

# **The IBA Data Format IDF**

version 0.0 - 08 September 2009

# CONTENTS

<b>1</b>	<b>A UNIVERSAL DATA FORMAT FOR ION BEAM ANALYSIS</b>	<b>3</b>
1.1	INTRODUCTION	3
1.2	THE IBA DATA FORMAT GENERAL PHILOSOPHY	4
1.3	EXTENSIBILITY	6
1.4	QUANTITIES AND UNITS	9
1.5	DESCRIPTION OF THE IDF STRUCTURE	11

# 1 A universal data format for Ion Beam Analysis

## 1.1 Introduction

Data acquisition systems and data analysis codes all implement different formats to represent and store the data, the experimental conditions, any analysis parameters, the description of the sample, and others. Data acquisition systems normally support specific (sometimes proprietary) formats, and experimentalists often convert these to their own favourite formats. Popular codes developed their own data formats to read and write all this information. This is a strong barrier against interchange of data between people from different laboratories, or that use different set-ups to measure their data, different computers to store their data, and different data analysis programs to analyse their data.

We propose and describe a new IBA data format, that aspires to support all forms of IBA data in a well defined simple format permitting easy exchange between sites and analysis programs. Its main characteristics should be: 1) It can be read and understood by inspection of the file, i.e. it is ASCII. This implies it is portable amongst all hardware platforms, and it is self-explanatory 2) It is flexible enough to cater to all IBA data needs, so a file entry can correspond to one given sample, or to a group of similar samples, or to one experimental run, or it may contain simulations only, or anything else that the user may need. 3) It is easily extendable; while the most commonly used features are already included in the new data format, new needs are easy to accommodate. For instance, most data analysis codes use similar layered sample descriptions, but the codes treat roughness or equation overlays in different non-equivalent ways; code-specific extensions to the basic data format are allowed, solving this problem.

The International Atomic Energy Agency sponsored a review of existing codes for IBA data analysis, and then an intercomparison exercise of the different codes [1,2]. In the first meeting of the IAEA Intercomparison of IBA Software exercise [3], that took place in December 2003, it was already recognised the need for an IDF, and also that XML [4] fulfilled the desired characteristics. A final decision to establish the IDF was taken when the development of a universal IBA data format became the first milestone of the European Commission-supported SPIRIT [5] Integrated Infrastructure Initiative, which integrates 11 leading European ion beam facilities from six member states and two associated states.

A similar approach was taken by the NeXus International Advisory Committee, that develops NeXus, an XML-based common data format for neutron, x-ray, and muon science [6]. The possibility of integrating IBA-specific requirements in NeXus was considered, as NeXus is being developed since 1994 by a large group of developers and users and offers many software facilities. This was discarded because NeXus is oriented towards instruments, i.e. experimental facilities, which are notoriously different in IBA and in the neutron and x-ray scattering world; to establish an IBA extension to NeXus, each separate IBA instrument would have to be defined and submitted to the NeXus International Advisory Committee (NIAC) for revision and ratification, which is a process that typically takes years,

even not considering that the NIAC holds no IBA expertise.

## ***1.2 The IBA Data Format general philosophy***

The extensible mark-up language (XML) was developed by the World Wide Web Consortium (W3C) to structure, store, and transport data. Although often associated with the internet, XML is by no means a specific internet language, and there are many non-internet XML applications.

In particular, XML has no pre-defined tags (keywords to carry specific information). In fact, the IDF is an XML schema, that is, a definition of all the elements (also called tags) allowed, how they are structured, which information they can contain, and how it can be extended. A given file is an IDF file if it is validated against the IDF schema. The schema contains no specific information about any experiment; it just defines the structure of the IDF files. In fact, the IDF schema *is* the definition of the IDF data format.

For instance, we show in Fig. 1a) an IDF file, where most elements are collapsed and their content is not shown. The only elements that are fully expanded are those that define the beam (note that in a complex case, the “beam” group would include definition of foils, slits, charge state distribution, and others). We show in Fig. 1b) the part of the IDF schema that defines the element “spectrum”, which normally will contain all the experimental conditions as well as the spectrum as such, and also any data processing made, such as a fitted curve together with any scattering cross sections or stopping powers used in the calculation (and will not contain information about the sample structure, reserved for the element “structure” defined elsewhere). This element “spectrum” structures the relevant information in daughter (higher-level) elements such as “beam”, “geometry”, “data”, or “process”. Each one of these is defined elsewhere in the schema, and contains further elements in a structured well-defined way. Note that not all tags defined in the schema need to be in the IDF file.

With this example in mind, the general philosophy of IDF is:

- The basic elements of IDF are data items and data groups. Data items are e.g. “layerthickness”, or “concentration”; these data items can be real numbers, integer, or even text (e.g. “geometrytype” can be “IBM”, “Cornell”, or “general”). Data groups are logically organised (e.g. “geometry”, “stoppingpowers”, or “structure”), and can contain other data groups as well as data items.

- The information is structured as a tree. Each data group is a node of the tree. Standard tree parsing methods can be applied. The root element is always “idf”.

- The elements appear, inside the group to which they belong, in fixed order that must be respected. These pre-defined elements support most, but not all, IBA needs. However, all elements are optional, they need not appear. The apparent complexity of the data groups does not in fact exist in normal cases and simple experiments.

a)

```
-<idf>
+<users></users>
+<notes></notes>
-<sample>
+<users></users>
+<notes></notes>
+<elementsandmolecules></elementsandmolecules>
+<structure></structure>
-<spectra>
- <spectrum>
+ <notes></notes>
- <beam>
<beamparticle>4He</beamparticle>
<beamZ>2</beamZ>
<beammass units="amu">4.0</beammass>
<beamenergy units="keV">1500.</beamenergy>
<beamfluence units="uC">10</beamfluence>
</beam>
+ <geometry></geometry>
+ <instrument></instrument>
+ <calibrations></calibrations>
+ <reactions></reactions>
+ <process></process>
</spectrum>
</spectra>
</sample>
</idf>
```

b)

```
<xs:element name="spectrum">
  <xs:complexType>
    <xs:sequence>
      <xs:group ref="usersnotes"/>
      <xs:element ref="log" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="environment" minOccurs="0" maxOccurs="1"/>
      <xs:group ref="experimental"/>
      <xs:element ref="reactions" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="data" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="process" minOccurs="0" maxOccurs="1"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:group name="experimental">
  <xs:sequence>
    <xs:element ref="beam" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="geometry" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="instrument" minOccurs="0" maxOccurs="1"/>
    <xs:element ref="calibrations" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:group>
```

Fig. 1 a). An IDF file, where most elements are collapsed and their content is not shown. b) Part of the IDF schema that defines the element "spectrum"

- Each data group can start with the elements “users” and “notes”. The group “users” should be for information on e.g. sample owner, experimentalist, data analyst.

