.

## SIARD Suite



Das *SiardEdit*-Hauptfenster passt sich sofort an die gewählte Sprache an. Die Benutzersprache wird beim Verlassen des Programms in der Datei `<home>/java/siard suite.properties` gespeichert. Dabei ist mit `<home>` unter LINUX das Verzeichnis `$HOME`, und unter Windows XP das Verzeichnis `C:\Dokumente und Einstellungen\<userid>` und unter Windows 7 das Verzeichnis `C:\Benutzer\<userid>` gemeint. (Mit `<userid>` ist der Name gemeint, mit dem sich der Benutzer bei Windows angemeldet hat.)

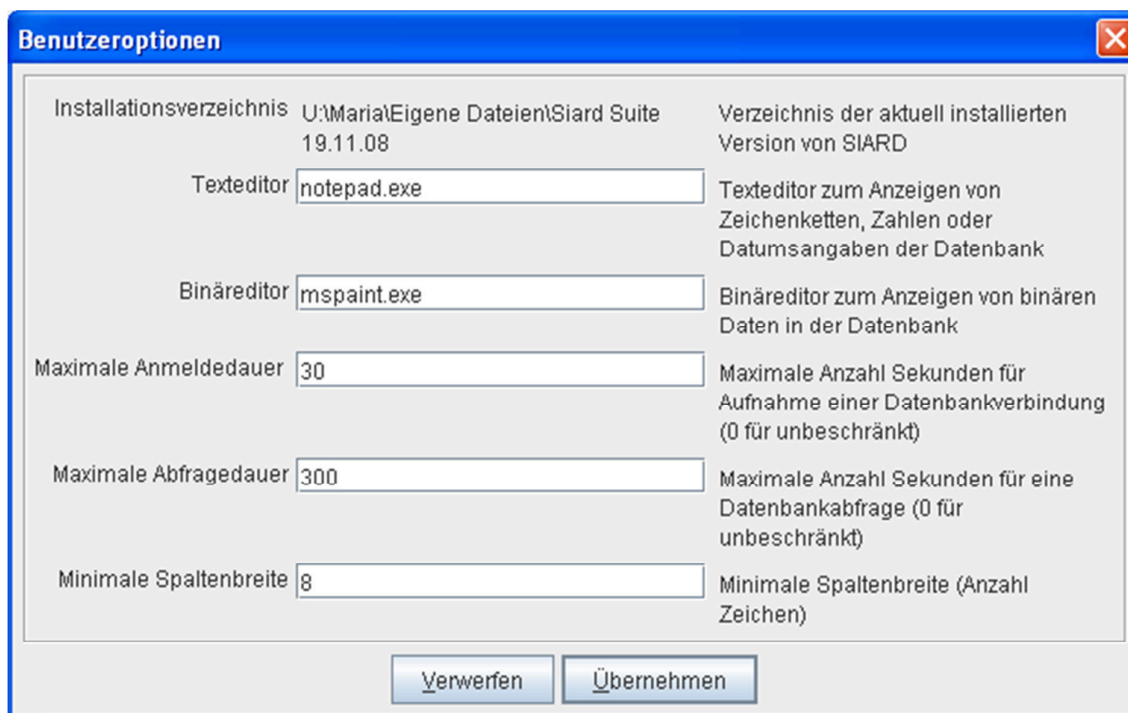
Expertenanwender können auch ein anderes Verzeichnis erzwingen, indem sie beim Aufruf von JAVA “`-Duser.dir=<anderesVerzeichnis>`” auf die Befehlszeile schreiben.

### *Integritätsprüfung*

Im Menü Extras kann man mit dem Befehl Integritätsprüfung verifizieren, dass die Primärdaten nicht verändert wurden, seit ihre Signatur (MD5 oder SHA-1) berechnet und in den Metadaten gespeichert wurde. Von *SIARD Suite* werden die Primärdaten nur einmal unmittelbar nach dem erfolgreichen Download aus einer Datenbank mit einer solchen Signatur „versiegelt“.

### *Optionen*

Im Menü Extras findet man den Befehl Optionen. Hier kann man einige benutzerspezifische Einstellungen eingeben:



Die Einträge zum *Texteditor* und zum *Binäreditor* werden verwendet, wenn auf einer Primärdatenzone vom Typ Zeichenkette bzw. vom Typ Binärdaten doppelgeklickt wird. Der Zelleninhalt wird in einer temporären Datei gespeichert und mit der hier angeführten Anwendung geöffnet. Besonders der Binäreditor muss jeweils entsprechend dem Typ der CLOB-Daten sinngemäss angepasst werden.

Die *maximale Anmeldedauer* steuert, nach wievielen Sekunden ein Anmeldeversuch bei einer Datenbank abgebrochen wird. Sie sollte nur dann abgeändert werden, wenn man Grund zur Annahme hat, dass das Netzwerk oder die Datenbank sehr langsam reagieren.

Die *maximale Abfragedauer* sollte man für das Herunterladen sehr grosser Datenbanken auf Null setzen, da die Abfragen in diesem Fall länger als die 5 Minuten der Grundeinstellung dauern können.

Die *minimale Spaltenbreite* steuert die Darstellung der Tabellen im rechten unteren Bereich des Fensters.

