Format Description

Track-it
XML Format

Version 1.2

D950.299.00/02 en 2020-04 Bi/Schi

Disclaimer

This description is provided to the user one-time. PTW-Freiburg is not obliged to update, extend or adapt this description. The content of this description does not entitle the user to make any assumptions or inferences on the completeness of this description.

Any conclusions drawn from this document are the sole responsibility of the user.

Safety Information

⚠ CAUTION

Improper handling.

Maloperation!

If you provide Track-it with data formatted according to this document, this document serves as a supplement to the Track-it user manual. In this case, use the document only in combination with the Track-it user manual.

Follow the General Information and Safety Information given in that user manual.

PTW-Freiburg
Physikalisch-Technische Werkstätten
Dr. Pychlau GmbH
Lörracher Str. 7
79115 Freiburg
GERMANY
Tel.: +49 761 49055-0
Fax: +49 761 49055-70
info@ptwdosimetry.com
www.ptwdosimetry.com

Contents

Overview

This document describes the XML format used to export data to the Track-it database.

The XML file describes measurements in a defined format that allows them to be imported into the Track-it database. As an option, the original measured values used to calculate the analysis values can additionally be saved.

⚠ CAUTION

Transferring data with unsuitable formatting.

Incorrect Operation!

It is mandatory to observe the information given in section "Transferring an XML File to Track-it" if you want to transfer your own data to the Track-it database. Otherwise, it is possible that data from PTW software can no longer be exported to Track-it.

XML Structure

```
Measurements
   Measurement 1
      AdminData (administrative properties of a measurement)
      MeasData (measurement data, basis of the analysis data, option)
      AnalyzeData (analysis data of the measurement)

   Measurement 2
      AdminData
      MeasData
      AnalyzeData
   …
DataTypes
   DataType 1
   DataType 2
   …
RadiationUnits
   RadiationUnit 1 (radiation unit)
   …
MeasuringDevices [option]
   MeasuringDevice 1 (measuring device)
   …
MeasuringSoftwares [option]
   MeasuringSoftware 1 (analysis software)
   …
Limits [option]
   Limit 1 (limit values of the analysis data)
   …
```

Informal Description

PTW

PTW is the root element. It contains the format version ("Version"), the document creation date ("LastModified"), the optional author ID ("Author"), and the content area ("Content").

./Version

File format version number in the "major.minor" format.

Example:

```
<Version>1.2</Version>
```

./LastModified

Indication of date and time according to ISO 8601. The time must be expressed with the difference to the Coordinated Universal Time (UTC).

Example:

```
<LastModified>2014-01-17T10:59:26.0701348+01:00</LastModified>
```

./Author

[option]

Information about the source program that created the XML file, e.g., the name and version of the export component.

Example:

```
<Author>Track-it XML Creator 1.0.0.0</Author>
```

PTW/Content

Contains the descriptions of the measurements ("Measurements"), data types ("DataTypes"), radiation units ("RadiationUnits"), measuring devices "MeasuringDevices"), analysis software ("MeasuringSoftwares") used, and optional limit values ("Limits").

PTW/Content/Measurements

Container element for measurements ("Measurement").

PTW/Content/Measurements/Measurement

Contains administrative data ("AdminData"), measurement data ("MeasData"), and analysis values ("AnalyzeData") of a measurement.

Attributes:

radiation-unit-ref:
[required]
References one of the radiation units defined in section "RadiationUnits" where the appropriate attribute "id" is set.

measuring-device-ref:
[option]
References one of the measuring devices defined in section "MeasuringDevices" where the appropriate attribute "id" is set.

measuring-software-ref:
[option]
References one of the analysis software programs defined in section "MeasuringSoftwares" where the appropriate attribute "id" is set.

guid:
[required]
unique identifier of a measurement. When importing measurement data into the database, those with an ID identical to that of an existing measurement will be skipped. This means that existing measurements in the database will never be overwritten. The structure of this ID is not defined. It can be determined by the exporting program. The following points should be observed:

→ If no ID is specified, new measurements will be created in each case. This is usually not desirable because exporting the same measurement more than once is not unlikely (e.g., export of all measurements in overlapping time windows).

