Microbeam Analysis — Hyper-Dimensional Data File Specification (HMSA)

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 *has 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/TC202, *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} ($\sim$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.
	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.

5.3.4 XML `standalone` attribute

The `standalone` attribute of the XML declaration shall have the value "yes". HMSA XML files do not support external document type definitions.

5.4 Document `root` element

5.4.1 General

The `root` element of the HMSA XML file shall be named and formatted with required attributes as follows:

```
<MSAHyperDimensionalDataFile Version="1.02" xml:lang="en-US"
UID="193581B9DD220ABB">
```

The attributes of the root element are described below.

5.4.2   The `Version` attribute

The HMSA version shall be declared as "1.02" in the `Version` attribute.

5.4.3   The `XML:lang` attribute

The default language of the document shall be US English, which shall be declared using an `xml:lang` attribute of the document root element with a value of "en-US".

5.4.4   The `UID` attribute

A pseudo-unique identifier shall be provided in the `UID` attribute in the form of 16 hexadecimal characters (0-9, A-F), representing a 64-bit binary value. This 64-bit unique identifier, which is stored in both the XML and binary HMSA files, serves two purposes:

a)   To verify that a HMSA file and XML file match. This is required because HMSA files cannot be decoded without the XML description, and using the wrong XML description can result in corrupted results or undefined software behavior.

b)   To allow software to search for a missing component of the file pair such as a renamed or moved file.

To ensure maximum efficacy of the UID mechanism, software that writes or modifies HMSA files shall create new UIDs when:

— Creating a new HMSA/XML dataset.

— Modifying any contents of either the HMSA or XML files.

— Extracting a subset of a HMSA file.

The UID may be retained unchanged when:

— Creating an exact copy of a HMSA/XML pair.

— Renaming a HMSA/XML pair.

To further guarantee the integrity of HMSA UIDs, the following is required of UID generation algorithms:

— The output domain of the algorithm should span every possible 64-bit value.

— The output shall not be a predictable or reproducible sequence of UIDs.

The recommended method of generating a UID is to use a one-way cryptographic hash function, such as the NIST-published SHA-1 algorithm, with a diverse set of inputs to ensure sufficient hash entropy.

5.5 XML Parameter element formats

5.5.1 General

The format of elements and attributes used to store arbitrary parameters in the HMSA XML file are defined below.

5.5.2 Numerical values

This specification defines a uniform scheme for representing numerical values as text within HMSA XML files, so as to simplify software implementations and reducing the likelihood of misinterpretation of experimental parameters. The values of numerical parameters shall be written in decimal, without any

digit grouping markers such as commas or whitespaces. The use of leading or trailing whitespaces is permitted. Text encoding of floating-point values shall follow the IEEE 754-1985 standard for binary <-> decimal conversion. Furthermore:

— Radix/decimal point marker shall be the full stop character (U+002E).

— Exponents shall be denoted by either 'E' or 'e'.

The HMSA XML format does not explicitly define data types or binary precision for numerical parameters, other than for arrays (see 5.5.3: Arrays of values). To minimize the risk of data truncation or misinterpretation, software that reads HMSA XML files should interpret all integer values as signed 64-bit integers and should interpret all floating point values as double-precision 64 bit floats.

The expected data types and precision of individual element values may be defined within this specification, and in such instances the declared data type shall be used by software when reading or interpreting interpret those elements. One example is the `<DataLength>` element in the `<Dataset>` element, for which the data type shall be a 64-bit integer (See 8.2: The `<DataLength>` and `<DataOffset>` elements).

5.5.3 Arrays of values

Arrays of values shall be written as comma separated values. For example:

```
<Fibonacci ArrayType="int" Count="6">1, 1, 2, 3, 5, 8</Fibonacci>
```

The number of values in the array shall be specified using a `Count` attribute. The use of the `Count` attribute name is reserved for the purpose of specifying array sizes and shall not be used for any other purpose in a HMSA XML file.

If all the array values are numbers of the same data type, the data type should be specified using an `ArrayType` attribute, with one of the values as shown in Table 2:

Table 2 — Array type attribute values

ArrayType	Description
"int"	Signed 32-bit integer
"int64"	Signed 64-bit integer
"float"	32-bit IEEE 754 single-precision floating point number
"float64"	64-bit IEEE 754 double-precision floating point number

The values of the `ArrayType` attributes *shall* be written in lower case. If the `ArrayType` attribute is not specified, the type of the array values should be assumed to be text.

5.5.4 Physical units

For numerical values with physical units, the units should be defined using a `Unit` attribute. Units shall be provided in SI units, SI derived units (e.g., "Pa", "Å"), or one of the customary technique-specific units defined in Annex B (e.g., "counts", "wt%"). Units shall be declared in abbreviated form, with optional single-character SI prefix codes (e.g., "kV", for kilovolt). The list of permitted prefixes is also included in Annex B.

Condition objects defined in Annex A specify the physical units that shall be used for parameters within those objects. The precise formats of the unit text shall be consistent with the definitions in the appendices.

To preserve scientific accuracy, it is critical that HMSA files use a consistent scheme for specifying compound units that is readable and writeable by both humans and computers. Aesthetically pleasing representations such as kg.m.s^{-2} are difficult to type and are prone to display or interpretation errors when moving between software packages. To avoid confusion, HMSA files shall therefore use only the full stop '.' (U+002E), solidus '/' (U+0047) and numerals 0-9 (U+0030 - U+0039) to represent compound units such as "kg.m/s2". The use of the hyphen-minus sign '-' (U+002D) to indicate negative exponents is permitted only for inverse singular units, such as inverse centimetres (cm-1), but not compound units (e.g. "m/s2", not "m.s-2"). Other methods of superscript markup such as the circumflex accent ^ (U+005E) shall not be used. The use of brackets in unit definitions is not permitted.

The Unicode character set defines a number of specific code points for scientific symbols, which are visually identical to non-scientific code points. For example, the Unicode Latin capital A with ring above 'Å' (U+00C5) is visually indistinguishable from the Unicode Ångström symbol 'Å' (U+212B). The casual use of one or the other symbol for the same quantity poses a risk to software compatibility. Consequently, to avoid confusion and maximize compatibility, the lowest code point shall be used in cases where a unit symbol can be written in two or more visually indistinguishable characters. Required character substitutions are provided in Annex C.

When defining concentrations, it is mandatory to specify whether the measurement is molar or atomic (mol%), volumetric (vol%) or mass or weight (wt%). Similarly, when using parts per million or parts per billion notations for concentration, the nature of the measurement shall be specified (e.g., mol_ppm, vol_ppm, wt_ppm.)

5.5.5 Alternative language attributes

In addition to the US English text, values in other languages may be specified using `altlang- xx[-YY...]` attributes, where 'xx' is the language code and 'YY…' the locale, as in the form of IETF language tags (i.e. 'en-US'). For example, the author may be specified as:

```
<Author alt-lang-ru="Фёдор Михайлович Достое́вский">
   Fyodor Mikhailovich Dostoyevsky
</Author>
```

This method should be used only to provide proper nouns in appropriate native languages, such as the names of authors, organizations, or places. The translations should be the culturally appropriate names that would be expected by native speakers of each language and need not be literal translations or transliterations.

The use of the prefix `alt-lang-` in attribute names is reserved for such proper-noun attributes and shall not be used in other attribute names.

5.5.6 Special characters

In accordance with the XML specification, the following characters given in Table 3 shall not be used in the names or values of elements or attributes.

Table 3 — Non-permitted element or attribute characters

Character	Code point	XML entity
<	U+003C	<
>	U+003E	>
"	U+0022	"
'	U+0027	'

&	U+0026	&

When writing XML files, occurrences of these characters in value strings shall be converted to their respective XML entities, as given above. Upon loading of XML files, following structural parsing, occurrences of these XML entities in strings shall be converted back to their corresponding character values before being presented to users or other software.

5.5.7 Ordering of elements

The order in which elements are listed within the XML file is not specified in general, meaning XML elements may be sorted in any order within their parent XML element unless otherwise specified. A notable example of where the ordering of elements is specified is for the contents of the `<MSAHyperDimensionalDataFile>` document root element (see 5.4: Document `root` element), where the child elements shall be in the following order: `<Header>`, `<Conditions>`, then `<Dataset>`. A further example is in the ordering of the elements within the `<Dimensions>` list of the `<Dataset>` object, which serves to define the ordering of data in the binary HMSA file (see 8.4.2: Ordering of dimensions). Condition templates may also define a required ordering of elements.

6 The `<Header>` list element

6.1 General

The `<Header>` list element contains metadata that principally identifies the title of the document, the author/ownership of the data, and the date/time of collection. Header information shall not contain parameters that are required for the interpretation of the experimental data.

6.2 Header items are optional

In keeping with the principle of minimalism (4.5: Minimalism), all items in the `<Header>` list element are optional. Some elements, such as the `<Checksum>` (6.3: The `<Checksum>` element), should be included, but are not mandatory. Software that reads HMSA XML files should not require the presence of any items in the `<Header>` list to open, display or process files.