## 7.3.4 ?

### Hilfe

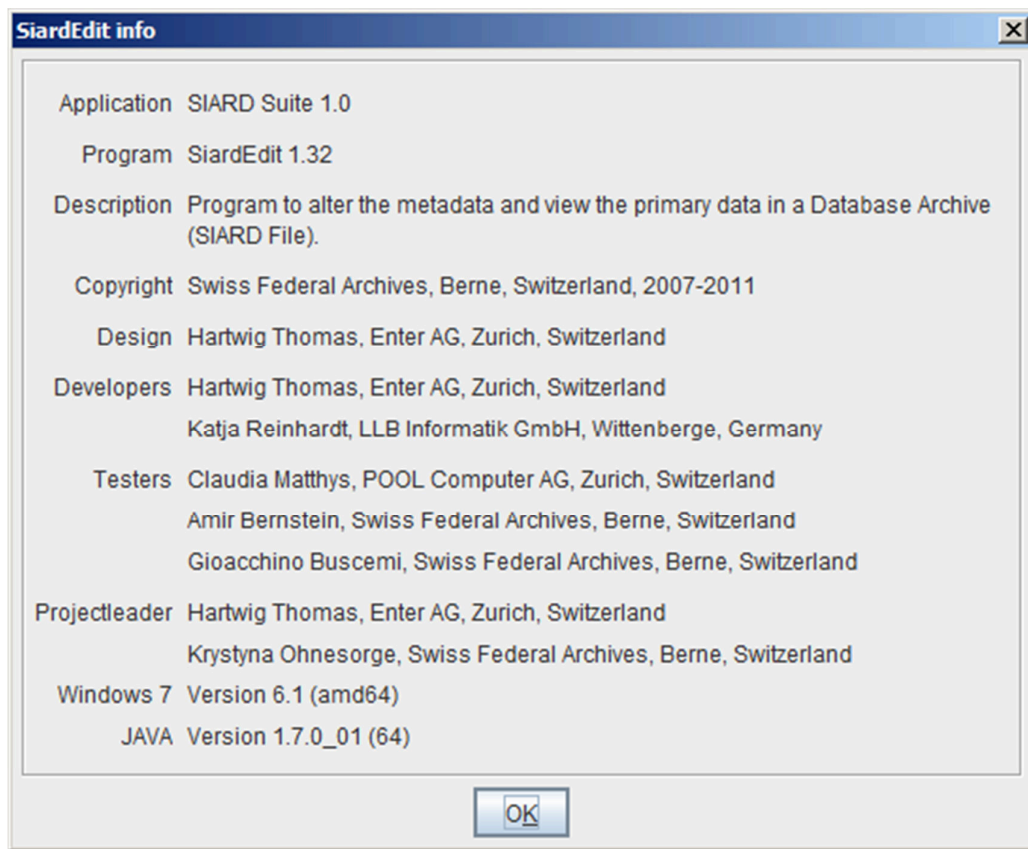
Beim Aufruf dieses Menüpunkts wird das Inhaltsverzeichnis des Hilfesystems angezeigt.

### Info

Im Menü ? findet sich die Urheberrechtsnotiz für *SiardEdit*:



## SIARD Suite



Ausserdem werden hier die Architektur des Betriebssystems und der JAVA Runtime angezeigt.

### 7.3.5 Kontextabhängige Hilfe

Grundsätzlich steht den Benutzern überall eine kontextabhängige Hilfe zur Verfügung, die aufgerufen wird, wenn man F1 drückt. Die darauf angezeigte Hilfe bezieht sich auf das Element oder dasjenige Fenster, welches gerade Tastatur- oder Mauseingaben entgegennimmt.

Wenn der Menüpunkt Hilfe im Menü ? gewählt wird, kommt man zu dieser Erklärung der Hilfefunktion.

Im Hilfsmenü kann man die Einzelseiten vom Inhaltsverzeichnis aus aufrufen oder nach Stichwörtern suchen. Eine angezeigte Seite wird jeweils mit dem Inhaltsverzeichnis synchronisiert.

## 8 SiardFromDb

Das Programm *SiardFromDb* ist ein Befehlszeilenprogramm, welches eine Datenbank in ein SIARD-Archiv extrahiert. Mit *SiardFromDb* kann man aus einer Datenbank

- ein „volles“ SIARD-Archiv (Metadaten und Primärdaten) herstellen, welches auf der Datenbank basiert (option "full"),
- ein „leeres“ SIARD-Archiv (ohne Primärdaten) herstellen mit einer Schemadefinition, welche auf der Datenbank basiert (option "empty").

### 8.1 Aufruf

```
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>]
```

Als *<siardpath>* ist das Verzeichnis einzugeben, wo SIARD installiert wurde. Im Unterverzeichnis *bin* des Installationsverzeichnisses befindet sich die Datei *SiardEdit.jar* mit der Klasse *ch.admin.bar.siard.SiardFromDb*, deren *main()* hier mit dem Befehl *java* (unter Windows nimmt man besser *javaw*) ausgeführt wird.

Wenn die Option *-h* (*help*) auf der Befehlszeile steht, wird die Aufrufsyntax angezeigt.

Da der Aufruf recht viele Parameter enthält, kann man ihn mit einer Konfigurationsdatei vereinfachen, in der alle Parameter mit Ausnahme des Passworts gespeichert sind. Wenn diese vorhanden ist, reduziert sich der Aufruf auf:

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

Wenn alle Parameter und zusätzlich eine Konfigurationsdatei angegeben wurden, werden die Parameter in der Konfigurationsdatei durch diejenigen auf der Befehlszeile ergänzt. Dies vereinfacht das Wiederholen von SIARD-Downloads.

Wenn die Option *-e* (*empty*) angegeben wird, werden nur die Metadaten archiviert.

### 8.2 Aufrufparameter

Parameter	Bedeutung
<data source>	Access, ODBC, Oracle, "SQL Server", MySQL oder "DB/2"
<database name>	Name der Datenbank je nach Verbindung verschieden strukturiert: Access: Dateiname der MS Access Datei ODBC (abgekündigt): konfigurierter Datasourcenname (DSN) der ODBC-Verbindung Oracle: z.B. dbhost.enternet.ch:1521:SIARD1 SQL Server: z.B. dbhost.enternet.ch:1433/SIARD MySQL: z.B. dbhost.enternet.ch:3306/SIARD DB/2: z.B. dbhost.enternet.ch:50000/SIARD

## SIARD Suite

<i>Parameter</i>	<i>Bedeutung</i>
<database user>	UserId für die Download-Session
<database password>	Passwort für die Download-Session
<siard file>	Dateiname des zu erzeugenden SIARD-Archivs
<query timeout>	Timeout für SQL Abfragen in Sekunden (0 für unbeschränkt)
<template file>	SIARD-Datei, die als Vorlage benutzt werden soll, um manuell einzugebende Felder zu füllen

### 8.3 Bemerkungen

Das Programm *SiardFromDb* sollte auf einem „Snapshot“ der Datenbank ausgeführt werden, welcher sich während der Archivierung nicht ändert.

Die Archivierung ist entweder als Ganzes erfolgreich oder scheitert gänzlich.