→ If a previously exported measurement is exported again, the same ID must be entered. Otherwise, the measurement would be registered as new in the database.

Proposals for IDs:

- A random GUID is generated and permanently retained together with the data by the exporting program so it can be entered again when the data are re-exported.
- The exporting program generates an ID composed of its own name and measurement parameters (e.g., time of the measurement, measuring device, …). A new ID can be generated for each export.
  Proposed format:
  `PROGRAM_NAME_TIME_STAMP(of the measurement, user-defined but unique to the seconds)_MEASURING_DEVICE_RADIATION_UNIT`
  e. g.
  `MultiCheck_2014-05-14T10:59:26_OCTAVIUS729_LinacA`.
  The database accepts almost all characters in an ID, including spaces and special characters. Only characters not allowed in XML attributes should not be used. However, a simple format should be selected for reasons of consistency and transparency.

```
<Measurement
  guid="3c3a3073-c7e6-4c35-a242-2866078027c8"
  radiation-unit-ref="Linac1"
  measuring-device-ref="Device1"
  measuring-software-ref="Software1">
  (…)
</Measurement>
```

PTW/Content/Measurements/Measurement/AdminData

Administrative data associated with a complete measurement. These are the measurement data ("Date"), a comment ("Comment", optional) as well as additional parameters that can be used as filter criteria ("Parameters").

./Date

Indication of date and time according to ISO 8601. The time must be expressed with the difference to the Coordinated Universal Time (UTC). If the UTC time difference is not indicated, the time difference of the current time zone will be applied during the import.

Example:

```
<Date>1999-12-31T23:59:59+01:00</Date>
```

./Comment

[option]

Free-text field for comments about the measurement.

Example:

```
<Comment>Comment</Comment>
```

PTW/Content/Measurements/Measurement/AdminData/Parameters

[option]

Parameters ("Parameter") associated with a measurement which can be used as filter criteria in Track-it for selection of the displayed analysis values.

./Parameter

[option]

Description of a parameter with the following possible attributes:

Attributes:

name:
[required, unique]
Name of the parameter as it appears on the Track-it user interface.

unit:
[option]
Unit of the values (any text).

valuetype:
[option]
One of the parameter value types defined in the appendix. This type determines the display formatting and input in a client. If a value type is not specified, the value "String" will be assumed.

precision :
[option]
Determines the display precision only for value type="Double". If this value is not specified, all values will be formatted with 3 decimal places.

The **value** of a parameter is indicated within a node, e. g. <Parameter>Photons</Parameter>

Examples:

```
<Parameter name="Modality" valuetype="Modality">Photons</Parameter>
<Parameter name="Energy" valuetype="Double" precision="1">6.0</Parameter>
<Parameter name="FFF" valuetype="Boolean">True</Parameter>
<Parameter name="Field size" unit="cm" valuetype="FieldSize">10x10</Parameter>
<Parameter name="SSD" valuetype="Double" unit="cm" precision="1">110.0</Parameter>
<Parameter name="Collimator angle" valuetype="Double" unit="°" precision="0">0</Parameter>
<Parameter name="Gantry angle" valuetype="Double" unit="°" precision="0">45</Parameter>
<Parameter name="Wedge angle" valuetype="Double" unit="°" precision="0">90</Parameter>
```

Additional Information for Attributes:

Parameter names (name)
To allow Track-it to display identical parameters from different applications together, **all the attributes** of the parameters must be identical.

Parameter value types (valuetype)
The display options for the parameters vary with the specified value type.

valuetype	Description
String	Text
Boolean	Boolean value: "True" or "False" example: contact open or closed
Long	Integer value
Double	Floating point value
Area	Field size Width and depth are saved, separated by an "x".
Modality	Modality list: "Photons", "Electrons", "Cobalt", "Protons", "Neutrons", "CIons", "HeavyParticle"

PTW/Content/Measurements/Measurement/MeasData

[option]

Grouping of all measurement data sets ("MeasValues") on which the analysis is based.

PTW/Content/Measurements/Measurement/MeasData/MeasValues

[option]

Contains the values associated with a scan or measuring quantity.

