

XLUM-file format Documentation

Sebastian Kreutzer, Steve Grehl, Michael Höhne, ... <further authors>

version: 1.0 [2022-03-30]

Contents

1	Motivation and principles	1
2	General description	1
3	Detailed description	2
3.1	The <curve> level	2
3.2	The <record> level	6
3.3	The <sequence> level	8
3.4	The <sample> level	9
3.5	The <xlum> level	10
3.6	General parameters (all nodes)	11
4	Special cases	12
4.1	Pulsing data	12
5	Practical examples	12

1 Motivation and principles

This document details the **XLUM** format, an XML based file format for long-term data preservation and exchange of luminescence data. The format is readable by humans and machines, and the data can be easily checked with any text reader on any major operating system. This design allows a platform-independent operation. The following documentation provides essential information on the file **XLUM** format and accounts for the needs for individual flexible data analysis, even with self-written software applications.

Three simple design principles underpin the format specification:

- Stored are data recorded over **time** or time derived instances,
- stored are data on a technical component level,
- data stored in a file are self-consistent.

2 General description

The **XLUM**-format is an XML derivative. The base data structure is a tree with five nodes storing the data. Each level has its unique denotation.

The levels and a short introduction is given in the table below. A detailed description can be found in the following sections. Nodes represent each level in the structure. **Only <curve> nodes contain physical quantities, such as luminescence data** all other nodes are parent nodes of <curve> to structure the dataset and ship additional metadata.

The first level, `xlum` is a supernode to enable storage of luminescence data in arbitrary files following the XML scheme. As a side effect this also allows custom file endings different from `*.xlum`.

Table 1: Node overview

Node name	Number of samples	Number of aliquots	Number of records	Number of data curves
<code><xlum></code>	inf	inf	inf	inf
<code><sample></code>	1	inf	inf	inf
<code><sequence></code>	1	1	inf	inf
<code><record></code>	1	1	1	inf
<code><curve></code>	1	1	1	1

A node has a name, attributes, and data stored in it. The data stored in the attributes describe the state of this level. The stored data in the node describes the process(es) assigned to the node. A minimal example is shown in the listing. The documentation provides an overview starting with the leaf node description and going up to the root from there. Further notes:

- The format version bases on XML version 1.0 (Fith Edition)
- File encoding should always be UTF-8
- This specification lists only mandatory attributes. Additional, custom attributes are explicitly supported.
- Parser supporting the XLUM format must not crash when encountering non-specified node attributes. However, they may skip them.

```
<?xml version="1.0" encoding="UTF-8"?>
<xlum>
  <sample>
    <sequence>
      <record>
        <curve>
          1,2,...,n
        </curve>
      </record>
    </sequence>
  </sample>
</xlum>
```

3 Detailed description

3.1 The `<curve>` level

The `curve` level is the deepest node (leaf) and has no further sub-levels. A curve holds the predefined/simulated or measured output of one single technical component. For example, a typical thermoluminescence measurement may consist of one or many curves.

1. **The three curve example:** (a) Time against temperature recorded by a thermocouple (temperature sensor), (b) time against photon counts recorded by a photomultiplier tube,
(c) time against a predefined heating ramp.
2. **The one curve example:** Time against temperature. In this case temperature is a processed quantity, because measurements happen over a time instant. However, for compatibility reasons, this would be allowed although it is not preferred.

In both cases, all mentioned `curve` nodes belong to one parent `record`. In case 1, the record contains one curve and three in case 2. Ideally, curves represent technical components and not processed quantities.

3.1.1 Value storage

Values in the `<curve>` node are stored comma separated. **Only quantities detected by a detector/sensor are stored in this node.** This includes simulated values. Figure 1 shows three types of (luminescence) detectors, distinguished by their capacity to record spatially resolved information. Regardless of the detector, recorded quantities are stored in the `<curve>` node separated by a comma. Example:

```
<curve>59.5167,109.5164,149.7003,109.5170,156.2654</curve>
```

Values in the node are real numbers ($-1e-307 \leq x \leq 1e+307$; $x \in \mathbb{R}$). Scientific ‘E’/‘e’ notations of numbers are explicitly allowed, example: $1e+2$ for 100 or $1e-2$ for 0.01. Decimal separator is a . (dot).

