



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement des Innern EDI

Schweizerisches Bundesarchiv BAR

Abteilung Informationsüberlieferung

Dienst Sicherung und Archivierungslösungen

Siard Suite Manuale

Version 1.0-SNAPSHOT

Table des matières

| | |
|--|---|
| 1. Einleitung | 1 |
| 2. Kontext | 1 |
| 3. Systemanforderungen und Installation | 1 |
| 3.1. Native Installer | 1 |
| 3.2. Zip Archive | 1 |
| 4. Siard Suite verwenden | 2 |
| 4.1. Siard Suite GUI | 2 |
| 4.2. Siard Suite CLI | 3 |
| 4.3. SiardFromDb | 3 |
| 4.4. SiardToDb | 6 |
| 5. Tips und Tricks | 7 |
| 5.1. Vorbereiten einer Datenbank für das Herunterladen | 7 |
| 5.2. Vorbereiten einer Datenbank für das Hochladen | 8 |
| 5.3. Time, Timestamps und Zeitzonen | 8 |

1. Einleitung

Der Standard SIARD (Software-Independent Archival of Relational Databases) definiert ein Format um Inhalte relationaler Datenbanken langfristig zu archivieren. Die Anwendung "Siard Suite" bietet eine grafische Benutzeroberfläche zum Archivieren, Wiederherstellen, Durchsuchen und Exportieren von SIARD Archiven.

2. Kontext

In Zusammenarbeit mit dem EU-Projekt E-ARK und der KOST wurde zwischen 2015 und 2018 die Version 2.1 des SIARD Formats spezifiziert. Wie schon zuvor die Version 1.0 wurde auch SIARD Format 2.1 als Standard eCH-0165 vom Verein "eCH E-Government Standards" homologiert. Im Jahr 2021 hat der DILCIS-Vorstand das erweiterte SIARD-Format 2.2 entwickelt und genehmigt.

Siard Suite ist die Referenzimplementation für das Archivieren relationaler Datenbanken im standardisierten SIARD Format 2.2. Weiter kann Siard Suite auch SIARD Archive in den Formaten SIARD 2.1 und SIARD 1.0 einlesen und zu SIARD 2.2 konvertieren.

3. Systemanforderungen und Installation

Die Applikation Siard Suite ist eine Java FX Applikation. Als solche kann sie grundsätzlich auf jedem System ausgeführt werden, auf welchem ein Java Runtime Environment 18 (JRE) mit der JavaFX Extension installiert ist.

Zusätzlich stehen native Installer für Windows, Mac und Linux zur Verfügung, welche die entsprechende Runtime beinhalten, so dass nicht notwendigerweise eine eigene JRE installiert werden muss.

In der Folge sind die einzelnen Distributionen beschrieben. Alle Distributionen können hier heruntergeladen werden: <https://github.com/sfa-siard/siard-suite/releases>

3.1. Native Installer

Für Windows, Mac und Linux können installierbare Pakete heruntergeladen und damit die Applikation SIARD Suite auf dem System installiert werden.

Allenfalls werden dazu auf dem System Administratorrechte benötigt.

Hinweis: für diese Installationsart steht das Siard CLI (Command Line Interface) nicht zur Verfügung.

3.2. Zip Archive

Anstelle der nativen Installer kann auch eine als ZIP Archiv bereitgestellte Distribution verwendet werden. Es steht keine Distribution mit inkludierter Runtime zur Verfügung. Entsprechend muss eine JRE v18 mit Java FX Erweiterung auf dem System installiert sein. Entsprechende

Distributionen sind u.a. hier erhältlich:

<https://www.azul.com/downloads/?package=jdk-fx#download-openjdk>

<https://aws.amazon.com/correto>

Die heruntergeladenen ZIP-Archive können an einem beliebigen Ort entpackt werden und dann mittels der entsprechenden Startskripte ausgeführt werden:

| Skriptname | Beschreibung |
|----------------------|---|
| siard-suite (.bat) | Starten der grafischen Benutzeroberfläche für Unix bzw. Windows Systeme |
| siard-from-db (.bat) | Archivieren einer Datenbank |
| siard-to-db (.bat) | Upload eines SIARD Archives in eine Datenbank |