If no items are defined within the `<Header>` list, the empty header list shall be specified as either an empty element (`<Header />`), or as a conventional matched pair of elements with no contents (`<Header></Header>`). XML parsers for HMSA XML files shall support both styles of empty element declaration.

6.3 The `<Checksum>` element

The `<Header>` list should include a `<Checksum>` element to allow software to verify that the binary HMSA file exactly matches that specified in the XML file. The `<Checksum>` element, if provided, shall take the following form:

```
<Checksum Algorithm="SHA-1">
   53AAD59C05D59A40AD746D6928EA6D2D526865FD
</Checksum>
```

The contents of the `<Checksum>` element shall be the hexadecimal-encoded (A-F, 0-9) checksum digest of the entire binary HMSA file. The algorithm used to generate the checksum shall be declared using the `Algorithm` attribute. The checksum algorithm should be one of the following algorithms:

— SUM32 (sum of all bytes in the binary HMSA file, truncated to a 32-bit / 8-hexadecimal character value)

— SHA-1 (recommended)

The 'SUM32' algorithm is provided for basic protection against single-bit and some multiple-bit errors but does not protect against multiple-bit errors with zero sum change. For this reason, the 'SHA-1' algorithm is recommended, as it provides strong detection of any form of modification and is furthermore a widely supported standard with libraries and implementations available in most programming languages and platforms.

6.4 The `<Title>`, `<Author>` and `<Owner>` elements

The title, author, and legal owner of the document should be specified within the `<Header>` list like so:

```
<Title>Very Important Results</Title>
<Author>J. Smith</Author>
<Owner>ARCN Inc.</Owner>
```

These elements may be provided in languages other than US English using an alternative language attribute `alt-lang-xx[-YY]` (5.5.5: Alternative language attributes). For example, the name of the author Leo Tolstoy may be provided in his native Russian Cyrillic script as:

```
<Author alt-lang-ru="Лев Никола́евич Толсто́й">Leo Nikolaevich
Tolstoy</Author>
```

6.5 The `<Date>`, `<Time>` and `<Timezone>` elements

The date and time of the creation of the HMSA file should be stored in `<Date>`, `<Time>` and `<Timezone>` elements, of the following format:

```
<Date>1985-10-26</Date>
<Time>20:04:00</Time>
<Timezone>UTC-8 US Pacific Standard Time</Timezone>
```

The `<Date>` and `<Time>` values shall be written in the ISO 8601 date/time format, with the date as YYYY-MM-DD, and the time as HH:MM:SS in 24 hour format. The `<Timezone>` value shall be given in terms of Universal Coordinated Time as "UTC", "UTC±HH", or "UTC±HH:MM", with the timezone offsets given in hours (HH), hours and minutes (HH:MM), or omitted if the offset is zero. Examples include "UTC", "UTC+10", and "UTC- 03:30". Following the timezone UTC offset, the two-character ISO 3166-1 alpha-2 country code and full formal timezone name may be given, such as in "UTC-4 CA Atlantic Standard Time".

Dates shall be encoded according to the Gregorian calendar in the common era (CE / AD).

6.6 The `<ArbitraryData>` element

Within the binary HMSA file, applications may elect to store blocks of arbitrary and proprietary binary or text data. The location and size of these arbitrary data blocks should be declared in the `<Header>` list element using one or more `<ArbitraryData>` elements. The `<ArbitraryData>` element allows compatible applications to find the arbitrary data blocks in the HMSA binary file, and also allows third party applications to preserve custom or proprietary data blocks when modifying or saving HMSA files. For example, should one wish to incorporate a fully formatted TIFF image into the HMSA data file this can be accomplished by embedding it within an `<ArbitraryData>` element without loss, while at the same time including formatted hyperspectral data. However, blocks of arbitrary data may be inserted in HMSA binary files without corresponding `<ArbitraryData>` declarations.

The `<ArbitraryData>` XML element shall contain `<DataOffset>` and `<DataLength>` elements, which respectively define the absolute position and size (in bytes) of the data block in the HMSA binary

file, as 64-bit integers. These elements are defined analogously to those used in the dataset definition (8.2: The `<DataLength>` and `<DataOffset>` elements). A `Name` attribute may be provided to identify the block, and a `<Format>` element may be used to describe the formatting of the arbitrary data block. An example `<ArbitraryData>` definition is provided below:

```
<ArbitraryData Name="Example Corp. data block #1">
    <DataOffset>176126333</DataOffset>
    <DataLength>3321</DataLength>
    <Format>MAC table</Format>
</ArbitraryData>
```

Additional attributes or XML elements may be specified within the `<ArbitraryData>` element but are not defined by this specification.

The first dataset object in a HMSA binary file is present at an offset of 8 bytes (i.e., immediately following the UID, see 4.2.2: HMSA general structure), but subsequent datasets may be present at any offset that does not overlap with another dataset. Hence, blocks of arbitrary data may be placed in the binary HMSA file after any dataset. This specification places no restrictions on the number or size of arbitrary data blocks present in a HMSA file.

Blocks of arbitrary data in the binary HMSA file should commence with a unique identifier or 'magic number', so that applications that read arbitrary data from HMSA files may verify the expected formatting of the arbitrary data block. Blocks of arbitrary data in the binary HMSA file should not contain absolute position references to other locations within the file, as third-party applications may modify the ordering of datasets and arbitrary data blocks. Relative position references to locations within the same arbitrary data block may be used.

Applications that save or modify HMSA/XML files may — but are not required to — preserve blocks of arbitrary data stored in the HMSA binary files. If an application does not preserve arbitrary data blocks when saving HMSA binary files, it should remove any `<ArbitraryData>` elements from the `<Header>` list of the XML files. As arbitrary data blocks may be removed from files, applications that read such arbitrary data should verify that the expected position of the arbitrary data block lies outside the position range of any declared dataset, and should validate any unique identifier or 'magic number' that is given at the start of an arbitrary data block in the HMSA binary file.

6.7 Other optional header elements

The header may optionally include any number of other metadata elements, such as:

— `<Client>`

— `<AuthorSoftware>`

— `<Location>`

— `<Comment>`

The formats and conventions of these optional elements are not defined, and these values shall not be required for the proper display or interpretation of the experimental data or conditions. Any scientifically meaningful metadata shall be stored within an appropriate element within the `<Conditions>` list (7: The `<Conditions>` list element).

7   The `<Conditions>` list element

7.1 General

The `<Conditions>` element is a list of experimental parameters that may assist in the scientific interpretation of the experimental data, such as: calibrations, acquisition details, instrument operating parameters, etc. These experimental parameters or conditions are entirely optional and are not required to determine the position and structure of datasets in the binary HMSA file, which is instead described by the `<Dataset>` element (See 8: The `<Dataset>` element).

Conditions are technique-specific, and so there will be a diverse range of possible condition elements. Templates for some common conditions are discussed in 7.3: Condition templates and classes. Examples of the condition elements that may be found in HMSA XML files is provided in 7.5: Typical conditions and detailed descriptions of each example condition are given in Annex A, while fully worked examples of HMSA XML files are provided in Annex D.

In order to keep the list of defined and proposed `<Conditions>` parameters manageable as well as instrument/manufacturer neutral an on-line database of reviewed and endorsed `<Conditions>` will be maintained on-line at:

http://www.microscopy.org/resources/scientific_data/

All condition templates shall have the following base structure:

```
<TemplateName Class="ClassName" ID="UniqueStringOfText">
    [...]
</TemplateName>
```

The `Class` and `ID` attributes are optional and may not be present for all elements in the `<Conditions>` list. The templates and class names are further described in 7.3: Condition templates and classes, and the `ID` attribute is described in 7.4: Condition identifiers. Note that the `<Conditions>` list may contain any number of entries with the same template name and/or class name. However, the `ID` attribute, if present, shall be a unique value within the XML file.

7.2 Conditions are optional

Because of the limitless number of potentially useful condition objects, it is not practical to assume that all software shall read or understand all condition types. Consequently, the HMSA/XML file format has been designed such that all conditions are optional. Software that reads HMSA files should be able to read and display datasets without having to parse and understand any or all of the associated conditions (albeit without calibration or further interpretation.) Conditions therefore shall not contain any information that is required to load the dataset from the file, as the position and layout of the dataset object in the HMSA file is completely defined in the relevant dataset object (8: The `<Dataset>` element).

This requirement is intended to ensure a universal base level of support for common dataset types, so that, for example, a program that can read and display any 2D spectral map dataset should work with all 2D spectral map datasets, from any technique (EELS, XEDS, CL, etc.)

7.3 Condition templates and classes

The instrumentation, experimental configurations, and techniques used to generate microscopy data sets are extremely varied. It is not possible to define a generic set of condition templates that can be used to describe any possible microscopy experiment in full detail. HMSA therefore takes a hierarchical approach to the description of experimental conditions. The top-level condition elements are defined to have generic names that represent concepts relevant to virtually any microscopy experiment. Using a software

analogy, these top-level condition templates represent the base classes for the data objects needed to describe any given experiment at an abstract level. However, particular instruments and particular techniques have settings and parameters that are only relevant within their particular contexts, so HMSA defines a mechanism for defining experiment-specific variants of each of the top-level condition elements. Taking a cue from class hierarchies in object-oriented software, context-specific condition elements are derived from the generic templates by adding a Class attribute that specifies the precise nature of the information captured within a particular condition element. The Class attribute tells both human and software readers of the HMSA/XML file what subsidiary condition elements and parameters are likely to be encountered within the associated condition object.

