

The specification for the location of LOBs according to SIARD

The specification for the location of LOBs (large objects) according to the SIARD File Format specification version 2.1 (and 2.2) can appear somewhat complex, which is mostly due to its flexibility and backwards compatibility.

In order to clarify the understanding of the locations of LOBs according to SIARD the document "LOB folder handling" Feb. 1st, 2018 was written by Hartwig Thomas, Enter AG on behalf of the SFA.

It states that : "The LOB (large object) folder handling in SIARD Format 2.1 needs to be clarified and illustrated with some examples."

The specification of the SIARD File Format version 2.2 is based on version 2.1 and the clarifying document.

Based on the above mentioned clarifying document and specifications the DILCIS Board states the following regarding the location of LOBs:

The location of LOBs is specified at three levels:

- cell¹ level - SIARD-2.2: 6.2 Large object data cells
- column level (aka 'local level') - SIARD-2.2: 5.6 Column level metadata
- database level (aka 'global level') - SIARD-2.2: 5.1 Database level metadata

Locations can be inside (inside the SIARD file) or outside.

The column level and the database level is for outside location (that is outside the SIARD file).

Locations can be absolute or relative. Relative locations are relative to the location of the upper level.

The cell level

If no cell level location is given, i.e. there is no *file* attribute in the <c[n]> element then no LOB exists for the cell!

If a cell level location is given it is the value of the *file* attribute of the <c[n]> element (of type blobType or clobType) in the <row> element in the file table[k].xml.

The location at cell level must be relative - i.e. absolute location at cell level is not allowed.

The column level

If no column level location is given (i.e. there is no <lobFolder> element in the <column> element for the column in question in the file metadata.xml) then the location is inside the SIARD file.

If a column level location is given the location is outside, either absolute - or relative (recommended) to the database level.

The database level

If no database level location is given (i.e. there is no <lobFolder> element in

¹the intersection of a column and a row

the <siardArchive> element in the file metadata.xml) then the location is relative to the SIARD file.

If the database level location is given the location is outside, either absolute - or relative to the location of the SIARD file in the file system.

Coexistence of inside and outside location of LOBs

Inside and outside location of LOBs can exist for the same database and the same table. If the column level location is not given (no lobFolder element) for a column in question, the LOB location is inside - even if a database level location is given (lobFolder element exists), since other columns with LOBs (in the same table or other tables) may have a column level location using a relative location to the database level location.

This permits maximum flexibility in assigning inside and outside location and preserving relocatability.

Strong recommendation: Use relative location for all LOBs

It is strongly recommended to choose all level locations as relative - according to SIARD-2.2: 5.1 Database level metadata.

Thus, on relocation of the SIARD file (or its information package), only the database level location needs to be changed.

(Note that due to space limitations in the file system where the SIARD file resides an absolute database level location to another file system is often needed.)

Concatening relative external locations of LOBs

The relative locations at cell level, the relative locations at column level, and the relative or absolute locations at database level must be concatenated to get the full location of a LOB (the target URI).

Example:

```
<c4 file="seg_0/t2_c4_r1.bin" length="10151" /></row>  
<column>...<lobFolder>s0_t2_c4/</lobFolder>...</column>  
<siardArchive>...<lobFolder>./Northwind_lobs/</lobFolder>
```

gives the full location:

```
./Northwind_lobs/s0_t2_c4/seg_0/t2_c4_r1.bin
```

where the database level location is relative to the location of the SIARD file in the file system.

```
Northwind.siard  
Northwind_lobs/  
    s0_t2_c4/  
        seg_0/  
            t2_c4_r1.bin
```

Appendix

Overview of combinations of locations of LOBs

Recall that:

The cell level location must only be relative, not absolute.

If a column level location does not exist (nil) then the location of the LOBs in the cells for that column are inside the SIARD file, regardless if a database level location exists or not.

If a database level location exists then the column level location must be relative (or nil).

Note that this interpretation means that the existence of a database level location implies that LOBs outside the SIARD file can only reside in one file system hierarchy.

In other words it implies that one cannot have the LOBs from one column stored on one server and LOBs from another column stored on another server. (Such a need requires no database level location and column level location to be absolute).

