mzDB: Mass Spectrometry SQLite Database

Status of This Document

This document presents the completed 0.6.0 specification for the mzDB (Mass Spectrometry SQLite Database) data format developed by the Proteomics French Infrastructure (ProFI). Distribution is unlimited.

Version of This Document

The current version of this document is: Version 0.6.0.0; February 18, 2014.

The version of this document matches the format version with one trailing decimal point and integer to denote specification documentation updates that do not correspond to a format update. Thus the version numbers correspond to:
majorVersion.minorVersion.maintenanceVersion.documentationOnlyUpdateVersion.

Contents

1. Introduction

1.1 Background

Mass spectrometry is a popular method to analyse bio-molecules by measuring the intact mass-to-charge ratios of their in-situ generated ionised forms or the mass-to-charge ratios of in-situ-generated fragments of these ions. The resulting mass spectra are used for a variety of purposes, among which is the identification, characterization, and absolute or relative quantification of the analysed molecules. The processing steps to achieve these goals typically involve semi-automatic computational analysis of the recorded mass spectra and sometimes also of the associated metadata (e.g., elution characteristics if the instrument is coupled to a chromatography system). The result of the processing can be assigned a score, rank or confidence measure.

Differences inherent in the use of a variety of instruments, different experimental conditions under which analyses are performed, and potential automatic data preprocessing steps by the instrument software can influence the actual measurements and therefore the results after processing. Additionally, most instruments output their acquired data in a very specific and often proprietary format. These proprietary formats are then typically transformed into so-called peak lists to be analysed by identification and characterisation software. Data reduction such as peak centroiding and deisotoping is often performed during this transformation from proprietary formats to peak lists. In addition, these peak list file formats lack information about the precursor MS signals and about the associated metadata (i.e., instrument settings and description, acquisition mode, etc) compared to the files they were derived from. The peak lists are then used as inputs for subsequent analysis. The many different and often proprietary formats make integration or comparison of mass spectrometer output data difficult or impossible, and the use of the heavily processed and data-poor peak lists is often suboptimal.

This document addresses this problem with the presentation of the mzDB SQLite format, which is designed to hold the data output of a mass spectrometer as well as a systematic description of the conditions under which this data was acquired and transformed. The following target objectives can be defined for the format:

T1. *The discovery of relevant results,* so that, for example, data sets in a database or public repository that use a particular technique or combination of techniques can be identified and studied by experimentalists during experiment design or data analysis.
T2. *The sharing of best practice*, whereby, for example, approaches that have been successful at analysing low abundance analytes can be captured alongside the results produced.
T3. *The evaluation of results*, whereby, for example, the number and quality of the spectra recorded from a sample can be assessed in the light of the experimental conditions.
T4. *The sharing of data sets,* so that, for example, public repositories can import or export data, multi-site projects can share results to support integrated analysis, or meta-analyses can be performed by third parties from previously published data.
T5. *The most comprehensive support of the instruments output,* so that data can be captured in profile mode, centroid mode, and other relevant forms of biomolecular mass spectrometry data representation
T6. *The efficient processing of MS data,* by leveraging the multiple indexing strategies that are provided.

The presented format is an interesting alternative to mzML for the long-term archiving, sharing and fast processing of mass spectrometry data.

The description of mass spectrometry data output and its experimental context requires that models include: (i) the actual data acquired, to a sufficient precision, as well as its associated metadata; and (ii) an adequate description of the instrument characteristics, its configuration and possible preprocessing steps applied. This document details both these parts, as they are required to support the tasks T1 to T6 above.

This document defines a specification and is not a tutorial. As such, the presentation of technical details is deliberately direct. The role of the text is to describe the format model and justify design decisions made. This document does not provide comprehensive examples of the format in use.

Example documents are provided separately and should be examined in conjunction with this document. It is anticipated that tutorial material will be developed in the future to aid implementation. Although the present specification document describes constraints and guidelines related to the mzDB format content as well as the availability of tools helping to read and write mzDB files, it does not describe any implementation constraints or specifications such as coding language or operating system for software that will generate and/or read mzDB data.

1.2 Previous formats

During 2003 – 2005, two data formats to store mass spectrometer output in an open, vendor-neutral, XML format were developed. The mzData format [mzData] was developed by the PSI, primarily as a data exchange and archive format. The mzXML format [mzXML] was developed at the Institute for Systems Biology (ISB), primarily in order to streamline data processing software. Both formats are used extensively but it has been noted that having two formats for essentially the same information causes unnecessary confusion in the community and adds complexity to software developers as often both formats must be supported. Therefore the designers of mzData and mzXML, including representatives of instrument vendors, analysis software developers and end users, have joined under the auspices of the PSI and jointly developed a single format intended to replace the previous two: the mzML format.

The main challenge in uniting the mzData and mzXML formats was indeed resolving the opposing philosophies rather than fundamental technical issues. On the contrary, the mzDB format is a radical technical change because it is no more based on the eXtendedMarkup Language but on the SQLite data format. The latter is used in many computational projects (http://www.sqlite.org/famous.html) and compatible with most programming languages. mzDB has an internal data structure allowing it to achieve efficient queries in both time and m/z dimensions. This makes it particularly suitable for the processing of LC-MS data.

1.3 Design Philosophy

Since the mzML format is the official PSI standard for mass spectrometry data representation, we decided to design mzDB in order to have a semantic and a data structure as close as possible to the mzML ones. We also kept some design principles that guided the mzML development including some disgressions:

1. *Keep the format simple*. Metadata representation is as simple as in an mzML document. However the MS data structure of mzDB is a little bit more complex because peaks are stored into bounding boxes (see section X.X) instead of full scan records.
2. *Minimize alternate ways of encoding the same information*. Such flexibility, while sometimes touted as a benefit for some products, is bad for data formats.
3. *Build in some flexibility for encoding new important information but keep the format stable*. There is a strong desire from companies that develop software for their customers to keep the data format stable over long periods of time with updates to an auxiliary file.

It should be noted that beyond these principles the main design goal of the mzDB was to obtain an efficient and compact data format leveraging multiple indexing strategies.

There was great temptation to optimize the compactness of metadata. There are many enhancements that have been suggested, but we decided to keep a strong compatibility with the mzML syntax. The enhancements not considered compatible with this goal may be entertained for a future version of mzDB.

The remainder of this document is structured as follows. Section 2.1 describes a number of concepts and information about the implementation of mzDB, including aspects of terminology, design issues, the controlled vocabulary, etc. The SQL schema is presented in Section 3; The XML schemata are presented in Section 4; Some conclusions are presented in Section 5.

2. Implementation of the Format

2.1 Concepts and Terminology

This document assumes familiarity with SQL and XML data modelling. The main mzDB structure is described using a relational model. Metadata are described using XML schema (www.w3.org/XML/Schema).

The keywords "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" are to be interpreted as described in RFC-2119 (Bradner 1997).

2.2 The PSI Mass Spectrometry Controlled Vocabulary (CV)

A comprehensive collection of terms have been defined (mostly extracted from vocabulary and definitions in chapter 12 of the IUPAC nomenclature book) and structured in an mzML-friendly way, hopefully facilitating the browsing of the terms. Almost all first-level branch terms (the direct children of the root term) have a homonymous XML element in mzML. Their children, the second-level terms, are relevant topics or categories which need CV support for their description. The leaf nodes under their respective parent categories should be used in a cvParam under the appropriate XML element in mzML schema.

Some terms describe attributes that must be coupled with a numerical value attribute in the CvParam element (e.g. dwell time MS:1000039) and optionally a unit for that value (e.g. second MS:1000502). The terms that require a value are denoted by having a "datatype" key-value pair in the CV itself; the use of the 'object attribute' (MS:1000547) term to denote this is now deprecated. Similarly, terms that need to be qualified with units are denoted by have a "needs_units" key in the CV itself.

As recommended by the PSI CV guidelines, psi-ms.obo should be dynamically maintained via the psidev-ms-vocab mailing lists that allow any user to request new terms in agreement with the community involved. Once a consensus is reached among the community the new terms are added within few days. If there is no obvious consensus, the CV coordinators committee should vote and make a decision. A new psi-ms.obo should then be released by updating the file on the CVS server without changing the name of the file (this would alter the propagation of the file to the OBO website and to other ontology services that rely on file stable URI). For this reason an internal version number with two decimals (x.y.z) should be increased:

- x should be increased when a first level term are renamed added deleted or rearranged in the structure. Such rearrangement is supposed to be rare and is very likely to have repercussion on the mapping.
- y should be increased when any other term except the first level one is altered.
- z should be increased when there is no term addition or deletion but just editing on the definitions or other minor changes.

It was decided that the CV not contain "unknown" terms as much as possible, thus there is no "unknown instrument", etc. There are two cases where it is tempting to use unknown:

1) The information is really known, but none of the existing terms fit. In this case, instead of choosing "unknown", a user should send email to the psidev-ms-vocab list proposing a new term. The CV committee should approve and add the desired term, or point out an existing synonym that should be used. Then once the term has been assigned an accession number and official name, then the user should begin using that.
2) The second case is there the information is truly not known. We want to avoid cases where the user picks something just to appease a validator. In this case, the use of the base class is recommended, e.g.:

```
 <instrumentConfiguration id="unknownInstrument">
   <cvParam cvRef="MS" accession="MS:1000031" name="instrument model"/>
</instrumentConfiguration>
```

The rationale is that the most specific information available is provided, i.e. that a mass spectrometer of some model generated the data, but it is not known which one. This also avoids the situation where the tags are optional. If optional tags are not provided, it is never clear whether the information is truly unknown, or the writer simple forgot to write the information or was too lazy to write the information.

In the above example, if the writer knows that the instrument was one from a specific vendor, but not the model, then the vendor model term should be used instead of the completely generic “instrument model”.

To obsolete a term, the following must be done:

- Put OBSOLETE at the beginning of the definition
- Add a comment to the term describing the reason for obsoleting.
- Set the OBO-format is_obsolete tag to true

This is a summary of the procedure given in http://psidev.info/files/CommunityPractice-revised.doc.

If a term name needs to be changed, the accession number should stay the same and the term name simply changed. One should NOT obsolete the term and create a new one with the revised name.

