INTERNATIONAL STANDARD

ISO 5820

First edition

Microbeam analysis — Hyper-dimensional data file specification (HMSA)

PROOF/ÉPREUVE

Reference number
ISO 5820:2023(E)

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 202, *Microbeam analysis*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Most if not all commercial microanalysis systems acquire and store data in proprietary formats. This hinders the transfer of data between instruments and or between laboratories, such as might be required for multi-technique analyses, round robin studies or collaborations. It is possible that even software from the same manufacturer but for different generations of instruments does not store data in compatible formats. This makes the archiving of data extremely difficult beyond the lifetime of the supported system. The format in this document has been developed by an independent group of experts from the Microscopy Society of America (MSA), the US Micro-Analysis Society (MAS), and the Australian Microbeam Analysis Society (AMAS) to be fully transferrable and archivable. It is independent of instrument manufacturer, computer hardware and operating system.

An existing standard (ISO 22029) allows for platform independent transfer and archiving of simple x-ray spectral data, but the increasing capabilities of microanalysis systems to acquire multi-dimensional signals in parallel has made this standard insufficient to meet all current needs. This standard has been written to meet these expanded requirements.

Microbeam analysis — Hyper-dimensional data file specification (HMSA)

1 Scope

The MSA/MAS/AMAS hyper-dimensional data file specification (HMSA, for short) is a platform-independent data format to permit the exchange of hyper-dimensional microscopy and microanalytical data between different software applications. The applications include, but are not limited to:

— Hyper-spectral maps, such as electron energy loss spectroscopy (EELS), energy dispersive x-ray spectrometry (XEDS), or cathodoluminescence spectroscopy (CL).

— 'Hyper-image' maps, such as pattern maps using electron backscatter diffraction (EBSD) or convergent beam electron diffraction (CBED).

— 3-dimensional maps, such as confocal microscopy, or focused ion beam (FIB) serial section maps.

— 4-dimensional maps, such as double-tilt electron tomography.

— Time-resolved microscopy and spectroscopy.

In addition to storing hyper-dimensional data, the HMSA file format is applicable for storing conventional microscopy and microanalysis data, such as spectra, line profiles, images, and quantitative analyses, as well as experimental conditions and other metadata.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

No terms and definitions are listed in this document.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

4 Overview

4.1 Design Considerations

The following requirements were considered in the design of this file format:

a) Modern experimental apparatus produce data with high dimensionality, such as spectral maps and 3D serial section maps. Therefore, this file format shall store data of high dimensionality.

b) High dimensionality data is necessarily very large, and consequently difficult and time consuming to store or transfer over networks. The file format shall therefore be as compact as is reasonably practical.

c) Many microanalytical techniques produce structurally similar hyperdimensional data. To simplify implementation of common tools, this file format shall use a common format to store data produced by different analytical techniques.

d) The data format shall preserve the scientific accuracy and meaning of the data. Therefore, the file format shall store data without loss of precision and include sufficient experimental parameters to permit the correct interpretation of the data.

e) To achieve the intended mission of being a widely supported exchange format, the file format shall achieve acceptance from instrument and software vendors, and from the microanalysis community. Consequently, the file format shall be useful, easy to understand, and easy to implement.

f) Furthermore, as the file format is intended for exchange, it shall be readable (and implementable) in any commonly available programming languages and environments. The format shall therefore be platform independent, and not require any proprietary or special software or hardware.

4.2 Binary and XML file pair

4.2.1 General

To satisfy the above requirements, the MSA/MAS/AMAS hyper-dimensional data file format uses a pair of files; a simple binary file to efficiently store the experimental data, and a text-based XML file to store the experimental conditions. The advantages of this dual format are:

— The structure of the binary file format is simple, unambiguous, and precisely defined in a human readable format within the XML file.

— High dimensionality experimental data is binary encoded for space efficiency, whilst also being easy to read and write programmatically.

— Experimental conditions are stored in a human-readable and self-descriptive format. Conditions are stored in a hierarchical structure to logically classify related settings.

— No special libraries are required to read or write HMSA/XML files. For convenience, XML libraries may be used, and are freely available on most programming environments.

4.2.2 HMSA general structure

The HMSA file is a binary file format consisting of an 8 byte (64 bit) unique identifier (5.4.4: The `UID` attribute), followed by one or more dataset objects. The location, size and layout of the binary dataset objects are described in the dataset definitions within the XML file (8: The `<Dataset>` element), and are not described within the binary HMSA file. The values contained within the HMSA file datasets cannot therefore be read or interpreted without the corresponding dataset definition within the XML file.