4. Siard Suite verwenden

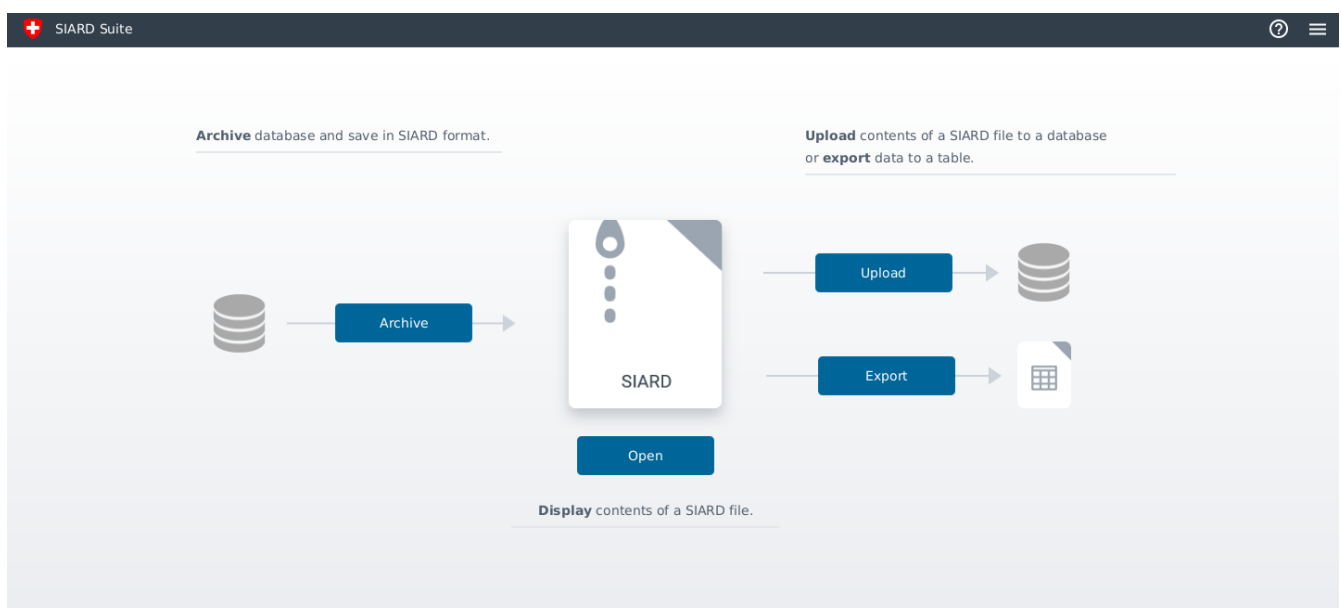
Die Siard Suite besteht aus folgenden Komponenten: * Grafische Benutzeroberfläche (GUI) * SiardFromDb (CLI) * SiardToDB (CLI)

Zweck und Verwendung der Komponenten wird in der Folge erklärt.

4.1. Siard Suite GUI

Die grafische Benutzeroberfläche ist die zentrale Anwendung um SIARD Archive zu erstellen und später auch wieder zu sichten, ohne dass die Primärdaten des Archives geändert werden können. Für komplexe Analysen der archivierten Daten sollte das Archiv wieder in eine Datenbank hochgeladen werden.

Der Startscreen bietet dein Einstiegspunkt in die Applikation. Hier können die vier wesentlichen Funktionalitäten der Applikation initiiert werden:



- Datenbank archivieren
- SIARD Archiv öffnen
- SIARD Archiv in eine Datenbank hochladen
- Daten eines SIARD Archives exportieren

Die Applikation führt den Benutzer durch die jeweiligen Prozesse.