For consistency reasons, the value in the node span an array with dimensions defined through the node attributes `xValues`, `yValues` and `tValues` (see Fig. 1).

1. The simplest form is a detector that records no spatial information. For instance, a photomultiplier tube only knows about count recorded over time. Alternatively, the detector can be any sensor, measuring, e.g., temperature, pressure or current.
2. A spectrometer (here a camera in front of a spectrograph) is another type of detector with spatial information, here wavelengths split by the spectrograph hitting different pixel areas of the detector.
3. The most complex from is a detector that records events (e.g., luminescence), spatially resolved. Such as a camera.

In the examples and the Fig.1 the detectors detect luminescence, however, any type of sensor, electronic component recording information of relevance for the measurement of luminescence are suitable. Typical examples:

- Temperature sensor (thermocouple), recording the temperature of the heating element
- Photo diode, monitoring the optical power density at the sample position
- Resistor, measuring the current of a LED
- Pressure sensor, monitoring the atmosphere in the measurement chamber
- Inductive sensor, monitoring closing and opening of a shutter

3.1.2 Node attributes

Table 2: Specified curve attributes. Attributes in [] are optional, hence NA is not allowed.

Identifier	Type	Allows NA?	Example	Information
component	string	yes	"EMI 9235QB15"	name of the technical component. NA allowed only for "predefined"
startDate	date	no	"2021-07-14T22:59:35.0Z"	ISO 8601-1:2019: YYYY-MM-DDThh:mm:ss[.mmm]Z recalculated to Zulu time
curveType	string	no	"predefined"	Values allowed are only "predefined" or "measured"
duration	number	no	"20.000"	Duration of the detection in seconds or a fraction of it
offset	number	no	"10.000"	Before the detection starts in seconds or a fraction of it

Identifier	Type	Allows NA?	Example	Information
xValues	number	yes	"1.0,2.0,3.0"	x-coordinate values of the detector
yValues	number	yes	"1.0,2.0,3.0"	y-coordinate values of detector
tValues	number	no	"1.0,2.0,3.0"	time values in seconds or fractions of it. Values must be positive
vLabel	string	no	"Luminescence"	Measured physical quantity
xLabel	string	yes	"pixel"	label of the x-coordinate values
yLabel	string	yes	"pixel"	label of the y-coordinate values
tLabel	string	no	"time"	label of t-values, usually 'time'
vUnit	string	no	"cts"	label of the v-values, for luminescence usually photon counts
xUnit	string	yes	"nm"	SI unit or equivalent for x-values
yUnit	string	yes	"px"	SI unit or equivalent for y-values
tUnit	string	no	"s"	SI unit or equivalent for t-values
[detectionWindow]	string	no	"375"	Centre wavelength if applicable. can be set to NA
[filter]	list (string)	no	"Hoya U340; Delta BP 365/50EX"	Filter names separated by ;

3.1.3 Example

In order to define a suitable standard only a few attributes are required. A few of them are useful for every kind of component, others are only meaningful in combination. For instance, providing information on the detection window is not meaningful for a heating element.

```
{...}
<curve component="heating element" startDate="2021-02-14T225700.0Z" curveType="predefined" duration=
  offset="0" xValues="NA" yValues="NA" vLabel="temperature" tValues="1,2,3,4,5" xLabel="NA" yLabel="NA"
  tLabel="time" vLabel="temperature" xUnit="" yUnit="" vUnit="K" tUnit="s">
    293,303,313,323,333
  </curve>
<curve component="thermo couple" startDate="2021-02-14T225700.0Z" curveType="measured" duration="5"
  offset="0" xValues="NA" yValues="NA" tValues="1,2,3,4,5" xLabel="NA" yLabel="NA"
  vLabel="temperature" tLabel="time" xUnit="" yUnit="" vUnit="K" tUnit="s">
    293,303,313,323,333
  </curve>
<curve component="EMI 9123QB15" startDate="2021-02-14T225700.0Z" curveType="measured" duration="5"
  offset="0" xValues="NA" yValues="NA" tValues="1,2,3,4,5" xLabel="NA" yLabel="NA"
  vLabel="luminescence" tLabel="time" xUnit="" yUnit="" vUnit="cts" tUnit="s"
  detectionWindow="380 nm" filter="Hoya U340 + Delta 365/40">
    20,25,32,41,46
  </curve>
```