Attributes:

name:
[required]
Name of the value on the display, e. g., "Profile Gun-Target"

type:
[required]
One of the following measurement data types. This type defines the display format in the program.

Additional Information for Attributes:

Measurement data types (type)
The display options vary with the specified measurement data type.

type	Description
String	Character string The character string must be available as a UTF-8 encoded string that, like all other values, must subsequently be encoded in Base64.
Boolean	Boolean value: 0 = false, otherwise true
Long	Integer value
Double	Floating point value
Profile	Profile with values, positions, and chamber numbers
PDD	Depth dose curve with values and positions
UserDefined	Binary data set. If the *name* attribute contains a file name, the appropriate application will be started according to the file extension.

All numeric values must be available as double arrays (64-bit floating point numbers).

./Values

Contains Base64-encoded values. Directly supported are series of measured values (depth dose curves, profiles, …), single measured values (temperature, air pressure, …), character strings, and binary data.

Attributes:

unit:
[option]
Unit of the values.

Example:

```
<Values unit="Gy">AAADwPwAA(…)AADAWEA=</Values>
```

./Positions

Contains Base64-encoded position values.

In addition to the values ("Values"), position information ("Positions") can be provided. The number of position values must be the same as that of measured values. Therefore, they only make sense for measured values of the "Profile" or "PDD" type (refer to section "Values").

Attributes:

unit:
[option]
Unit of the values.

Example:

```
<Positions unit="mm">AAAAAAB(...)AAAAAAQGpA</Positions>
```

PTW/Content/Measurements/Measurement/AnalyzeData

[option]

Grouping of all analysis values ("AnalyzeValue") of a measurement.

PTW/Content/Measurements/Measurement/AnalyzeData/AnalyzeValue

[option]

An analysis value of a measurement. Only these values can be displayed over time.

Attributes:

data-type-ref:
[required]
References one of the data types defined in section "DataTypes" where the appropriate attribute "id" is set.

Example:

```
<AnalyzeValue data-type-ref="foo">(…)</AnalyzeValue>
```

./Value

[required]

Single floating point value.

The following values can be used for values assigned to a data type of "Boolean" value type:

- "0" or "False"
- "1" or "True"
- "2" or "Warning"

Example:

```
<Value>3.14</Value>
```

./Comment

[option]

Free-text field for comments about the analysis value.

Example:

```
<Comment>Comment</Comment>
```

PTW/Content/DataTypes

[option]

Contains descriptions of data types ("DataType") that may be referenced by analysis data and limit values.

These data types will not overwrite data types that already exist in the database. In this case, measurement data will be linked with the existing data types in the database.

PTW/Content/DataTypes/DataType

[option]

Its name ("Name") and its definition ("Definition") are unique identifiers of a data type.

Attributes:

id:
[required]
In the XML file, data types are identified by means of the "id", when referenced from other elements. The content of this attribute is not saved in the database. Therefore, when generating the XML file, the attribute can be selected from all characters permitted in XML attributes.

Proposals:

- Progressive counter
- Combination of data type name and data type definition

Example:

```
<DataType id="foo">
  (…)
</DataType>
```

./Name

[required, unique with definition]

Name of the data type.

Example:

```
<Name>Flatness</Name>
```

To allow Track-it to display identical analysis parameters from different applications all together, **all attributes** of a data type (combination of "Name" and "Definition") must be identical.

./Definition

[option, unique with name]

The definition from the standard by which the values of this type are analyzed (unique when combined with "Name").

Example:

```
<Definition>IEC 60976</Definition>
```

./Unit

[option]

Unit of the corresponding analysis value. Indicated as a character string.

Example:

```
<Unit>Gy/min</Unit>
```

./ValueType

[option]

The category of data associated with the data type. One of the types defined subsequently.
If a value type is not specified, the value "Double" will be assumed.

Example:

```
<ValueType>Double</ValueType>
```

The display options for the analysis data vary with the specified data type.

ValueType	Description
Boolean	Logical value, e.g., contact closed or open
Long	Integer value
Double	Floating point value

./Precision

[option]

Display precision for value type "Double". Three decimal places will be used unless specified otherwise.