4.1.1. Einstellungen

Unter dem Menüpunkt "Optionen" kann die Applikation konfiguriert werden.

Options

X

Login Timeout (s)

0

Query Timeout (s)

0

Cancel

Save

Folgende Einstellungen können vorgenommen werden:

| Option | Beschreibung |
|-------------------|---|
| Login Timeout (s) | Maximale Wartezeit auf eine Datenbankverbindung (in Sekunden). 0 für unbegrenzt |
| Query Timeout (s) | Maximale Wartezeit auf eine Datenbankabfrage (0 für unbegrenzt) |

4.2. Siard Suite CLI

Neben der grafischen Benutzeroberfläche können Datenbanken auch per Kommandozeile archiviert bzw. ein Archiv wieder in eine Datenbank eingelesen werden.

Dazu stehen in der ZIP Distribution die Scripte `siard-from-db` und `siard-to-db` zur Verfügung.

4.3. SiardFromDb

Mit `siard-from-db` kann man

- ein SIARD-Archive (Metadaten und Primärdaten) aus einer Datenbank (Option -s), und/oder
- eine SIARD Metadaten XML mit der Definition des Datenbankschemas (option -e). erzeugen.

4.3.1. Aufruf

In einem Terminal in das Verzeichnis wechseln, wo das ZIP Archiv entpackt wurde und dort in des Verzeichnis **bin** öffnen und dann ausführen mit:

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

4.3.2. Argumente

| Argument | Beschreibung |
|-----------------------|--|
| -o | Überschreiben (overwrite) der Ausgabe Datei(en), wenn sie existieren |
| -v | Views (Ansichten) als Tabellen archivieren |
| <login timeout> | Timeout in Sekunden für die Anmeldung (login) (0 für unbeschränkt) |
| <query timeout> | Timeout in Sekunden für die Abfrage (0 für unbeschränkt) |
| <import meta data> | Name der Metadaten XML-Datei, die als Muster zu berücksichtigen ist |
| <external LOB folder> | Ordner wo die Daten der grössten LOB (large object) Spalte der Datenbank extern zu speichern sind (gleichnamige Inhalte werden überschrieben!) |
| <mime type> | MIME-Typ der Daten der grössten LOB-Spalte der Datenbank (beeinflusst die Wahl der Namenerweiterung der extern gespeicherten LOBs) |

| Argument | Beschreibung |
|---------------------|--|
| <JDBC URL> | <p>JDBC URL der Datenbank, die heruntergeladen werden soll</p> <p>Bespiele:</p> <p>MS Access: jdbc:access:<path>/<filename>.mdb</p> <p>DB/2: jdbc:<host>:50000/<db-name></p> <p>MySQL: jdbc:mysql://<host>:3306/<db-name></p> <p>Oracle: jdbc:oracle:thin:@<host>:1521:orcl</p> <p>Postgres: jdbc:postgresql://<host>:5432/<db-name></p> <p>SQL Server: jdbc:sqlserver://<host>/<db-name>:1433</p> |
| <database user> | Datenbank User |
| <database password> | Datenbank Passwort |
| <siard file> | Name der zu erzeugenden .siard Datei |
| <export meta data> | Name der zu exportierenden .xml Metadaten-Datei |

4.3.3. Bemerkungen

Mindestens einer der beiden Parameter **siard-file** bzw. **export meta data** muss gesetzt sein (es ist auch möglich beide Parameter zu verwenden).

Hinweis: Während des Archivierungsprozesses sollte der Inhalt der Datenbank nicht geändert werden - verwenden sie gegebenenfalls einen Snapshot der Datenbank.

Die Option Views als Tabellen archivieren hat normalerweise redundante Speicherung derselben Daten zur Folge und ist deshalb nicht empfohlen. Sie ist dann nützlich wenn der für das Archivieren genutzte Datenbank-User die Inhalte der Views (Ansichten) anzeigen kann, nicht aber die der Basistabellen.