XLUM count value storage principles for different detectors in <curve> node

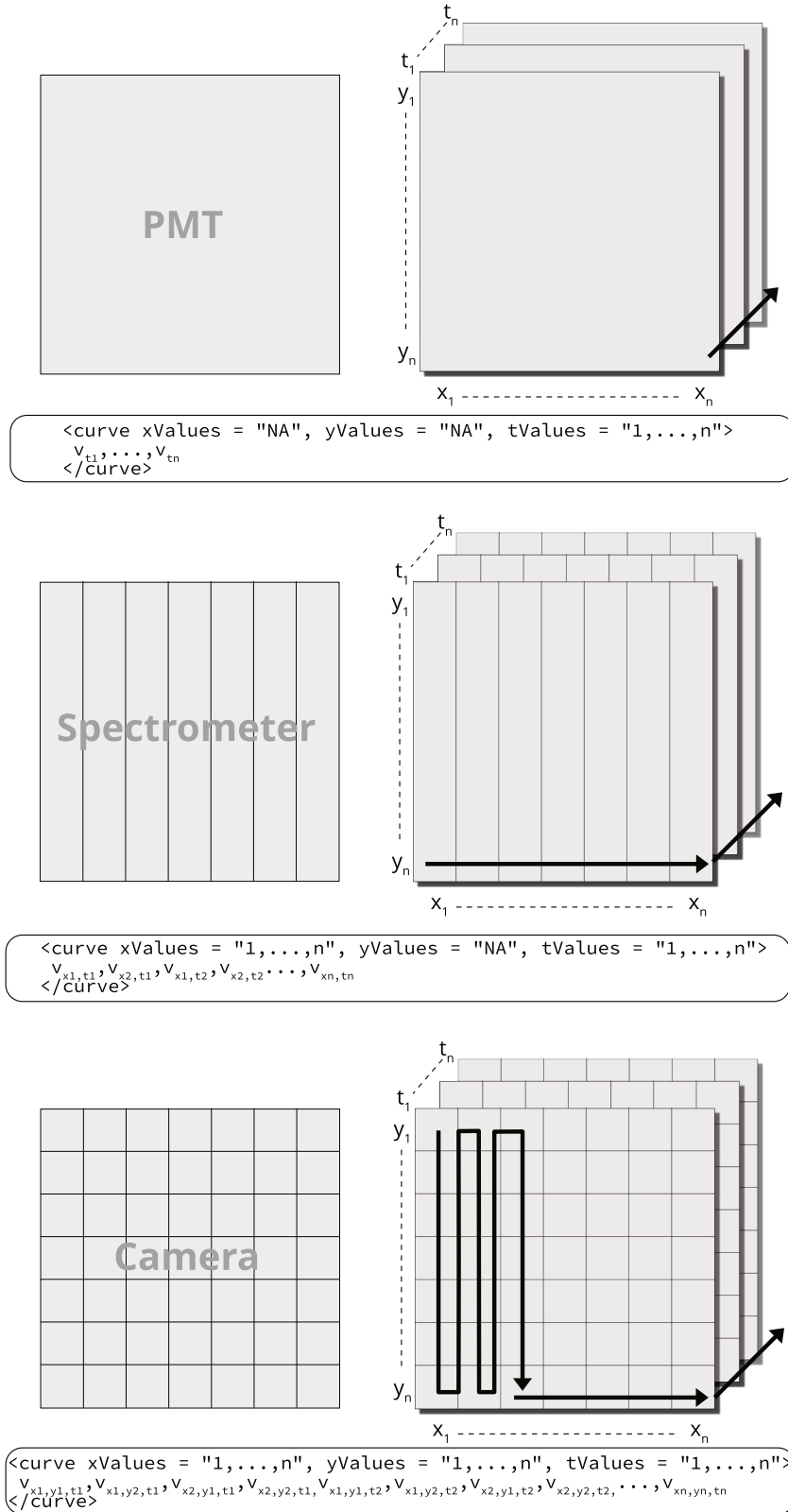


Figure 1: The three cases for data storage depending on the detector. The black solid arrows indicate how the array is constructed from the values stored in the node.

```
</curve>
{...}
```

One curve is related to one technical device. To define an accurate time window, the parameters *duration* and *offset* should be used. These parameters are related to the start of the parent **<record>** node.

3.2 The **<record>** level

A **<record>** defines one process involving many components, hence **<record>** may contain $1 \leq n \leq Inf; n \in \mathbb{Z}$ **<curve>** nodes. A record can also be understood as one step in a measurement sequence, e.g., a TL measurement. All curves within the record should have been detected within the same time frame defined through the attributes **<startDate>** and **<endDate>**.

3.2.1 Attributes

Table 3: Allowed attributes for **<record>** node.

Identifier	Type	Allows NA?	Example	Information
startDate	date	no	"2021-07-14T22:59:35.028601-1:2019:ISOZ601-1:2019:YYYY-MM-DDThh:mm:ss[.mmm]Z"	recalculated to Zulu time. Time when the record starts
endDate	date	no	"2021-07-14T22:59:35.028601-1:2019:ISOZ601-1:2019:YYYY-MM-DDThh:mm:ss[.mmm]Z"	recalculated to Zulu time. Time when this measurement step was finished
recordType	string	no	"TL"	valid values according to the recordType table
description	string	yes	"green stimulated luminescence"	allows a longer description of the record
sequenceStepNumber	integer	yes	5	index in the sequence
sampleCondition	string	yes	dose	valid values according to the table below