Example:

```
<Precision>3</Precision>
```

PTW/Content/RadiationUnits

[required]

Contains descriptions of radiation units ("RadiationUnit") that may be referenced by measurements.

PTW/Content/RadiationUnits/RadiationUnit

[required]

Its name ("Name") is the unique identifier of a radiation unit.

Attributes:

id:
[required]
Radiation units are identified by means of their "id" attribute when referenced from other elements. The content of the attribute is not saved in the database. It is only relevant for processing of the XML file. Therefore, when generating the XML file, the attribute can be selected from all characters permitted in XML attributes.

Proposals:

- Progressive counter
- Name of the radiation unit

Example:

```
<RadiationUnit id="Linac1">
  (…)
</RadiationUnit>
```

./Name

[required, unique]

The name of the radiation unit.

Example:

```
<Name>Linac A</Name>
```

PTW/Content/MeasuringDevices

[option]

Contains descriptions of measuring devices ("MeasuringDevice") that may be referenced by measurements.

PTW/Content/MeasuringDevices/MeasuringDevice

[option]

Its name ("Name") is the unique identifier of a measuring device.

Attributes:

id:
[required]:
Measuring devices are identified by means of their "id" attribute when referenced from other elements. The content of the attribute is not saved in the database. It is only relevant for processing of the XML file. Therefore, when generating the XML file, the attribute can be selected from all characters permitted in XML attributes.

Proposals:

- Progressive counter
- Name of the measuring device

Example:

```
<MeasuringDevice id="Device1">
  (…)
</MeasuringDevice>
```

./Name

[required, unique]

The name of the measuring device.

Example:

```
<Name>QUICKCHECK webline</Name>
```

PTW/Content/MeasuringSoftwares

[option]

Contains descriptions of analysis software programs ("MeasuringSoftware") that may be referenced by measurements.

PTW/Content/MeasuringSoftwares/MeasuringSoftware

[option]

Its name ("Name") is the unique identifier of an analysis software.

Attributes:

id:
[required]
Software programs are identified by means of their "id" attribute when referenced from other elements. The content of the attribute is not saved in the database. It is only relevant for processing of the XML file. Therefore, when generating the XML file, the attribute can be selected from all characters permitted in XML attributes.

Proposals:

- Progressive counter
- Name of the analysis software

Example:

```
<MeasuringSoftware id="Software1">
  (…)
</MeasuringSoftware>
```

./Name

[required, unique]

The name of the analysis software.

Example:

```
<Name>QUICKCHECK</Name>
```

PTW/Content/Limits

[option]

Optional limit value definitions.

PTW/Content/Limits/Limit

[option]

A limit consists of a lower limit ("LimitLower") and an upper limit ("LimitUpper") that serve as statistical analysis parameters. Similar to the identical section "Parameters", validity restrictions for a limit can be specified in the measurement node.

Attributes:

data-type-ref [required]:
Unique identification of the software. References one of the data types defined in section "DataTypes" where the appropriate attribute "id" is set.

radiation-unit-ref [option]:
Link to a radiation unit

measuring-device-ref [option]:
Link to a measuring device

An import will not overwrite limits with the same filter parameters and references that already exist in the database. This ensures that user-defined limits in the database will not be modified.

Example:

```
<Limit data-type-ref="foo" radiation-unit-ref="Linac1" measuring-device-ref="Device1">
  (…)
  <Parameters>
    <Parameter name="Modality" valuetype="Modality">Electrons</Parameter>
    <Parameter name="Energy" valuetype="Double" precision="1">12</Parameter>
    <Parameter name="FFF" valuetype="Boolean">False</Parameter>
  </Parameters>
</Limit>
```

./LimitLower
./LimitUpper

Limits for evaluation and display of analysis values. Indicated as a numeric value.

Example:

```
<LimitLower>0.0</LimitLower>
<LimitUpper>2.0</LimitUpper>
```

./Name

An optional name for the limit.