Der Archivierungsprozess ist entweder gänzlich erfolgreich oder er scheitert gänzlich.

Für grosse Datenbanken wird empfohlen, zuerst die Metadaten-XML herunterzuladen. Diese enthält alle Metadaten und Tabellengrößen und hilft so der Abschätzung der für den Download benötigten Ausführungszeit. Ausserdem sollte man die Option -q 0 für grosse Tabellen verwenden, da es unmöglich ist, abzuschätzen, wie viele Sekunden eine Größenabfrage dauern wird.

4.3.4. Archivbenutzer

Es ist generell keine gute Idee, den Datenbankadministrator (DBA, root, dbo, ...) mit allen Rechten für das Herunterladen eines SIARD-Archivs zu benutzen. Der Umfang der im SIARD-Archiv gespeicherten Daten ist durch die Objekte definiert, auf welche der Archivbenutzer lesend zugreifen darf. Der globale DBA kann normalerweise alle Datenbanken auf dem System und ausserdem viele Systemtabellen lesen, die nicht zu archivieren sind. Deshalb ist es wichtig, einen passenden, allenfalls neu zu erzeugenden, Archivbenutzer für das Herunterladen zu benutzen.

4.4. SiardToDb

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

Aufruf

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

4.4.1. Argumente

| Argument | Beschreibung |
|---------------------|--|
| -o | Überschreiben (overwrite) von gleichnamigen Typen und/oder Tabellen in der Datenbank, sofern solche existieren |
| <login timeout> | Timeout in Sekunden für die Anmeldung (login) (0 für unbeschränkt) |
| <query timeout> | Timeout in Sekunden für eine Abfrage (query) (0 für unbeschränkt) |
| <siard file> | Name der hochzuladenden .siard Datei |
| <JDBC URL> | JDBC URL der Zieldatenbank Beispiele: MS Access: jdbc:access:<paht>/<file-name>.mdb DB/2: jdbc:<host>:50000/<db-name> MySQL: jdbc:mysql://dbserver.enterag.ch:3306/<db-name> Oracle: jdbc:oracle:thin:@<host>:1521:orcl Postgres: jdbc:postgres://<host>:5432/<db-name> SQL Server: jdbc:sqlserver://<host>\<db-name>:1433 |
| <database user> | Datenbank User |
| <database password> | Datenbank Passwort |
| <schema> | Schemaname im SIARD-Archiv |

| | |
|----------------|--|
| <mappedschema> | in der Datenbank zu benutzender Schemaname |
|----------------|--|

4.4.2. Bemerkungen

Da ältere Datenbanken nicht SQL:2008-konform sind, ist es unvermeidlich, dass etliche manuelle Vorbereitungen für das Hochladen notwendig sind. Es gibt keine Schema-Objekte in MS Access. Benutzer und Schemas sind untrennbar in Oracle. Schemaobjekte und Datenbanken sind dasselbe in MySQL. Deshalb müssen alle Schemas vor dem Hochladen manuell erzeugt werden. Ausserdem muss der Datenbankbenutzer das Recht haben, Tabellen und Typen in diesen Schemas zu erzeugen. Da dies nicht immer einfach möglich ist, werden Datenbankschemas entsprechend der Abbildungsliste auf der Befehlszeile den Schemanamen zugeordnet.

Das Hochladen erzeugt nur Tabellen und Typen und versucht Primär- und Fremdschlüssel zu aktivieren. Keine anderen Objekte werden in der Zieldatenbank erzeugt. Falls die Schlüssel nicht aktiviert werden konnten, wird eine Warnung ausgegeben, das Hochladen aber trotzdem als erfolgreich abgeschlossen betrachtet. Auch ohne diese Schlüssel kann man SQL SELECT Abfragen der Datenbank ausführen.

Ausserdem werden beim Hochladen eventuell gewisse Abstriche gemacht. Bei MS Access landen alle Tabellen in derselben MDB/ACCDB. 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 Tabellename 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 SiardGui konsultiert, wird man die korrekten Bezeichnungen und Werte zuordnen können.