The following ontologies or controlled vocabularies specified below may also be suitable or required in certain instances:

- Unit Ontology (http://www.obofoundry.org/cgi-bin/detail.cgi?id=unit)
- ChEBI (http://www.ebi.ac.uk/chebi/)
- OBI (Ontology of Biological Investigations - http://obi.sourceforge.net/) (formerly called FuGO)

2.3 Conventions

Date timestamps must be encoded as in the ISO-8601 specification (http://www.w3.org/TR/NOTE-datetime) such as 2007-06-27T15:23:45.00035.

Metadata (XML parameter tree) designs follow the principles detailed in the mzML specifications.

2.4 Other supporting materials

This document cannot be fully judged on its own. It is important to study the accompanying sample instance documents, controlled vocabulary, schema files, and the software that implements this version of mzDB.

All these files and programs are available at: http://github.com/mzdb

They are:

Filename	Description
mzDB_0.5.0.architect	The database diagram in the Power Architect format
mzDB_0.5.0.sql	The database SQL script
mzDB_0.5.0.html	HTML documentation of the database schema

2.5 Open Issues

We note that usage of JSON was considered and rejected for the representation of metadata in the current version of mzDB. It remains a possibility that metadata of mzDB 2.0 will be stored using JSON.

2.6 Comments on Specific Use Cases

As for mzML, many special use cases were considered during the development of mzDB. Most of these use cases have a corresponding example file that exercise the relevant part of the schema and provide a reference implementation example. Authors of mzDB writing software are encouraged to examine the examples that accompany this format release before implementing the writer. In the subsections below, we comment on a few of the notable use cases that were considered.

2.6.1 Selected Reaction Monitoring (SRM)

There was considerable discussion on how to encode SRM experiments. There seem to be two major contenders: encoding them as tiny MS/MS-like spectra; or encoding them directly as complete chromatograms. We decided to follow the mzML specifications namely encode each SRM scan as a mini MS/MS-like spectra with a precursor corresponding to the Q1 m/z and a small spectrum encoding one or more Q3 m/z values that correspond to the Q1 m/z. We note that these mini scans may be a single (centroided) value per Q3 m/z, or the mini scans may be profile mode scans surrounding the Q3 m/z. For example, it is entirely permissible to monitor two Q3 m/z values for a single Q1 m/z, and encode profile mode scans for both Q3 regions in a single spectrum.

The mzDB format also contains a “chromatogram” table that is capable of containing a full description of and the data for a chromatogram. The chromatogram may be simply be a total ion current (TIC) chromatogram of an ordinary MS1 or MS/MS run, or a chromatogram corresponding to a Q1,Q3 pair in a SRM run.

It has been resolved that all SRM runs must be encoded as mini MS/MS-like spectra using the “scan” table. Optionally, the same information may also be encoded using the “chromatogram” table as a speed-enhancing feature. At present, it has been decided that SRM output may not be encoded ***only*** in the “chromatogram” form. The goal is to avoid having two different ways of encoding the same data. Readers can always count on the mini MS/MS-like spectra and may only optionally support the “chromatogram” constructs. This is merely a policy decision, not one dictated by the schema.

Note: SRM is often referred to as multiple reaction monitoring (MRM), but that term is an obsolete synonym of SRM according to IUPAC.

2.6.2 Profile (continuous) spectra vs. centroided (peak list) (peak picked) spectra vs. fitted (parameterized peaks) spectra

Mass spectra typically come in two major flavors: profile and centroided. Profile spectra represent the scanned data in a (sometimes only approximately) regularly spaced format, sometimes with gaps. Centroided spectra present the scanned data only by specifying the location and intensity of individual detected peaks, usually after subjecting the profile spectrum to a peak-picking algorithm. We have added a third mode of data representation obtained after processing by a peak-fitting algorithm: the fitted mode. The latter present the scanned data by specifying the location and intensity of individual detected peaks along with the left and right HWHMs (Half Width at Half-Maximum) of these peaks.

The mzDB format can encode either format with the specification of the used mode (using the data_encoding table). However, it is not allowed to encode the same spectrum using different modes in the same file. The recommended workflow if both spectra are desired is to encode the profile spectra in one file and the processed data in a second file (with appropriate annotations as to what was done). It is permissible to have some spectra in one mode and different ones in another; for example MS level 1 spectra may be profile mode, while MS level 2 spectra may be peak picked in the same file.

3. Database schema

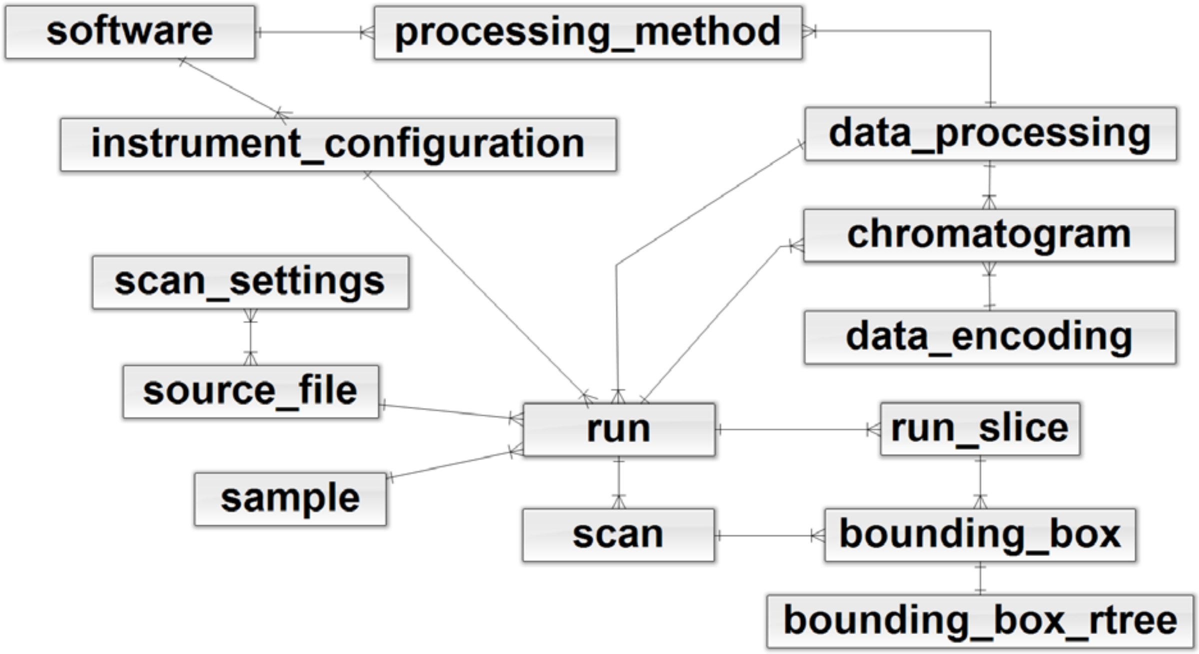


Figure 1: partial ER (Entity-relationship) diagram of the mzDB format. Each box represents a database table. Some tables related to metadata were voluntarily not shown for the sake of clarity.

Below is appended the full HTML documentation of the database schema. Tables are presented in ascendant alphabetical order.

3.1 bounding_box

The Bounding Box (BB) table aims to store a region of the run. A BB refers to a run slice which describes the m/z width of this BB. The first and last scan id describe the height in time dimension of the BB. Data points may be accumulated in m/z and time dimensions. In the simplest case a BB may store peaks of a full MS spectrum, implying that first and last scans correspond to the same record.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
data	data	BLOB		NOT NULL
A BLOB containing the data points using a given structure representation (see data_enconding table).				
run_slice_id (FK)	run_slice_id	INTEGER		NOT NULL
A reference to the corresponding run_slice.				
first_spectrum_id (FK)	first_spectrum_id	INTEGER		NOT NULL
A reference to the first spectrum of the BB.				
last_spectrum_id (FK)	last_spectrum_id	INTEGER		NOT NULL
A reference to the last spectrum of the BB.				

References

- spectrum through (first_spectrum_id)
- spectrum through (last_spectrum_id)

- run_slice through (run_slice_id)

Referenced By

- bounding_box_rtree referencing (id)
- bounding_box_msn_rtree referencing (id)

3.2 bounding_box_msn_rtree

A virtual table used to index the bounding boxes with the help of a R*Tree structure automatically generated by SQLite. For more detailed information see http://www.sqlite.org/rtree.html. The aim of this table is to index in 4 dimensions LC-MS/MS data, such as those generated by DIA experiments.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK) (FK)	id	INTEGER	PK	NOT NULL
The identifier of the bounding box that corresponds to this index.				
min_ms_level	min_ms_level	INTEGER		NOT NULL
Minimum MS level of the corresponding BB.				
max_ms_level	max_ms_level	INTEGER		NOT NULL
Maximum MS level of the corresponding BB.				
min_parent_mz	min_parent_mz	INTEGER		NOT NULL
Minimum parent m/z of the corresponding BB.				
max_parent_mz	max_parent_mz	INTEGER		NOT NULL
Maximum parent m/z of the corresponding BB.				
min_mz	min_mz	REAL		NOT NULL
Minimum m/z of the corresponding BB.				
max_mz	max_mz	REAL		NOT NULL
Maximum m/z of the corresponding BB.				
min_time	min_time	REAL		NOT NULL
Minimum time of the corresponding BB.				
max_time	max_time	REAL		NOT NULL
Maximum time of the corresponding BB.				

References

- bounding_box through (id)

3.3 bounding_box_rtree

A virtual table used to index the bounding boxes with the help of a R*Tree structure automatically generated by SQLite. For more detailed information see http://www.sqlite.org/rtree.html. The aim of this table is to index in two dimensions LC-MS data, such as those generated by DDA experiments.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK) (FK)	id	INTEGER	PK	NOT NULL
The identifier of the bounding box that corresponds to this index.				
min_mz	min_mz	REAL		NOT NULL
Minimum m/z of the corresponding BB.				
max_mz	max_mz	REAL		NOT NULL
Maximum m/z of the corresponding BB.				
min_time	min_time	REAL		NOT NULL

Minimum time of the corresponding BB.			
max_time	max_time	REAL	NOT NULL
Maximum time of the corresponding BB.			