- Any number of non-IDF elements can be inserted in any data group, at the end. They are clearly identified with a prefix, that is unique to each specific code or user that wishes to have their own elements. For instance, all NDF specific tags will be identified with the prefix “NDF:”. This makes the IDF fully flexible and extensible.

- All elements that describe a quantity have optional attributes such as “quantity” (e.g. “energy”, “thickness”, “arealdensity”) “units” (e.g. “keV”, “nm”, “1e15at/cm2”) or “uncertainty” (and even “uncertaintyA” and “uncertaintyB” for types A and B, respectively, and “uncertaintyplus” and “uncertaintyminus” for the upper and lower limits, respectively).

- All elements that describe a spread have as optional attribute “mode”, which can be “FWHM”, “sigma”, or “variance”.

- Not all units are allowed, since it would lead to severe parsing problems by software. For each quantity, only certain units are allowed. This is described in full below.

- The IDF allows two basic ways of structuring data and information. The first one has as top level tag “sample”; all the information relating to one given sample is given, including structure, any spectra measured (which can be more than one, or even none if it is a pure simulation), all experimental conditions for the different measurements, all calculations and simulations. In an IDF file, there can be any number of “sample” entries. The second way is batch mode; this is appropriate when for instance many different samples were measured with the same or similar experimental conditions (as in an experimental run). This way is more concise, as information on the experimental conditions does not need to be repeated for each sample measured. A full explanation is given below.

### ***1.3 Extensibility***

As said above, any number of non-IDF elements can be inserted in any data group, at the end. They are clearly identified with a prefix, that is unique to each specific code or user that wishes to have their own elements. For instance, all NDF specific tags will be identified with the prefix “NDF:”. This makes the IDF fully flexible and extensible.

In XML, extensibility is provided by the schema, that can allow (or forbid) elements not defined in the schema to be included in the instance files. In IDF, this is allowed, but the new elements must be defined in a separate, user-defined, schema. That is, user-specific tags must be defined in a user-specific schema, which is then referred to in the instance file.

For instance, to include NDF-specific tags, the root element “idf” becomes

```

<idf
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://idf.schemas.itn.pt idfv0.xsd http://ndf.schemas.itn.pt ndfv0.xsd"
xmlns="http://idf.schemas.itn.pt"
xmlns:idf="http://idf.schemas.itn.pt"
xmlns:ndf="http://ndf.schemas.itn.pt ">

```

The second line (`xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`) is required by XML, so that the XML expression `"xsi:schemaLocation"` can be used in the third line (`xsi:schemaLocation=" http://idf.schemas.itn.pt idfv0.xsd http://ndf.schemas.itn.pt ndfv0.xsd"`).

In this third line, we associate to the IDF file two schemas: the standard IDF schema `idfv0.xsd`, and the schema for NDF, `ndfv0.xsd`. Note that each of these schemas comes with an associated Uniform Resource Identifier (URI, which is a string that identifies an internet resource; the most common URI is the Uniform Resource Locator (URL) which identifies an internet domain address). URIs are used because they are unique; they are a way to uniquely identify one group of elements. The URI associated to a given schema does not even need to actually exist, it just needs to be unique. However, it is common to give in the URI documentation and information related to the schema.

The fourth line (`xmlns="http://idf.schemas.itn.pt"`) serves to define that all tags that do not have user-specific prefixes belong to the IDF schema.

The fifth line (`xmlns:idf="http://idf.schemas.itn.pt"`) says basically the same as the fourth, but it explicitly defines the `"idf"` prefix.

The sixth line (`xmlns:ndf="http://ndf.schemas.itn.pt"`) specifies that the the prefix `"ndf"` is to be used for all tags that belong to the NDF schema.

This leads, for instance in calculation 25 of the IAEA intercomparison, that requires code-specific straggling options, to have:

```

- <idf xsi:schemaLocation="http://schemas.itn.pt/ndf/Nuno/TBA_UDF/test1wb.xsd http://schemas.itn.pt/ndf/NDF/
+ <users></users>
+ <notes></notes>
- <sample>
+ <users></users>
+ <notes></notes>
+ <elementsandmolecules></elementsandmolecules>
+ <structure></structure>
- <spectra>
- <spectrum>
+ <notes></notes>
+ <beam></beam>
+ <geometry></geometry>
+ <instrument></instrument>
+ <calibrations></calibrations>
+ <reactions></reactions>
- <process>
+ <crosssections></crosssections>
+ <stoppingpowers></stoppingpowers>
+ <energyspread></energyspread>
+ <simulations></simulations>
- <ndf:stragglings>
- <ndf:stragglingsdefault>
+ <ndf:stragglingsfactor units="none">1.</ndf:stragglingsfactor>
+ <ndf:stragglingsoffset units="keV">0.</ndf:stragglingsoffset>
- <ndf:stragglingsmodel>
+ <ndf:code>wdepth</ndf:code>
- <ndf:wdepthparameters>
+ <ndf:wdepthfile>26.dpt</ndf:wdepthfile>
+ <ndf:wdepthenergylevel>0</ndf:wdepthenergylevel>
</ndf:wdepthparameters>
</ndf:stragglingsmodel>
</ndf:stragglingsdefault>
</ndf:stragglings>
+ <ndf:doublesscattering>normal</ndf:doublesscattering>
</process>
</spectrum>
</spectra>
</sample>
</idf>

```

Fig. 2. Part of the IDF file created by NDF for calculation 25 of the IAEA intercomparison exercise, including NDF-specific tags, identified by the prefix NDF. (Note that the schema location is not the same as describe above, because to produce the figure we used local files.)



## 1.4 Quantities and units

Accepting all possible units for all quantities would lead to a nearly impossible task of supporting and converting different sets of units. Instead, a reduced set of units for each quantity was defined as standard in IDF. On top of these, all quantities can be given with units “other”, “arbitrary”, and “none”. The IDF standard accepted units are (u stands for micro):

### time

s  
ns  
us

### density

g/cm<sup>3</sup>  
1e22at/cm<sup>3</sup>

### pressure

atm  
bar  
mbar  
Torr  
mTorr  
mmHg

### temperature

°C  
K

### solid angle

sr  
msr  
srad  
msrad

### angle

degree  
rad  
mrad

### fluence

uC  
puC (particle micro Coulomb)  
C  
pC (particle Coulomb)  
uCoulomb  
puCoulomb (particle micro Coulomb)  
Coulomb  
pCoulomb (particle Coulomb)  
#particles

### current

nA  
nAmpere  
Ampere

### mass

amu  
g  
kg

### charge

q

**stopping power**

eV/A  
keV/A  
MeV/A  
eV/Angstrom  
keV/Angstrom  
MeV/Angstrom  
eV/nm  
keV/nm  
MeV/nm  
eV/um  
keV/um  
MeV/um  
eV/cm  
keV/cm  
MeV/cm  
eV/m  
keV/m  
MeV/m  
eV/(ug/cm2)  
keV/(ug/cm2)  
MeV/(ug/cm2)  
eV/(1e15at/cm2)  
keV/(1e15at/cm2)  
MeV/(1e15at/cm2)

**cross section**

barn  
b  
mbarn  
mb  
cm2  
rr (ratio to Rutherford)

**concentration**

at%  
mol%  
wt%  
fraction  
relative

**energy**

eV  
keV  
MeV

**areal density**

ug/cm2  
1e15at/cm2

**thickness, length**

A  
Angstrom  
nm  
um  
mm  
cm  
m

Note however that quantities that describe the sample structure, thickness of foils, can be given both as areal density and as thickness.