Bei grossen Tabellen empfiehlt es sich, vorgängig eine leere SIARD-Datei herunterzuladen. So erhält man Einblick in alle Metadaten und die Tabellengrössen. Diese helfen, die benötigte Download-Zeit abzuschätzen. Ausserdem sollte man für grosse Tabellen jeweils die Option `-q 0` angeben, weil nicht abschätzbar ist, wieviele Sekunden eine Grössenabfrage auf einer grossen Tabelle benötigt.

Die Umwandlung der TIMES und TIMESTAMPS in der Datenbank ist abhängig von der lokalen Zeitzone. Wenn man etwa die Zeit 15:30 in Zürich abspeichert, wird in der XML-Datei die UTC-Zeit 14:30 (im Winter!) abgespeichert. Sind die Zeiten der Datenbank unverändert als UTC-Zeiten zu interpretieren, muss man *SiardFromDb* mit der Option

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

starten.

### 8.4 Vorbereitung Datenbank

Je nach Datenbanktyp muss man die Datenbank für die Archivierung verschieden vorbereiten. Man muss sich darüber im Klaren sein, dass die Wahl der für den Download verwendeten Datenbank-UserId über den Umfang der zu archivierenden Datenbank entscheidet. Die resultierende SIARD-Datei wird diejenigen Datenbankobjekte enthalten, welche für den Datenbank-User sichtbar sind. Aus diesem Grund ist es normalerweise keine gute Idee, einen Datenbank-Administrator (*SYS* oder *SYSTEM* für Oracle, *dbo* für SQL Server, *mysql* oder *information\_schema* für MySQL, *dbinst1* oder *SYSIBM* für DB/2) für den Download zu benutzen. Diese Benutzer haben normalerweise zu viel Leserechte auch Systemobjekte, die nicht zu archivieren sind. Für MS Access Datenbanken ist dagegen die Archivierung möglichst des gesamten Inhalts der Datenbank erwünscht. Darum wird in diesem Fall die Verwendung von *Admin* als Datenbank-UserId empfohlen.

Die Tatsache, dass die Wahl der Datenbank-UserId den Umfang der archivierten Datenbank bestimmt kann dazu verwendet werden, den Umfang der Daten, die in die SIARD-Datei gelangen, genau zu kontrollieren: Man erzeugt einen neuen Archivierungs-User in der Datenbank und räumt ihm Lesezugriff auf genau die Objekte ein, die archiviert werden sollen.

#### 8.4.1 Vorbereitung „pure JAVA“-Zugriff auf MS Access Datei

In diesem Fall sind normalerweise keine Vorbereitungsschritte nötig. Die Datei muss einfach im Dateisystem lesbar sein.

## SIARD Suite

Die allermeisten MS Access Datenbanken kann man mit UserId *Admin* und Passwort „“ öffnen. Um den Zugang zu besser geschützten Datenbanken zu erhalten, muss man die Dokumentation des Entwicklers konsultieren.

### 8.4.2 Vorbereitung ODBC-Zugriff auf MS Access Datei (abgekündigt)

#### *Accdb- und Mdb-Dateien*

MS Access 2007 unterstützt zwei Dateiformate: das „alte“ Format der Dateien mit Endung *.mdb* und ein „neues“ mit Endung *.accdb*. Accdb-Dateien unterstützen neue Datentypen – insbesondere mehrwertige Felder und „Anlage“-Felder (ebenfalls mehrwertig). Accdb-Dateien verbieten ausserdem den Zugriff auf Systemtabellen und verunmöglichen so den Zugang zu den darin gespeicherten Metadaten über die ODBC- (und somit auch über die JDBC-) Schnittstelle. Für Mdb-Dateien konnte man Zugriff auf die Systemtabellen *MSysObjects*, *MSysRelationships* und *MSysQueries* manuell einrichten. Dieser machte es möglich, dass *SIARD Suite* auf zusätzliche Metadaten zugreifen konnte, die nicht über die JDBC-Klasse *DatabaseMetaData* zugänglich sind.

*SIARD Suite* behandelt diese neue Sachlage folgendermassen: Falls auf die *MSys*-Tabellen lesend zugegriffen werden kann, wird ihre Zusatzinformation im SIARD Archiv gespeichert. Andernfalls können weder Fremdschlüssel noch Views gespeichert werden. Für Archivierungszwecke ist dieser Verlust an Metainformation wohl oft verschmerzbar, da keine Primärdaten betroffen sind.

Problematischer ist der neue Datentyp *Anlage* in den Accdb-Dateien. Microsoft hat festgelegt, dass deren Inhalt in keiner Weise über die ODBC- (und somit die JDBC-) Schnittstelle ausgelesen werden kann. Die aktuelle Version des SIARD Formats könnte solche mehrwertigen BLOB-Felder auch nicht angemessen archivieren. Aus diesem Grund übernimmt *SIARD Suite* – wie auch die Microsoft-eigenen Migrations-Werkzeuge – solche Felder beim Download nur als Textfelder, welche Listen von Dateinamen der Anlagen enthalten. Der Inhalt der Anlagen jedoch geht dabei verloren! Unglücklicherweise hindert Microsoft die *SIARD Suite* – und alle anderen ODBC- oder JDBC-Clients daran, diesen Verlust auch nur zu detektieren. Es bleibt also Aufgabe des Archivierers, Vorkehrungen zu treffen, dass Anlagen in Accdb-Dateien separat archiviert werden.

Falls Mdb-Dateien archiviert werden müssen, hat der Archivierer die Wahl, ob er nur diejenigen Metadaten archivieren will, die bei einer Accdb-Datei archiviert würden – ohne Fremdschlüssel und Views –, oder ob er die Mdb-Datei vorgängig so vorbereiten will, dass die Systemtabellen für den *Admin*-Benutzer zugänglich sind.