References

- bounding_box through (id)

3.4 chromatogram

A chromatogram may be simply be a total ion current (TIC) chromatogram of an ordinary MS1 or MS/MS run, or a chromatogram corresponding to a Q1,Q3 pair in a SRM run.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
name	name	VARCHAR(0)		NOT NULL
A unique name for this chromatogram. Examples: sic, tic, ms1_tic, ms2_tic, bpc, transition_xxx_xxx				
activation_type	activation_type	VARCHAR(10)		NOT NULL
The type of activation used for fragmentation. This field stores only a 3 letters label (e.g. CID, ETD, HCD), but more detailled information could be provided using the param_tree field.				
data_points	data_points	BLOB		NOT NULL
A BLOB containing the data points using a given data representation (see data_enconding table).				
param_tree	param_tree	CLOB		NOT NULL
This field must contain the table cvParams and userParams as an XML string (see XML schema specifications).				
precursor	precursor	CLOB		
This field can contain optionally an XML string describing the method of precursor ion selection and activation (see XML schema specifications).				
product	product	CLOB		
This field can contain optionally an XML string describing the method of product ion selection and activation in a precursor ion scan (see XML schema specifications).				
shared_param_tree_id (FK)	shared_param_tree_id	INTEGER		
An optional reference to a shared_param_tree, which is a reusable container of one or more cvParams.				
run_id (FK)	run_id	INTEGER		NOT NULL
This field must reference the 'id' of the corresponding run.				
data_processing_id (FK)	data_processing_id	INTEGER		
This field can optionally reference the 'id' of the appropriate data_processing.				
data_encoding_id (FK)	data_encoding_id	INTEGER		NOT NULL
This field must reference the 'id' of the appropriate data_encoding.				

References

- run through (run_id)
- data_encoding through (data_encoding_id)
- shared_param_tree through (shared_param_tree_id)
- data_processing through (data_processing_id)

3.5 cv

Information about an ontology or CV source and a short 'lookup' tag to refer to.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	VARCHAR(10)	PK	NOT NULL
The short label to be used as a reference tag with which to refer to this particular Controlled Vocabulary source description.				
full_name	full_name	VARCHAR(0)		NOT NULL
The usual name for the resource (e.g. The PSI-MS Controlled Vocabulary).				
version	version	VARCHAR(10)		
The version of the CV from which the referred-to terms are drawn.				
uri	uri	VARCHAR(0)		NOT NULL
The URI for the resource.				

Referenced By

- cv_unit referencing (id)
- cv_term referencing (id)

3.6 cv_term

This table lists the CV terms used in the mzDB file.

Logical Column Name	Physical Column Name	Type	PK	Nullable
accession (PK)	accession	VARCHAR(0)	PK	NOT NULL
The accession number of the referred-to term in the named resource (e.g.: MS:000012).				
name	name	VARCHAR(0)		NOT NULL
The actual name for the parameter, from the referred-to controlled vocabulary. This should be the preferred name associated with the specified accession number.				
unit_accession (FK)	unit_accession	VARCHAR(0)		
This field can optionally reference the CV accession number of the appropriate unit.				
cv_id (FK)	cv_id	VARCHAR(10)		NOT NULL
This field must reference the 'id' of the corresponding CV resource.				

References

- cv through (cv_id)
- cv_unit through (unit_accession)

3.7 cv_unit

This table lists the CV units used in the mzDB file.

Logical Column Name	Physical Column Name	Type	PK	Nullable
accession (PK)	accession	VARCHAR(0)	PK	NOT NULL
The unit accession number (e.g., 'UO:0000266' for 'electron volt').				
name	name	VARCHAR(0)		NOT NULL
The unit name (e.g., 'electron volt' for 'UO:0000266').				
cv_id (FK)	cv_id	VARCHAR(10)		NOT NULL
This field must reference the 'id' of the corresponding CV resource.				

References

- cv through (cv_id)

Referenced By

- cv_term referencing (accession)
- user_term referencing (accession)

3.8 data_encoding

Describes data encoding parameters used for data points storage.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
mode	mode	VARCHAR(10)		NOT NULL
Signal can be stored using 3 different modes: profile, centroided, fitted. Profile mode represent the scanned data in a (sometimes only approcimately) regularly spaced format, sometimes with gaps. Centroided mode present the scanned data only by specifying the location and intensity of individual detected peaks. Fitted mode is an enhencement of the centroided mode with additional information for each detected peak such as the left and right HWHMs.				
compression	compression	VARCHAR(0)		
This field provides optionally the name of the algorithm used for data compression (e.g. zlib, lzma...).				
byte_order	byte_order	VARCHAR(13)		NOT NULL
The byte order used for data encoding: little_endian or big_endian.				
mz_precision	mz_precision	INTEGER		NOT NULL
The floating number precision used for encoding of m/z values. Must be 32 or 64.				
intensity_precision	intensity_precision	INTEGER		NOT NULL
The floating number precision used for encoding of intensity values. Must be 32 or 64.				
param_tree	param_tree	CLOB		
This field must contain the table cvParams and userParams as an XML string (see XML schema specifications).				

Referenced By

- spectrum referencing (id)
- chromatogram referencing (id)

3.9 data_processing

Description of the way in which particular software were used. Variable methods should be described in the appropriate acquisition section - if no acquisition-specific details are found, then this information serves as the default.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
name	name	VARCHAR(0)		NOT NULL
A unique name for this data processing.				

Referenced By

- processing_method referencing (id)
- spectrum referencing (id)
- run referencing (id)
- run referencing (id)
- chromatogram referencing (id)

3.10 instrument_configuration

Description of a particular hardware configuration of a mass spectrometer. Each configuration must have one (and only one) of the three different components used for an analysis. For hybrid instruments, such as an LTQ-FT, there must be one configuration for each permutation of the components that is used in the document.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
name	name	VARCHAR(0)		NOT NULL
A unique name for this instrument configuration.				
param_tree	param_tree	CLOB		NOT NULL
This field must contain the table cvParams and userParams as an XML string (see XML schema specifications).				
component_list	component_list	CLOB		NOT NULL
This field must contain the componentList as an XML string (see XML schema specifications).				
shared_param_tree_id (FK)	shared_param_tree_id	INTEGER		
An optional reference to a shared_param_tree, which is a reusable container of one or more cvParams.				
software_id (FK)	software_id	INTEGER		NOT NULL
This field must be used to reference the 'id' of the appropriate software.				

References

- software through (software_id)
- shared_param_tree through (shared_param_tree_id)

Referenced By

- run referencing (id)
- spectrum referencing (id)

3.11 mzdb

Information pertaining to the entire mzDB file (i.e. not specific to any part of the data set) is stored here.

Logical Column Name	Physical Column Name	Type	PK	Nullable
version (PK)	version	VARCHAR(10)	PK	NOT NULL
The version of the mzDB format using tho following notation: majorVersion.minorVersion.maintenanceVersion				
creation_timestamp	creation_timestamp	VARCHAR(0)		NOT NULL
The creation date of the mzDB file in the ISO-8601 notation (e.g. 2007-06-27T15:23:45.00035).				

file_content	file_content	CLOB		NOT NULL
This field must contain the fileContent as an XML string (see XML schema specifications).				
contact	contact	CLOB		NOT NULL
This field must contain the contacts as an XML string (see XML schema specifications).				
param_tree	param_tree	CLOB		NOT NULL
This field must contain the table cvParams and userParams as an XML string (see XML schema specifications). These user params must be defined: BB_height_ms1, BB_height_msn, BB_width_ms1, BB_width_msn, is_no_loss.				

3.12 param_tree_schema

Each record describes the schema of the params that can be set in the param_tree column of the corresponding table or in the data column of the shared_param_tree table. This schema can then be used by the client application to validate the data structure to be stored.

Logical Column Name	Physical Column Name	Type	PK	Nullable
name (PK)	name	VARCHAR(0)	PK	NOT NULL
The name of the schema.				
type	type	VARCHAR(10)		NOT NULL
The serialization format used for param tree data. May be XSD or JSON, but only XSD is currently supported.				
schema	schema	CLOB		NOT NULL
This field must contain the schema of the corresponding param_tree.				

Referenced By

- shared_param_tree referencing (name)
- table_param_tree_schema referencing (name)

3.13 processing_method

Description of the way in which a particular software was used. The default peak processing method describes the base method used in the generation of a particular mzML file. Variable methods should be described in the appropriate acquisition section - if no acquisition-specific details are found, then this information serves as the default.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
order	order	INTEGER		NOT NULL
This field allows a series of consecutive steps to be placed in the correct order. Processing methods are ordered across all data processings.				
param_tree	param_tree	CLOB		NOT NULL
This field must contain the table cvParams and userParams as an XML string (see XML schema specifications).				
shared_param_tree_id (FK)	shared_param_tree_id	INTEGER		
An optional reference to a shared_param_tree, which is a reusable container of one or more cvParams.				
data_processing_id (FK)	data_processing_id	INTEGER		NOT NULL
This field must reference the 'id' of the corresponding data processing.				
software_id (FK)	software_id	INTEGER		NOT NULL
This field must reference the 'id' of the appropriate software.				