Example:

```
<Name>Flatness Electrons</Name>
```

PTW/Content/Limits/Limit/Parameters

[option]

Refer to PTW/Content/Measurements/Measurement/AdminData/Parameters

./Parameter

[option]

Refer to PTW/Content/Measurements/Measurement/AdminData/Parameters

Encoding

XML Character Set

The UTF-8 character set is used.

Floating point numbers

Floating point numbers use the point as the decimal separator. Numbers can be written in decimal form or scientific notation.

Base64

For encoding, the values must be available as double arrays (64-bit floating point numbers) in the little-endian format.

Date indication

Date and time are always indicated according to ISO 8601. They must be expressed with the difference to the Coordinated Universal Time (UTC).

Sample file

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<PTW>
  <Version>1.0.0.0</Version>
  <LastModified>2016-03-16T12:02:28.3479937+01:00</LastModified>
  <Author>QcwToTrackItConverter, Version=1.0.0.169</Author>
  <Content>
    <DataTypes>
      <DataType id="flatness2d(relative)_iec60976">
        <Name>Flatness 2D (relative)</Name>
        <ValueType>Double</ValueType>
        <Definition>IEC 60976</Definition>
      </DataType>
      </DataTypes>
    <Limits>
      <Limit data-type-ref="flatness2d(relative)_iec60976" measuring-device-ref="1" measuring-software-ref="1" radiation-unit-
ref="1">
        <LimitLower>9.8000E+01</LimitLower>
        <LimitUpper>1.0200E+02</LimitUpper>
        <BaseLine>1.0000E+02</BaseLine>
        <Parameters />
      </Limit>
      </Limits>
    <RadiationUnits>
      <RadiationUnit id="1">
        <Name>TB1</Name>
      </RadiationUnit>
    </RadiationUnits>
    <MeasuringDevices>
      <MeasuringDevice id="1">
        <Name>QUICKCHECK webline</Name>
      </MeasuringDevice>
    </MeasuringDevices>
    <MeasuringSoftwares>
      <MeasuringSoftware id="1">
        <Name>QUICKCHECK</Name>
      </MeasuringSoftware>
    </MeasuringSoftwares>
    <Measurements>
      <Measurement guid="1344951372" radiation-unit-ref="1" measuring-device-ref="1" measuring-software-ref="1">
        <AdminData>
          <Date>2012-08-14T13:36:12.0000000+02:00</Date>
          <Comment></Comment>
          <Parameters>
            <Parameter name="Modality" valuetype="Modality">Electrons</Parameter>
            <Parameter name="Energy" valuetype="Double" unit="MV/MeV" precision="1">6</Parameter>
            <Parameter name="Field size" valuetype="Area" unit="cm x cm">20.0x20.0</Parameter>
            <Parameter name="Field shape" valuetype="String">square</Parameter>
            <Parameter name="Gantry angle" valuetype="Double" unit="°" precision="0">0</Parameter>
            <Parameter name="Wedge angle" valuetype="Double" unit="°" precision="0">0</Parameter>
            <Parameter name="SDD" valuetype="Double" unit="mm" precision="0">1000</Parameter>
          </Parameters>
        </AdminData>
        <AnalyzeData>
           <AnalyzeValue data-type-ref="flatness2d(relative)_iec60976">
            <Value>0.0000E+00</Value>
          </AnalyzeValue>
        </AnalyzeData>
        <MeasData>
          <MeasValues name="G10 dose" type="Double">
            <Values unit="Gy">YHZPHhbqAEA=</Values>
          </MeasValues>
          <MeasValues name="L10 dose" type="Double">
            <Values unit="Gy">P1dbsb/sAEA=</Values>
          </MeasValues>
          <MeasValues name="T10 dose" type="Double">
            <Values unit="Gy">bVZ9rrbiAEA=</Values>
          </MeasValues>
          <MeasValues name="R10 dose" type="Double">
            <Values unit="Gy">+FPjpZvEAEA=</Values>
          </MeasValues>
          <MeasValues name="Temperature" type="Double">
            <Values unit="°C">WDm0yHbeOEA=</Values>
          </MeasValues>
          <MeasValues name="Pressure" type="Double">
            <Values unit="hPa">mpmZmZnpjkA=</Values>
          </MeasValues>
          <MeasValues name="Device ID 1" type="String">
            <Values unit="">UVVJQ0tDSEVDSyB3ZWJsaW5lIDU1Nw==</Values>
          </MeasValues>
          <MeasValues name="Software ID 1" type="String">
            <Values unit="">UVVJQ0tDSEVDSyAxLjUuMQ==</Values>
```