Location of LOBs – database location nil

Level	Path	In/Out	Level	Path	In/Out	Level	Path	In/Out
database	<i>nil</i>							
column	<i>nil</i>		column	<i>rel</i>		column	<i>abs</i>	
cell	<i>nil</i>	nil	cell	<i>nil</i>	nil	cell	<i>nil</i>	nil
cell	<i>rel</i>	in	cell	<i>rel</i>	out	cell	<i>rel</i>	out
cell	<i>abs</i>	ERR	cell	<i>abs</i>	ERR	cell	<i>abs</i>	ERR

Location of LOBs – database location relative

Level	Path	In/Out	Level	Path	In/Out	Level	Path	In/Out
database	<i>rel</i>							
column	<i>nil</i>		column	<i>rel</i>		column	<i>abs</i>	
cell	<i>nil</i>	nil	cell	<i>nil</i>	nil	cell	<i>nil</i>	nil
cell	<i>rel</i>	in	cell	<i>rel</i>	out	cell	<i>rel</i>	ERR
cell	<i>abs</i>	ERR	cell	<i>abs</i>	ERR	cell	<i>abs</i>	ERR

Location of LOBs – database location absolute

Level	Path	In/Out	Level	Path	In/Out	Level	Path	In/Out
database	<i>abs</i>							
column	<i>nil</i>		column	<i>rel</i>		column	<i>abs</i>	
cell	<i>nil</i>	nil	cell	<i>nil</i>	nil	cell	<i>nil</i>	nil
cell	<i>rel</i>	in	cell	<i>rel</i>	out	cell	<i>rel</i>	ERR
cell	<i>abs</i>	ERR	cell	<i>abs</i>	ERR	cell	<i>abs</i>	ERR

Appendix

[RFC 3986](#)

[5.4.](#) Reference Resolution Examples

Within a representation with a well defined base URI of

```
http://a/b/c/d;p?q
```

a relative reference is transformed to its target URI as follows.

[5.4.1.](#) Normal Examples

```
"g:h"           = "g:h"
"g"             = "http://a/b/c/g"
"/g"           = "http://a/b/c/g"
"g/"           = "http://a/b/c/g/"
"/g"           = "http://a/g"
"//g"          = "http://g"
"?y"           = "http://a/b/c/d;p?y"
"g?y"          = "http://a/b/c/g?y"
"#s"           = "http://a/b/c/d;p?q#s"
"g#s"          = "http://a/b/c/g#s"
"g?y#s"        = "http://a/b/c/g?y#s"
";x"           = "http://a/b/c/;x"
"g;x"          = "http://a/b/c/g;x"
"g;x?y#s"      = "http://a/b/c/g;x?y#s"
""             = "http://a/b/c/d;p?q"
"."            = "http://a/b/c/"
"./"           = "http://a/b/c/"
".."           = "http://a/b/"
"../"          = "http://a/b/"
"../g"         = "http://a/b/g"
"../.."        = "http://a/"
"../../"       = "http://a/"
"../..g"       = "http://a/g"
```

-- END OF QUOTATION --

Our interpretation of transformation of RFC 3986 5.4.:

Within a representation with a well defined base URI of

```
http://a/b/c/Northwind.siard;p?q
```

a relative reference is transformed to its target URI as follows.

```
"Northwind_lobs/" = "http://a/b/c/Northwind_lobs/"
"./Northwind_lobs/" = "http://a/b/c/Northwind_lobs/"
```

[RFC 8089](#)[E.2.](#) DOS and Windows Drive Letters

On Windows- or DOS-like file systems, an absolute file path can begin with a drive letter. To facilitate this, the "local-path" rule in [Section 2](#) can be replaced with the following:

```
local-path      = [ drive-letter ] path-absolute
drive-letter    = ALPHA ":"
```

The "ALPHA" rule is defined in [\[RFC5234\]](#).

This is intended to support the minimal representation of a local file in a DOS- or Windows-like environment, with no authority field and an absolute path that begins with a drive letter. For example:

- o "file:c:/path/to/file"

URIs of the form "file:///c:/path/to/file" are already supported by the "path-absolute" rule.

Note that comparison of drive letters in DOS or Windows file paths is case insensitive. In some usages of file URIs, drive letters are canonicalized by converting them to uppercase; other usages treat URIs that differ only in the case of the drive letter as identical.

[E.2.1.](#) Relative Resolution

To mimic the behavior of DOS- or Windows-like file systems, relative references beginning with a slash "/" can be resolved relative to the drive letter when present; resolution of "." dot segments (per [Section 5.2.4 of \[RFC3986\]](#)) can be modified to not ever overwrite the drive letter.

For example:

```
base URI:  file:///c:/path/to/file.txt
rel. ref.: /some/other/thing.bmp
resolved:  file:///c:/some/other/thing.bmp
```

-- END OF QUOTATION --