References

- software through (software_id)
- shared_param_tree through (shared_param_tree_id)
- data_processing through (data_processing_id)

3.14 run

A run in mzDB should correspond to a coherent set of spectra acquired on an instrument.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK) Automatically incremented primary key.	id	INTEGER	PK	NOT NULL
name A unique name for this run.	name	VARCHAR(0)		NOT NULL
start_timestamp The optional start timestamp of the run in the ISO-8601 notation (e.g. 2007-06-27T15:23:45.00035).	start_timestamp	VARCHAR(0)		
param_tree This field can contain optionally the table cvParams and userParams as an XML string (see XML schema specifications).	param_tree	CLOB		
shared_param_tree_id (FK) An optional reference to a shared_param_tree, which is a reusable container of one or more cvParams.	shared_param_tree_id	INTEGER		
sample_id (FK) This field must reference the 'id' of the appropriate sample.	sample_id	INTEGER		
default_instrument_config_id (FK) This field MUST reference the 'id' of the default instrument configuration. If a scan does not reference an instrument configuration, it implicitly refers to this configuration.	default_instrument_config_id	INTEGER		NOT NULL
default_source_file_id (FK) This field can optionally reference the 'id' of the default source file. If a spectrum or scan does not reference a source file and this field is set, then it implicitly refers to this source file.	default_source_file_id	INTEGER		
default_scan_processing_id (FK) This field MUST reference the 'id' of the default data processing for the scans. If an acquisition does not reference any data processing, it implicitly refers to this data processing. This field is required because the minimum amount of data processing that any format will undergo is "conversion to mzDB".	default_scan_processing_id	INTEGER		NOT NULL
default_chrom_processing_id (FK) This field MUST reference the 'id' of the default data processing for the chromatograms. If an acquisition does not reference any data processing, it implicitly refers to this data processing. This field is required because the minimum amount of data processing that any format will undergo is "conversion to mzDB".	default_chrom_processing_id	INTEGER		NOT NULL

References

- source_file through (default_source_file_id)
- sample through (sample_id)
- instrument_configuration through (default_instrument_config_id)
- shared_param_tree through (shared_param_tree_id)
- data_processing through (default_chrom_processing_id)
- data_processing through (default_scan_processing_id)

Referenced By

- spectrum referencing (id)

- run_slice referencing (id)
- chromatogram referencing (id)

3.15 run_slice

A "run slice" is a region of the run containing MS peaks in a given m/z scan window and in the whole duration of the run.
Constraints: UNIQUE(number, ms_level)

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
ms_level	ms_level	INTEGER		NOT NULL
Stage of MS achieved for this run slice in a multi stage mass spectrometry experiment.				
number	number	INTEGER		NOT NULL
This field allows a series of consecutive run slices to be placed in the correct order.				
begin_mz	begin_mz	REAL		NOT NULL
The theoretical minimum m/z of the run slice.				
end_mz	end_mz	REAL		NOT NULL
The theoretical maximum m/z of the run slice.				
param_tree	param_tree	CLOB		
This field can contain optionally the table cvParams and userParams as an XML string (see XML schema specifications).				
run_id (FK)	run_id	INTEGER		NOT NULL
This field must reference the 'id' of the corresponding run.				

References

- run through (run_id)

Referenced By

- bounding_box referencing (id)

3.16 sample

Description of the samples used to generate the dataset.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
name	name	VARCHAR(0)		NOT NULL
A unique name for the sample description.				
param_tree	param_tree	CLOB		
This field can contain optionally the table cvParams and userParams as an XML string (see XML schema specifications).				
shared_param_tree_id (FK)	shared_param_tree_id	INTEGER		
An optional reference to a shared_param_tree, which is a reusable container of one or more cvParams.				

References

- shared_param_tree through (shared_param_tree_id)

Referenced By

- run referencing (id)

3.17 scan_settings

Description of the acquisition settings of the instrument prior to the start of the run.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
param_tree	param_tree	CLOB		
This field can contain optionally the table cvParams and userParams as an XML string (see XML schema specifications).				
shared_param_tree_id (FK)	shared_param_tree_id	INTEGER		
An optional reference to a shared_param_tree, which is a reusable container of one or more cvParams.				

References

- shared_param_tree through (shared_param_tree_id)

Referenced By

- source_file_scan_settings_map referencing (id)
- target referencing (id)

3.18 shared_param_tree

A tree of CVParam and UserParam elements that can be shared with some mzDB tables.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
data	data	CLOB		NOT NULL
This field must contain a tree of cvParams and userParams as an XML string. The later must valid regarding the schema defined in the correspondig param_tre_shema record.				
schema_name (FK)	schema_name	VARCHAR(0)		NOT NULL
This field must reference the name of the appropriate param tree schema.				

References

- param_tree_schema through (schema_name)

Referenced By

- instrument_configuration referencing (id)
- spectrum referencing (id)
- scan_settings referencing (id)
- software referencing (id)

- run referencing (id)
- target referencing (id)
- sample referencing (id)
- processing_method referencing (id)
- source_file referencing (id)
- chromatogram referencing (id)

3.19 software

Software used to acquire and/or process the data.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK) Automatically incremented primary key.	id	INTEGER	PK	NOT NULL
name A unique name for this software.	name	VARCHAR(0)		NOT NULL
version The software version.	version	VARCHAR(0)		NOT NULL
param_tree This field must contain the table cvParams and userParams as an XML string (see XML schema specifications).	param_tree	CLOB		NOT NULL
shared_param_tree_id (FK) An optional reference to a shared_param_tree, which is a reusable container of one or more cvParams.	shared_param_tree_id	INTEGER		

References

- shared_param_tree through (shared_param_tree_id)

Referenced By

- instrument_configuration referencing (id)
- processing_method referencing (id)

3.20 source_file

Descriptions of the source files this mzDB was generated or derived from.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK) Automatically incremented primary key.	id	INTEGER	PK	NOT NULL
name Name of the source file, without reference to location (either URI or local path).	name	VARCHAR(0)		NOT NULL
location URI-formatted location where the file was retrieved.	location	VARCHAR(0)		NOT NULL
param_tree This field must contain the table cvParams and userParams as an XML string (see XML schema specifications).	param_tree	CLOB		NOT NULL
shared_param_tree_id (FK) An optional reference to a shared_param_tree, which is a reusable container of one or more cvParams.	shared_param_tree_id	INTEGER		

References

- shared_param_tree through (shared_param_tree_id)

Referenced By

- run referencing (id)
- source_file_scan_settings_map referencing (id)
- spectrum referencing (id)

3.21 source_file_scan_settings_map

Mapping between the source files and the acquisition settings.

Logical Column Name	Physical Column Name	Type	PK	Nullable
scan_settings_id (PK) (FK)	scan_settings_id	INTEGER	PK	NOT NULL
This field must reference the 'id' of the corresponding scan settings.				
source_file_id (PK) (FK)	source_file_id	INTEGER	PK	NOT NULL
This field must reference the 'id' of the corresponding source file.				

References

- source_file through (source_file_id)
- scan_settings through (scan_settings_id)

3.22 spectrum

The structure that captures the generation of a peak list. Also describes some of the parameters for the mass spectrometer for a given acquisition (or list of acquisitions). Subsidiary acquisition data points are stored in corresponding bounding boxes.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
initial_id	initial_id	INTEGER		NOT NULL
The native identifier of the spectrum as an integer.				
title	title	VARCHAR(0)		NOT NULL
A title describing this spectrum. It may correspond to the 'filter string' CV param.				
cycle	cycle	INTEGER		NOT NULL
The cycle number this spectrum refers to. A given cycle will typically contains an MS1 scan and all related MS2 scans.				
time	time	REAL		NOT NULL
The scan starting time in seconds.				
ms_level	ms_level	INTEGER		NOT NULL
Stage of MS achieved for this spectrum in a multi stage mass spectrometry experiment.				
activation_type	activation_type	VARCHAR(10)		NOT NULL
The type of activation used for fragmentation. This field stores only a 3 letters label (e.g. CID, ETD, HCD), but more detailled information could be provided using the param_tree field.				
tic	tic	REAL		NOT NULL
The total ion current of this spectrum.				
base_peak_mz	base_peak_moz	DOUBLE		NOT NULL

The m/z value of the most intense peak.			
base_peak_intensity	base_peak_intensity	REAL	NOT NULL
The intensity value of the most intense peak.			
main_precursor_mz	main_precursor_mz	DOUBLE	
The m/z value of the main precursor ion (first entry of precursor list).			
main_precursor_charge	main_precursor_charge	INTEGER	
The charge value of the main precursor ion (first entry of precursor list).			
data_points_count	data_points_count	INTEGER	NOT NULL
The number of data points for this spectrum.			
param_tree	param_tree	CLOB	NOT NULL
This field must contain the table cvParams and userParams as an XML string (see XML schema specifications).			
scan_list	scan_list	CLOB	
List and descriptions of scans.			
precursor_list	precursor_list	CLOB	
List and descriptions of precursor isolations to the spectrum currently being described, ordered.			
product_list	product_list	CLOB	
List and descriptions of product isolations to the spectrum currently being described, ordered.			
shared_param_tree_id (FK)	shared_param_tree_id	INTEGER	
An optional reference to a shared_param_tree, which is a reusable container of one or more cvParams.			
instrument_configuration_id (FK)	instrument_configuration_id	INTEGER	
This field can optionally reference the 'id' of the appropriate instrument configuration.			
source_file_id (FK)	source_file_id	INTEGER	
This field can optionally reference the 'id' of the appropriate source_file.			
run_id (FK)	run_id	INTEGER	NOT NULL
This field must reference the 'id' of the corresponding run.			
data_processing_id (FK)	data_processing_id	INTEGER	
This field can optionally reference the 'id' of the appropriate data_processing (see processing_method table).			
data_encoding_id (FK)	data_encoding_id	INTEGER	NOT NULL
This field must reference the 'id' of the appropriate data_encoding.			
bb_first_spectrum_id (FK)	bb_first_spectrum_id	INTEGER	NOT NULL
The first spectrum id of the bounding box (BB) this scan refers to. In the mzDB a BB is only linked to the first and last spectra of the stored data region. Thus to know the BB corresponding to each spectrum of the run one has to use this foreign key referencing a spectrum that is effectively linked to a BB.			

References

- spectrum through (bb_first_spectrum_id)
- run through (run_id)
- source_file through (source_file_id)
- instrument_configuration through (instrument_configuration_id)
- data_encoding through (data_encoding_id)
- shared_param_tree through (shared_param_tree_id)
- data_processing through (data_processing_id)

Referenced By

- spectrum referencing (id)
- bounding_box referencing (id)
- bounding_box referencing (id)

3.23 table_param_tree_schema

This table allows to 'logically' map a given table to its corresponding param_tree schema.

Logical Column Name	Physical Column Name	Type	PK	Nullable
table_name (PK)	table_name	VARCHAR(0)	PK	NOT NULL
The name of the table.				
schema_name (FK)	schema_name	VARCHAR(0)		NOT NULL
The name of the schema.				

References

- param_tree_schema through (schema_name)

3.24 target

Contains the target list (or 'inclusion list') configured prior to the run.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
param_tree	param_tree	CLOB		NOT NULL
This field must contain the table cvParams and userParams as an XML string (see XML schema specifications).				
shared_param_tree_id (FK)	shared_param_tree_id	INTEGER		
An optional reference to a shared_param_tree, which is a reusable container of one or more cvParams.				
scan_settings_id (FK)	scan_settings_id	INTEGER		NOT NULL
Automatically incremented primary key.				