## 1.5 Description of the IDF structure

Sentences in italics are comments about a given group of entries. Cursive text describes the contents of a given tag.

It should be kept in mind that all elements that describe a quantity have the following optional attributes, in order:

quantity - e.g. “energy”, “thickness”, arealdensity”

units - e.g. “keV”, “nm”, “1e15at/cm2”

uncertainty

uncertaintyA - for type A uncertainty

uncertaintyB - for types B uncertainty

uncertaintyplus - for the upper limit of the quantity

uncertaintyminus - for the lower limit of the quantity

Furthermore, the elements that describe a spread (such as the beam energy spread, beam angular spread, energy resolution, layer thickness inhomogeneity), have the following optional attribute, located between “quantity” and “units”:

mode - options are “FWHM”, “sigma”, and “variance”

`<idf` *this is the root element of the IDF data format. It includes technical information on the schema and namespace URI used, that we omit here for clarity.*

`<users` *every single group of tags can start with the group “users”, where information on e.g. sample owner, experimentalist, data analyst can be given. Any number of tags “user” can be given. We will omit this group from most of this description for the sake of conciseness.*

`<user` name of user, such as sample owner, experimentalist, data analyst `>user`

`>users`

`<notes` *every single group of tags can have the group “notes” just after the group “users”, where any information or comments can be given. Any number of tags “note” can be given. We will omit this group from most of this description for the sake of conciseness.*

`<note` any text `>note`

`>notes`

`<attributes` *these are attributes of the IDF file*

`<users` `>users`

`<notes` `>notes`

`<idfversion` version of IDF that this file conforms to `>idfversion`

`<filename` name of the IDF file `>filename`

`<createtime` Date file was created in XML format: 2002-05-30 or 2002-05-30T09:00:00 `>createtime`

`<updatetimes` any number of “updatetime” tags can be given.

`<users` `>users`

`<notes` `>notes`

`<updatetime` Date file was updated `>updatetime`

`>updatetimes`

`>attributes`

`<sample` *The IDF allows two basic ways of structuring data and information. The first one has as top level tag “sample”; all the information relating to one given sample is given, including structure, any spectra measured (which can be more than one, or even none if it is a pure simulation), all experimental conditions for the different measurements, all calculations and simulations. In an IDF file, there can be any number of “sample” entries.*

`<elementsandmolecules` *here, the elements and molecules present in the sample are given. If both “elements” and “molecules” are present, care should be taken to ensure that they are consistent with each other*

«nelements» number of elements present; redundant with content of “elements” tag «nelements»

«elements» here, one “element” tag is given per element present in the sample

«element»

«name» element present, such as  $^{28}\text{Si}$ , Si (natural isotopic distribution), or natSi «name»

«density» density of the element «density»

element»

elements»

«nmolecules» number of molecules present; redundant with content of “molecules” tag «nmolecules»

«molecules» here, one “molecule” tag is given per molecule present in the sample. “molecule” can also be taken to be a single element. The “molecules” group should have complete information about all the sample components, for instance Si 1 O 2, Si, Ti 1 O 2 and Ti could be the “molecule” tags defined.

«molecule» one given molecule (or element) is given

«name» molecule present with element name and stoichiometry separated by a white space, e.g. Si 1 O 2; SiO<sub>2</sub> not allowed since it would lead to confusions in molecules with specified isotopes

such as Si 1 <sup>18</sup>O 2 «name»

«density» density of the molecule «density»

molecule»

molecules»

elementsandmolecules»

«structure» here the structure of the sample is given (normally this is a result of processing and analysing the data, so it could be in the “process” group below). The normal way of describing a sample structure is a layered description. A second way is the so-called “point by point” profile, in which for each element a table (depth, concentration) is calculated by directly converting the yield in each channel into a concentration. This second method is popular for instance in HI-ERDA.

«crystalstructure» text describing the crystalline structure of the sample «crystalstructure»

«layeredstructure» this is the traditional layered structure

«nlayers» number of layers in the sample. Not strictly necessary, since it should match the number of layer entries; if it does not, it is ignored and the number of layer entries should be taken «nlayers»

«layers» contains several “layer” tags, in order of appearance: the first layer entry is the surfacemost layer, the last is the bottommost (substrate) layer-

«layer» a given layer

«layerthickness» layer thickness «layerthickness»

«layeruniformity» uniformity of the layer thickness «layeruniformity»

«layerdensity» layer density; this overrides what was given for elements or molecules «layerdensity»

«layerelements» the elements present in the layer are given

«layerelement» one given element present in the layer

«name» element present «name»

«concentration» concentration of the element «concentration»

layerelement»

layerelements»

«layermolecules» the molecules present in the layer are given. If both “layerelements” and “layermolecules” are present, care should be taken to ensure that they are consistent with each other

`<layermolecule` *one given molecule present in the layer*  
`<name` *molecule present* `<name`  
`<concentration` *concentration of the molecule* `<concentration`  
`>layermolecule`  
`>layermolecules`  
`>layer`  
`>layers`  
`<equationoverlays` *where code-specific equation description of concentrations is to be included* `<equationoverlays`  
`<roughness` *where code-specific description of roughness is to be included* `<roughness`  
`>layeredstructure`  
`<pointbypointstructure` *this is the second way of giving a sample definition. This can very well coexist with the layered description; for instance, the layered description may have been used to produce a fit, while nevertheless a point by point yield calculation was performed*  
`<elements` *the point by point profile is given separately for each element*  
`<element` *point by point profile for one given element*  
`<elementname` *element present* `<elementname`  
`<pbp` *this is the point by point profile as such, containing as many pbpitem as there are points in the “depth, concentration” table for this element*  
`<pbpitem` *one point by point depth/concentration item*  
`<depth` *depth for which concentration is given* `<depth`  
`<concentration` *concentration of the element at the given depth* `<concentration`  
`>pbpitem`  
`>pbp`  
`>element`  
`>elements`  
`>pointbypointstructure`  
`>structure`  
`<spectra` *as many spectra as desired can be given*  
`<spectrum` *here one spectrum (experimental and all calculations based on it) is given, including all experimental conditions.*  
`<log` *Log parameters such as real and live time are often useful in the calculation of effective charge, as well as of pile-up*  
`<realtime` *real time, in XML period of time format e.g. PT1H25M21S* `<realtime`  
`<livetime` *live time, in XML period of time format e.g. PT1H25M10S* `<livetime`  
`<deadtime` *dead time, in XML period of time format e.g. PT11S* `<deadtime`  
`<starttime` *time at which the experiment started e.g. 2002-05-30T09:00:00* `<starttime`  
`<stoptime` *time at which the experiment ended* `<stoptime`  
`>log`  
`<environment` *“temperature” and “pressure” are not very useful parameters. The “environment” group is here so users who have special requirements have a place to introduce user-specific environment-related tags.*  
`<temperature` *temperature during experiment* `<temperature`