#### *Architekturfragen (32-bit / 64-bit)*

Wenn *SIARD Suite* mit einer 32-bit JAVA Runtime gestartet wurde, kann nur auf 32-bit ODBC-Treiber zugegriffen werden. Wenn man auf einem 64-bit Windows ein 64-bit MS Access installiert hat, richtet man mit dem Standard-ODBC-Administrator (*C:\Windows\system32\odbcad32.exe*) nur 64-bit-ODBC-Zugänge ein. Diese sind also für *SIARD Suite* nur zugänglich, wenn es mit einer 64-bit JAVA Runtime gestartet wurde. Wenn man einen 32-bit-ODBC-Zugang einrichten möchte, muss man den 32-bit ODBC-Administrator (*C:\Windows\SysWOW64\odbcad32.exe*) benutzen. Diese 32-bit-ODBC-Zugänge sind dann für *SIARD Suite* nur zugänglich, wenn es mit einer 32-bit JAVA Runtime gestartet wurde.

Die Batchdateien *siardedit.cmd*, *siardfromdb.cmd* und *siardtodb.cmd* (und der von der Installation eingerichtete Shortcut auf dem Desktop) benutzen die 64-bit JAVA Runtime, sofern eine solche installiert ist. Wenn gleichzeitig eine 32-bit JAVA Runtime verfügbar ist, kann man *SIARD Suite* manuell mit dieser starten, indem man die Batchdateien mit dem fakultativen Parameter */32* aufruft. Wenn beim Download einer MS Access Datei mit dem 64-bit-ODBC Probleme auftreten, lohnt es sich, dasselbe mit dem 32-bit-ODBC auszuprobieren. Momentan (2014) hat der 64-bit-ODBC-Treiber noch viele Bugs.

## SIARD Suite

Systemtabellen von Mdb-Dateien zugänglich machen

1. Die Systemtabellen *MSysObjects*, *MSysRelationships* und *MSysQueries* für den Archivierungsbenutzer (meistens *Admin*) lesbar machen.

MS Access vor 2007:

Zu diesem Zweck unter *Extras / Optionen / Ansicht* die Systemobjekte sichtbar machen, und darauf unter *Extras / Sicherheit / Benutzer- und Gruppenberechtigungen* die Tabellen *MSysObjects*, *MSysRelationships* und *MSysQueries* für den Benutzer *Admin* lesbar machen.

MS Access 2007:

Unter *Office Schaltfläche / Access-Optionen / Aktuelle Datenbank / Navigationsoptionen...* / *Systemobjekte anzeigen* die Systemobjekte sichtbar machen, und darauf unter *Datenbanktools / Verwalten / Benutzer und Berechtigungen / Benutzer- und Gruppenberechtigungen* die Tabellen *MSysObjects*, *MSysRelationships* und *MSysQueries* für den Benutzer *Admin* lesbar machen.

Die ganze Gruppe *Datenbanktools / Verwalten / Benutzer und Berechtigungen* fehlt, wenn die offene Datei eine Accdb-Datei ist. In diesem Fall muss man im „Immediate Window“ (aufzurufen mittels Ctrl-g) die Befehle

```
CurrentProject.Connection.Execute "GRANT SELECT ON MSysObjects TO Admin;"
CurrentProject.Connection.Execute "GRANT SELECT ON MSysRelationships TO Admin;"
CurrentProject.Connection.Execute "GRANT SELECT ON MSysQueries TO Admin;"
eingeben.
```

MS Access 2010 ff:

Wie MS Access 2007, aber statt *Office Schaltfläche* den Ribbon *Datei* benutzen.

2. ODBC-Verbindung einrichten, welche auf die MDB-Datei verweist (s. Stichwort ODBC in der Windows-Hilfe unter *Start / Hilfe* für Instruktionen, wie eine ODBC-Verbindung über *Start / Systemsteuerung / Verwaltung / Datenquellen(ODBC)* einzurichten ist), und deren erweiterte Eigenschaft (unter *Erweitert ...*) *ExtendedAnsiSQL=1* setzen. Die ODBC-Verbindung kann von Nicht-Administratoren als *Benutzer DSN* eingerichtet werden.

*Automatische Konfiguration und Konversion*

Ein quelloffenes Befehlsskript. *accesstosiard.js*<sup>20</sup>, welches als Ergänzung zur *SIARD Suite* entwickelt wurde, nimmt dem Benutzer alle diese Konfigurationsarbeiten ab und konvertiert einfach eine ACCDB- oder MDB-Datei in eine SIARD-Datei.

### 8.4.3 Vorbereitung Oracle

Eine Oracle-Datenbank konfiguriert man als Datenbankadministrator (DBA) am besten für den Export, indem man einen neuen Archivierungsbenutzer erzeugt und diesem die Leserechte auf alle zu exportierenden Datenbank-Objekte einräumt. Mit Hilfe dieses neuen Benutzerkennung prüft man anhand von Beispielabfragen, ob alle relevanten Referenzen sichtbar sind. Es ist sinnvoll, die Beispielabfragen für zukünftige Archivbenutzer zu kommentieren und zu dokumentieren.

---

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

## SIARD Suite

### *Troubleshooting Oracle*

Damit es gelingt, mit *SIARD Suite* eine Verbindung zur gewünschten Oracle-Datenbank herzustellen, müssen Berechtigungen, Firewalls etc. so eingerichtet sein, dass man mit folgendem Oracle-Befehl erfolgreich eine Verbindung aufbauen kann:

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

Dann ist in SIARD Suite der Datenbankname *<hostname>:<port>:<service>* zu verwenden.

#### 8.4.4 Vorbereitung SQL Server

Eine SQL-Server-Datenbank konfiguriert man als Datenbankadministrator (DBA) am besten für den Export, indem man einen neuen Archivierungsbenutzer erzeugt, diesem die Leserechte auf die zu exportierende Datenbank einräumt.

Wenn man dem Archivierungsbenutzer keine Eigentümer-Eigenschaften an den zu archivierenden Schemas zugesteht, kann dieser die bestehenden Privilegien nicht lesen und die gespeicherte Information über Benutzer und Rollen kann unvollständig sein.

Wenn der Archivbenutzer SELECT-Rechte erhält, so wird immer automatisch auch – wohl meistens unerwünschterweise – das Schema „dbo“ archiviert. Die besten Einstellungen für den Archivierungsbenutzer sind also:

Unter Datenbank-Permissions nur CONNECT-Recht einräumen. Unter User-Eigenschaften Archivierungsbenutzer als Eigentümer aller zu archivierenden Schemas definieren. (Achtung: dabei werden die bisherigen Eigentümer „enteignet“. Man muss diese Einstellung nach der Archivierung allenfalls wieder rückgängig machen.)