References

- scan_settings through (scan_settings_id)
- shared_param_tree through (shared_param_tree_id)

3.25 user_term

This table lists the uncontrolled user parameters used in the mzDB file.

Logical Column Name	Physical Column Name	Type	PK	Nullable
id (PK)	id	INTEGER	PK	NOT NULL
Automatically incremented primary key.				
name	name	VARCHAR(0)		NOT NULL
The unique name for the parameter.				
type	type	VARCHAR(0)		NOT NULL
The datatype of the parameter, where appropriate (e.g.: xsd:float).				
unit_accession (FK)	unit_accession	VARCHAR(0)		
This field can optionally reference the CV accession number of the appropriate unit.				

References

- cv_unit through (unit_accession)

4. Metadata XML schemata

4.1 Element <params>

Definition: A tree of CVParam and UserParam elements.

Type: dx:ParamGroupType

Subelements:

Subelement Name	min	max	Definition
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.
userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead

Example Context:

```
<params>
  <cvParam cvRef="MS" accession="MS:1000133" name="collision-induced dissociation"
value=""/>
  <cvParam cvRef="MS" accession="MS:1000045" name="collision energy" value="35"
unitCvRef="UO" unitAccession="UO:0000266" unitName="electronvolt"/>
  <cvParam cvRef="MS" accession="MS:1000419" name="collision gas" value="nitrogen"/>
</params>
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000579" name="MS1 spectrum" value=""/>
<cvParam cvRef="MS" accession="MS:1000130" name="positive scan" value=""/>
<cvParam cvRef="MS" accession="MS:1000580" name="MSn spectrum" value=""/>
<cvParam cvRef="MS" accession="MS:1000514" name="m/z array" unitCvRef="MS"
unitAccession="MS:1000040" unitName="m/z"/>
<cvParam cvRef="MS" accession="MS:1000523" name="64-bit float"/>
<cvParam cvRef="MS" accession="MS:1000576" name="no compression"/>
<cvParam cvRef="MS" accession="MS:1000515" name="intensity array" unitCvRef="MS"
unitAccession="MS:1000131" unitName="number of counts"/>
<cvParam cvRef="MS" accession="MS:1000521" name="32-bit float"/>
<cvParam cvRef="MS" accession="MS:1000516" name="charge array"/>
<cvParam cvRef="MS" accession="MS:1000448" name="LTQ FT" value=""/>
<cvParam cvRef="MS" accession="MS:1000529" name="instrument serial number"
value="SN06061F"/>
<cvParam cvRef="MS" accession="MS:1000032" name="customization" value="none"/>
<cvParam cvRef="MS" accession="MS:1000133" name="collision-induced dissociation"
value=""/>
<cvParam cvRef="MS" accession="MS:1000045" name="collision energy" value="35"
unitCvRef="UO" unitAccession="UO:0000266" unitName="electronvolt"/>
<cvParam cvRef="MS" accession="MS:1000419" name="collision gas" value="nitrogen"/>
```

Notes This element is referred as param_tree in the SQL schema.

4.2 Element <fileContent>

Definition: This summarizes the different types of spectra that can be expected in the file. This is expected to aid processing software in skipping files that do not contain appropriate spectrum types for it. It should also describe the nativeID format used in the file by referring to an appropriate CV term.

Type: dx:ParamGroupType

Attributes: none

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.

cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.
userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead

Example Context:

```
<fileContent>
  <cvParam cvRef="MS" accession="MS:1000580" name="MSn spectrum" value=""/>
  <userParam name="ProteoWizard" value="Thermo RAW data converted to mzML, with additional
MIAPE parameters added for illustration"/>
</fileContent>
```

cvParam Mapping Rules:

```
MUST supply a *child* term of MS:1000524 (data file content) one or more times
  e.g.: MS:1000235 (total ion current chromatogram)
  e.g.: MS:1000235 (total ion current chromatogram)
  e.g.: MS:1000322 (charge inversion mass spectrum)
  e.g.: MS:1000322 (charge inversion mass spectrum)
  e.g.: MS:1000325 (constant neutral gain spectrum)
  e.g.: MS:1000325 (constant neutral gain spectrum)
  e.g.: MS:1000326 (constant neutral loss spectrum)
  e.g.: MS:1000326 (constant neutral loss spectrum)
  e.g.: MS:1000328 (e/2 mass spectrum)
  e.g.: MS:1000341 (precursor ion spectrum)
  et al.
MAY supply a *child* term of MS:1000525 (spectrum representation) only once
  e.g.: MS:1000127 (centroid spectrum)
  e.g.: MS:1000128 (profile spectrum)
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000580" name="MSn spectrum" value=""/>
<cvParam cvRef="MS" accession="MS:1000127" name="centroid spectrum" value=""/>
<cvParam cvRef="MS" accession="MS:1000326" name="constant neutral loss spectrum"/>
```

4.3 Element <contact>

Definition: Structure allowing the use of a controlled (cvParam) or uncontrolled vocabulary (userParam), or a reference to a predefined set of these in this mzML file (paramGroupRef).

Type: dx:ParamGroupType

Attributes: none

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.
userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead

Example Context:

```
<contact>
  <cvParam cvRef="MS" accession="MS:1000586" name="contact name" value="William
Pennington"/>
  <cvParam cvRef="MS" accession="MS:1000590" name="contact organization"
value="Higglesworth University"/>
  <cvParam cvRef="MS" accession="MS:1000587" name="contact address" value="12
Higglesworth Avenue, 12045, HI, USA"/>
  <cvParam cvRef="MS" accession="MS:1000588" name="contact URL"
value="http://www.higglesworth.edu/"/>
  <cvParam cvRef="MS" accession="MS:1000589" name="contact email"
value="wpennington@higglesworth.edu"/>
</contact>
```

cvParam Mapping Rules:

```
MAY supply a *child* term of MS:1000585 (contact person attribute) one or more times
  e.g.: MS:1000586 (contact name)
  e.g.: MS:1000587 (contact address)
  e.g.: MS:1000588 (contact URL)
  e.g.: MS:1000589 (contact email)
  e.g.: MS:1000590 (contact organization)
MUST supply term MS:1000590 (contact organization)  only once
MUST supply term MS:1000586 (contact name)  only once
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000586" name="contact name" value="William
Pennington"/>
<cvParam cvRef="MS" accession="MS:1000590" name="contact organization"
value="Higglesworth University"/>
<cvParam cvRef="MS" accession="MS:1000587" name="contact address" value="12 Higglesworth
Avenue, 12045, HI, USA"/>
<cvParam cvRef="MS" accession="MS:1000588" name="contact URL"
value="http://www.higglesworth.edu/"/>
<cvParam cvRef="MS" accession="MS:1000589" name="contact email"
value="wpennington@higglesworth.edu"/>
```

4.4 Element <componentList>

Definition: List with the different components used in the mass spectrometer. At least one source, one mass analyzer and one detector need to be specified.

Type: dx:ComponentListType

Attributes:

Attribute Name	Data Type	Use	Definition
count	xs:nonNegativeInteger	required	The number of components in this list.

Subelements:

Subelement Name	min	max	Definition
source	1	unlim	A source component.
analyzer	1	unlim	A mass analyzer (or mass filter) component.
detector	1	unlim	A detector component.

Example Context:

```
<componentList count="3">
  <source order="1">
    <cvParam cvRef="MS" accession="MS:1000073" name="electrospray ionization" value=""/>
    <cvParam cvRef="MS" accession="MS:1000057" name="electrospray inlet" value=""/>
    <cvParam cvRef="MS" accession="MS:1000486" name="source potential" value="4.20"
unitCvRef="UO" unitAccession="UO:0000218" unitName="volt"/>
  </source>
  <analyzer order="2">
  ...
</componentList>
```

4.5 Element <source>

Definition: A source component.

Type: dx:SourceComponentType

Attributes:

Attribute Name	Data Type	Use	Definition
order	xs:int	required	This attribute MUST be used to indicate the order in which the components are encountered from source to detector (e.g., in a Q-TOF, the quadrupole would have the lower order number, and the TOF the higher number of the two).

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.

userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead

Example Context:

```
<source order="1">
  <cvParam cvRef="MS" accession="MS:1000073" name="electrospray ionization" value=""/>
  <cvParam cvRef="MS" accession="MS:1000057" name="electrospray inlet" value=""/>
  <cvParam cvRef="MS" accession="MS:1000486" name="source potential" value="4.20"
unitCvRef="UO" unitAccession="UO:0000218" unitName="volt"/>
</source>
```

cvParam Mapping Rules:

Path componentList/source

```
MAY supply a *child* term of MS:1000482 (source attribute) one or more times
  e.g.: MS:1000392 (ionization efficiency)
  e.g.: MS:1000486 (source potential)
MUST supply term MS:1000008 (ionization type) or any of its children only once
  e.g.: MS:1000070 (atmospheric pressure chemical ionization)
  e.g.: MS:1000071 (chemical ionization)
  e.g.: MS:1000074 (fast atom bombardment ionization)
  e.g.: MS:1000075 (matrix-assisted laser desorption ionization)
  e.g.: MS:1000227 (multiphoton ionization)
  e.g.: MS:1000239 (atmospheric pressure matrix-assisted laser desorption ionization)
  e.g.: MS:1000255 (flowing afterglow)
  e.g.: MS:1000257 (field desorption)
  e.g.: MS:1000258 (field ionization)
  e.g.: MS:1000259 (glow discharge ionization)
  et al.
MAY supply a *child* term of MS:1000007 (inlet type) only once
  e.g.: MS:1000055 (continuous flow fast atom bombardment)
  e.g.: MS:1000056 (direct inlet)
  e.g.: MS:1000058 (flow injection analysis)
  e.g.: MS:1000059 (inductively coupled plasma)
  e.g.: MS:1000060 (infusion)
  e.g.: MS:1000061 (jet separator)
  e.g.: MS:1000062 (membrane separator)
  e.g.: MS:1000063 (moving belt)
  e.g.: MS:1000064 (moving wire)
  e.g.: MS:1000065 (open split)
  et al.
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000398" name="nanoelectrospray" value=""/>
<cvParam cvRef="MS" accession="MS:1000073" name="electrospray ionization" value=""/>
<cvParam cvRef="MS" accession="MS:1000057" name="electrospray inlet" value=""/>
<cvParam cvRef="MS" accession="MS:1000486" name="source potential" value="4.20"
unitCvRef="UO" unitAccession="UO:0000218" unitName="volt"/>
```

4.6 Element <analyzer>

Definition: A mass analyzer (or mass filter) component.