```
          </MeasValues>
        </MeasData>
      </Measurement>
    </Measurements>
  </Content>
</PTW>
```

BASE64 Decoding C# Sample

```
namespace PTWBase64Example
{
     using System;

     class Program
     {
           static void Main()
           {
                  short l_sBase64Short;
                  float l_fBase64Float;

                  // Base 64 short example
                 string l_strBase64Short = "OTA="; // 12345
                  byte[] l_oBase64Short = Convert.FromBase64String(l_strBase64Short);
                  l_sBase64Short = BitConverter.ToInt16(l_oBase64Short, 0);
                  Console.WriteLine(l_sBase64Short);

                  // Base 64 float example
                  string l_strBase64Float = "1en2Qg=="; // 123.4567
                  byte[] l_oBase64Float = Convert.FromBase64String(l_strBase64Float);
                  l_fBase64Float = BitConverter.ToSingle(l_oBase64Float, 0);
                  Console.WriteLine(l_fBase64Float);

                  // Base 64 short array example
                  string l_strBase64ShortArray = "0gQuFg=="; // 1234;5678
                  byte[] l_oBase64ShortArray = Convert.FromBase64String(l_strBase64ShortArray);
                  for (int i = 0; i < 2; i++)
                  {
                         l_sBase64Short = BitConverter.ToInt16(l_oBase64ShortArray, sizeof(short) * i);
                         Console.WriteLine(l_sBase64Short);
                  }

                  // Base 64 float array example
                  string l_strBase64FloatArray = "1en2QluRakM="; // 123.4567;234.5678
                  byte[] l_oBase64FloatArray = Convert.FromBase64String(l_strBase64FloatArray);
                  for (int i = 0; i < 2; i++)
                  {
                         l_fBase64Float = BitConverter.ToSingle(l_oBase64FloatArray, sizeof(float) * i);
                         Console.WriteLine(l_fBase64Float);
                  }
           }
     }
}
```

Transferring an XML File to Track-it

PTW programs transfer the analysis results with the function "Export to Track-it". XML files can also be transferred directly to Track-it by using the Tool "TrackItExporter.exe". PTW applications that support exporting to Track-it install this tool in the following directory:

`C:\Program Files (x86)\PTW\Tool\ExportToDatabase`

All transfer parameters can optionally be passed as command line arguments.

- `-i`: Path to the XML input file
- `-o`: URL of the Track-it server (port must be specified), e.g. http://localhost:8080
- `-u`: Track-it user name
- `-p`: Password of the Track-it user
- `-m`: Show or suppress the graphical user interface. `0`: show user interface (default); `1`: suppress user interface
- `-h`: Show command line options and possible return values.

Examples:

Export a file to the Track-it server. The user interface requests the user name and password if needed.

```
TrackItExporter.exe -i "C:\Path\To\MyTrackItExport.xml" -o "http://localhost:8080"
```

Export without graphical user interface. In this example `start /wait` is used to be able to output the return value (`%errorlevel%`) after the transfer completed or failed.

```
start /wait TrackItExporter.exe -i "C:\Path\To\MyTrackItExport.xml" -o "http://localhost:8080" -u
Administrator -p Track-it -m 1
echo %errorlevel%
```

Important:

⚠ CAUTION

Transferring data with unsuitable formatting.

Incorrect Operation!

PTW programs already export a large number of data types and parameters. It is mandatory to use the symbol "*" as a prefix if you want to transfer your own data to the Track-it database. Otherwise, it is possible that data from PTW software can no longer be exported to Track-it.

You will find the list of data types and parameters exported by PTW programs in the appendix of the user manuals concerned.

Example of a custom data type:

```
<DataType id="MyData">
  <Name>*MyFlatness</Name>
  <Definition>*MyProtocol</Definition>
</DataType>
```