### *Troubleshooting SQL Server*

Damit es gelingt, mit *SIARD Suite* eine Verbindung zur gewünschten SQL Server-Datenbank herzustellen, müssen Berechtigungen, Firewalls etc. so eingerichtet sein, dass man mit folgendem SQL Server-Befehl erfolgreich eine Verbindung aufbauen kann:

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

Dann ist in SIARD Suite der Datenbankname *<hostname>:<port>/<database>* zu verwenden.

#### 8.4.5 Vorbereitung MySQL

Eine MySQL-Datenbank konfiguriert man als Datenbankadministrator (DBA) am besten für den Export, indem man einen neuen Archivierungsbenutzer erzeugt und diesem die Leserechte auf alle zu exportierenden Datenbank-Objekte einräumt.

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

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

(MySQL behandelt Datenbank und Schema als austauschbare Begriffe. Statt <SCHEMA> ist hier etwa der Datenbankname einzugeben.)

Mit Hilfe dieser neuen Benutzerkennung prüft man anhand von Beispielabfragen, ob alle relevanten Referenzen sichtbar sind. Es ist sinnvoll, die Beispielabfragen für zukünftige Archivbenutzer zu kommentieren und zu dokumentieren.

### *Troubleshooting MySQL*

Damit es gelingt, mit *SIARD Suite* eine Verbindung zur gewünschten MySQL-Datenbank herzustellen, müssen Berechtigungen, Firewalls etc. so eingerichtet sein, dass man mit folgendem MySQL-Befehl erfolgreich eine Verbindung aufbauen kann:

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

## SIARD Suite

Dann ist in *SIARD Suite* der Datenbankname `<hostname>:<port>/<database>` zu verwenden. (Achtung: Kein Abstand zwischen `-p` und `<password>!`)

Um die Zugriffsrechte eines Benutzers einer MySQL-Daten zu verstehen, muss man vier Tabellen im `INFORMATION_SCHEMA` beachten: `USER_PRIVILEGES` (globale Zugriffsrechte), `SCHEMA_PRIVILEGES` (Zugriffsrechte auf Schemas), `TABLE_PRIVILEGES` (Zugriffsrechte auf Tabellen) und `COLUMN_PRIVILEGES` (Zugriffsrechte auf Spalten).

### 8.4.6 Vorbereitung DB/2

DB/2 kennt als Benutzer nur die OS-Benutzer. Man muss also im OS einen geeigneten Archivbenutzer einrichten und diesem mit `GRANT ON DATABASE` alle Rechte an der zu archivierenden Datenbank einräumen.

Achtung: Unter LINUX darf die Benutzerkennung nur Kleinbuchstaben enthalten. Sie darf höchstens acht Zeichen lang sein!

#### Troubleshooting DB/2

Auf der Arbeitsstation muss der DB/2-Client installiert sein. Unter Windows muss man dann *db2cmd* ausführen, um Zugriff auf die Befehlszeilenumgebung *db2* zu haben. Unter LINUX hat man diesen Zugriff direkt.

Mittels der Befehlszeilenumgebung *db2* muss man dann einen TCP/IP-Knoten mittels `CATALOG` einrichten und dann mittels `CONNECT` eine Verbindung aufbauen.

Zu diesem Zweck startet man das DB/2 Befehlszeilenfenster oder die *db2* Befehlszeilenumgebung und gibt Folgendes ein:

#### a) LIST NODE DIRECTORY

Dies sollte die auf dem Client registrierten Nodes auflisten. Falls kein Node für die gewünschte DB/2-Instanz existiert, muss man zuerst einen schaffen mit dem Befehl:

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

Hierbei nimmt man als `<nodename>` am besten den ersten Teil des DNS-Namen des Servers, der auch als Hostname bezeichnet wird. Als `<dnsname>` ist der volle DNS-Name des Servers anzugeben. Als `<port>` die Portnummer, auf welcher der DB/2-Listener Anfragen erwartet (Standard: 50000). Wenn man jetzt noch einmal den Befehl `LIST NODE DIRECTORY` ausführt, muss der neue `<nodename>` erscheinen.

#### b) ATTACH TO <nodename> USER <siarduser>

Hierbei ist `<nodename>` der Knotenname, welcher die DB/2-Instanz auf diesem Client bezeichnet. Als `<siarduser>` ist die UserId des Archivbenutzers der Datenbank auf dieser Instanz einzugeben. Auf den `ATTACH`-Befehl hin, wird man nach dem Passwort für diesen Benutzer gefragt.

#### c) LIST DATABASE DIRECTORY

Damit werden alle Datenbanken auf der DB/2-Instanz angezeigt, zu der man sich soeben verbunden hat.

#### d) CONNECT TO <database> USER <siarduser>

Als `<siarduser>` ist die UserId des Archivbenutzers dieser Datenbank einzugeben. Auf den `CONNECT`-Befehl hin, wird man nach dem Passwort für diesen Benutzer gefragt. Damit beziehen sich die weiteren Befehle (SQL Statements) auf diese Datenbank.

## **SIARD Suite**

Falls dies erfolgreich war, kann man den Download mit *SIARD Suite* mit <dnsname>:<port>/<database> und dem Archivbenutzer <siarduser> starten.



## 9 SiardToDb

Das Programm *SiardToDb* ist ein Befehlszeilenprogramm, welches ein SIARD-Archiv zu Recherchezwecken in eine Datenbank-Instanz lädt.

### 9.1 Aufruf

```
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>]*
```

Als *<siardpath>* ist das Verzeichnis einzugeben, wo SIARD installiert wurde. Im Unterverzeichnis *bin* des Installationsverzeichnisses befindet sich die Datei *SiardEdit.jar* mit der Klasse *ch.admin.bar.siard.SiardToDb*, deren *main()* hier mit dem Befehl *java* (unter Windows nimmt man besser *javaw*) ausgeführt wird.

Wenn die Option *-h* (*help*) auf der Befehlszeile steht, wird die Aufrufsyntax angezeigt.