Type: dx:AnalyzerComponentType

Attributes:

Attribute Name	Data Type	Use	Definition
order	xs:int	required	This attribute MUST be used to indicate the order in which the components are encountered from source to detector (e.g., in a Q-TOF, the quadrupole would have the lower order number, and the TOF the higher number of the two).

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.
userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before

			using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead

Example Context:

```
<analyzer order="2">
  <cvParam cvRef="MS" accession="MS:1000079" name="fourier transform ion cyclotron
resonance mass spectrometer" value=""/>
</analyzer>
```

cvParam Mapping Rules:

Path componentList/analyzer

```
MAY supply a *child* term of MS:1000480 (mass analyzer attribute) one or more times
  e.g.: MS:1000014 (accuracy)
  e.g.: MS:1000022 (TOF Total Path Length)
  e.g.: MS:1000024 (final MS exponent)
  e.g.: MS:1000025 (magnetic field strength)
  e.g.: MS:1000105 (reflectron off)
  e.g.: MS:1000106 (reflectron on)
MUST supply term MS:1000443 (mass analyzer type) or any of its children only once
  e.g.: MS:1000078 (axial ejection linear ion trap)
  e.g.: MS:1000079 (fourier transform ion cyclotron resonance mass spectrometer)
  e.g.: MS:1000080 (magnetic sector)
  e.g.: MS:1000081 (quadrupole)
  e.g.: MS:1000082 (quadrupole ion trap)
  e.g.: MS:1000083 (radial ejection linear ion trap)
  e.g.: MS:1000084 (time-of-flight)
  e.g.: MS:1000254 (electrostatic energy analyzer)
  e.g.: MS:1000284 (stored waveform inverse fourier transform)
  e.g.: MS:1000288 (cyclotron)
  et al.
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000082" name="quadrupole ion trap" value=""/>
<cvParam cvRef="MS" accession="MS:1000081" name="quadrupole"/>
<cvParam cvRef="MS" accession="MS:1000084" name="time-of-flight"/>
<cvParam cvRef="MS" accession="MS:1000079" name="fourier transform ion cyclotron resonance
mass spectrometer" value=""/>
<cvParam cvRef="MS" accession="MS:1000083" name="radial ejection linear ion trap"
value=""/>
```

4.7 Element <detector>

Definition: A detector component.

Type: dx:DetectorComponentType

Attributes:

Attribute Name	Data Type	Use	Definition
order	xs:int	required	This attribute MUST be used to indicate the order in which the components are encountered from source to detector (e.g., in a Q-TOF, the quadrupole would have the lower order number, and the TOF the higher number of the two).

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.
userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead

Example Context:

```
        <detector order="4">
          <cvParam cvRef="MS" accession="MS:1000114" name="microchannel plate detector"/>
        </detector>
```

cvParam

Path componentList/detector

```
MUST supply term MS:1000026 (detector type) or any of its children only once
```

Mapping Rules:

```
  e.g.: MS:1000107 (channeltron)
  e.g.: MS:1000108 (conversion dynode electron multiplier)
  e.g.: MS:1000109 (conversion dynode photomultiplier)
  e.g.: MS:1000110 (daly detector)
  e.g.: MS:1000111 (electron multiplier tube)
  e.g.: MS:1000112 (faraday cup)
  e.g.: MS:1000113 (focal plane array)
  e.g.: MS:1000114 (microchannel plate detector)
  e.g.: MS:1000115 (multi-collector)
  e.g.: MS:1000116 (photomultiplier)
  et al.
MAY supply a *child* term of MS:1000481 (detector attribute) one or more times
  e.g.: MS:1000028 (detector resolution)
  e.g.: MS:1000029 (sampling frequency)
MAY supply a *child* term of MS:1000027 (detector acquisition mode) one or more times
  e.g.: MS:1000117 (analog-digital converter)
  e.g.: MS:1000118 (pulse counting)
  e.g.: MS:1000119 (time-digital converter)
  e.g.: MS:1000120 (transient recorder)
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000253" name="electron multiplier" value=""/>
<cvParam cvRef="MS" accession="MS:1000114" name="microchannel plate detector"/>
<cvParam cvRef="MS" accession="MS:1000624" name="inductive detector" value=""/>
```

4.8 Element <scanList>

Definition: List and descriptions of scans.

Type: dx:ScanListType

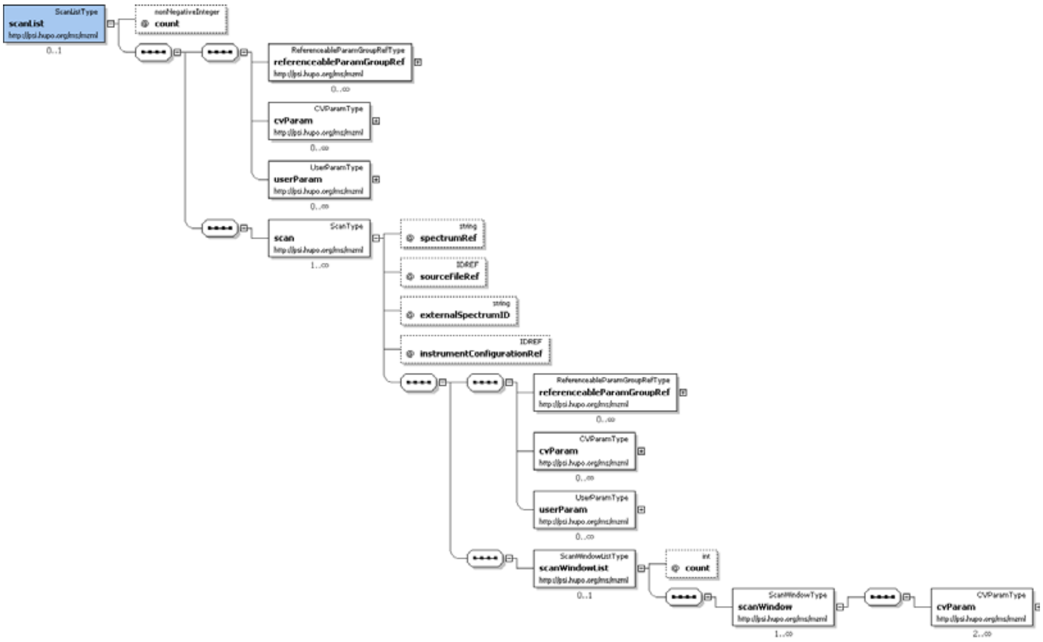
Attributes:

Attribute Name	Data Type	Use	Definition
count	xs:nonNegativeInteger	required	the number of scans defined in this list.

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.
userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead
scan	1	unlim	Scan or acquisition from original raw file used to create this peak list, as specified in sourceFile.

Graphical Context:


ScanListType
scanList
http://psi.hupo.org/ms/mzml
0..1
nonNegativeInteger
count
ReferenceableParamGroupRefType
referenceableParamGroupRef
http://psi.hupo.org/ms/mzml
0..∞
CVParamType
cvParam
http://psi.hupo.org/ms/mzml
0..∞
UserParamType
userParam
http://psi.hupo.org/ms/mzml
0..∞
ScanType
scan
http://psi.hupo.org/ms/mzml
1..∞
string
spectrumRef
IDREF
sourceFileRef
string
externalSpectrumID
IDREF
instrumentConfigurationRef
ReferenceableParamGroupRefType
referenceableParamGroupRef
http://psi.hupo.org/ms/mzml
0..∞
CVParamType
cvParam
http://psi.hupo.org/ms/mzml
0..∞
UserParamType
userParam
http://psi.hupo.org/ms/mzml
0..∞
ScanWindowListType
scanWindowList
http://psi.hupo.org/ms/mzml
0..1
int
count
ScanWindowType
scanWindow
http://psi.hupo.org/ms/mzml
1..∞
CVParamType
cvParam
http://psi.hupo.org/ms/mzml
2..∞

Example Context:

```
<scanList count="1">
  <cvParam cvRef="MS" accession="MS:1000795" name="no combination" value=""/>
  <scan instrumentConfigurationRef="IC2">
    <cvParam cvRef="MS" accession="MS:1000016" name="scan start time"
value="0.011218333333333334" unitCvRef="UO" unitAccession="UO:0000031" unitName="minute"/>
    <cvParam cvRef="MS" accession="MS:1000512" name="filter string" value="ITMS + c ESI d
Full ms2 810.79@cid35.00 [210.00-1635.00]"/>
    <cvParam cvRef="MS" accession="MS:1000616" name="preset scan configuration" value="3"/>
    <userParam name="[Thermo Trailer Extra]Monoisotopic M/Z:" value="0" type="xsd:float"/>
  ...
</scanList>
```

cvParam Mapping Rules:

```
MUST supply a *child* term of MS:1000570 (spectra combination) only once
  e.g.: MS:1000571 (sum of spectra)
  e.g.: MS:1000573 (median of spectra)
  e.g.: MS:1000575 (mean of spectra)
  e.g.: MS:1000795 (no combination)
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000795" name="no combination" value=""/>
<cvParam cvRef="MS" accession="MS:1000571" name="sum of spectra"/>
```

4.9 Element <precursorList>

Definition: List and descriptions of precursor isolations to the spectrum currently being described, ordered.