Blocks of arbitrary and proprietary binary or text data also may be placed in the binary HMSA file. These arbitrary data blocks may be used to store proprietary application-specific data, or ancillary experimental data that cannot be formatted as a HMSA data set object (8: The `<Dataset>` element). The formatting of these arbitrary data blocks in the HMSA file are not defined by this specification, but the location and size of the arbitrary block should be declared in the <Header> section of the XML file using one or more `<ArbitraryData>` elements (6.6: The `<ArbitraryData>` element).

The byte ordering of the HMSA binary file shall be little-endian (Intel/Windows style).

4.2.3 XML general structure

The XML file consists of human-readable hierarchical text, using a subset of the XML version 1.0 format.

The structures within the XML file are strictly defined and self-descriptive, so that the XML file can be read and interpreted correctly without a finely detailed study of the specification. This strict definition does, however, require software that writes the XML files to diligently adhere to the specification.

The structure of the XML file is described in detail in 5: XML file specification.

4.2.4 HMSA-XML association

Because the XML file is required to interpret the HMSA file, the HMSA/XML files shall be associated in such a way that software that loads a HMSA file can readily and unambiguously locate the associated XML file. The principal method by which the HMSA and XML files are associated together is by file name. The HMSA/XML file pairs shall share the same file name except for their file extensions, such as "Spodumene.HMSA" and "Spodumene.XML". The HMSA/XML file pairs should be transferred together, and stored in the same directory.

Users may inadvertently rename or move one member of the file pair, which would prevent software from finding the correct experimental conditions or binary data. To reduce this risk, the XML and HMSA files each contain an identifier that is, for all intents and purposes, unique to each individual pair of files. By comparing the unique identifiers (UIDs) given in the XML and HMSA file, software can be assured that binary data matches the description in the XML file, and vice versa. Furthermore, by searching the file system for XML or HMSA files containing the UID, software may automatically find renamed or relocated files. This pseudo-unique identifier is a 64-bit code, providing a possible 2^{64} (~1,84 x 10^{19}) unique values. The UID is described further in 5.4.4: The UID attribute.

4.3 Hyper-dimensional data

The HMSA format is designed to store data that may be structured as a regular N-dimensional array. This design readily supports common microanalytical dataset types such as spectra, grayscale and color images, hyperspectral maps, ‘hyper-image’ maps (an image per pixel), 3-dimensional analyses by confocal microscopy or serial sectioning, as well as irregular sequences of the above. Table 1 summarizes the dimensionality of common dataset types:

Table 1 — Dimensionality of common data types

Dimensions	Example datasets
1	A single spectrum.
	A sequence of single-valued measurements, such as an x-ray intensity line profile, or time sequence of vacuum pressure.
2	A 2D grayscale image.

Dimensions	Example datasets
	A sequence of spectra.
3	A 2D color image or hyper-spectral map.
	A sequence of 2D grayscale images.
	A 3D grayscale image, such as from a FIB-SEM serial section.
4	A 2D ‘hyper-image’ map of 2D measurements, such as an EBSD pattern map.
	A 3D color image or spectral map, such as from a FIB-SEM serial section.
5	A 3D map of 2D measurements, such as an EBSD pattern map from a FIB-SEM serial section.

This specification does not restrict the number or size of dimensions in a HMSA dataset. The number, identity and sizes of the dataset dimensions are defined by the `<Dataset>` element in the XML file (see 8: The `<Dataset>` element). Examples of common dataset dimensions are defined in Annex E.

4.4 Unicode and internationalization

The HMSA XML file format requires the use of the UTF-8 Unicode character encoding, permitting native-language representations of the non-English names for authors, organizations, specimens, locations, etc. However, for maximum interoperability, the names of XML elements and attributes shall be given in US English using the ASCII character set. Furthermore, the values of elements shall be given in US English where possible, with non- English text provided as an alternative translation to the English text using an `alt-lang- [xx][-YY]` attribute (5.5.5: Alternative language attributes).

In addition to supporting non-English scripts, the use of Unicode for the HMSA XML file allows the use of scientifically meaningful non-Latin characters such as α, μ, and Å. However, these characters may not be typeable on many standard keyboards, and so they should only be used when no unambiguous Latin character equivalent is available. Please refer to Annex C for a list of permitted Unicode characters in units and unit prefixes.

In cases where the Unicode character set includes multiple code points for visually indistinguishable glyphs, HMSA XML files shall consistently use one code point in preference to any alternatives (see Annex C).