The top-level condition templates defined by HMSA emerge from the following generic picture of microscopy experiments. Any experiment is carried out within or through use of an **instrument**. Within the instrument, some type of **probe** is prepared and controlled to impinge on a **specimen** that is maintained within a particular **environment**. Signals emitted or transmitted by the specimen are captured and manipulated by means of one or more **measurement modes** of the instrument. Each measurement mode terminates in one or more **detectors** that generate digitized signals recorded as a numeric dataset. A complete experiment may involve repeated **acquisition** of detector readouts as one or more instrument parameters are varied. Physical interpretation of the resulting multi-dimensional dataset requires a **calibration** for each of the axes of the dataset as well as for its intensity values.

In light of the above picture, HMSA defines the following top-level condition templates:

— `<Instrument>`

— `<Probe>`

— `<Specimen>`

— `<SpecimenEnvironment>`

— `<MeasurementMode>`

— `<Detector>`

— `<Acquisition>`

— `<Calibration>`

Condition templates are typically refined for a specific context by specifying a Class attribute. For example, the <Probe Class="SEM"> condition is a particular class of the <Probe> condition template, one that contains parameters relevant to SEM-based experiments and configurations. This class may be further refined for a particular type or model of SEM column by appending a solidus '/' (U+002F) and a subclass name to the Class attribute, such as <Probe Class="SEM/FEG">.

Each subclass inherits the required and optional parameters of the parent template/class, as well as any restrictions on parameter values. Required parameters shall not be removed by subclasses, nor shall any restrictions on parameter ranges be violated. Consequently, an object of type <Probe Class="SEM/FEG"> is both a valid <Probe Class="SEM"> object, and a valid <Probe> object. This class hierarchy system is intended to ensure that software that can interpret a condition object such as <Probe Class="SEM"> can validly interpret all derived subclasses, even if no additional parameters are read or understood. Furthermore, the class hierarchy allows for conditions to be extended in the future as required, by adding new sub-classes with well-defined sets of additional parameters.

To ensure class names are unambiguous and universally typeable, class names shall contain only Latin characters and digits from the ASCII subset of the Unicode character set (A-Z, a-z, 0-9), and the hyphen-minus '-' (U+002D). The solidus '/' (U+002F) shall only be used to delimit class/subclass names.

A partial list of endorsed condition templates and classes, which is not exhaustive, is provided in Annex A. It is expected that users of different techniques, or different vendors, may extend these templates/classes to suit their particular needs (See 3: Terms and definitions). Submission of definitions for review and endorsement by the HMSA committee is recommended to keep parameters well defined and whenever possible instrument neutral.

7.4 Condition identifiers

XML elements in the `<Conditions>` list may be declared with a unique identifier string using the `ID` attribute. The purpose of the `ID` attribute is to allow `<Dataset>` definitions to identify relevant condition objects. For example, within the `<Dimensions>` list of a `<Dataset>` definition, each dimension object may specify the `ID` attribute value of the condition object that provides additional calibration information for that dimension (see 8.4: The `<Dimensions>` list element). Furthermore, in multi-dataset files, `<Dataset>` objects may include an `<IncludeConditions>` list element, which defines the `ID` values for the subset of conditions that apply to each dataset (see 8.5: The `<IncludeConditions>` list element).

If provided, the value of `ID` attributes shall be unique within the scope of the XML file, as required by the XML 1.01 specification. The `ID` attribute is reserved for use with top-level members of the `<Conditions>` list, and shall not be used for nested elements, nor for any element in the HMSA XML file outside of the `<Conditions>` list element.

For maximum compatibility, the `ID` string should only contain characters in the Unicode range U+0032 to U+007E, corresponding to printable ASCII characters (excluding special characters, see 5.5.6: Special characters).

7.5 Typical conditions

Although it is not necessary to include any conditions in a HMSA XML file, it is expected that HMSA XML files will typically include a sufficient set of experimental conditions to permit a meaningful interpretation of the dataset. For example, an image dataset may include conditions to define the pixel size or field of view. For spectra or spectral maps, the conditions may include calibration data to convert spectrum channels into energy or wavelength.

This specification defines a range of experimental conditions for common instrument components and experimental method, which are described in detail in Annex A: Condition templates and classes. The list below provides an overview of the salient conditions that may be included in typical HMSA XML file:

— An `<Instrument>` condition, which defines the manufacturer and model of the instrument.

— An `<Acquisition>` condition, which defines the date / time, duration and specimen position of the measurement.

— A `<Probe>` condition, which describes the analytical probe used to collect a HMSA dataset, such as settings for electron or ion columns, lasers, etc. For electron microscopes, the `<Probe Class="EM">` sub-class is defined, which includes parameters for accelerating voltage, beam current, and other probe-forming conditions.

— A `<Detector>` condition, which describes the type and configuration of the detector used to collect the HMSA dataset, such as a camera, spectrometer, or other type of sensor. Specific sub-classes are

defined for common detector types, such as `<Detector Class="Camera">` and `<Detector Class="XEDS">`.

— A `<Calibration>` condition, which describes the energy or wavelength calibration for channels in spectrum or spectral map datasets.

— A `<MeasurementMode>` condition, which describes additional instrument configuration parameters that are not related only to the probe or detector. For example, in a TEM diffraction pattern dataset, this condition may describe parameters relating to the intermediate and projector lens, camera length, etc.

— One or more `<Axis>` conditions, for defining the spatial dimensions of the dataset, including step size and direction.

— A `<Specimen>` condition, which describes the identity, ownership, and optionally composition of the specimen.

— A `<SpecimenEnvironment>` condition, which describes the temperature and ambient pressure at the specimen and may be useful for experiments at low or high temperatures, or in controlled atmospheres.

For more information about specific condition templates, please refer to Annex A: Condition templates and classes. Complete examples of XML files with a typical range of conditions are provided in Annex D: Example files.

8   The `<Dataset>` element

8.1 General

The `<Dataset>` element describes the location, size, and layout of the binary data within the associated HMSA file. Multiple `<Dataset>` elements are permitted in HMSA XML files, but applications are not required to parse more than the first dataset.

By design, dataset definitions do not contain extraneous data that is unrelated to the format of the binary data, such as experimental parameters to assist with the interpretation or display of the data. This arrangement ensures that common dataset types can be used across a range of techniques. For instance, the dataset definition for a spectral map will be identical regardless of whether the dataset was collected via XEDS, CL, EELS, Raman, etc.

All datasets have the following base structure:

```
<Dataset Name="Example">
   <DataOffset>123</DataOffset>
   <DataLength>456</DataLength>
   <DatumType>uint16</DatumType>
   <Dimensions>
      [ one or more dimension definitions ]
   </Dimensions>
   <IncludeConditions>
      [ zero or more references to conditions ]
   </IncludeConditions>
</Dataset>
```

The format of the HMSA binary data is that of a regular multidimensional array, with the dimension size and ordering defined by the contents of the `<Dimensions>` list. Experimental data that cannot be represented as a regular multidimensional array may be stored in the HMSA binary file as arbitrary data blocks (6.6: The `<ArbitraryData>` element).

By default, it is assumed that all conditions in the `<Conditions>` list apply to every dataset in the file. Optionally, datasets may explicitly specify a subset of conditions that apply using the `<IncludeConditions>` list, which may be necessary in multi-dataset files with multiple instances of the same condition template (8.5: The `<IncludeConditions>` list element).

The elements of the base dataset object are defined in detail in the following clauses.

8.2 The `<DataLength>` and `<DataOffset>` elements

The size of the binary data in a HMSA file corresponding to a `<Dataset>` object shall be defined in a `<DataLength>` element. The value of `<DataLength>` shall be measured in bytes, with an assumed minimum precision of 64 bits.

The location of the first dataset in the HMSA binary file shall be 8 bytes from the start of the file, meaning there is no padding between the 8-byte UID and the first dataset. The location of subsequent datasets within the HMSA file shall be defined by the `<DataOffset>` element, which is measured in bytes from the start of the file, in a minimum precision of 64 bits. The `<DataOffset>` element may be omitted from the definition of the first `<Dataset>` in the XML file, but shall be included for all subsequent datasets.

If more than one dataset is present in the file, the locations of subsequent datasets shall not overlap other datasets in the file, and may be:

— Non-contiguous. Padding is permitted between datasets, which may be used to store arbitrary or proprietary data that is not defined in this specification (6.6: The `<ArbitraryData>` element).

— Out of order. For example, dataset 2 can come after dataset 3 in the HMSA file.

8.3 The `<DatumType>` element

The data type of an individual numerical measurement within the dataset shall be declared using the `<DatumType>` element, like so:

`<DatumType>int</DatumType>`

For spectra and spectral maps, this element declares the data type of a spectrum channel. For images and hyperimage maps, this is the type of an image pixel. For multi-channel images, such as color micrographs, the datum type represents the type of a single channel within a pixel. A color image is therefore defined in the same way as a hyperspectral map, albeit with few channels (typically 3, for red green and blue.)

