

# The mzXML Schema version 3.0



Patrick G.A. Pedrioli<sup>2</sup>, James S. Eddes<sup>1</sup>, Jimmy K. Eng<sup>3</sup>, Nichole L. King<sup>1</sup>, Brian Pratt<sup>4</sup>, David Shteynberg<sup>1</sup>, Joshua M. Tasman<sup>1</sup>, Ning Zhang<sup>1</sup>, Ruedi Aebersold<sup>2</sup>

<sup>1</sup>Institute for Systems Biology, Seattle, WA; <sup>2</sup>Institute for Molecular Systems Biology (ETH), Zurich, Switzerland; <sup>3</sup>Fred Hutchinson Cancer Research Center, Seattle, WA; <sup>4</sup>Insilicos LLC, Seattle, WA

## Overview

The XML Schema for the mzXML format has been updated to encapsulate feature requests collected from the community (through public forums on the Sashimi Sourceforge site, direct discussion with scientists, and published reviews of the format). Support for the most commonly requested features has been added to the XML Schema. Tools to read/write the format have been or are in the process of being updated while maintaining backward compatibility.

We describe here the changes in the format, highlight advantages of the changes, and discuss the future plans.

## Introduction

Different MS vendors use multiple, incompatible representations to store the data acquired by different MS instruments. This impedes the downstream analysis of data generated in an MS-based proteomics experiment and has prompted the creation of alternative, standardized data representations. Recently, two such formats have emerged, mzXML and mzData, and quickly gained community acceptance. In this poster we officially introduce version 3.0 of the mzXML Schema and discuss the importance of the major updates.

## Methods

Based on feedback received through various channels, the XML Schema for the mzXML format has been updated to encapsulate the most commonly requested modifications.

The peaks element now has two new attributes: *contentType* and *compressionType*.

Possible values for *contentType* are:

- m/z int* - a list of m/z, intensity pairs
- m/z* - list of m/z data points
- intensity* - corresponding list of m/z data point intensities
- S/N* - corresponding list of m/z data point signal to noise values
- charge* - corresponding list of m/z data point charge states
- m/z ruler* - intensities of m/z data points at defined-interval m/z deltas (Figure 1).
- TOF* - time of flight data points

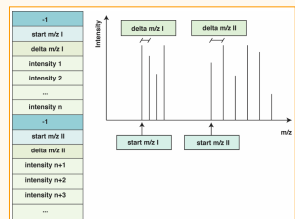


Figure 1. The *m/z ruler contentType* is intended for the representation of regularly spaced m/z data points (e.g. profile data) yet is flexible enough to record data which is not regularly spaced over the entire spectrum. In practice, the beginning of each "cluster" of regularly spaced m/z values is signaled with an element of value -1. The next two elements define the initial m/z value and the delta offset for the cluster. Remaining elements record the intensities of the cluster's m/z coordinates. We are currently investigating the feasibility of transforming irregularly spaced data into regularly spaced data by inserting artificial m/z intensity values.

Possible values for *compressionType* are:

- none - none
- zlib - zlib compression

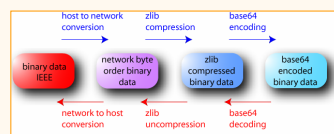


Figure 2. The *peaks* data (the actual spectrum data points represented using one of the above *contentTypes*) may optionally be compressed during conversion using the lossless data-compression library zlib. Zlib is a free library available for a broad range of disparate computer hardware and operating systems.

The *pairOrder* attribute has been retired from the schema.

## Results

The main points that have been addressed in this revision of the mzXML format are as follows:

- The instrument type can now be associated with each scan element; important for hybrid instruments where MS or MS/MS scans in a single run may have been acquired using different detectors.
  - msInstrument* is now unbound
  - msInstrumentID* attribute is added to *msInstrument* element
  - msInstrumentIDKey* (bound to *msInstrumentID*) is added to *msInstrument* element
  - msInstrumentID* added to scan element
  - Keyref onlyValidMsInstrumentIDKey* (bound to *msInstrumentID*) is added to scan element
- Ability to associate extra information to each m/z-intensity pair (e.g. signal to noise ratios, peak charge states, peak areas, etc.).
- Ability to store 64 bit long m/z values to account for higher mass accuracy of instrumentation.
- The introduction of a novel, space-optimized representation of profile data.
- The ability to optionally compress the spectrum data points.

The last two points address one of the biggest community concerns regarding XML-based MS data representations: the increase in file size compared to the native data. Through the application of the new *m/z ruler* data representation and the use of a lossless compression algorithm *zlib*, one can now generate mzXML files that are in some cases smaller than the corresponding native data file.

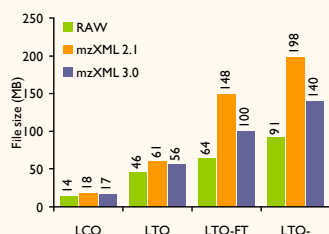


Figure 3. The ReAdW converter was used to convert RAW files from 4 common instruments to both mzXML 2.1 and mzXML 3.0. In all instances, the size of the mzXML 3.0 file was smaller than the corresponding mzXML 2.1 file. For LCQ and LTQ RAW files, using *zlib* compression of the peaks data results in only a modest reduction in the size of the mzXML (approx. 5-10%). This is due to the LCQ and LTQ files consisting of MS and MS/MS spectra acquired in centroid mode. The most dramatic reduction in file size is observed when using *zlib* to compress profile spectrum data before encoding into mzXML (approx. 30%). Both the LTQ-FT and LTQ-Orbitrap RAW data files contain profile MS and centroided MS/MS scans.