« pressure » pressure during experiment » pressure »

environment »

« beam » *the “beam” group includes things that respect to the beam before it hits the sample, so including any slits and foils located before the sample*

« beamparticle » full isotope should be specified e.g. 4He » beamparticle »

« beamZ » atomic number of beam particle, this is redundant with beamparticle » beamZ »

« beammass » atomic mass of beam particle » beammass »

« beamenergy » initial beam energy; for RNRA, NRP, PIGE, and techniques where the beam energy is scanned, this tag should be left unused » beamenergy »

« beamenergyspread » initial energy spread of the beam » beamenergyspread »

« beamchargestate » *this is the initial beam charge state before the beam hits any foils or the sample. It can be a single value or a distribution.*

« beamchargestates » *as many beamcsstate entries are given as charge states of the initial beam. The fraction needs not be normalised*

« beamcsstate » *one single beam charge state with the corresponding fraction*

« beamcscharge » a given charge state of the beam » beamcscharge »

« beamcsfraction » initial fraction of the beam with the given charge state » beamcsfraction »

beamcsstate »

beamchargestates »

beamchargestate »

« beamfluence » total accumulated beam fluence (usually known as “charge”) » beamfluence »

« beamcurrent » average beam current during the experiment » beamcurrent »

« beamangularspread » initial angular spread of the beam » beamangularspread »

« beamshape » *this is the beam shape before any slits; if no slits are specified, this is the final beam shape, otherwise the slits must be considered*

« shape » initial shape of the beam, the options are “square”, “circular”, “rectangular”, “elliptical”, “other”. If it is “other”, then l1 and l2 lose meaning. If it is circular or square, only l1 is needed, l2 is ignored » shape »

« l1 » dimension of beam (e.g. side or diameter) in the plane defined by the normal to the sample and the incident beam propagation axis, in the direction normal to the incident beam propagation » l1 »

« l2 » dimension of beam in the orthogonal plane, in the direction normal to the incident beam propagation » l2 »

beamshape »

« slitsbeforesample » *any slits that are placed before the sample (to guide or to define the beam) are given here.*

« slit » *one given slit. The order is not important, the relative position of slits is given by the “slitdistancetosample” tag.*

« slitshape »

« shape » shape of the slit as seen by the beam, the options are “square”, “circular”, “rectangular”, “elliptical”, “other”. If it is “other”, then l1 and l2 lose meaning. If it is circular or square, only l1 is needed, l2 is ignored. » shape »

« l1 dimension of slit (e.g. side or diameter) in the plane defined by the normal to the sample and the incident beam propagation axis, in the direction normal to the incident beam propagation » l1 »

« l2 dimension of slit in the orthogonal plane, in the direction normal to the incident beam propagation » l2 »

« l3 dimension of slit in the direction of beam propagation » l3 »

slitshape »

« slitdistancetosample distance between the slit and the sample » slitdistancetosample »

slit »

slitsbeforesample »

« beamfoil *if the beam crosses a foil before hitting the sample, it is given here.*

« foildistancetosample distance between the foil and the sample » foildistancetosample »

« foillayers *a layered description similar to the sample is adopted.*

« layer

« layerthickness thickness of layer » layerthickness »

« layeruniformity uniformity of layer thickness » layeruniformity »

« layerdensity density of layer » layerdensity »

« layerelements

« layerelement

« name element present in the layer » name »

« concentration concentration of the given element » concentration »

layerelement »

layerelements »

layer »

foillayers »

beamfoil »

beam »

« geometry *The experimental geometry is defined here as it pertains to scattering; the normal parameters considered in codes are used here, plus the definition of the spot in the sample. Further details on the geometry are found in the beam and instrument entries*

« geometrytype type of geometry. Options are: Cornell, IBM, and general. if general is specified, then a given exit angle must be specified. If Cornell or IBM are specified, then the exit angle is ignored » geometrytype »

« incidenceangle angle of incidence from sample normal » incidenceangle »

« scatteringangle scattering angle » scatteringangle »

« exitangle angle of exit from sample normal » exitangle »

« solidangle solid angle of detector » solidangle »

« spot *defines the beam spot on the sample*

« shape shape of the beam spot on the sample, the options are “square”, “circular”, “rectangular”, “elliptical”, “other”. If it is “other”, then l1 and l2 lose meaning. If it is circular or square, only l1 is needed, l2 is ignored. » shape »

« l1 dimension of spot of beam in the sample (e.g. side or diameter) in the plane defined by the beam propagation axes (in and out) l1 »

« l2 dimension of spot of beam in the orthogonal plane l2 »

spot »

geometry »

« instrument here are the things that pertain to the detection as such, i.e. after the detected beam leaves the sample, including stopping foils and slits after the sample, and the detector as such.

« detectortype type of detector, Options are, but this will need expanding:

SSB - Silicon surface barrier or equivalent

ToF - time of flight

EdE - telescope with a foil where a small energy loss serves as particle identifier and a final detector

Bragg - Bragg chamber

MagSpec - magnetic spectrometer

other - anything not yet in the list- detectortype »

« detector

« shape shape of detector as seen by the beam, the options are “square”, “circular”, “rectangular”, “elliptical”, “other”. If it is “other”, then l1 and l2 lose meaning. If it is circular or square, only l1 is needed, l2 is ignored. shape »

« l1 dimension of detector (e.g. side or diameter) in the plane defined by the sample normal and the beam propagation axis l1 »

« l2 dimension of detector in the orthogonal plane l2 »

« l3 dimension of active region of detector in the direction of beam propagation l3 »

detector »

« detectordeadlayerthickness thickness of the dead layer of the (SSB)

detector detectordeadlayerthickness »

« systemresolution resolution of the detecting system

« resolutionparameter polynomial coefficient on beam energy expressed in the same units as the beam energy, starting with a0, then a1, a2, etc resolutionparameter »

systemresolution »

« shapingtime shaping time of the amplifier shapingtime »

« purtime characteristic time of the pileup rejection system purtime »

« distancedetectortosample distance between the detector and the sample distancedetectortosample »

« slitsaftersample similar definition as already given for the slits before the sample

« slit

« slitshape

« shape shape of the slit as seen by the beam shape »

« l1 dimension of slit (e.g. side or diameter) in the plane defined by the normal to the sample and the exit beam propagation axis, in the direction normal to the incident beam propagation l1 »