The `<DatumType>` element shall take one of the following values as shown in Table 4:

Table 4 — `<DatumType>` elhement values

DatumType	Size (B)	Description
"byte"	1	Unsigned 8-bit integer
"int16"	2	Signed 16-bit integer
"uint16"	2	Unsigned 16-bit integer
"int"	4	Signed 32-bit integer

"uint"	4	Unsigned 32-bit integer
"int64"	8	Signed 64-bit integer
"float"	4	32-bit IEEE 754 single-precision floating point number
"float64"	8	64-bit IEEE 754 double-precision floating point number

8.4 The `<Dimensions>` list element

8.4.1 General

The `<Dimensions>` list element is used to define the number and ordering of dimensions in the dataset, as well as the name, size and (optionally) calibration conditions for each dimension.

An example dimension list is given below for a simple 2-dimensional (X/Y) image that is 512x400 pixels in size, with three color channels:

```
<Dimensions>
    <Color>3</Color>
    <X>512</X>
    <Y>400</Y>
</Dimensions>
```

The numerical value of each item in the `<Dimensions>` list defines the size (i.e., number of measurement ordinals) of that dimension, such as the number of channels in a spectrum, or the number of pixels in the x dimension of an image. The size of all dimensions shall be greater than or equal to 1.

The ordering of items in the `<Dimensions>` list determines the ordering of measurements in the binary HMSA file (see 8.4.2: Ordering of dimensions). Equations for converting from measurement ordinals to file positions are given in 8.4.3: Coordinate mapping equations.

The format of the dimensions list is simple and terse by design. No parameters that may be useful for interpreting the dimension, such as energy calibration, may be provided directly in the `<Dimensions>` list. Instead, dimensions may identify an object in the `<Conditions>` list that provides additional calibration information (see 8.4.4: Identity and calibration of dimensions). Examples of common dataset dimensions for a range of different techniques are provided in Annex E.

8.4.2 Ordering of dimensions

Unlike the majority of the elements within the HMSA XML file, the ordering of items in the `<Dimensions>` list is meaningful and defines the sequence in which the data dimensions are written to the binary HMSA file. The data of a complete sequence of measurements of the first dimension is stored contiguously, before stepping to the next coordinate in the second collection dimension, and storing another complete sequence of the first dimension, and so on. As an example, consider the following `<Dimensions>` list for a 2-dimensional X/Y image that is 512x400 pixels in size:

```
<Dimensions>
    <X>512</X>
    <Y>400</Y>
</Dimensions>
```

As the <X> dimension is listed first, the data is written to the file as sequence of x-dimension scanlines, with pixel (x,y) coordinates in the following sequence: (0,0)…(511,0), (0,1)…(511,1), (0, 2)…(511,2), etc.

The ordering of items in the `<Dimensions>` list also affects how the data in multi-channel images are stored, such as color images or spectral maps. If the channel dimension is listed first (as below), the

complete multi-channel data from a pixel will be stored contiguously in the HMSA binary file (e.g., spectrum-by-spectrum).

```
<Dimensions>
    <Channel>4096</Channel>
    <X>512</X>
    <Y>400</Y>
</Dimensions>
```

For some techniques, it can be appropriate to store multi-channel images in the HMSA file as a set of contiguous image planes, with one image per channel. In such cases, the channel dimension will be listed last in the `<Dimensions>` list, as below:

```
<Dimensions>
    <X>512</X>
    <Y>400</Y>
    <Channel>4096</Channel>
</Dimensions>
```

This specification does not mandate a strict ordering of dimensions for all possible dataset types. However, to encourage compatibility and simplify implementations, the following recommendations should be followed:

— Dimensions should, where possible, be listed in order of acquisition speed. Thus, dimensions that are acquired quickly (or simultaneously) should be listed before dimensions that are acquired slowly. For example, in a sequence of CCD images, the X and Y dimension of the CCD are acquired essentially simultaneously and should be listed before the sequence dimension. This convention ensures the data is stored in the order in which it is acquired, which is convenient for data acquisition applications.

— Cartesian axes should be ordered in `<X>`, `<Y>` and then `<Z>`. Cartesian axes should either be listed all together as the first set of dimensions in the <Dimensions> list, or as the last set, and should not be interspersed with other non-spatial dimensions such as energy or wavelength.

— Color images should be stored with contiguous pixels (i.e., pixel-by-pixel), meaning the dimensions are listed in order of `<Color>`, `<X>`, then `<Y>`.

8.4.3 Coordinate mapping equations

Following the ordering of dimensions defined in the preceding clause, a generalized equation may be defined to determine the location within the HMSA binary file of any numeric measurement, which may be useful for out-of-core processing of large datasets, including random seeking. The equation below gives the offset, measured in bytes from the start of the HMSA binary file, for any numeric measurement in a dataset of up to five dimensions, where the coordinates for the measurement in each dimension are $c_0...c_4$, and the size (i.e., number of ordinals) of each dimension are $n_0...n_4$:

Offset = DataOffset + DatumTypeSize x (c_4 + n_4 x (c_3 + n_3 x (c_2 + n_2 x (c_1 + n_1 x c_0)))

This equation assumes a zero base for all coordinates, such that a dimension coordinate cx takes values of 0...*nx* - 1. The value of *DatumTypeSize* is the size in bytes of each numeric measurement and is determined from the `<DatumType>` element in the dataset definition (8.3: The `<DatumType>` element). The value of *DataOffset* is the offset relative to the start of the HMSA binary file and is given by the `<DataOffset>` element in the dataset definition (8.2: The `<DataLength>` and `<DataOffset>` elements).

The offset equation for data of higher dimensionality may be derived by induction by adding additional dimension and coordinate terms. Similarly, the equation may be simplified for data of lower dimensionality by setting the appropriate dimension size $(n_0...n_4)$ to 1 and corresponding coordinates $(c_0...c_4)$ to zero. Examples for salient dataset types are given below.

EXAMPLE   **Single spectrum (1 dimension)**

The equation below gives the offset, in bytes from the start of the HMSA binary file, of channel c in a spectrum of nc channels.

Offset = DataOffset + DatumTypeSize x c

EXAMPLE   **Single grayscale image (2 dimensions)**

The equation below gives the offset, in bytes from the start of the HMSA binary file, of a pixel with coordinates (x, y) in a single grayscale image measurement of nx x ny pixels.

Offset = DataOffset + DatumTypeSize Åx (x + nx x y)

This equation assumes x is the first dimension listed in the <Dimensions> list, and y is the second, as below:

```
<Dimensions>
    <X> [nx] </X>
    <Y> [ny] </Y>
</Dimensions>
```

EXAMPLE   **Hyperspectral map or color image (3 dimensions)**

The equation below gives the offset, in bytes from the start of the HMSA binary file, of a channel c in a pixel with coordinates (x, y) in a multi-channel image (e.g., color image, hyperspectral map) of nx x ny pixels and nc channels:

Offset = DataOffset + DatumTypeSize x (c + nc x (x + nx x y))

This equation assumes the channel dimension is the first dimension listed in the <Dimensions> list, x is the second, and y is the third, as below. This ordering of dimensions amounts to a spectrum-by-spectrum format in the HMSA file, wherein the complete multichannel data for each pixel is stored contiguously.

```
<Dimensions>
    <Channel> nc] </Channel>
    <X> [nx] </X>
    <Y> [ny] </Y>
</Dimensions>
```

If the multi-channel image in the HMSA file is stored as a set of contiguous image planes, with one image per channel, the channel dimension will be the last dimension in the <Dimensions> list, as below:

```
<Dimensions>
    <X> [nx] </X>
    <Y> [ny] </Y>
    <Channel> [nc] </Channel>
</Dimensions>
```

For such image-by-image datasets, the coordinate mapping equation becomes:

Offset = DataOffset + DatumTypeSize x (x + nx x (y + ny x c))

8.4.4 Identity and calibration of dimensions

The names of elements within the `<Dimensions>` list provide a simple, human-readable indication of the identity of each of the dataset dimensions. For example, a 2-dimensional (X/Y) hyperspectral map may have three dimensions named `<Channel>`, `<X>` and `<Y>`, like so:

```
<Dimensions>
    <Channel>2000</Channel>
    <X>512</X>
    <Y>400</Y>
</Dimensions>
```

The range of supported names for elements within the `<Dimensions>` list is not strictly defined in this specification. However, to encourage compatibility, examples and use cases of common dimensions are given in Annex E.