5. Tips und Tricks

5.1. Vorbereiten einer Datenbank für das Herunterladen

SIARD Suite lädt alle Daten herunter, die für den Datenbank-User sichtbar sind, welcher für die Verbindung benutzt wird. Die Wahl eines geeigneten Datenbank-Users bestimmt deshalb den Umfang der archivierten Datenbank. Oft steht ein geeigneter "technischer Datenbank-User" einer Datenbank-Anwendung zur Verfügung, mit der die Datenbank bewirtschaftet wird. Dieser hat normalerweise genau die Zugriffsrechte, die für die Archivierung der Datenbank benötigt werden.

Wenn jedoch kein solcher Datenbank-User verfügbar ist, sollte nicht der Datenbank-Administrator (database administrator, DBA, dbo, root, SYSTEM, sa, ...) für das Herunterladen benutzt werden. Denn dieser hat lesend Zugriff auf viele Systemtabellen und Tabellen aus anderen Datenbanken, die nicht archiviert werden sollen. Stattdessen wird empfohlen, einen Datenbank-User zum Zweck der Archivierung zu erzeugen. Diesem Archivierunguser sind Leseprivilegien auf alle Schemas, Tabellen, Views und Typen einzuräumen, die zur zu archivierenden Datenbank gehören. Die Dokumentation des DBMS gibt Auskunft, wie man einen User erzeugt und ihn mit den notwendigen

Privilegien ausstattet.

Wenn ein geeigneter User bestimmt oder erzeugt wurde, kann das Herunterladen der Datenbank mit dessen Anmeldeinformationen vorgenommen werden.

5.2. Vorbereiten einer Datenbank für das Hochladen

Für das Hochladen einer Datenbank in eine Instanz eines DBMS mit SIARD Suite müssen geeignete Datenbank-Schemas in der Zieldatenbank zur Verfügung stehen. Diese Schemas können dann in der Schema-Zuordnung im Dialog für das Hochladen (oder im Teil Schema-Zuordnung der Befehlszeile von SiardToDb) mit den Schemas im SIARD-Archiv assoziiert werden.

Idealerweise sind die Zielschemas leer. Es ist aber je nach Sicherheitsvorgaben beim DBMS nicht immer möglich, Schemas nach Belieben zu erzeugen oder auszuwählen. Wenn die Zielschemas nicht leer sind, werden die archivierten Daten nur hochgeladen, wenn entweder keine Namenskollisionen bei den Typen und Tabellen vorkommen, oder aber explizit die Option Überschreiben gewählt wurde.

Der Datenbank-User, dessen Anmeldeinformationen für das Hochladen benutzt werden, muss das Privileg haben, in den Zielschemas Typen und Tabellen zu erzeugen und Daten in diese Tabellen einzufügen. Die DBMS-Dokumentation gibt Auskunft, wie man Schemas erzeugt und wie man dem Datenbank-User die nötigen Privilegien einräumt. Bequemerweise nutzt man die Anmeldeinformationen des Datenbank-Administrators, sofern diese verfügbar sind. Denn dieser hat normalerweise alle notwendigen Privilegien. In diesem Fall sollte aber die Option Überschreiben nicht gewählt werden. Andernfalls ist das Risiko zu gross, dass wichtige Tabellen oder Typen im System überschrieben werden.

5.3. Time, Timestamps und Zeitzonen

Die Konversion von TIMES und TIMESTAMPS in der Datenbank hängt von der lokalen Zeitzone ab. Die UTC Zeit 14:30 im SIARD Archiv wird beim Hochladen in Zürich als lokale Zeit 15:30 in die Datenbank geladen – im Winter. Um diese Konversion zu unterdrücken, muss SiardToDb mit der Option `-Duser.timezone=GMT` gestartet werden, welche SIARD veranlasst, alle Datenbankzeiten als UTC-Zeiten zu interpretieren