Da der Aufruf recht viele Parameter enthält, kann man ihn mit einer Konfigurationsdatei vereinfachen, in der alle Parameter mit Ausnahme des Passworts gespeichert sind. Wenn diese vorhanden ist, reduziert sich der Aufruf auf:

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

Wenn alle Parameter und zusätzlich eine Konfigurationsdatei angegeben wurden, werden die Parameter in der Konfigurationsdatei durch diejenigen auf der Befehlszeile ergänzt. Dies vereinfacht das Wiederholen von SIARD-Uploads.

### 9.2 Aufrufparameter

Parameter	Bedeutung
<data source>	Access, ODBC, Oracle, "SQL Server", MySQL oder "DB/2"
<database name>	Name der Datenbank je nach Verbindung verschieden strukturiert: Access: Dateiname der MS Access Datei ODBC (abgekündigt): konfigurierter Datasourcenname (DSN) der ODBC-Verbindung Oracle: z.B. dbhost.enternet.ch:1521:SIARD1 SQL Server: z.B. dbhost.enternet.ch:1433/SIARD MySQL: z.B. dbhost.enternet.ch:3306/SIARD DB/2: z.B. dbhost.enternet.ch:50000/SIARD
<database user>	UserId für die Upload-Session
<database password>	Passwort für die Upload-Session
<siard file>	Dateiname des hinaufzuladenden SIARD-Archivs
<query timeout>	Timeout für SQL Abfragen in Sekunden (0 für unbeschränkt)

<i>Parameter</i>	<i>Bedeutung</i>
<schema>	Schemaname im SIARD Archiv (beliebig oft zusammen mit darauffolgendem <mappedschema> anzugeben)
<mappedschema>	Schemaname, der für den Upload zu verwenden ist.  Für alle Schemas, denen kein anderer Schemaname zugeordnet wurde, wird ihr eigener Name verwendet.

### 9.3 Bemerkungen

Da die älteren Datenbanken nicht SQL:1999-konform sind, lässt sich etlicher manueller Konfigurationsaufwand als Vorbereitung für den Upload leider nicht vermeiden. Unter MS Access gibt es keine Schema-Objekte. Unter SQL Server 2000 und Oracle 8 und 9 sind Benutzer und Schema-Objekte nicht getrennt.

Beim Hinaufladen werden nur Tabellen erzeugt. Alle anderen Datenbank-Objekte werden nicht generiert. Ausserdem werden beim Hinaufladen evtl. gewisse Abstriche gemacht. Bei MS Access landen alle Tabellen in derselben MDB. Bei Oracle werden alle Namen, die länger sind als 30 Zeichen gekürzt. Zur Vermeidung von Kollisionen werden Tabellen- und Spaltennamen um einen Zähler ergänzt. (So wird etwa aus der Tabelle „Ein viel zu langer Tabellenname für Oracle“ zum Beispiel „Ein viel zu langer Tabellen01“.)

Wo die maximale Präzision und die maximale Anzahl Dezimalstellen (etwa unter MS Access) kleiner ist als benötigt, werden die Werte mit geringerer Präzision hinaufgeladen. SIARD behilft sich so gut es eben im Zieldatenbanksystem möglich ist. Wenn man die Metadaten der Datenbank mittels *SiardEdit* konsultiert, wird man die korrekten Bezeichnungen und Werte zuordnen können.

Die Umwandlung der TIMES und TIMESTAMPS in der Datenbank ist abhängig von der lokalen Zeitzone. Ist etwa die (UTC-) Zeit 14:30 in der SIARD Datei Zürich abgespeichert, wird die lokale Zeit 15:30 (im Winter!) in die Datenbank geladen. Sind die Zeiten der SIARD-Datei unverändert als UTC-Zeiten in die Datenbank zu laden, muss man *SiardToDb* mit der Option

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

starten.

### 9.4 Vorbereitungen Datenbank

Je nach Datenbanktyp muss man die Datenbank für den Upload verschieden vorbereiten.

#### 9.4.1 Vorbereitung „pure JAVA“-Zugriff auf MS Access Datei

Um eine SIARD-Datei in eine MS Access Datenbank laden zu können, muss man mittels MS Access eine leere Mdb- oder Accdb-Datei erzeugen.

#### 9.4.2 Vorbereitung ODBC-Zugriff auf MS Access Datei (abgekündigt)

Um eine SIARD-Datei in eine MS Access-Datenbank laden zu können, muss man folgendermassen vorgehen:

1. Mittels MS Access eine leere Mdb- oder Accdb-Datei erzeugen.
2. Die Systemtabellen *MSysObjects*, *MSysRelationships* und *MSysQueries* für den Archivierungsbenutzer *Admin* lesbar machen.

MS Access vor 2007:

Zu diesem Zweck unter *Extras / Optionen / Ansicht* die Systemobjekte sichtbar machen, und darauf unter *Extras / Sicherheit / Benutzer- und Gruppenberechtigungen*

## SIARD Suite

die Tabellen *MSysObjects*, *MSysRelationships* und *MSysQueries* für den Benutzer *Admin* lesbar machen.

MS Access 2007:

Unter *Office Schaltfläche / Access-Optionen / Aktuelle Datenbank / Navigationsoptionen...* / *Systemobjekte anzeigen* die Systemobjekte sichtbar machen, und darauf unter *Datenbanktools / Verwalten / Benutzer und Berechtigungen / Benutzer- und Gruppenberechtigungen* die Tabellen *MSysObjects*, *MSysRelationships* und *MSysQueries* für den Benutzer *Admin* lesbar machen.

Die ganze Gruppe *Datenbanktools / Verwalten / Benutzer und Berechtigungen* fehlt, wenn die offene Datei eine Accdb-Datei ist. In diesem Fall muss man im „Immediate Window“ (aufzurufen mittels Ctrl-g) die Befehle

```
CurrentProject.Connection.Execute "GRANT SELECT ON MSysObjects TO Admin;"
CurrentProject.Connection.Execute "GRANT SELECT ON MSysRelationships TO Admin;"
CurrentProject.Connection.Execute "GRANT SELECT ON MSysQueries TO Admin;"
eingeben.
```

MS Access 2010 ff:

Wie MS Access 2007, aber statt *Office Schaltfläche* den Ribbon *Datei* benutzen.