To ensure the `<Dataset>` definition is simple and universal, the items in the `<Dimensions>` list contain no information that may otherwise be useful in converting the measurement ordinals of the dimensions into physical quantities such as energy or position. Instead, the relevant calibration parameters may be provided as condition objects in the `<Conditions>` list, such as `<Calibration>` conditions. The calibration conditions associated with each dimension, if defined, shall have a unique `ID` attribute value, which by default should be the same value as the name of the dimension. Using the above hyperspectral map example, the conditions list should contain three conditions with `ID` attribute values of "Channel", "X" and "Y", like so:

```
<Conditions>
    <Calibration Class="LinearDispersion" ID="X">
        <Unit>um</Unit>
        <Gradient>0.390625</Gradient>
    </Calibration>
    <Calibration Class="LinearDispersion" ID="Y">
        <Unit>um</Unit>
        <Gradient>0.413125</Gradient>
    </Calibration>
    <Calibration Class="LinearDispersion" ID="Channel">
        <Quantity>Photoelectron binding energy</Quantity>
        <Unit>eV</Unit>
        <Intercept>5</Intercept>
        <Gradient>0.1</Gradient>
    </Calibration>
</Conditions>
```

It is not a requirement that all dimensions have associated calibration conditions. Dataset dimensions may be left uncalibrated, in which case the only indication to the identity of the dimension is the name of the dimension element in the `<Dimensions>` list.

If a dimension has an associated calibration condition with ID attribute value that is not equal to the name of that dimension, the `ID` of the condition shall be provided in the dimension declaration as a `ConditionID` attribute, like so:

```
<Dimensions>
    <Channel ConditionID="XEDS calibration">4096</Channel>
    <X>512</X>
    <Y>400</Y>
</Dimensions>
```

In the above example, the calibration condition associated with the `<Channel>` dimension shall have an `ID` attribute of "XEDS calibration", like so:

```
<Calibration Class="LinearDispersion" ID="XEDS calibration">
    <Quantity>Energy</Quantity>
    <Unit>eV</Unit>
    <Intercept>480</Intercept>
    <Gradient>5</Gradient>
</Calibration>
```

The `ConditionID` mechanism is provided to allow multi-dataset files to re-use the same dimension names in multiple datasets, but with different calibration conditions.

The names of the dimensions should be sufficient to allow a basic interpretation of the data of each dataset, as applications are not required to parse or understand any calibration conditions for the dataset dimensions. To maximize compatibility and commonality between techniques, generic names for dimensions should be used where possible. Examples of common dataset dimensions names are provided in Annex E, and include:

— `<X>`, `<Y>` and `<Z>`, for Cartesian dimensions relating to specimen coordinates.

— `<Color>`, for the color channels of a color image.

— `<Channel>`, for the multi-channel dimension of a spectrum or spectral map.

The names of dimensions are further restricted by the formatting requirements of XML elements, and shall not contain spaces. More expressive and scientifically meaningful names for the physical quantities of each dimension, such as "Energy loss" or "Binding energy", should be defined using the `<Quantity>` elements of the associated `<Calibration>` conditions.

8.5 The `<IncludeConditions>` list element

The `<IncludeConditions>` element is an optional element in the dataset definition that may contain zero or more references to the conditions that should be used to interpret the data in the dataset. If the `<IncludeConditions>` list is not defined or is empty, all conditions specified in the `<Conditions>` list are assumed to apply to the dataset. Furthermore, conditions with no `ID` attribute are assumed to apply to all datasets, regardless of the contents of the `<IncludeConditions>` list.

Condition references in the `<IncludeConditions>`, if used, shall take the following form:

```
<ConditionTemplateName>ConditionIdentifier</ConditionTemplateName>
```

...where `<ConditionTemplateName>` matches the template name for the condition, such has `<Probe>`, `<Detector>`, etc. The *ConditionIdentifier* value shall match the `ID` attribute of the element referenced in the `<Conditions>` list.

For example, to reference a condition defined in the `<Conditions>` list thusly:

```
<Detector Class="XEDS" ID="Windowless SDD">
    [...]
</Detector>
```

...the entry in the dataset's `<IncludeConditions>` list would be:

```
 <Detector>Windowless SDD</Detector>
```

Annex A
(normative)

Condition templates and classes

A.1 General

Although it is not necessary to include any conditions in a HMSA XML file, it is expected that the HMSA XML files will typically include a sufficient set of experimental conditions to permit a meaningful interpretation of the dataset. For example, an image dataset may include conditions to define the detector type, pixel size or field of view. For spectra or spectral maps, the conditions may include calibration data to convert spectrum channels into energy or wavelength.

This annex defines a few specific classes of the top-level elements of the `<Conditions>` list described in 7. The `<Conditions>` element list. Due to a highly varied and extensive nature of conditions required to describe any given microscopy data set, this document cannot practically include details of all templates and conditions. A committee-endorsed set of `<Conditions>` and their parameters is available on-line at the HMSA standards committee WWW site. Additional proposed parameters for `<Conditions>` can also be submitted there by the community.

http://www.microscopy.org/resources/scientific_data/

A.2 `<Instrument>`

A.2.1 General

The `<Instrument>` condition template is a generic object that describes the type of instrument used to collect a HMSA dataset.

A.2.2 The `<Manufacturer>` and `<Model>` elements

The name of the manufacturer of the instrument and the model shall be provided using the `<Manufacturer>` and `<Model>` elements, like so:

```
<Manufacturer>Example Inc.</Manufacturer>
<Model>Example Model 123</Model>
```

A.2.3 The `<SerialNumber>` element

The serial number of the instrument may be provided as a text string using the `<SerialNumber>` element, like so:

```
<SerialNumber>12345-abc-67890</SerialNumber>
```

Example:

```
<Instrument>
   <Manufacturer>ARCN Microscopes Inc.</Manufacturer>
   <Model>Model 6000</Model>
   <SerialNumber>12345-abc-67890</SerialNumber>
</Instrument>
```

25

A.3 `<Probe>`

The `<Probe>` condition template is a generic object that includes any parameters that describe the formation and properties of the probe that impinged on the sample to produce the signals recorded in the data set. This would include gun parameters for electron sources, wavelengths and/or spectral characteristics for light sources, and settings of any optical elements involved in probe formation. Such parameters are highly technique-specific, so this abstract condition template should not be used directly. Instead, it is recommended to use a `<Probe>` template with a specific Class attribute, such as `<Probe Class="EM">`, `<Probe Class="LM">`, or `<Probe Class="SPM">`.

A.4 `<Probe Class="EM">`

A.4.1 General

The `<Probe Class="EM">` condition template describes the probe formation setup and characteristics for an electron beam column.

A.4.2 Required elements:

A.4.2.1 The `<ProbeEnergy>` element

The probe energy of the source, measured as landing energy at the specimen, shall be given using the `<ProbeEnergy>` and `<unit>` elements, like so:

`<ProbeEnergy Unit="keV">15.</ProbeEnergy>`

A.4.3 Optional elements:

A.4.3.1 The `<GunType>` element

The type of electron gun should be defined using the `<GunType>` element like so:

`<GunType>Schottky FEG</GunType>`

Recognized values of the gun type are:

— "W filament", for conventional tungsten filament thermionic emitters,

— "LaB6", for lanthanum hexaboride thermionic emitters,

— "Cold FEG", for cold-cathode field emission guns, and;

— "Schottky FEG", for warm field emission guns.

A.4.3.2 The `<EmissionCurrent>` element

The emission current of the electron gun may be defined using the `<EmissionCurrent>` element like so:

`<EmissionCurrent Unit="uA">12345</EmissionCurrent>`

A.4.3.3 The `<FilamentCurrent>` element

The emitter filament heater current may be defined using the `<FilamentCurrent>` element like so:

`<FilamentCurrent Unit="A">1.234</FilamentCurrent>`

A.4.3.4 The `<ExtractorBias>` element

The bias voltage on the gun extractor may be defined using the `<ExtractorBias>` element like so:

```
<ExtractorBias Unit="V">4200</ExtractorBias>
```

A.4.3.5 The `<GunPressure>` element

The gun chamber vacuum pressure may be defined using the `<GunPressure>` element like so:

```
<GunPressure Unit="Pa">3.14E-10</GunPressure>
```

A.4.3.6 The `<ProbeDiameter>` element

The diameter of the probe at the point of incidence on the specimen may be defined using the `<ProbeDiameter>` element like so:

```
<ProbeDiameter Unit="nm">12345</ProbeDiameter>
```

A.4.3.7 The `<ProbeCurrent>` element

The incident probe current, should be defined using the `<ProbeCurrent>` element like so:

```
<ProbeCurrent Unit="nA">47.59</ProbeCurrent>
```

A.4.3.8 The `<ProbeConvergenceAngle>` element

The convergence semi-angle of the incident probe at the specimen may be defined using the `<ConvergenceAngle>` element, like so:

```
<ProbeConvergenceAngle Unit="mrad">1.5</ProbeConvergenceAngle>
```

A.4.3.9 The `<Aperture>` element(s)

The diameter of apertures in the probe forming optics may be defined using one or more <Aperture> objects.

The identity of the aperture shall be specified using the `<Name>` element, such as "Objective" for the objective lens aperture. The `<Selection>` element is optional and may be used to record the index of the selected aperture on an aperture strip or turret.

```
<Aperture>
   <Name>Objective</Name>
   <Diameter Unit="um">20</Diameter>
   <Selection>2</Selection>
</Aperture>
```

Note that apertures may also be defined in the `<MeasurementMode>` and `<Detector>` conditions.