Type: dx:PrecursorListType

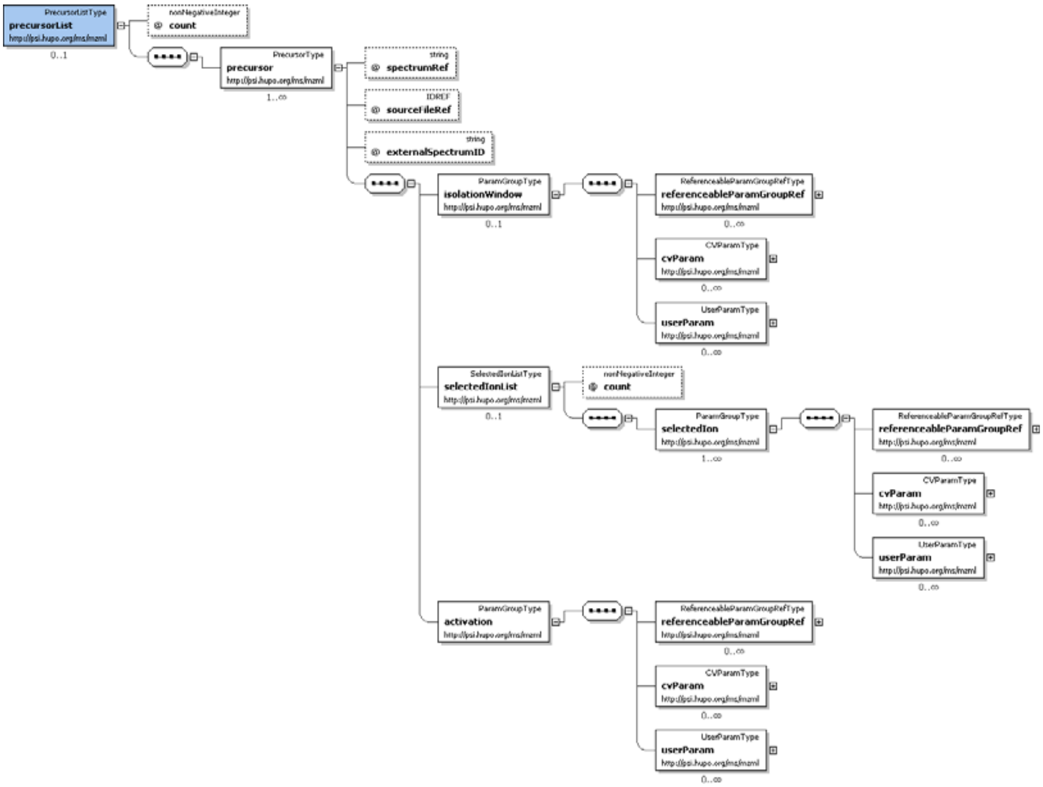
Attributes:

Attribute Name	Data Type	Use	Definition
count	xs:nonNegativeInteger	required	The number of precursor isolations in this list.

Subelements:

Subelement Name	min	max	Definition
precursor	1	unlim	The method of precursor ion selection and activation

Graphical Context:


PrecursorListType
precursorList
http://psi.hupo.org/ms/mzml
0..1
nonNegativeInteger
count
PrecursorType
precursor
http://psi.hupo.org/ms/mzml
1..∞
string
spectrumRef
IDREF
sourceFileRef
string
externalSpectrumID
ParamGroupType
isolationWindow
http://psi.hupo.org/ms/mzml
0..1
ReferenceableParamGroupRefType
referenceableParamGroupRef
http://psi.hupo.org/ms/mzml
0..∞
CVParamType
cvParam
http://psi.hupo.org/ms/mzml
0..∞
UserParamType
userParam
http://psi.hupo.org/ms/mzml
0..∞
SelectedIonListType
selectedIonList
http://psi.hupo.org/ms/mzml
0..1
nonNegativeInteger
count
ParamGroupType
selectedIon
http://psi.hupo.org/ms/mzml
1..∞
ReferenceableParamGroupRefType
referenceableParamGroupRef
http://psi.hupo.org/ms/mzml
0..∞
CVParamType
cvParam
http://psi.hupo.org/ms/mzml
0..∞
UserParamType
userParam
http://psi.hupo.org/ms/mzml
0..∞
ParamGroupType
activation
http://psi.hupo.org/ms/mzml
ReferenceableParamGroupRefType
referenceableParamGroupRef
http://psi.hupo.org/ms/mzml
0..∞
CVParamType
cvParam
http://psi.hupo.org/ms/mzml
0..∞
UserParamType
userParam
http://psi.hupo.org/ms/mzml
0..∞

Example Context:

```
<precursorList count="1">
  <precursor spectrumRef="controllerType=0 controllerNumber=1 scan=16">
    <isolationWindow>
      <cvParam cvRef="MS" accession="MS:1000827" name="isolation window target m/z"
value="811.40999999999997" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
      <cvParam cvRef="MS" accession="MS:1000828" name="isolation window lower offset"
value="0.5" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
      <cvParam cvRef="MS" accession="MS:1000829" name="isolation window upper offset"
value="0.5" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
    </isolationWindow>
  ...
</precursorList>
```

4.10 Element <productList>

Definition: List and descriptions of product isolations to the spectrum currently being described, ordered.

Type: dx:ProductListType

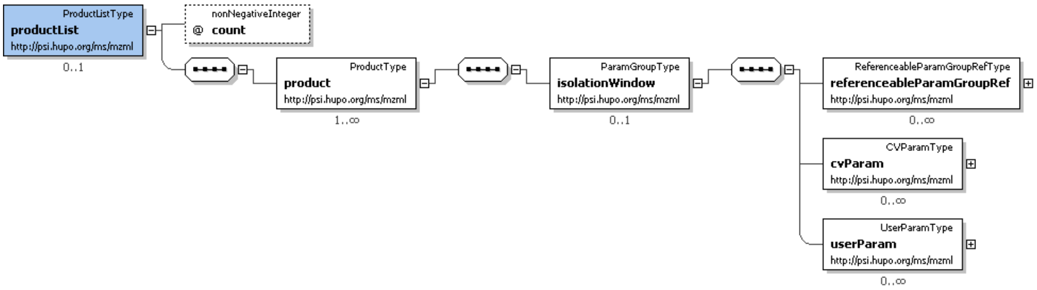
Attributes:

Attribute Name	Data Type	Use	Definition
count	xs:nonNegativeInteger	required	The number of product isolations in this list.

Subelements:

Subelement Name	min	max	Definition
product	1	unlim	The method of product ion selection and activation in a precursor ion scan

Graphical Context:


ProductListType
productList
http://psi.hupo.org/ms/mzml
0..1
nonNegativeInteger
@ count
ProductType
product
http://psi.hupo.org/ms/mzml
1..∞
ParamGroupType
isolationWindow
http://psi.hupo.org/ms/mzml
0..1
ReferenceableParamGroupRefType
referenceableParamGroupRef
http://psi.hupo.org/ms/mzml
0..∞
CVParamType
cvParam
http://psi.hupo.org/ms/mzml
0..∞
UserParamType
userParam
http://psi.hupo.org/ms/mzml
0..∞

Example Context:

```
<productList count="1">
 <product>
  <isolationWindow>
   <!-- Q3 transmission window -->
   <cvParam cvRef="MS" accession="MS:1000828" name="isolation window lower offset"
value="1.0" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
   <cvParam cvRef="MS" accession="MS:1000829" name="isolation window upper offset"
value="1.0" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
  </isolationWindow>
  ...
</productList>
```

4.11 Element <scan>

Definition: Scan or acquisition from original raw file used to create this peak list, as specified in sourceFile.

Type: dx:ScanType

Attributes:

Attribute Name	Data Type	Use	Definition
externalSpectrumID	xs:string	optional	For scans that are external to this document, this string MUST correspond to the 'id' attribute of a spectrum in the external document indicated by 'sourceFileRef'.
instrumentConfigurationRef	xs:IDREF	optional	This attribute can optionally reference the 'id' attribute of the appropriate instrument configuration.
sourceFileRef	xs:IDREF	optional	If this attribute is set, it MUST reference the 'id' attribute of a sourceFile representing the external document containing the spectrum referred to by 'externalSpectrumID'.
spectrumRef	xs:string	optional	For scans that are local to this document, this attribute can be used to reference the 'id' attribute of the spectrum corresponding to the scan.

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.
userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether

			there is an appropriate CV term available, and if so, use the CV term instead
scanWindowList	0	1	Container for a list of scan windows.

Example Context:

```
<scan instrumentConfigurationRef="LCQDeca">
  <cvParam cvRef="MS" accession="MS:1000016" name="scan start time"
value="42.049999999999997" unitCvRef="UO" unitAccession="UO:0000010" unitName="second"/>
  <cvParam cvRef="MS" accession="MS:1000512" name="filter string" value="+ c MALDI Full ms
[100.00-1000.00]"/>
  <scanWindowList count="1">
    <scanWindow>
      <cvParam cvRef="MS" accession="MS:1000501" name="scan window lower limit"
value="100" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
      <cvParam cvRef="MS" accession="MS:1000500" name="scan window upper limit"
value="1000" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
  ...
</scan>
```

cvParam Mapping Rules:

Path scanList/scan

```
MAY supply a *child* term of MS:1000503 (scan attribute) one or more times
  e.g.: MS:1000011 (mass resolution)
  e.g.: MS:1000015 (scan rate)
  e.g.: MS:1000016 (scan start time)
  e.g.: MS:1000502 (dwell time)
  e.g.: MS:1000512 (filter string)
  e.g.: MS:1000616 (preset scan configuration)
  e.g.: MS:1000800 (mass resolving power)
  e.g.: MS:1000803 (analyzer scan offset)
  e.g.: MS:1000826 (elution time)
MAY supply a *child* term of MS:1000018 (scan direction) only once
  e.g.: MS:1000092 (decreasing m/z scan)
  e.g.: MS:1000093 (increasing m/z scan)
MAY supply a *child* term of MS:1000019 (scan law) only once
  e.g.: MS:1000094 (exponential)
  e.g.: MS:1000095 (linear)
  e.g.: MS:1000096 (quadratic)
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000016" name="scan start time"
value="5.8905000000000003" unitCvRef="UO" unitAccession="UO:0000031" unitName="minute"/>
<cvParam cvRef="MS" accession="MS:1000512" name="filter string" value="+ c NSI Full ms [
400.00-1800.00]"/>
<cvParam cvRef="MS" accession="MS:1000616" name="preset scan configuration" value="3"/>
<cvParam cvRef="MS" accession="MS:1000803" name="analyzer scan offset" value="80"
unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
```

4.12 Element <precursor>

Definition: The method of precursor ion selection and activation

Type: dx:PrecursorType

Attributes:

Attribute Name	Data Type	Use	Definition
externalSpectrumID	xs:string	optional	For precursor spectra that are external to this document, this string MUST correspond to the 'id' attribute of a spectrum in the external document indicated by 'sourceFileRef'.
sourceFileRef	xs:IDREF	optional	For precursor spectra that are external to this document, this attribute MUST reference the 'id' attribute of a sourceFile representing that external document.
spectrumRef	xs:string	optional	For precursor spectra that are local to this document, this attribute MUST be used to reference the 'id' attribute of the spectrum corresponding to the precursor spectrum.