3. ODBC-Verbindung zur erzeugten leeren Upload-Datei einrichten (s. Stichwort ODBC in der Windows-Hilfe unter *Start / Hilfe* für Instruktionen, wie eine ODBC-Verbindung über *Start / Systemsteuerung / Verwaltung / Datenquellen (ODBC)* einzurichten ist), und deren erweiterte Eigenschaft (unter *Erweitert ...*) *ExtendedAnsiSQL=1* setzen. Die ODBC-Verbindung kann von Nicht-Administratoren als *Benutzer DSN* eingerichtet werden.

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

Wenn *SIARD Suite* mit einer 32-bit JAVA Runtime gestartet wurde, kann nur auf 32-bit ODBC-Treiber zugegriffen werden. Wenn man auf einem 64-bit Windows ein 64-bit MS Access installiert hat, richtet man mit dem Standard-ODBC-Administrator (*C:\Windows\system32\odbcad32.exe*) nur 64-bit-ODBC-Zugänge ein. Diese sind also für *SIARD Suite* nur zugänglich, wenn es mit einer 64-bit JAVA Runtime gestartet wurde. Wenn man einen 32-bit-ODBC-Zugang einrichten möchte, muss man den 32-bit ODBC-Administrator (*C:\Windows\SysWOW64\odbcad32.exe*) benutzen. Diese 32-bit-ODBC-Zugänge sind dann für *SIARD* nur zugänglich, wenn es mit einer 32-bit JAVA Runtime gestartet wurde.

Die Batchdateien *siardedit.cmd*, *siardfromdb.cmd* und *siardtodb.cmd* (und der von der Installation eingerichtete Shortcut auf dem Desktop) benutzen die 64-bit JAVA Runtime, sofern eine solche installiert ist. Wenn gleichzeitig eine 32-bit JAVA Runtime verfügbar ist, kann man *SIARD Suite* manuell mit dieser starten, indem man die Batchdateien mit dem fakultativen Parameter */32* aufruft. Wenn beim Upload einer MS Access Datei mit dem 64-bit-ODBC Probleme auftreten, lohnt es sich, dasselbe mit dem 32-bit-ODBC auszuprobieren. Momentan hat der 64-bit-ODBC-Treiber noch viele Bugs.

### 9.4.3 Vorbereitung Oracle

Um eine *SIARD*-Datei in eine Oracle-Datenbank laden zu können, muss man folgendermaßen vorgehen:

1. Als Systemadministrator für jedes Schema in der *SIARD*-Datei einen Benutzer (nur Grossbuchstaben, weil Oracle Benutzernamen mit Kleinbuchstaben nur fehlerhaft unterstützt) mit Hilfe des CREATE USER-Befehls erzeugen.
2. Einem Archivbenutzer (z.B. einem der Schemabbenutzer) das CONNECT und RESOURCE-Privileg einräumen.

## SIARD Suite

3. Dem Archivbenutzer alle Erzeugungs- und Schreibrechte auf alle Schemas einräumen.

Nur in Ausnahmefällen enthält die SIARD-Datei mehr als ein Schema. Meistens reduzieren sich die obigen Instruktionen also auf diese Zeilen:

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

Beim Upload werden evtl. lange Zeichenketten (z.B. VARCHAR(4000) auf UTF-16-Datenbanken, die maximal VARCHAR(2000) akzeptieren) in CLOBs umgewandelt. Ausserdem wird der Session-Parameter NLS\_LENGTH\_SEMANTICS für den Upload auf CHAR (Standard: BYTE) gesetzt.

### 9.4.4 Vorbereitung SQL Server

Um eine SIARD-Datei in eine SQL Server-Datenbank laden zu können, muss man folgendermassen vorgehen:

1. Als Systemadministrator eine leere Upload-Datenbank erzeugen. (Im Microsoft SQL Server Management Studio mittels *New Database ...* unter *Databases* oder mit Hilfe des CREATE DATABASE-Befehls).
2. Für den Archivbenutzer ein *Login* erzeugen. Diesem Archivbenutzer darf man keine Server-Privilegien erteilen.
3. Nur für SQL Server 2000: für jedes Schema in der Upload-Datenbank einen Benutzer erzeugen, welcher dem *Login* zugeordnet ist.
4. Die Upload-Datenbank räumt dem Archivbenutzer mindestens folgende Privilegien ein:
  - 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

## SIARD Suite

- s. Update
- t. View database state
- u. View definition

Unter SQL Server 2000 waren Benutzer und Schemas identisch. Unter SQL Server 2005/2010/2012 führt man für jedes Schema den Befehl

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

aus. In diesem Fall genügt es, dem Archivbenutzer nur die Create-Privilegien für die Datenbank einzuräumen.

### 9.4.5 Vorbereitung MySQL

Um eine SIARD-Datei in eine MySQL-Datenbank laden zu können, muss man folgendermaßen vorgehen:

1. Als Systemadministrator für jedes Schema in der SIARD-Datei eine Datenbank (=Schema) mit Hilfe des CREATE SCHEMA-Befehls erzeugen.
2. Dem Archivbenutzer alle Erzeugungs- und Schreibrechte auf alle Schemas einräumen.

Nur in Ausnahmefällen enthält die SIARD-Datei mehr als ein Schema. Meistens reduzieren sich die obigen Instruktionen also auf diese Zeilen:

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

### 9.4.6 Vorbereitung DB/2

Um eine SIARD-Datei in eine DB/2-Datenbank laden zu können, muss man folgendermaßen vorgehen:

Auf dem Betriebssystem, wo der DB/2-Service läuft, einen Archivuser (z.B.: *siardusr*) erzeugen.

Achtung: Wenn das Betriebssystem LINUX ist, darf die UserId des Archivusers nur aus Kleinbuchstaben bestehen und maximal acht Zeichen lang sein!

Als Datenbank-Administrator eine neue, leere Datenbank anlegen, welche den Inhalt der SIARD-Datei aufnehmen wird.