A.4.3.10 The `<Control>` element(s)

The `<Control>` element name may be used to document one or more dimensionless control settings, such as vendor-specific instrument configuration modes. If provided, the `<Control>` element shall be defined like so:

```
<Control Label="SpotSize">5</Control>
```

The `Label` used herein is defined in the context of the vendor specific instrument used. Note that `<Control>` elements may also be used in the `<MeasurementMode>` and `<Detector>` conditions.

A.4.3.11 The `<LensCurrent>` element(s)

The current through lenses in the probe forming optics may be defined using one or more `<LensCurrent>` elements, like so:

`<LensCurrent Unit="A" Name="Condenser">2.34</LensCurrent>`

The identity of the lens shall be specified using the `Name` attribute. Example `Name` values for probe-forming lenses include "Condenser" and "Objective". Note that `<LensCurrent>` elements may also be used in the `<MeasurementMode>` condition.

A.5 `<Probe Class="EM/SEM">`

A.5.1 General

The `<Probe Class="EM/SEM">` condition template describes the probe formation setup and characteristics for a dedicated scanning electron beam column.

A.5.2 Optional elements:

A.5.2.1 The `<WorkingDistance>` element

The working distance from the objective optic element to the specimen surface may be defined using the `<WorkingDistance>` element, like so:

`<WorkingDistance Unit="mm">10</WorkingDistance>`

A.6 `<Probe Class="EM/TEM">`

A.6.1 General

The `<Probe Class="EM/TEM">` condition template describes the probe formation setup and characteristics for a transmission electron beam column.

A.6.2 Optional elements:

A.6.2.1 The `<ProbeMode>` element

The probe formation mode of the illumination column shall be:

`<ProbeMode>CTEM</ProbeEnergy>`

Recognized values of the probe mode are:

— "CTEM", for conventional broad beam illumination mode,

— "STEM", for fine probe mode using the objective pre-field,

A.6.3 Example:

Below is an example of the `<Probe Class="EM/SEM">` condition for a warm field emission electron column such as an SEM/EPMA. Note additional optional condition parameters from the HMSA WWW site are used in this example.

```
<Probe Class="EM/SEM">
   <GunType>Schottky FEG</GunType>
   <ProbeEnergy Unit="keV">15.</ProbeEnergy>
   <BeamCurrent Unit="nA">47.59</BeamCurrent>
   <WorkingDistance Unit="mm">10</WorkingDistance>
   <GunPressure Unit="Pa">2.7E-9</GunPressure>
   <ProbeDiameter Unit="nm">10</ProbeDiameter>
   <ExtractorBias Unit="V">4200</ExtractorBias>
   <FilamentCurrent Unit="A">2.4</FilamentCurrent>
   <EmissionCurrent Unit="uA">240</EmissionCurrent>
</Probe>
```

A.7 `<Specimen>`

A.7.1 General

The `<Specimen>` conditions template defines a physical specimen, including the name, origin, composition, etc.

A.7.2 The `<Name>` element

The name or identifier of the specimen should be provided using the `<Name>` element, like so:

```
<Name>Cryolite EPMA standard #1</Name>
```

A.7.3 The `<Description>` element

A description of the specimen may be provided using the `<Description>` element, like so:

```
<Description>Natural cryolite standard</Description>
```

A.7.4 The `<Owner>` element

The owner of the specimen may be provided using the `<Owner>` element, like so:

```
<Owner>ARCN Geosciences Laboratory plc</Owner>
```

A.7.5 The `<Origin>` element

The origin (i.e., geographical location, source institution / vendor, etc.) may be provided using the `<Origin>` element, like so:

```
<Origin>Ilimaussaq complex, Narsaq peninsula, Greenland</Origin>
```

A.7.6 The `<Material>` element

If the specimen composition is known and uniform, the material may be defined using a `<Material>` element, like so:

```
<Material>
   <Formula>Na3AlF6</Formula>
   <Composition Class="Elemental" Unit="atoms">
```

```
      <Element Z="11">3</Element>
      <Element Z="13">1</Element>
      <Element Z="9">6</Element>
   </Composition>
   <Density Unit="g/cm3">2.97</Density>
</Material>
```

A.7.7 The `<Coating>` element

If one or more coatings were applied to the analysis surface of the specimen, the coating(s) should be defined using a <Coating> element containing one or more <Material> elements, like so:

```
<Coating>
   <Material>
      <MaterialName>Carbon</MaterialName>
      <Thickness Unit="nm">50</Thickness>
   </Material>
   [ Multiple coatings are allowed  ]
</Coating>
```

If multiple coating layers are present, the order of <Material> elements in the <Coating> shall be the order of the layers from top to bottom (i.e., towards the substrate).

A.7.8 The `<Thickness>` element

If the specimen thickness (excluding coatings) is known and uniform, it may be defined using the <Thickness> element, like so:

```
<Thickness Unit="nm">10.0</Thickness>
```

A.7.9 Example:

The following is an example of a bulk specimen of cryolite (Na3AlF6), which has been carbon coated:

```
<Specimen>
   <Name>Cryolite EPMA standard #1</Name>
   <Description>Natural cryolite standard</Description>
   <Owner>ARCN Geosciences Laboratory plc</Owner>
   <Origin>Ilimaussaq complex, Narsaq peninsula, Greenland</Origin>
   <Material>
      <Formula>Na3AlF6</Formula>
      <Density Unit="g/cm3">2.97</Density>
   </Material>
   <Coating>
      <Material>
          <MaterialName>Carbon</MaterialName>
          <Thickness Unit="nm">50</Thickness>
      </Material>
   </Coating>
</Specimen>
```

A.8 `<SpecimenEnvironment>`

A.8.1 General

The `<SpecimenEnvironment>` conditions template defines the environment of the specimen, such as temperature, pressure, and surrounding medium.

A.8.2 The `<Pressure>` element

The temperature of the specimen during analysis may be provided using the `<Pressure>` element, like so:

```
<Pressure Unit="Pa">1E-8</Pressure>
```

A.8.3 The `<Temperature>` element

The temperature of the specimen during analysis may be provided using the `<Temperature>` element, like so:

```
<Temperature Unit="degreesC">-20.0</Temperature>
```

A.8.4 The `<Medium>` element

The surrounding medium, such as gas or liquid in an environmental cell, may be provided using the `<Medium>` element, like so:

```
<Medium>Water vapor</Medium>
```

A.8.5 Example:

The following is an example of an oxidizing hot-stage specimen environment in an ESEM, where the specimen was exposed to 2 700 Pa of water vapor at 1 220 K.

```
<SpecimenEnvironment>
   <Temperature Unit="K">1220</Temperature>
   <Pressure Unit="Pa">2.7E3</Pressure>
   <Medium>Water vapor</Medium>
</SpecimenEnvironment>
```

A.9 `<MeasurementMode>`

A.9.1 General

The `<MeasurementMode>` condition template is a generic object that describes the configuration and system parameters involved in selecting a particular signal from the specimen and coupling it to its associated detector. In other words, it specifies the details of the measurement mode or technique used to capture a particular dataset from the sample.

The boundary between `<MeasurementMode>` and `<Detector>` parameters may not always be clear cut, particularly for measurements that merely involve insertion of a detector into a scattered beam path, such as SE or BSE imaging signals in SEM. In such cases, and even in some that are more complex, such as XEDS, all measurement mode parameters are simply lumped into a `<Detector>` condition. This is largely a matter of convention and the conceptual preferences on the part of the practitioners of a given technique.

The base `<MeasurementMode>` condition defines only generic elements and should not be used directly; instead, it is recommended that one use a specific class that has been tailored to the requirements of the technique employed, such as

— `<MeasurementMode Class="TEM/Imaging">`.

— `<MeasurementMode Class="TEM/Diffraction">`.

— `<MeasurementMode Class="TEM/EELS">`.

— `<MeasurementMode Class="EFTEM">`.

It is expected that additional sub-classes of `<MeasurementMode>` will be defined in the future by technique experts to meet the needs of their experimental methods (see 3: Terms and definitions).

A.9.2 Optional elements:

A.9.2.1 The `<Control>` element(s)

The `<Control>` element name may be used to document one or more dimensionless control settings, such as vendor-specific instrument configuration modes. If provided, the `<Control>` element shall be defined like so:

```
<Control Label="Mag Index">4</Control>
```

The `Label` used herein is defined in the context of the vendor specific instrument used. Note that `<Control>` elements may also be used in the `<Probe>` and `<Detector>` conditions.

A.10 `<MeasurementMode Class="TEM">`

A.10.1 General

The `<MeasurementMode Class="TEM ">` condition template describes the image formation setup and characteristics for a TEM column.

A.10.2 Optional elements:

A.10.2.1 The `<Aperture>` element(s)

The diameter of apertures in the image forming optics may be defined using one or more `<Aperture>` objects.

The identity of the aperture shall be specified using the `<Name>` element, such as "Objective" for the objective lens aperture. The `<Selection>` element is optional, and may be used to record the index of the selected aperture on an aperture strip or turret.

```
<Aperture>
   <Name>Objective</Name>
   <Diameter Unit="um">20</Diameter>
   <Selection>2</Selection>
</Aperture>
```

Note that apertures may also be defined in the `<Probe>` and `<Detector>` conditions.