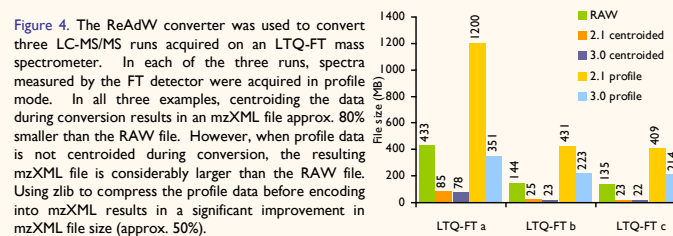


Figure 4. The ReAdW converter was used to convert three LC-MS/MS runs acquired on an LTQ-FT mass spectrometer. In each of the three runs, spectra measured by the FT detector were acquired in profile mode. In all three examples, centroiding the data during conversion results in an mzXML file approx. 80% smaller than the RAW file. However, when profile data is not centroided during conversion, the resulting mzXML file is considerably larger than the RAW file. Using *zlib* to compress the profile data before encoding into mzXML results in a significant improvement in mzXML file size (approx. 50%).

The most dramatic improvement can be seen for LTQ-FT a. This run is composed of all profile data (acquired by both IT and FT detectors), resulting in dramatically larger RAW and subsequent mzXML 2.1 file. Using a combination of the new *m/z ruler* data representation for the profile IT spectra and *zlib* compression, both new features of mzXML 3.0, we can now achieve an xml representation of our profile mass spectrometry data that is approx. 20% smaller in size than the original RAW file.

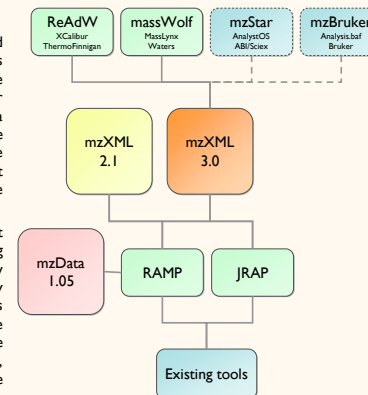
The new schema can be found at:

[http://sashimi.sourceforge.net/schema\\_revision/mzXML\\_3.0/mzXML\\_3.0.xsd](http://sashimi.sourceforge.net/schema_revision/mzXML_3.0/mzXML_3.0.xsd)

## Compatibility

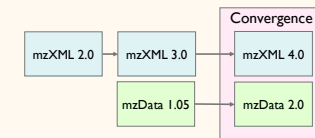
The open source mzXML converters ReAdW and massWolf, for XCalibur and MassLynx native formats respectively, have been updated to write to the mzXML 3.0 format. The mzStar converter for conversion of AnalystOS data files is currently a development priority and is expected to be available soon. The open source parsers RAMP and JRAP have both been updated to read the new format whilst maintaining compatibility with previous releases of the schema. RAMP is also compatible with mzData 1.05.

Where possible, changes to the parsers have been kept transparent to the end users. This facilitates upgrading existing analytical tools to take advantage of the new mzXML compression without having to substantially modify the source code. For example, the Trans Proteomic Pipeline, an analytical software pipeline developed at the Institute for Systems Biology for the interpretation of quantitative MS-base proteomics data, supports mzXML 3.0 files with a simple re-compile against the new RAMP library without any other changes.



## Convergence of mzXML & PSI's mzData

mzXML and PSI's mzData formats were initially designed for slightly different purposes. The intent of mzXML is to encapsulate unprocessed, raw peak lists whereas mzData was initially developed to standardize the format of processed peak lists to be used as the input to search engines. The main reason why the two formats co-exist is timing.



The Seattle Proteome Center has been actively developing, using and distributing both the mzXML format and analytical tools since 2003 when mzData was still a draft proposal and in development. However, recently there has been significant progress in the mzData schema and the necessity for having both mzXML and mzData no longer exist. mzXML 3.0 will be the last major release of mzXML before the planned convergence of mzXML and mzData, integrating the best features from both formats. We fully support such a merger and will tailor our tools to make use of the merged format.

## Conclusions

- The changes in the mzXML 3.0 format are practical, functional improvements requested by the community.
- The flexible peaks element along with *zlib* data compression addresses the most problematic issue with XML based mass spectrometry data formats which is file size.
- The ReAdW and massWolf converters have been updated to write mzXML 3.0.
- The need for two open xml standards for mass spectrometry data has been reduced with the recent improvements to the mzData schema.
- mzXML 3.0 represents the last release of the schema before the planned merger with PSI's mzData format.

## References

- 1) A common open representation of mass spectrometry data and its application to proteomics research. Nat Biotechnol. 2004 Nov;22(11):1459-66.
- 2) PSI-MS: Mass Spectrometry Standards Working Group <http://psidev.sourceforge.net/ms/>
- 3) Sashimi: <http://sashimi.sourceforge.net/>
- 4) Seattle Proteome Center: <http://proteomecenter.org/>
- 5) Insilicos: <http://insilicos.com/>

This work was supported in part by funding from the NHLBI, NIH, under contract No. N01-HV-28179.