« l2 dimension of slit in the orthogonal plane, in the direction normal to the incident beam propagation l2 »

« l3 dimension of slit in the direction of beam propagation l3 »

slitshape »



« slitdistancetosample » distance between the slit and the sample » slitdistancetosample »

slit »

slitsaftersample »

« stoppingfoil » *similar definition as already given for the foil before the sample*

« foildistancetosample » distance between the foil and the sample » foildistancetosample »

« foillayers »

« layer »

« layerthickness » thickness of layer » layerthickness »

« layeruniformity » uniformity of layer thickness » layeruniformity »

« layerdensity » density of layer » layerdensity »

« layerelements »

« layerelement »

« name » element present in the layer » name »

« concentration » concentration of the given element » concentration »

layerelement »

layerelements »

layer »

foillayers »

stoppingfoil »

instrument »

« calibrations » *detector efficiencies and energy calibrations could also be given in the “instrument” group. However, the calibrations depend not only on the instrument, but also on the detected particle.*

« detectorefficiencies » *the detector can have a given efficiency, that can depend on detected energy and on the detected particle (if more that one particle is detected). These are given here.*

« detectorefficiency » *Detector efficiency for one given detected particle.*

« efficiencyon » particle for which the detection efficiency is given » efficiencyon »

« efficiencymode » efficiency can be introduced as a table energy/efficiency, or for given x-ray lines in PIXE. Options are “energy” and “xrayline” » efficiencymode »

« efficiencyparameters » *this can be a table (energy,efficiency), with as many values as there are “efficiencyparameter” entries; or, for PIXE/PIGE, the efficiency for given lines can be given.*

« efficiencyparameter »

« xrayline » x-ray line for which efficiency is given » xrayline »

« energy » energy for which efficiency is given » energy »

« efficiency » efficiency for given energy or x-ray line » efficiency »

efficiencyparameter »

efficiencyparameters »

detectorefficiency »

detectorefficiencies »

« energycalibrations » *the energy calibration can be given in different modes, and can be different for different particle detected.*

« energycalibration » *one given energy calibration for one detected particle*

`calibrationion` particle for which energy calibration is given. This is only necessary if more than one ion is detected simultaneously `calibrationion`

`calibrationmode` calibration mode. The options are “energy”, “PH” if pulse height spectra are considered, “time” for time of flight experiments, and “other” `calibrationmode`

`calibrationparameters` *the calibration is given as a polynomial approximation, on channel number; a series of “calibrationparameter” tags has the polynomial coefficients*

`calibrationparameter` polynomial coefficient on channel, starting with a0, then a1, a2, etc `calibrationparameter`

`calibrationparameters`

`energycalibration`

`energycalibrations`

`calibrations`

`reactions` *Here the reactions that occur are listed. For RBS, ERDA, and PIXE, this is not strictly necessary, because the reactions can be deduced from kinematics. For NRA and NRP this is more important, even if some codes may have built-in lists of possible reactions. This could be a good place to specify cross section data to be used. However, conceptually these are separate things. One thing is reactions that occur and particles detected that make up the data, a different thing is how this data is analysed.*

`technique` technique considered, options are:  
 RBS - including forward scattering and non-Rutherford elastic scattering  
 ERDA - no matter what the beam  
 NRA - for non-resonant i.e. particle-particle reactions, in which a single beam energy is used and an energy spectrum is collected  
 NRP - for resonant reactions, in which the result is the total yield for normally a sharp resonance, and often an excitation function is measured using a range of initial beam energies. This includes PIGE and RNRA.  
 PIXE - no matter what the beam  
 other  
 This is the main technique. Note however, for instance in ERDA, the forward scattered primary beam can also be detected. `technique`

`reactionlist` *all reactions that are of interest are described here. However, e.g. for RBS, ERDA, and PIXE, in the absence of nuclear reactions, there is no need to specify the reactions.*

`reaction` *one given reaction*

`initialtargetparticle` target particle hit by the beam. For specific reactions a given isotope should be used, otherwise e.g. Si or natSi mean natural isotopic distribution `initialtargetparticle`

`incidentparticle` incident beam particle `incidentparticle`

`exitparticle` particle that is detected `exitparticle`

`finaltargetparticle` final target particle `finaltargetparticle`

`reactionQ` reaction heat `reactionQ`

`reaction`

`reactionlist`

`reactions`

`data` *There are many different types of data that can be measured, and it is difficult to find a universal representation. The IDF supports most normal types of data and some exotic ones as well*

`datamode` data mode; The options are:

y - one single quantity was recorded. This can be for instance yield in each channel, where the channel number is implied and not explicitly written

xy - two quantities are recorded, for instance channel and yield

xrayline - yield for given lines; the lines are identified by a string, not a number

xxymatrix - a matrix is recorded, for instance yield as a function of time of flight and energy.

That is,  $z = f(z,y)$ ,  $z$  (the yield) is a function of two variables.

xyytable - a table is recorded, so  $z = f(x)$  and  $y = f(x)$ .

listmode - a time stamp followed by any number of variables recorded at that time (for instance, time of flight and energy)

complex - multidimensional table, with any number of variables being a function of any number of variables:  $y_1=f(x_1,\dots,x_n)$ ,  $y_m=f(x_1,\dots,x_n)$ . This can in principle accommodate any type of data

**datamode**

**datadimensions** *this is only needed for complex datamode. Two integer values are needed, the first is the number of x variables, the second is the number of y dependent variables*

**datadimensionx** number of x variables in "complex" data mode **datadimensionx**

**datadimensiony** number of y variables in "complex" data mode **datadimensiony**

**datadimensions**

**xaxes** *x axis names and units are defined*

**axis**

**axisname** name of an x axis **axisname**

**axisunit** units of an x axis (not restricted to IDF standard units) **axisunit**

**axis**

**xaxes**

**yaxes** *y axis names and units are defined*

**axis**

**axisname** name of an y axis **axisname**

**axisunit** units of an y axis (not restricted to IDF standard units) **axisunit**

**axis**

**yaxes**

**ndataentries** number of data entries given (e.g. number of channels). Not strictly necessary, since it should match the number of dataentry entries; if it does not, it is ignored and the number of dataentry entries should be taken **ndataentries**

**dataentries** *each data entry contains several dataitem entries; 1 for datamode option y (one single quantity), 2 for xy (two quantities), in general one per quantity in order that they were defined, so its number should match what is in datamode, or datadimensionx+datadimensiony*

**dataentry**

**timestamp** time stamp used in list mode **timestamp**

**xrayline** x-ray or  $\gamma$ -ray line for each a yield is given **xrayline**

**dataitem** general data item **dataitem**

**dataentry**

**dataentries**

**data**

**process** *here come the things that relate to the data analysis, including cross sections and stopping powers used, straggling models, simulations made, and anything else*

**crosssections** *everything that relates to cross sections should be here*

**crosssectiondefault** *this defines the defaults on how to handle cross sections. The defaults can be overridden by inputting given cross sections for given reactions*

« Rutherford logical (true or false), whether Rutherford cross sections are the default or not. “true” only makes sense for RBS and ERDA » Rutherford »

« screening defines default for screening. Andersen, Ecuyer, or none » screening »

« computercode if a computer code was used to calculate or produce the cross section, its details should be given here

« codename name of the code used » codename »

« codeversion version of the code used » codeversion »

« codebuilddate build date of the code used in XML format, e.g. 2002-05-30 or 2002-05-30T09:00:00 » codebuilddate »

« codereleasedate release date of the code used in XML format, e.g. 2002-05-30 or 2002-05-30T09:00:00 » codereleasedate »

computercode »

crosssectiondefault »

« crosssectionentries here cross sections can be introduced, either as a file name or as (energy,cross section) tabular data. Both of these override the crosssectiondefault.