A.10.2.2 The `<LensCurrent>` element(s)

The current through lenses in the image forming and projection optics may be defined using one or more `<LensCurrent>` elements, like so:

```
<LensCurrent Unit="A" Name="Intermediate 2">1.37</LensCurrent>
```

The identity of the lens shall be specified using the `Name` attribute. Example `Name` values for probe-forming lenses include "Condenser" and "Objective". Note that `<LensCurrent>` elements may also be used in the <Probe> condition.

A.11 `<MeasurementMode Class="TEM/Imaging">`

A.11.1 General

The `<MeasurementMode Class="TEM/Imaging">` condition template describes the image formation setup and characteristics for a TEM column.

A.11.2 Optional elements:

A.11.2.1 The `<Defocus>` element

The defocus of the objective lens may be defined using the `<Defocus>` element like so:

```
<Defocus Unit="nm">72</Defocus>
```

A.11.2.2 The `<AcceptanceAngle>` element

The acceptance angle determined by the objective aperture can be defined using the `<AcceptanceAngle>` element like so:

```
<AcceptanceAngle Unit="mrad">4.5</AcceptanceAngle>
```

A.11.2.3 The `<NominalMagnification>` element

The nominal magnification of the imaging column may be defined using the `<NominalMagnification>` element like so:

```
<NominalMagnification>12000</NominalMagnification>
```

A.11.3 Example:

Below is an example of the `<MeasurementMode Class="TEM/Imaging">` condition for a TEM operating in conventional imaging mode:

```
<MeasurementMode Class="TEM/Imaging">
   <Defocus " Unit="nm">420</Defocus>
   <Control Label="Focus coarse">9C53</Control>
   <Control Label="Focus fine">46B2</Control>
   <LensCurrent Name="Objective" Unit="A">
      2.89
   </LensCurrent>
   <Aperture>
      <Name>Objective</Name>
      <Diameter Unit="um">20</Diameter>
      <Index>2</Index>
   </Aperture>
   <AcceptanceAngle Unit="mrad">4.5</AcceptanceAngle>
   <NominalMagnification>10000</NominalMagnification>
</MeasurementMode>
```

A.12 **`<Detector>`**

A.12.1 General

The `<Detector>` condition template is a generic object that describes the type and configuration of a detector used to collect a HMSA dataset.

A.12.2 Optional elements:

A.12.2.1 The **`<Manufacturer>`** and **`<Model>`** elements

The name of the manufacturer of the detector and the model should be provided using the `<Manufacturer>` and `<Model>` elements, like so:

```
<Manufacturer>Example Inc.</Manufacturer>
<Model>Example Model 123</Model>
```

A.12.2.2 The **`<SerialNumber>`** element

The serial number of the detector may be provided as a text string using the `<SerialNumber>` element, like so:

```
<SerialNumber>12345-abc-67890</SerialNumber>
```

A.12.2.3 The **`<SignalType>`** element

The type of signal measured by the detector should be defined using the `<SignalType>` element, like so:

```
<SignalType>EDS</SignalType>
```

Historically many of the labels used for `<SignalType>` have been interpreted both as a mode as well as a detector signal and changing this mindset is beyond the scope of this document. Since `<Conditions>` are optional elements, it is expected that the analyst will appreciate the labels used in context of the experiment and its data.

The recognized values of the `<SignalType>` element are the same as those defined in the EMSA/MAS spectrum file specification, namely:

— EDS = Energy Dispersive Spectroscopy

— WDS = Wavelength Dispersive Spectroscopy

— ELS = Energy Loss Spectroscopy

— AES = Auger Electron Spectroscopy

— PES = Photo Electron Spectroscopy

— XRF = X-ray Fluorescence Spectroscopy

— CLS = Cathodoluminescence Spectroscopy

— GAM = Gamma Ray Spectroscopy

Furthermore, this specification defines additional values for `<SignalType>` not included in the EMSA/MAS spectrum file specification:

— BEI = Backscattered Electron Imaging

— CBED = Convergent Beam Electron Diffraction

— EBSD = Electron Back-Scatter Diffraction

— EDIF = Electron Diffraction (generic)

— LEED = Low-Energy Electron Diffraction

— OPR = Optical, Reflected light

— OPT = Optical, Transmitted light

— PIXE = Proton Induced X-ray Emission Spectrometry

— RHEED = Reflection High-Energy Electron Diffraction

— SEI = Secondary Electron Imaging

— SXES = Soft X-ray Emission Spectroscopy

— TEI = Transmission Electron Imaging

To propose additional values, please contact the HMSA working group (see 3: Terms and definitions).

A.12.2.4 The `<MeasurementUnit>` element

The physical units of the quantity measured by the detector (typically intensity) may be defined using the `<MeasurementUnit>`, like so:

`<MeasurementUnit>counts</MeasurementUnit>`

The value of the `<MeasurementUnit>` element should be an SI, SI-derived or non-SI unit defined in Annex B, or a compound unit thereof (5.5.4: Physical units). If the `<MeasurementUnit>` element is not specified, a default value of "counts" shall be assumed.

A.12.2.5 The `<CollectionMode>` element

The `<CollectionMode>` element may be provided to describe the time sequencing of measurements by the detector. If provided, the `<CollectionMode>` element shall be of the form:

`<CollectionMode>Parallel</CollectionMode>`

Recognized `<CollectionMode>` values include:

— "Parallel", for detectors that measure their entire range essentially simultaneously (e.g., XEDS, CCD), and;

— "Serial" for spectrometers that measure their range sequentially (e.g., WDS, AES, CL monochromators).

A.12.2.6 The `<Distance>` element

The `<Distance>` element may be used to define the linear distance from the centre of the detector face to the point of incidence of the incident probe on the specimen surface. If provided, the `<Distance>` element shall be defined like so:

```
<Distance Unit="mm">50</Distance>
```

A.12.2.7 The `<Area>` element

The `<Area>` element may be used to define the surface area of the detector element. If provided, the `<Area>` element shall be defined like so:

```
<Area Unit="mm2">20</Area>
```

A.12.2.8 The `<SolidAngle>` element

The `<SolidAngle>` element may be used to define the solid angle subtended by the detector as measured from the point of incidence of the incident probe on the specimen surface. If provided, the `<SolidAngle>` element shall be defined like so:

```
<SolidAngle Unit="sr">1.</SolidAngle>
```

A.12.2.9 The `<SemiAngle>` element

For radially-symmetric detectors facing at normal angles towards the specimen, the `<SemiAngle>` element may be used to define half of the inside cone angle subtended by the detector as measured from the point of incidence of the incident probe on the specimen surface. If provided, the `<SemiAngle>` element shall be defined like so:

```
<SemiAngle Unit="mrad">3.4</SemiAngle>
```

A.12.2.10 The `<Temperature>` element

The `<Temperature>` element may be used to define the temperature of the detector element, like so:

```
<Temperature Unit="degreesC">-60.0</Temperature>
```

A.12.2.11 The `<Elevation>` element

The `<Elevation>` element should be used to define the elevation angle of the detector above the plane normal to the incident probe, as measured from the point of incidence on the specimen. As the `<Elevation>` element is measured relative to the incident probe, it is invariant with regards to rotation or tilt of the specimen. The `<Elevation>` element shall be measured in units of degrees, like so:

```
<Elevation Unit="degrees">45.</Elevation>
```

A.12.2.12 The `<Azimuth>` element

The azimuth angle of the detector may be defined using the `<Azimuth>` element, which is defined as the rotation around the incident probe vector, independent of specimen rotation or tilt. Zero azimuth occurs in the direction of the negative Y axis of the instrument (conventionally towards the rear of the instrument), and azimuth increases clockwise as viewed along the incident probe vector.

```
<Azimuth Unit="degrees">0.</Azimuth>
```

A.12.2.13 The `<DetectorName>` element

The `<DetectorName>` element may be used to identify a particular detector on an instrument, such as distinguishing between the multiple WDS and/or EDS detectors on an EPMA. If provided, the `<DetectorName>` element shall be defined as a text string like so:

```
<DetectorName>Ch1</DetectorName>
```

The value or interpretation of the `<DetectorName>` element is instrument specific, and is not defined in this specification.

A.12.2.14 The `<Aperture>` element(s)

The diameter of apertures in the detector optical path may be defined using one or more `<Aperture>` objects, like so:

```
<Aperture>
    <Name>XEDS aperture</Name>
    <Diameter Unit="mm">5</Diameter>
    <Selection>4</Selection>
</Aperture>
```

The identity of the aperture shall be specified by the `<Name>` element if multiple apertures are present in the detector definition. The `<Selection>` parameter is optional, and may be used to record the index of the selected aperture on an aperture strip or turret (if applicable).

A.12.2.15 The `<Control>` element(s)

The `<Control>` element name may optionally be used within `<Detector>` conditions to document one or more dimensionless detector-specific control parameters, such as software brightness and contrast adjustments. If provided, the `<Control>` element shall be defined like so:

```
<Control Label="DigitalContrast">100</Control>
```

The identity of the control parameter is described by the Label attribute. Please refer to `<Control>` in Annex A for more information about the usage of the `<Control>` parameter.