The **recordType** names are not carved in stone, but should be chosen to best describe a record in one word or using an accepted abbreviation (e.g., OSL for optically stimulated luminescence). However, it is important to keep in mind that the **recordType** makes **no** claim on the technical components involved, but is solely set to ease the understanding of the record. The hard physical information is stored in the **<curve>** node.

Table 4: Valid `recordTypes`

recordType	Information
"bleaching"	any kind of bleaching
"irradiation"	irradiation with ionising source
"atmosphereExchange"	atmosphere exchange
"heating"	any kind for heating without signal detection
"spectrometer"	spectrometer
"camera"	camera, e.g. any kind of imaging other than spectrometry
"TL"	thermoluminescence measurement
"ITL"	Isothermal luminescence
"TM-OSL"	thermally modulated luminescence
"RF"	radiofluorescence
"UV-RF"	radiofluorescence with detection in the UV wavelength range
"IR-RF"	radiofluorescence with detection in the infrared wavelength range
"IR-PL"	infrared photoluminescence
"OSL"	optically stimulated luminescence
"GSL"	green optically stimulated luminescence
"VSL"	violet optically stimulated luminescence
"YSL"	yellow optically stimulated luminescence
"POSL"	pulsed optical stimulated luminescence
"pause"	pause
"custom"	any undefined or custom record type not listed above

valid entries for `sampleConditions` are:

Table 5: Valid values for `sampleConditions`

sampleCondition	Information
"Natural"	Natural
"Natural+Dose"	Natural+Dose
"Bleach"	Bleach
"Bleach+Dose"	Bleach+Dose
"Nat.(Bleach)"	Nat.(Bleach)
"Nat.+Dose(Bleach)"	Nat.+Dose(Bleach)
"Dose"	Dose
"Background"	Background

Please note that the attribute `sampleCondition` is of informative nature, accounting for information stored in BIN/BINX-files. The field can be set to `NA`.