« crosssectionentry cross section for one given reaction

« crosssectionframe frame of reference where cross section is given, cm (for center of mass) or lab (for laboratory) » crosssectionframe »

« crosssectiontype type of cross section, either total or differential » crosssectiontype »

« crosssectionenergyunits energy units for given cross section » crosssectionenergyunits »

« crosssectiondataunits cross section units » crosssectiondataunits »

« reaction reaction for which cross section is given. For RBS, only the initial target particle needs be given. For ERDA, only the exit particle is required

« initialtargetparticle target particle hit by the beam. For specific reactions a given isotope should be used, otherwise e.g. Si or natSi mean natural isotopic distribution » initialtargetparticle »

« incidentparticle incident beam particle » incidentparticle »

« exitparticle particle that is detected » exitparticle »

« finaltargetparticle final target particle » finaltargetparticle »

« reactionQ reaction heat » reactionQ »

reaction »

« crosssection there are three ways of inputting a cross section, only one of (crosssectionfile/crosssectiondata/crosssectionoverride) can be present

« crosssectionfile This is the first possible way of inputting a given cross section. Here a given file name containing a cross section can be given.

« filename name of the file » filename »

« fileformat format of the file (e.g. “r33”, or “other”) » fileformat »

« filesource text defining where the file comes from » filesource »

crosssectionfile »

« crosssectiondata this is the second possible way of inputting a given cross section. Here actual cross section data can be given in full

« crosssectionitem one point of a (energy, cross section table)

`<crosssectionenergy` energy for which cross section is given `<crosssectionenergy`  
`<crosssectionvalue` cross section at given energy `<crosssectionvalue`  
`<crosssectionitem`  
`<crosssectiondata`  
`<crosssectionoverride` *This is the third possible way of inputting a given cross section. Here a given model can be given, overriding the default*  
`<Rutherford` `<Rutherford`  
`<screening` `<screening`  
`<computercode`  
`<codename` `<codename`  
`<codeversion` `<codeversion`  
`<codebuilddate` `<codebuilddate`  
`<codereleasedate` `<codereleasedate`  
`<computercode`  
`<crosssectionoverride`  
`<crosssection`  
`<crosssectionentry`  
`<crosssectionentries`  
`<crosssections`  
`<stoppingpowers` *everything that relates to stopping powers should be here*  
`<stoppingpowerdefault` *This defines the defaults on how to handle stopping powers. The defaults can be overridden by inputting given cross sections for given reactions. The main defaults should match the usual codes.*  
`<computercode` *if a computer code was used to calculate or produce the cross section, its details should be given here*  
`<codename` name of the code used `<codename`  
`<codeversion` version of the code used `<codeversion`  
`<codebuilddate` build date of the code used in XML format, e.g. 2002-05-30 or 2002-05-30T09:00:00 `<codebuilddate`  
`<codereleasedate` release date of the code used in XML format, e.g. 2002-05-30 or 2002-05-30T09:00:00 `<codereleasedate`  
`<computercode`  
`<stoppingpowerdefault`  
`<stoppingpowerentries` *stopping power will be given, either as a file name or as (energy, stopping power) tabular data. Both of these override the stoppingpowerdefault*  
`<stoppingpowerentry` *stopping power for one given ion/element pair*  
`<ion` ion for which stopping power is given `<ion`  
`<ionchargestate` charge state of ion for which stopping power is given `<ionchargestate`  
`<targetelement` target element for which stopping power is given `<targetelement`  
`<stoppingmode` stopping power mode: electronic, nuclear, or total `<stoppingmode`

« stoppingpowerenergyunits (energy units for given stopping power. However, for SRIM files, this is overridden by what is in the file) stoppingpowerenergyunits »

« stoppingpowerdataunits (stopping power units. However, for SRIM files, this is overridden by what is in the file) stoppingpowerdataunits »

« stoppingpower (there are three ways of inputing a stopping power, only one of stoppingpowerfile|stoppingpowerdata|stoppingpoweroverride) can be present

« stoppingpowerfile »

« filename (name of the file) filename »

« fileformat (format of the file (e.g. "SRIM", or "other")) fileformat »

« filesource (text defining where the file comes from) filesource »

stoppingpowerfile »

« stoppingpowerdata (This is the second possible way of inputing a given stopping power. Here actual stopping power data can be given in tabulated form

« stoppingpoweritem (one point of a (energy, stopping power table)

« stoppingpowerenergy (energy for which stopping power is given) stoppingpowerenergy »

« stoppingpowervalue (stopping power at given energy) stoppingpowervalue »

stoppingpoweritem »

stoppingpowerdata »

« stoppingpoweroverride (This is the third possible way of inputing a given stopping power. Here a given model can be given, overriding the default

« computercode »

« codename ( ) codename »

« codeversion ( ) codeversion »

« codebuilddate ( ) codebuilddate »

« codereleasedate ( ) codereleasedate »

computercode »

stoppingpoweroverride »

stoppingpower »

stoppingpowerentry »

stoppingpowerentries »

stoppingpowers »

« energyspread (This group is mainly a series of flags that inform where some effect that affect the energy spread of the beam are included in any calculation. Actual calculation requires a series of values taken from different places in the file

« energylossstragglng (Basic model to calculate energy loss stragglng, e.g. Bohr, Chu, none) energylossstragglng »

« multiplescattering (logical flag, true/false if the effect of multiple scattering on energy spread is to be calculated or not) multiplescattering »

« Dopplereffect (logical flag, true/false if the effect of the Doppler effect on energy spread is to be calculated or not) Dopplereffect »

« beamangularspread logical flag, true/false if the effect of the beam angular spread on energy spread is to be calculated or not » beamangularspread ▶

« geometricspread ▶

« beamsize logical flag, true/false if the effect of the finite beam size on energy spread is to be calculated or not » beamsize ▶