4.5 Minimalism

The purpose of the HMSA file format is to enable the convenient exchange of scientific data between different software packages. To succeed in this purpose, the HMSA file format shall be unambiguous in its specification, and easy to implement. To this end, the HMSA XML file format has been designed with a minimalist core of mandatory features that are necessary only to properly determine the layout of the hyper-dimensional dataset(s) in the HMSA binary data file. The structure of the dataset definition in the XML file is strictly defined to exclude all experimental parameters, thereby making it universal for all dataset types (8: The `<Dataset>` element).

All useful experimental conditions (such as spectrometer gain and offset) and other metadata (such as author or date) are recommended, but optional. Nevertheless, to ensure compatibility, the structure and format of these optional conditions and metadata elements are defined in this document (6: The `<Header>` list element, and 7: The `<Conditions>` list element).

The absolute minimum effort possible to produce a conformant HMSA XML file is demonstrated in the ‘baseline’ HMSA XML example files in Annex D. These files contain no optional elements such as conditions or metadata. Important conditions such as microscope settings and spectrometer calibration are not included, meaning that – for example - spectra can only be interpreted as raw channels, and the user is responsible for determining energy calibration and accelerating voltage. For reference, the same files are also provided in typical form in Annex D and include all common experimental conditions and metadata.

4.6 Extensibility

In addition to being simple and easy to implement (4.5: Minimalism), a key feature of the HMSA file format is that it is *extensible*. Although this specification enumerates a number of common condition objects (Annex A), the specification permits the unlimited use of additional, un-specified experimental conditions to be stored in the HMSA XML file (7: The `<Conditions>` list element). Critically, the well-formed, hierarchical and self-descriptive nature of XML allows these additional conditions to be included without imposing an additional burden on applications to support any or all of these conditions. In effect, applications are not required to read, write or interpret any conditions, but may elect to provide additional scientific meaning or interpretation to the data by including additional conditions to any degree of detail.

For example, consider the case of a typical XEDS spectral map collected in an SEM. A typical HMSA file would include conditions for spectrometer calibration and beam accelerating voltage. This information is sufficient for a basic interpretation of the map data, such as peak identification in spectra and generating elemental region of interest (ROI) images. A more detailed file may also include a Faraday cup beam current measurement, and even intensity measurements from standard reference materials so as to allow quantification of elemental compositions. An extreme example may also include all electron gun conditions, lens currents, and the like, so as to allow the comparison or monitoring of microscope and detector performance between instruments or over time. However, not all SEMs have Faraday cups, and nor do all experiments require quantification or performance monitoring, and thus these elements are purely optional.

In addition to supporting unlimited experimental conditions, the HMSA specification also supports the inclusion of multiple binary datasets in a single HMSA/XML file pair. Typical usage cases for multiple dataset files are:

— The storage of multi-detector maps, such as simultaneous XEDS+EELS in a TEM, XEDS+EBSD in a SEM, or WDS+XEDS+CL in an EPMA.

— The storage of auxiliary map data that is helpful for the interpretation of the primary dataset, such as a beam current/flux map, a specimen thickness map, or a detector saturation/dead-time map.

— The storage of reference spectra with spectral maps.

Support for multiple datasets is provided in such a way as to impose no additional burden on applications that expect only single-dataset files. Applications are not required to support multiple datasets.

4.7 What HMSA does not do

To reduce the complexity of implementing HMSA support, certain features or usage cases have been excluded:

— HMSA is not intended to be a general long-term archival format for all relevant or extraneous data from a set of experiments. HMSA is intended to store the data, and optionally the relevant conditions, from a single experiment, on a single apparatus, from a single specimen, collected over a single contiguous time interval.

— No compression is to be used on either the XML or HMSA file, as compression algorithms may be proprietary or unavailable in some environments. Users may elect to compress the XML/HMSA file pair for transmission or storage at their own discretion, but HMSA-compatible software should not write compressed HMSA/XML files.

— The format is not primarily intended to be an efficient ‘working’ format for applications, and so it has not been specifically optimized for minimum memory footprint, maximum read/write speed, efficient random seeking, etc.

— HMSA is not intended to support all possible experimental techniques. Whilst a reasonable effort has been made to support a broad range of experimental dataset types, the HMSA format may not be particularly amenable to some types of experimental data (sparse spectra, for example.)

5 XML File Specification

5.1 XML general structure

The XML file consists of human-readable hierarchical text, using a subset of the XML version 1.0 format (5.2: XML specification). The structures within the XML file are strictly defined and self-descriptive, so that the XML file can be read and interpreted correctly without a finely detailed study of the specification. This strict definition does, however, require software that writes the XML files to diligently adhere to the specification.