3.2.2 Example

The following examples consists of two records.

```
{...}
<record startDate="2021-02-14T22:59:35.0z" endDate="2021-02-14T23:10:00.0z"
  recordType="GSL" description="green OSL CW" comment="standard green OSL step"
  sequenceStepNumber="1" sampleCondition="NA">
  <curve ...>
    {...}
  </curve>
```

```

    <curve ...>
      {...}
    </curve>
  </record>
  <record startDate="2021-02-14T23:10:10.0z" endDate="2021-02-14T23:20:00.0z"
    recordType="bleach" description="bleaching with solar simulator" comment="NA"
    sequenceStepNumber="2" sampleCondition="NA">
    <curve ...>
      {...}
    </curve>
  </record>
{...}

```

3.3 The <sequence> level

A sequence describes multiple measurements sequentially used at **one** aliquot, for instance as cup/disc with multiple grains or a single grain. Hence, another aliquot from the **same** sample needs to be wrapped in a new <sequence> node. Naturally the information stored at that level can be very verbose, but are limited to a few required attributes only. In the luminescence context a sequence can fit, e.g., a SAR protocol, but also a succession of protocols measured on one aliquot.

3.3.1 Attributes

Table 6: Mandatory attributes <sequence> node

Identifier	Type	Allows NA?	Example	Information
startDate	Date	no	"2021-08-14T22:59:35.0Z"	start date of the sequence (can be identical with first <record> and <curve> entry)
position	numeric	no	"42"	position of aliquot in the reader, can be 1 in case samples are changed manually
name	string	yes	"SAR measurement"	name of the sequence
fileName	string	yes	"SAR_OSL.seq"	sequence file so far applicable
software	string	yes	"Sequence Editor 4.4"	software used to write the sequence if applicable
readerName	string	yes	"Risø OSL/TL"	name of the luminescence reader employed for the measurement
readerSN	string	yes	"234-23"	reader serial number
readerFW	string	yes	"reader OSL 12.3"	reader firmware version

3.3.2 Example


```

{...}
<sequence startDate="2021-08-14T22:59:35.0Z", position="1" name="SAR measuremt"
  fileName="Testsequence.seq" software="LexStudio 2, 2.37">
  <record ...>
    <curve ...>
      {...}
    </curve>
    {...}
  </record>
  {...}
</sequence>
{...}

```

3.4 The <sample> level

The sample constitutes the top level for one particular sample. The sample level may contain an infinite number of sequences measured for one sample.

3.4.1 Attributes

Table 7: Mandatory attributes <sample> node

Identifier	Data type	Allows NA?	Example	Information
name	number	yes	"LUM-21321"	unique sample identifier
mineral	string	yes	"quartz"	dominant mineral composition of the sample
latitude	numeric	yes	"52.4091392"	latitude in decimal degrees of the sample origin
longitude	numeric	yes	"-4.0702446"	longitude in decimal degrees of the sample origin
altitude	numeric	yes	"50"	altitude in m above sea level
doi	string	yes	"10.1016/j.radmeas.2017.02.003"	DOI with further information on the sample, if applicable

3.4.2 Example

```

{...}
<sample name="LUM-21321" mineral="quartz" latitude="52.4091392" longitude="-4.0702446"
  altitude="50" doi="10.1016/j.radmeas.2017.02.003">
  <sequence ...>
    <record ...>
      <curve ...>
        {...}
      </curve>
      {...}
    </record>
  </sequence>
{...}

```

```

    </record>
    {...}
  </sequence>
  {...}
</sample>
{...}

```

3.5 The <xlum> level

The <xlum> is the format parent node and it can contain an infinite number of samples. Only this level allows to set a licence for the distribution and usage of the data. All data within a particular <xlum> are considered distributed under the licence. You have to create new instances of the <xlum> to set different licences.