« detectoraperture logical flag, true/false if the effect of the finite detector aperture size on energy spread is to be calculated or not » detectoraperture ▶

geometricspread ▶

« Tschalareffect logical flag, true/false if the Tschalar effect is to be calculated or not » Tschalareffect ▶

energyspread ▶

« simulations this is where simulations are given. Several types of simulations can be defined, depending on the "simulationtype" flag:

*total: structure of the simulated data is identical to what was specified in the "dataentries" group of the "data" group. In principle, the x entries should be exactly the same as given in the "data" group; and to write them here again is simple redundancy, for the purpose of ease of extracting the simulated data together with the x axes values*

*partialement: the structure of the simulated data is similar to what was specified in the "dataentries" group of the "data" group. The x and y axes are exactly the same, but the simulation is for scattering off one single target element. This is useful to output in a simple way the calculate partial spectrum of a given element in RBS or ERDA. In principle, the x entries should be exactly the same as given in the "data" group; and to write them here again is simple redundancy, for the purpose of ease of extracting the simulated data together with the x axes values. If the targetlayer is also specified, then the simulation is for the contribution of the given element in the specified layer only.*

*partial: This is the third way to present a simulation. When simulationtype is "partial", then the structure of the simulated data is similar to what was specified in the "dataentries" group of the "data" group. The x axes are exactly the same, but one single y axis was simulated and presented. This is useful when two quantities were measured (for instance time of flight and energy), but only one is simulated (for instance in ToF ERDA, where a pre-processing separates the detected ions. In principle, the x entries should be exactly the same as given in the "data" group; and to write them here again is simple redundancy, for the purpose of ease of extracting the simulated data together with the x axes values. The y axis that is simulated must be specified.*

*reaction: This is the fourth way to present a simulation. When simulationtype is "reaction", then one given reaction is simulated and presented. This becomes quite independent of the structure given in the "data" group, because the quantities can be all different. Suppose an RBS spectrum given in "data" as channel and total yield. Here, the simulated partial spectrum of one of the elements is to be presented as energy and elemental yield. Also, this can be used for a pure simulation, even for a complete spectrum; if in "reaction" only the incidentparticle is given, then all reactions are considered to be included*

« simulation one given simulation.

« simulationtype total simulationtype ▶

« dataentries as many data entries as for the data

« dataentry with exactly the same structure as for the data

« dataitem same quantity and units as for the data » dataitem ▶

dataentry ▶

dataentries ▶

« simulationtype partialement simulationtype ▶

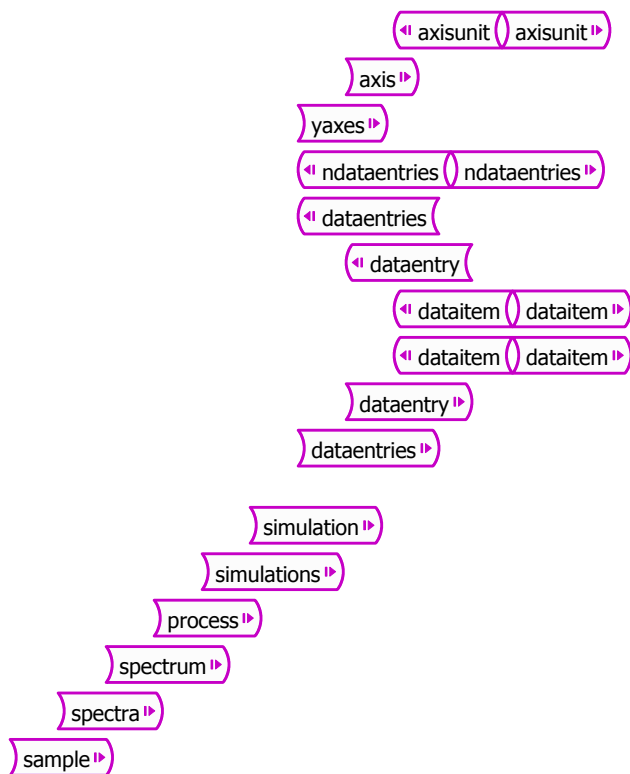
<initialtargetparticle the simulation is for this initial target particle  
 only >initialtargetparticle >  
 <targetlayer Integer. If this tag is present, then the simulation is for the contribution of  
 the given element in the specified layer only. >targetlayer >  
 <dataentries as many data entries as for the data  
   <dataentry with exactly the same structure as for the data  
     <dataitem same quantity and units as for the data >dataitem >  
   >dataentry >  
 >dataentries >

<simulationtype partial >simulationtype >  
 <simulatedy integer. the y axis that is simulated must be specified; if there were 2 y  
 quantities, the "1" given means it is the first one that is simulated >simulatedy >  
 <dataentries as many data entries as for the data  
   <dataentry with similar structure as for the data, but only the one y data item  
   specified in the tag simulatedy is given  
     <dataitem same quantity and units as for the data >dataitem >  
   >dataentry >  
 >dataentries >

The last method to present a simulation, with simulationtype="reaction", implies defining the data structure completely, see the definitions in the "data" group above.

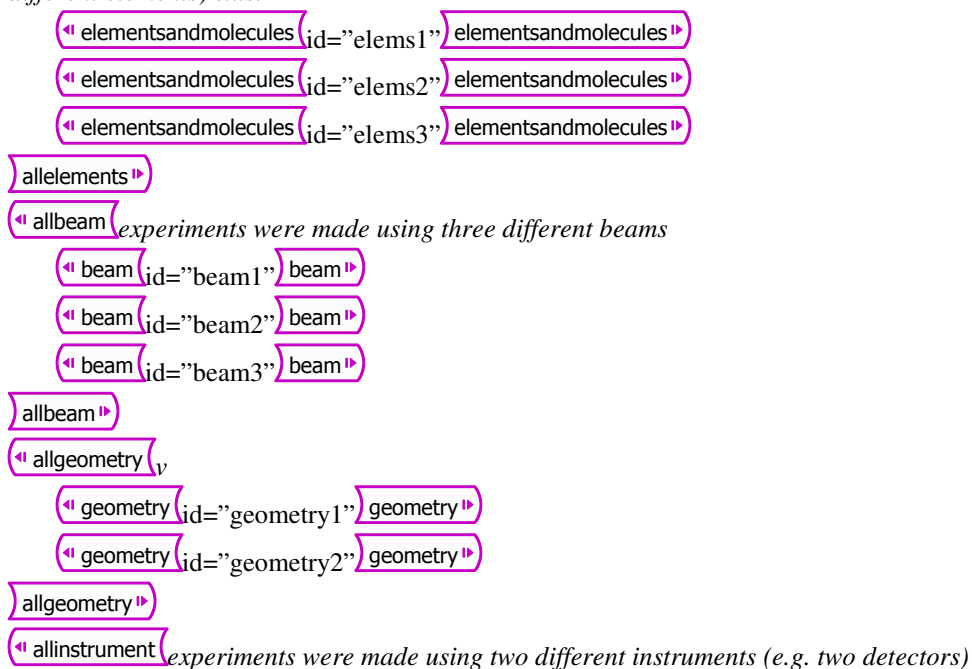
<simulationtype reaction >simulationtype >  
 <reaction  
   <initialtargetparticle >initialtargetparticle >  
   <incidentparticle >incidentparticle >  
   <exitparticle >exitparticle >  
   <finaltargetparticle >finaltargetparticle >  
 >reaction >  
 <datamode >datamode >  
 <datadimensions  
   <datadimensionx >datadimensionx >  
   <datadimensiony >datadimensiony >  
 >datadimensions >  
 <xaxes  
   <axis  
     <axisname >axisname >  
     <axisunit >axisunit >  
   >axis >  
 >xaxes >  
 <yaxes  
   <axis  
     <axisname >axisname >  
   >axis >  
 >yaxes >