Subelements:

Subelement Name	min	max	Definition
isolationWindow	0	1	This element captures the isolation (or 'selection') window configured to isolate one or more ions.

selectedIonList	0	1	A list of ions that were selected.
activation	1	1	The type and energy level used for activation.

Example Context:

```
<precursor spectrumRef="controllerType=0 controllerNumber=1 scan=16">
  <isolationWindow>
    <cvParam cvRef="MS" accession="MS:1000827" name="isolation window target m/z"
value="811.40999999999997" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
    <cvParam cvRef="MS" accession="MS:1000828" name="isolation window lower offset"
value="0.5" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
    <cvParam cvRef="MS" accession="MS:1000829" name="isolation window upper offset"
value="0.5" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
  </isolationWindow>
  <selectedIonList count="1">
  ...
</precursor>
```

4.13 Element <product>

Definition: The method of product ion selection and activation in a precursor ion scan

Type: dx:ProductType

Attributes: none

Subelements:

Subelement Name	min	max	Definition
isolationWindow	0	1	This element captures the isolation (or 'selection') window configured to isolate one or more ions.

Example Context:

```
 <product>
  <isolationWindow>
   <!-- Q3 transmission window -->
   <cvParam cvRef="MS" accession="MS:1000828" name="isolation window lower offset"
value="1.0" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
   <cvParam cvRef="MS" accession="MS:1000829" name="isolation window upper offset"
value="1.0" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
  </isolationWindow>
 </product>
```

4.14 Element <scanWindowList>

Definition: Container for a list of scan windows.

Type: dx:ScanWindowListType

Attributes:

Attribute Name	Data Type	Use	Definition
count	xs:int	required	The number of scan windows defined in this list.

Subelements:

Subelement Name	min	max	Definition
scanWindow	1	unlim	A range of m/z values over which the instrument scans and aquires a spectrum.

Example Context:

```
<scanWindowList count="1">
  <scanWindow>
    <cvParam cvRef="MS" accession="MS:1000501" name="scan window lower limit" value="400"
unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
    <cvParam cvRef="MS" accession="MS:1000500" name="scan window upper limit"
value="1800" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
  </scanWindow>
</scanWindowList>
```

4.15 Element <isolationWindow>

Definition: This element captures the isolation (or 'selection') window configured to isolate one or more ions.

Type: dx:ParamGroupType

Attributes: none

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.
userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead

Example Context:

```
<isolationWindow>
  <cvParam cvRef="MS" accession="MS:1000827" name="isolation window target m/z"
value="445.30000000000001" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
  <cvParam cvRef="MS" accession="MS:1000828" name="isolation window lower offset"
value="0.5" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
  <cvParam cvRef="MS" accession="MS:1000829" name="isolation window upper offset"
value="0.5" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
</isolationWindow>
```

cvParam Mapping Rules:

Path precursorList/precursor/isolationWindow

```
MUST supply a *child* term of MS:1000792 (isolation window attribute) one or more times
  e.g.: MS:1000827 (isolation window target m/z)
  e.g.: MS:1000828 (isolation window lower offset)
  e.g.: MS:1000829 (isolation window upper offset)
```

Path productList/product/isolationWindow

```
MUST supply a *child* term of MS:1000792 (isolation window attribute) one or more times
  e.g.: MS:1000827 (isolation window target m/z)
  e.g.: MS:1000828 (isolation window lower offset)
  e.g.: MS:1000829 (isolation window upper offset)
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000827" name="isolation window target m/z"
value="445.30000000000001" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
<cvParam cvRef="MS" accession="MS:1000828" name="isolation window lower offset"
value="0.5" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
<cvParam cvRef="MS" accession="MS:1000829" name="isolation window upper offset"
value="0.5" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
```

4.16 Element <selectedIonList>

Definition: A list of ions that were selected.

Type: dx:SelectedIonListType

Attributes:

Attribute Name	Data Type	Use	Definition
count	xs:nonNegativeInteger	required	The number of selected precursor ions defined in this list.

Subelements:

Subelement Name	min	max	Definition
selectedIon	1	unlim	Structure allowing the use of a controlled (cvParam) or uncontrolled vocabulary (userParam), or a reference to a predefined set of these in this mzML file (paramGroupRef).

Example Context:

```
            <selectedIonList count="1">
              <selectedIon>
                <cvParam cvRef="MS" accession="MS:1000744" name="selected ion m/z"
value="1082.5037" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
                <cvParam cvRef="MS" accession="MS:1000633" name="possible charge state"
value="2"/>
                <cvParam cvRef="MS" accession="MS:1000633" name="possible charge state"
value="3"/>
              </selectedIon>
            </selectedIonList>
```

4.17 Element <activation>

Definition: The type and energy level used for activation.

Type: dx:ParamGroupType

Attributes: none

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.
userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead

Example Context:

```
<activation>
  <cvParam cvRef="MS" accession="MS:1000133" name="collision-induced dissociation"
value=""/>
  <cvParam cvRef="MS" accession="MS:1000045" name="collision energy" value="35"
unitCvRef="UO" unitAccession="UO:0000266" unitName="electronvolt"/>
</activation>
```

cvParam Mapping Rules:

Path precursorList/precursor/activation

```
MAY supply a *child* term of MS:1000510 (precursor activation attribute) one or more
times
  e.g.: MS:1000045 (collision energy)
  e.g.: MS:1000245 (charge stripping)
  e.g.: MS:1000412 (buffer gas)
  e.g.: MS:1000419 (collision gas)
  e.g.: MS:1000509 (activation energy)
MUST supply term MS:1000044 (dissociation method) or any of its children one or more
times
  e.g.: MS:1000133 (collision-induced dissociation)
  e.g.: MS:1000134 (plasma desorption)
  e.g.: MS:1000135 (post-source decay)
  e.g.: MS:1000136 (surface-induced dissociation)
  e.g.: MS:1000242 (blackbody infrared radiative dissociation)
  e.g.: MS:1000250 (electron capture dissociation)
  e.g.: MS:1000262 (infrared multiphoton dissociation)
  e.g.: MS:1000282 (sustained off-resonance irradiation)
  e.g.: MS:1000422 (high-energy collision-induced dissociation)
  e.g.: MS:1000433 (low-energy collision-induced dissociation)
  et al.
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000133" name="collision-induced dissociation"
value=""/>
<cvParam cvRef="MS" accession="MS:1000045" name="collision energy" value="35"
unitCvRef="UO" unitAccession="UO:0000266" unitName="electronvolt"/>
<cvParam cvRef="MS" accession="MS:1000044" name="dissociation method"/>
```

4.18 Element <scanWindow>

Definition: A range of m/z values over which the instrument scans and aquires a spectrum.

Type: dx:ParamGroupType

Attributes: none

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.

userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead

Example Context:

```
 <scanWindow>
  <cvParam cvRef="MS" accession="MS:1000501" name="scan window lower limit" value="400"
unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
  <cvParam cvRef="MS" accession="MS:1000500" name="scan window upper limit" value="1600"
unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
 </scanWindow>
```

cvParam Mapping Rules:

Path scanList/scan/scanWindowList/scanWindow

```
MAY supply a *child* term of MS:1000549 (selection window attribute) one or more times
  e.g.: MS:1000500 (scan window upper limit)
  e.g.: MS:1000501 (scan window lower limit)
MUST supply term MS:1000500 (scan window upper limit)  only once
MUST supply term MS:1000501 (scan window lower limit)  only once
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000501" name="scan window lower limit" value="400"
unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
<cvParam cvRef="MS" accession="MS:1000500" name="scan window upper limit" value="1800"
unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
```

4.19 Element <selectedIon>

Definition: Structure allowing the use of a controlled (cvParam) or uncontrolled vocabulary (userParam), or a reference to a predefined set of these in this mzML file (paramGroupRef).

Type: dx:ParamGroupType

Attributes: none

Subelements:

Subelement Name	min	max	Definition
referenceableParamGroupRef	0	unlim	A reference to a previously defined ParamGroup, which is a reusable container of one or more cvParams.
cvParam	0	unlim	This element holds additional data or annotation. Only controlled values are allowed here.
userParam	0	unlim	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead

Example Context:

```
              <selectedIon>
                <cvParam cvRef="MS" accession="MS:1000744" name="selected ion m/z"
value="1082.5037" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
                <cvParam cvRef="MS" accession="MS:1000633" name="possible charge state"
value="2"/>
                <cvParam cvRef="MS" accession="MS:1000633" name="possible charge state"
value="3"/>
              </selectedIon>
```

cvParam Mapping Rules:

Path precursorList/precursor/selectedIonList/selectedIon

```
MUST supply a *child* term of MS:1000455 (ion selection attribute) one or more times
  e.g.: MS:1000041 (charge state)
  e.g.: MS:1000042 (intensity)
  e.g.: MS:1000633 (possible charge state)
  e.g.: MS:1000744 (selected ion m/z)
```

Example cvParams:

```
<cvParam cvRef="MS" accession="MS:1000744" name="selected ion m/z"
value="445.33999999999997" unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
<cvParam cvRef="MS" accession="MS:1000042" name="intensity" value="120053"/>
<cvParam cvRef="MS" accession="MS:1000041" name="charge state" value="2"/>
<cvParam cvRef="MS" accession="MS:1000633" name="possible charge state" value="2"/>
```

5. Conclusions

This document contains the specifications for using the mzDB format to represent mass spectrometry results, metadata and associated context. This specification, in conjunction with the SQL and XML schemata constitute a proposal for a standard from the Proteomics French Infrastructure.

6. Authors and Contributors

David Bouyssié, Marc Dubois, Sara Nasso