



Hartwig Thomas, 4. December 2013

Document version 1.10

SIARD Suite

Data Type Mapping for MS Access

Published by:

Swiss Federal Archives
Archivstrasse 24
3003 Bern
Switzerland

1 Introduction

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

On behalf of the Swiss Federal Archives, Enter AG implemented the software SIARD Suite which supports converting database content from live proprietary database systems to the normalized SIARD format as well as uploading database content in SIARD format to such a live database system.

Unfortunately most real database systems do not support the SQL:1999 standard fully, on which the SIARD format is based. Therefore SIARD Suite needs to normalize/denormalize the data types during the conversion process.

This document specifies, how the MS Access data are converted to the SIARD format and how SIARD data are converted to MS Access on upload.

The data type conversions specified here are valid for .mdb as well as for .accdb files.

The conversions are *idempotent*. I.e. after the initial download any number of up- and download can be executed without changing the data types or values.

2 Mapping of SIARD Datatypes

2.1 MS Access => SIARD

See also

http://msdn.microsoft.com/en-us/library/aa140015.aspx#acintsql_datatypes.

The new .accdb data types are not listed in the MS Access 2007 Developer's Reference because they are not visible to SQL.

<i>MS Access</i>	<i>JDBC (java.sql.Types)</i>	<i>SQL:1999 (SIARD)</i>	<i>XML</i>
CHAR	CHAR(255)	CHARACTER(255)	xs:string
CHAR(n)	CHAR(n)	CHARACTER(n)	xs:string
VARCHAR(n)	VARCHAR(n)	CHARACTER VARYING(n)	xs:string
TEXT	LONGVARCHAR (1073741823)	CHARACTER LARGE OBJECT	clobType
MEMO	LONGVARCHAR (1073741823)	CHARACTER LARGE OBJECT	clobType
GUID	OTHER(36)	CHARACTER VARYING(36)	xs:string
SMALLINT	SMALLINT	SMALLINT	xs:integer
INTEGER	INTEGER	INTEGER	xs:integer
COUNTER	INTEGER	INTEGER	xs:integer
CURRENCY	NUMERIC(19,4)	NUMERIC(19,4)	xs:decimal
DECIMAL(p,q)	NUMERIC(p,q)	NUMERIC(p,q)	xs:decimal
REAL	REAL(24)	REAL	xs:float
DOUBLE	DOUBLE(53)	DOUBLE PRECISION	xs:float
BIT	BIT(1)	BOOLEAN	xs:boolean
BINARY	BINARY(510)	BIT(4080)	xs:hexBinary
BINARY(n)	BINARY(n)	BIT(8*n)	xs:hexBinary
VARBINARY	VARBINARY(510)	BIT VARYING(4080)	xs:hexBinary
VARBINARY(n)	VARBINARY(n)	BIT VARYING(8*n)	xs:hexBinary
IMAGE	LONGVARBINARY	BINARY LARGE OBJECT	blobType

SIARD Suite

<i>MS Access</i>	<i>JDBC (java.sql.Types)</i>	<i>SQL:1999 (SIARD)</i>	<i>XML</i>
	(1073741823)		
HYPERLINK	LONGVARCHAR	CHARACTER LARGE OBJECT	clobType
MULTIVALUED	LONGVARCHAR	CHARACTER LARGE OBJECT	clobType
ATTACHMENT	LONGVARCHAR	CHARACTER LARGE OBJECT	clobType
DATETIME	TIMESTAMP(19)	TIMESTAMP	xs:dateTime

2.2 SIARD => MS Access

<i>XML</i>	<i>SQL:1999 (SIARD)</i>	<i>MS Access</i>
xs:decimal	NUMERIC	DECIMAL
xs:decimal	NUMERIC(n)	DECIMAL(n)
xs:decimal	NUMERIC(p,q)	DECIMAL(p,q)
xs:decimal	DECIMAL	DECIMAL
xs:decimal	DECIMAL(n)	DECIMAL(n)
xs:decimal	DECIMAL(p,q)	DECIMAL(max(p,28),q)
xs:integer	SMALLINT	SMALLINT
xs:integer	INTEGER	INTEGER
xs:integer	BIGINT	INTEGER
xs:float	REAL	REAL
xs:float	DOUBLE PRECISION	DOUBLE
xs:float	FLOAT(n)	n <= 24: REAL n > 24: DOUBLE
xs:boolean	BIT	BIT
xs:boolean	BOOLEAN	BIT

SIARD Suite

<i>XML</i>	<i>SQL:1999 (SIARD)</i>	<i>MS Access</i>
xs:hexBinary	BIT(n)	n/8 < 256: BINARY(ceil(n/8)) n/8 >= 256: IMAGE
xs:hexBinary	BIT VARYING(n)	n/8 < 256: VARBINARY(ceil(n/8)) n/8 >= 256: IMAGE
xs:hexBinary	BINARY LARGE OBJECT	IMAGE
xs:string	CHARACTER	CHAR(1)
xs:string	CHARACTER(n)	n < 256: CHAR(n) n >= 256: MEMO
xs:string	CHARACTER VARYING(n)	n < 256: CHAR(n) n >= 256: MEMO
xs:string	CHARACTER LARGE OBJECT	MEMO
xs:string	NATIONAL CHARACTER	CHAR(1)
xs:string	NATIONAL CHARACTER(n)	n < 256: CHAR(n) n >= 256: MEMO
xs:string	NATIONAL CHARACTER VARYING	N < 256: VARCHAR(n) N >= 256: MEMO
xs:string	NATIONAL CHARACTER LARGE OBJECT	MEMO
xs:string	XML	MEMO
xs:date	DATE	DATETIME
xs:time	TIME(p)	DATETIME
xs:dateTime	TIMESTAMP(p)	DATETIME

If a string is longer than 4000 characters then „clobType“ and „xs:string“ are replaced by an external reference to a text file.

If a binary array is longer than 2000 bytes then „blobType“ and „xs:hexBinary“ are replaced by an external reference to a binary file.

Characters that cannot be represented in UNICODE (Codes 0-8, 14-31, 127-159) as well as the escape character '\' and multiple space characters are escaped as \u00<xx> in XML. Less-than and ampersand characters are represented as entity references in XML.