Zu diesem Zweck startet man das DB/2 Befehlszeilenfenster oder die DB/2 Befehlszeilenumgebung und gibt Folgendes ein:

#### a) LIST NODE DIRECTORY

Dies sollte die auf dem Client registrierten Nodes auflisten. Falls kein Node für die gewünschte DB/2-Instanz existiert, muss man zuerst einen schaffen mit dem Befehl:

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

Hierbei nimmt man als <nodename> am besten den ersten Teil des DNS-Namen des Servers, der auch als Hostname bezeichnet wird. Als <dnsname> ist der volle DNS-Name des Servers anzugeben. Als <port> die Portnummer, auf welcher der DB/2-Listener Anfragen erwartet (Standard: 50000). Wenn man jetzt noch einmal den Befehl „LIST NODE DIRECTORY“ ausführt, muss der neue <nodename> erscheinen.

## SIARD Suite

b) ATTACH TO <nodename> USER <dbadm>

Hierbei ist <nodename> der Knotenname, welcher die DB/2-Instanz auf diesem Client bezeichnet. Als <dbadm> ist die UserId eines Datenbank-Administrators dieser Instanz einzugeben. Auf den ATTACH-Befehl hin, wird man nach dem Passwort für diesen Datenbank-Administrator gefragt.

c) LIST DATABASE DIRECTORY

Damit werden alle Datenbanken auf der DB/2-Instanz angezeigt, zu der man sich eben verbunden hat.

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

Den Namen <newdatabase> für die neue Datenbank auf der DB/2-Instanz kann man frei wählen. Er darf aber nicht schon vorhanden sein. Ausserdem darf er höchstens aus 8 Zeichen bestehen. Gross-/Kleinschreibung wird ignoriert.

Benutzung des Codeset UTF-8 ist nützlich, weil sonst eventuell Zeichenketten mit Umlauten nicht in die – bei DB/2 in Byte-Länge gemessenen – Felder passen. Die Angabe des Territoriums ist obligatorisch. Man kann auch andere ISO-Ländercodes (z.B. CH, US, ...) nehmen.

e) CONNECT TO <newdatabase> USER <dbadm>

Als <dbadm> ist die UserId eines Datenbank-Administrators dieser Instanz einzugeben. Auf den CONNECT-Befehl hin, wird man nach dem Passwort für diesen Datenbank-Administrator gefragt. Damit beziehen sich die weiteren Befehle (SQL Statements) auf die eben erzeugte neue Datenbank.

f) CREATE SCHEMA <schema> AUTHORIZATION <siarduser>

für jedes Schema in der SIARD-Datei. Hier ist als <siarduser> die als Archivbenutzer erzeugte UserId im Betriebssystem der DB/2-Instanz anzugeben.

g) Zum Testen verbindet man sich als <siarduser> mit der neuen Datenbank:

CONNECT TO <newdatabase> USER <siarduser>

Man wird dabei nach dem Passwort des <siarduser> gefragt.

Diese Aktionen müssen offensichtlich von einem Administrator des DB/2-Servers durchgeführt werden. Für alles Weitere wird nur der Archivbenutzer mit seinem Passwort benötigt.

## **10 Unterstützte Datenbanksysteme**

In der *SIARD Suite* werden die Datenbanksysteme „MS Access“, „Oracle“, „SQL Server“, „MySQL“ und „DB/2“ unterstützt. Für den Zugriff auf MS Access wird der „pure JAVA“ JDBC-Treiber EnterAccess oder die JDBC/ODBC-Brücke von SUN (abgekündigt) benutzt, für die anderen Datenbanksysteme werden datenbankspezifische JDBC-Treiber benutzt.

Aufgrund von Unzulänglichkeiten der JDBC- und ODBC-Treiber ist die Verwendung der ODBC-JDBC-Brücke nicht ratsam.

Da die verschiedenen Datenbanksysteme leicht unterschiedliche Datentypen unterstützen, ist die von SIARD Suite vorgenommene Normalisierung der Datentypen für jedes Datenbanksystem in einem separaten Dokument beschrieben.

### 11 Protokollierung

Wenn Probleme bei der Benutzung der *SIARD Suite* auftauchen, ist es nützlich, ein Ausführungsprotokoll zu speichern. Ein solches sollte eine Fehlermeldung an die Wartung begleiten. Um ein solches Protokoll aufzuzeichnen, geht man folgendermassen vor:

Im Verzeichnis *etc* der SIARD-Installation findet man die Datei *logging.properties*. In dieser Datei wird die Protokollierung mit Niveau INFO auf Konsole umgeleitet. Um die Protokollierung in eine Datei umzuleiten, entfernt man das Kommentarzeichen # vor

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

Ausserdem ersetzt man in der Zeile

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

den Dateinamen nach dem Gleichheitszeichen durch den vollständigen Pfad der gewünschten Protokolldatei. Dabei ist zu berücksichtigen, dass Backslashes (\) zu verdoppeln sind. Man kann aber auch unter Windows überall Slashes (/) benutzen. Zum Beispiel:

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

Man kann in dieser Datei auch das Protokollierungsniveau (.level) setzen. Erlaubte Werte sind: NONE, SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST, ALL. ALL erzeugt am meisten, für Nicht-Entwickler unverständliche, Einträge, SEVERE erzeugt nur Information über Systemfehler. Eine vernünftige Alltagseinstellung ist INFO.



## **12      Beschränkungen**

Das SIARD-Format unterliegt folgenden Beschränkungen:

1. Die Grösse der SIARD-Datei kann 18'446'744'073'709'551'615 Bytes (ca. 18 ExaBytes) nicht überschreiten (ZIP64-Beschränkung).
2. Die Anzahl der (Tabellen und LOB-) Dateien kann 4'294'967'295 (ca. 4 Milliarden) nicht überschreiten (ZIP64-Beschränkung).
3. Die Gesamtheit der Metadaten der Datenbank muss im JAVA-Speicher Platz haben.

Die ersten beiden Beschränkungen dürften keinerlei Einschränkung darstellen, da reale Datenbanken noch lange Zeit nicht so gross sein werden.

Die dritte Beschränkung hingegen kann bei einer ungünstigen Konstellation von verfügbarem Hauptspeicher und Komplexität der Datenbank-Metadaten durchaus einmal eintreten. Man kann dieser Beschränkung entgehen, indem man die SIARD-Anwendungen auf einer Maschine startet, die über genügend Hauptspeicher (z.B. 4 GB) verfügt und allenfalls den JAVA-Heap mittels dem JAVA-Befehlszeilenparameter „-Xmx2000m“ o.ä. auf der JAVA-Befehls vergrössert.