3.5.1 Attributes

Table 8: Mandatory attributes of <xlum> node

Identifier	Type	Allows NA?	Example	Information
xml:lang	string'	no	: <i>predefined</i>	ISO 639-1 language code, must not be changed
version	numeric	no	"1.0"	the xlum format version number relevant
flavour	string	no	"generic"	allows to specify a particular flavour of the format
author	list(string)	yes	"Max Planck; Marie Curie"	names of the author(s) of this dataset
license	string	yes	"CC BY 4.0"	license for the distribution of the dataset.
doi	string	yes	"10.5281/zenodo.596252"	A digital document identifier referencing the dataset already archive in a data repository

Table 9: Allowed licence types

licence	Licence name	Reference	Comment
"CC BY 4.0"	Creative commons	https://creativecommons.org	all flavours of the creative commons license scheme

licence	Licence name	Reference	Comment
"Copyright"	Copyright protected	-	data are copyright protected and cannot be used or distributed without the creator's agreement

3.5.2 Example

```
{...}
<xlum xml:lang="en" formatVersion="1.0" flavour="generic" author="Marie Curie; Max Planck"
  license="CC BY 4.0" doi="NA">
  <sample ...>
    <sequence ...>
      <record ...>
        <curve ...>
          {...}
        </curve>
        {...}
      </record>
    </sequence>
    {...}
  </sample>
  {...}
</xlum>
{...}
```

3.6 General parameters (all nodes)

A few nodes are available on all node levels. There are mainly of technical nature to ease the sequential of data storage during the measurement process.

Table 10: Attributes valid in all nodes

Identifier	Data type	Allows NA?	Example	Information
state	string	yes	"recording"	defines the state of the node; only used by equipment manufactures
parentID	number	yes	"201007145551910"	unique identifying the parent node, the super node has the ID 0, this is usually not needed but may help to store data
comment	String	yes	"what a wonderful sample"	comment field

4 Special cases

This section provides suggestions for attributes for particular types of measurements. These attributes are not compulsory, but may be used for the sake of consistency.

Either way, these metadata must not be required to store and analyse the data!

4.1 Pulsing data

In measurements with pulsed light stimulation data processing usually requires summations of photon counts on the hardware level, because it is neither meaningful nor feasible to store every single count in separate curves for photon arrival times of $10^{-6} \leq \tau \leq 10^{-3}$ s. Hence, multiple detector **<curve>** data may be stored per **<record>**, one for each (x) pulse(-s). The **<record>** node contains some additional metadata that may help to understand the data and the undertaken processing on the hardware level better.

Table 11: Additional metadata **<record>** node pulsing data

Identifier	Data type	Example	Information
onTime	numeric	"1E-01"	on time of stimulation per pulse, [s]
offTime	numeric	"1E-01"	on time of stimulation per pulse, [s]
nPulses	numeric	"10"	number of stimulation pulses total
summations	numeric	"2"	how many consecutive pulse records are summed (e.g. 10 pulses, 2 summations -> 5 records)
channelsPerPulse	numeric	"1000"	number of channels that are recorded in one curve
countsNormalized	numeric	"1"	if not 0, counts per channel are normalized to counts per second (helpful with uneven distributed channel times)

Also the curves contain additional metadata

Table 12: Additional metadata **<curve>** node pulsing data

Identifier	Data type	Example	Information
pulseID	numeric	"1"	current pulse number

5 Pratical examples

This section contains detailed examples for particular curve types in order to provide usable examples for the data format.

```
<?xml version="1.0" encoding="utf-8">
<xlum version="1.0" flavour="xlum" doiRef="">
  <sample ...>
    <sequence ...>
      <record ...>
        <curve ...>
          12,23,23,23,13,23
        </curve>
      </record>
```

```
</sequence>  
</sample>  
</xlum>
```