The XML files have the following general structure:

— An XML declaration

— An MSAHyperDimensionalDataFile root element, containing:

  — A Header element, containing:

    — Descriptive metadata such as the document title, collection date, author, etc.

  — A Conditions element, containing:

    — One or more items of experimental conditions that describe how the dataset is to be interpreted or displayed, such as microscope and spectrometer settings.

  — One or more Dataset elements, which formally define the address, ordering, and size of a binary data block within the HMSA file.

In XML, this looks like:

```
<?xml version="1.01" [...] ?>
<MSAHyperDimensionalDataFile [...] >
   <Header>
      [...]
   </Header>
   <Conditions>
      [...]
   </Conditions>
   <Dataset>
      [...]
   </Dataset>
</MSAHyperDimensionalDataFile>
```


The XML declaration, `<MSAHyperDimensionalDataFile>` document root element, `<Header>`, `<Conditions>` and `<Dataset>` elements are described in the following clauses:

— 5.3: XML declaration

— 5.4: Document root element

— 6: The `<Header>` list element

— 7: The `<Conditions>` list element

— 8: The `<Dataset>` element

5.2 XML specification

5.2.1 General

The HMSA XML file specification follows the W3C Extensible Markup Language (XML) 1.01 Recommendation (Fifth Edition), except where noted below (See https://www.w3.org/TR/xml/).

5.2.2 XML features not supported

To simplify the tasks of reading, writing and interpreting HMSA XML files, this specification excludes certain XML features that may complicate implementation for no benefit in this application. HMSA XML files shall not contain the following XML features declared in the XML 1.01 recommendation (clause numbers in parentheses):

— Comments (2.5)

— Processing instructions (2.6)

— CDATA sections (2.7)

— Document type definitions (2.8)

— Element type definitions (3.2)

— Conditional sections (3.4)

— Entity declarations (4.2)

— Notation declarations (4.7)

The HMSA XML format also explicitly does not support the following associated W3C XML specifications:

— XML Schema

— Namespaces in XML

5.2.3 XML conformance and validation

The W3C XML specification defines two levels of compliance; conformant, and valid. Conformant XML files satisfy all requirements of the XML specification, such as wellformedness. Valid XML files are conformant XML files, and also contain document type definitions (DTDs) that specify the structure and range of all elements in the XML file. Valid XML files can therefore be validated for completeness and correctness by a generic validating XML parser, without reference to an external specification of the file format. In effect, valid XML files are self-specifying.

In the interests of minimizing the size and complexity of HMSA XML files, XML document and element type definitions were excluded from the HMSA XML specification (5.2.2: XML features not supported). Consequently, HMSA XML documents are conformant XML files, but not *valid* XML files.

5.2.4 Character encodings

HMSA XML files shall only be encoded in the Unicode UTF-8 character encoding. To provide backwards compatibility with the ASCII character set, HMSA XML files should use the basic Latin characters and symbols in the range of U+0032 to U+007E in preference to visually similar Unicode characters when it is customary to do so, and whenever such substitution does not change the meaning or introduce ambiguity. For example, ‘Ka’ should be used to represent the Kα x-ray in the Siegbahn notation, and ‘um’ should be used to represent μm. Further character substitutions are specified in Annex C.

5.2.5 Byte order markers

Byte order markers (BOM) are not required for UTF-8 encoded text files but may be automatically inserted at the start of the file stream by certain text editors. Thus, HMSA XML files may, but should not, contain the UTF-8 BOM (0xEFBBBF), and shall not contain byte order markers for other character encodings (e.g., 0xFFFE for UTF-16LE on Windows, or 0xFEFF for UTF-16BE on Unix/Linux/Mac). HMSA XML parsers shall process and ignore UTF-8 BOM, if present.

5.2.6 Case sensitivity

As defined in the XML standard, the structure of an XML file is case sensitive. The names of all elements and attributes shall be written with the case specified in this document. The values of attributes and elements are also assumed to be case sensitive, unless specified otherwise in this document.

To avoid confusion, identifier attributes such as `Name` and `ID` shall have unique values in case-insensitive comparison.

5.3 XML declaration

5.3.1 General

The HMSA XML file shall begin with an XML declaration of the form:

```
<?xml version="1.01" encoding="UTF-8" standalone="yes" ?>
```

The attributes of the XML declaration are described below.

5.3.2 XML `version` attribute

The `version` attribute of the XML declaration shall have the value "1.0". XML version 1.1 or subsequent versions are not supported by this version of the HMSA/XML specification.

5.3.3 XML `character encoding` attribute

The `encoding` attribute of the XML declaration shall have the value "UTF-8". No other character encoding is permitted for HMSA XML files.