`<batch` The batch mode is the second basic way of structuring the information and data in IDF. This is appropriate when for instance many different samples were measured with the same or similar experimental conditions (as in an experimental run). This way is more concise, as information on the experimental conditions does not need to be repeated for each sample measured. The method is to define independently e.g. all possible “beam” groups, that is, all the different beams to be considered, all possible different “elementsandmolecules”, all possible “geometry” groups, all possible “instrument” groups, all possible “calibrations” groups. Each group is given a unique identifier, an attribute called “id”. Then, for each sample and spectrum, reference back to the appropriate group. This is made with an attribute called “ref” that points to the “id”. As many “batch” entries can be given as desired, and “batch” and “sample” entries can be mixed at will.

`<allelements` here, all the necessary elementsandmolecules groups are defined; each one is defined in exactly the same way as it was explained above. The only difference is that each group has an “id” attribute, such as `<elementsandmolecules id="elems1">`. In this example, three different types of samples (i.e., containing different elements) exist



```
« instrument (id="instrument1" instrument )»
```

```
« instrument (id="instrument2" instrument )»
```

```
allinstrument )»
```

```
« allcalibrations (experiments involved two different calibrations (e.g. one for each detector used)
```

```
« calibrations (id="calibrations1" calibrations )»
```

```
« calibrations (id="calibrations1" calibrations )»
```

```
allcalibrations )»
```

« allsample (now that all the required conditions have been defined, we can proceed with the samples. The structure is exactly the same as for the "sample" entries. The only difference is that the "elementsandmolecules", "beam", "geometry", "instrument", and "calibrations" groups are not explicitly written in full with all the corresponding groups and elements, instead they reference back to the "id" of the definitions given in the beginning of the "batch" group.

« sample (the "elementsandmolecules" group is not explicitly given with all its tags and elements, all that is needed is to reference back to what was defined in the "allelements" group.

```
« elementsandmolecules (ref="elems1" elementsandmolecules )»
```

```
« structure ( ) structure )»
```

« spectra (in this sample, three measurements were made with the same conditions except for the beam (e.g. three different beam energies)

« spectrum (the log, environment, reactions, and data groups are given in full; the beam, geometry, instrument, and calibrations groups are simply referenced back to the previous definitions

```
« log ( ) log )»
```

```
« environment ( ) environment )»
```

```
« beam (ref="beam1" beam )»
```

```
« geometry (ref=" geometry1" geometry )»
```

```
« instrument (ref=" instrument1" instrument )»
```

```
« calibrations (ref=" calibrations1" calibrations )»
```

```
« reactions ( ) reactions )»
```

```
« data ( ) data )»
```

```
spectrum )»
```

```
« spectrum (
```

```
« log ( ) log )»
```

```
« environment ( ) environment )»
```

```
« beam (ref="beam2" beam )»
```

```
« geometry (ref=" geometry1" geometry )»
```

```
« instrument (ref=" instrument1" instrument )»
```

```
« calibrations (ref=" calibrations1" calibrations )»
```

```
« reactions ( ) reactions )»
```

```
« data ( ) data )»
```

```
spectrum )»
```

```
« spectrum (
```

```
« log ( ) log )»
```

```
« environment ( ) environment )»
```

```

    « beam (ref="beam3" beam »
    « geometry (ref=" geometry1" geometry »
    « instrument (ref=" instrument1" instrument »
    « calibrations (ref=" calibrations1" calibrations »
    « reactions () reactions »
    « data () data »
  spectrum »
spectra »
sample »

```

```

« sample Different sample, one measurement only
  « elementsandmolecules (ref="elems2" elementsandmolecules »
  « structure () structure »
  « spectra
    « spectrum
      « log () log »
      « environment () environment »
      « beam (ref="beam1" beam »
      « geometry (ref=" geometry2" geometry »
      « instrument (ref=" instrument2" instrument »
      « calibrations (ref=" calibrations2" calibrations »
      « reactions () reactions »
      « data () data »
    spectrum »
  spectra »
sample »

```

« sample *For this sample two measurements are presented, with the same beam but different geometry, instrument, and calibrations, e.g. eventually two simultaneous detectors.*

```

  « elementsandmolecules (ref="elems3" elementsandmolecules »
  « structure () structure »
  « spectra
    « spectrum
      « log () log »
      « environment () environment »
      « beam (ref="beam1" beam »
      « geometry (ref=" geometry1" geometry »
      « instrument (ref=" instrument1" instrument »
      « calibrations (ref=" calibrations1" calibrations »
      « reactions () reactions »
      « data () data »
    spectrum »
  »

```

```
« spectrum (
  « log ( log )
  « environment ( environment )
  « beam (ref="beam1" beam )
  « geometry (ref=" geometry2" geometry )
  « instrument (ref=" instrument2" instrument )
  « calibrations (ref=" calibrations2" calibrations )
  « reactions ( reactions )
  « data ( data )
```

```
spectrum )
```

```
spectra )
```

```
sample )
```

```
allsample )
```

```
batch )
```

```
idf )
```

## References

- [1] E. Rauhala, N. P. Barradas, S. Fazinic, M. Mayer, E. Szilágyi, M. Thompson, Nucl. Instrum. Methods B244 (2006) 436
- [2] N.P. Barradas, K. Arstila, G. Battistig, M. Bianconi, N. Dytlewski, C. Jeynes, E. Kótai, G. Lulli, M. Mayer, E. Rauhala, E. Szilágyi, M. Thompson, Nucl. Instrum. Methods B262 (2007) 282
- [3] Consultants Meeting on Evaluation and Intercomparison of Software for Accelerator Science, 15-16 December 2003, Vienna, Austria
- [4] <http://www.w3.org/XML>. XML is a World Wide Web Consortium (W3C) recommendation.
- [5] <http://www.spirit-ion.eu>. SPIRIT represents an Integrated Infrastructure Initiative (I3) funded by the European Commission. The SPIRIT consortium integrates 11 leading ion beam facilities and 4 R&D providers from 6 member states and 2 associated states.
- [6] [http://www.nexusformat.org/Main\\_Page](http://www.nexusformat.org/Main_Page)