A.12.3 Example:

Examples of the `<Detector>` condition template are given for the following sub-classes:

— `<Detector Class="Camera">`

— `<Detector Class="CL">`

— `<Detector Class="WDS">`

— `<Detector Class="XEDS">`

A.13 `<Detector Class="Camera">`

A.13.1 General

The `<Detector Class="Camera">` condition template describes the calibration and collection mode of a camera used to collect a HMSA dataset, such as an EBSD or TEM camera.

A.13.2 Base template:

— `<Detector>`

A.13.3 Optional elements:

A.13.3.1 The `<FocalLength>` element

The focal length of the camera may be defined using the `<FocalLength>` element, like so:

`<FocalLength Unit="mm">10.</FocalLength>`

A.13.3.2 The `<ExposureTime>` element

The total exposure time of a frame, including integration, may be defined using the `<ExposureTime>` element, like so:

`<ExposureTime Unit="ms">200.</ExposureTime>`

A.13.3.3 The `<FrameIntegration>` element

The number of frames that are integrated to produce the output image may be defined using the `<FrameIntegration>` element, like so:

`<FrameIntegration>4</FrameIntegration>`

A.13.3.4 The `<Magnification>` element

The nominal indicated magnification of the camera and optics may be defined using the `<Magnification>` element, like so:

`<Magnification>200.</Magnification>`

A.13.3.5 The `<NumericalAperture>` element

The numerical aperture of the camera and optics may be defined using the `<NumericalAperture>` element, like so:

`<NumericalAperture>0.3</NumericalAperture>`

A.13.3.6 The `<PixelSize>` element

The physical size of the pixel elements of the camera may be defined using the `<PixelSize>` element, like so:

`<PixelSize Unit="um">7.4</PixelSize>`

A.13.4 Example:

Below is an example of a `<Detector  Class="Camera">` condition for a reflected-light optical microscope:

```
<Detector Class="Camera">
   <Manufacturer>ARCN Optical Microscopes Pty. Ltd.</Manufacturer>
   <Model>9000</Model>
   <SerialNumber>0774-X11</SerialNumber>
   <SignalType>OPR</SignalType>
   <Elevation Unit="degrees">90.</Elevation>
   <FocalLength Unit="mm">10.</FocalLength>
```

```
    <ExposureTime Unit="ms">40.</ExposureTime>
    <FrameIntegration>1</FrameIntegration>
    <Magnification>200.</Magnification>
    <NumericalAperture>0.3</NumericalAperture>
</Detector>
```

A.14 <Detector Class="CL">

A.14.1 General

The `<Detector Class="CL">` condition template describes the type and configuration of a cathodoluminescence spectrometer.

A.14.2 Base templates:

— `<Detector>`

A.14.3 Optional elements:

A.14.3.1 The `<DispersionElement>` element

The identity of the dispersion element of the CL spectrometer should be defined using the `<DispersionElement>` element, like so:

```
<DispersionElement>6M</DispersionElement>
```

This specification does not define recognized values for the `<DispersionElement>` element, as these are likely to be vendor-specific.

A.14.3.2 The `<Grating-d>` element

For CL spectrometers using diffraction gratings with constant d-spacing, the d-spacing should be defined using the `<Grating-d>` element, like so:

```
<Grating-d Unit="mm-1">800</Grating-d>
```

The d-spacing may be expressed as the distance between successive lines (`Unit`="um" or "nm"), or as lines per millimetre (`Unit`="mm-1").

A.14.3.3 The `<EntranceSlit>` element

The entrance slit of the spectrometer may be defined using the `<EntranceSlit>` element, like so:

```
<EntranceSlit Unit="um">50</EntranceSlit>
```

A.14.4 Example:

The example below is the definition of a Peltier-cooled grating CCD cathodoluminescence spectrometer, with a wavelength calibration defined as a third order polynomial:

```
<Detector Class="CL">
    <SignalType>CLS</SignalType>
    <CollectionMode>Parallel</CollectionMode>
    <Manufacturer>ARCN Optical Instruments, GmbH</Manufacturer>
    <Model>MX800</Model>
```

```
    <DispersionElement>H50</DispersionElement>
    <Temperature Unit="degreesC">-10</Temperature>
    <EntranceSlit Unit="um">200.</EntranceSlit>
</Detector>
```

Note that the wavelength/energy calibration of the detector, if defined, is stored as a separate top-level condition in the `<Conditions>` list, for example:

```
<Calibration Class="Polynomial" ID="CL calibration">
    <Quantity>Wavelength</Quantity>
    <Unit>nm</Unit>
    <Coefficients ArrayType="float" Count="3">
       199.945602, 0.79385, -0.00003
    </Coefficients>
</Calibration>
```

A.15 `<Detector Class="WDS">`

A.15.1 General

The `<Detector Class="WDS">` condition template describes the type and configuration of a wavelength dispersive x-ray spectrometer.

A.15.2 Base templates:

— `<Detector>`

A.15.3 Optional elements:

A.15.3.1 The `<DispersionElement>` element

The identity of the dispersion element should be defined using the `<DispersionElement>` element, like so:

```
<DispersionElement>TAP</DispersionElement>
```

The values of the `<DispersionElement>` may include traditional WDS crystals, such as:

— TAP, for thallium acid phthalate,

— RAP, for rubidium acid phthalate,

— KAP, for potassium acid phthalate,

— PET, for pentaerythritol, or;

— LIF, for lithium fluoride.

Additionally, the `<DispersionElement>` element may take values that are vendor-specific, including those of synthetic multi-layer elements (e.g., PC1, LDEB), or modified crystals (e.g., PETJ, LIFH).

A.15.3.2 The `<Crystal-2d>` element

The d-spacing of the dispersion element should be defined using the `<Crystal-2d>` element, like so:

```
<Crystal-2d Unit="Å">8.742</Crystal-2d>
```

A.15.3.3 The `<RowlandCircleDiameter>` element

The diameter of the Rowland circle of the WDS may be defined using the `<RowlandCircleDiameter>` element, like so:

```
<RowlandCircleDiameter Unit="mm">
   140.
</RowlandCircleDiameter>
```

A.15.3.4 The `<PulseHeightAnalyzer>` elements

The configuration of the pulse height analyzer (PHA) of the WDS spectrometer should be defined, like so:

```
<PulseHeightAnalyzer>
   <Bias Unit="V">1700.</Bias>
   <Gain>16.</Gain>
   <BaseLevel Unit="V">0.7</BaseLevel>
   <Window Unit="V">9.3</Window>
   <Mode>Integral</Mode>
</PulseHeightAnalyzer>
```

The PHA `<Mode>` element should be either "Integral" or "Differential".

Note that if the WDS spectrometer is performing a PHA scan of the bias, gain, window or baseline, the variable parameter shall be omitted from the `<PulseHeightAnalyzer>` definition. Instead, the variable parameter shall be defined as the calibration quantity in the `<Calibration>` object associated with the PHA scan dimension of the dataset.

A.15.3.5 The `<Counter>` element

The configuration of the x-ray counter may be defined using the `<Counter>` element, like so:

```
<Counter>
   <Type>Sealed Xe</Type>
   <Pressure Unit="Pa">1.01E5</Pressure>
   <Window>
      <Material>
          <MaterialName>Al</MaterialName>
          <Thickness Unit="nm">40</Thickness>
      </Material>
      <Material>
          <MaterialName>PET</MaterialName>
          <Thickness Unit="nm">200</Thickness>
      </Material>
      [ multiple layers are supported ]
   </Window>
</Counter>
```

The counter `<Type>` element for proportional counters may be "Sealed" or "Flow", followed by the gas medium, which may be "Xe", "Ar", and "P10" (for 10% methane in Ar). Scintillator-photomultiplier counters are supported but are not defined here.

The `<Pressure>` element may be used to define the gas pressure in a proportional counter. If the counter pressure value is regulated relative to ambient atmospheric pressure, such as may occur with a

flow type counter, the `<Pressure>` element shall include a `Relative="Ambient"` attribute. If this attribute is not present, the pressure shall be interpreted as absolute.

The `<Window>` and `<Material>` elements are defined identically to those in the `<Detector Class="XEDS">` condition template, including the list of recognized window materials (e.g., "Al", "Be", "PET", etc.)

A.15.3.6 The `<WDSPosition>` element

When the WDS spectrometer is acting as a monochromator with a fixed position or angle, such as when mapping or performing a PHA scan, the dataset may not contain a dimension corresponding to WDS position, and hence there may be no associated `<Calibration>` object to define the WDS position. Therefore, the WDS dispersion element position or angle should be separately defined in the `<Detector Class="WDS">` condition using the `<WDSPosition>` element, as below:

`<WDSPosition Unit="mm">95</WDSPosition>`

The `Unit` attribute may be "mm" for spectrometers with linear positions, or "degrees" for spectrometers with angular (2 theta) positions. Note that the `<WDSPosition>` element should not be confused with the `<Position>` element (defined in the base `<Detector>` condition template), which defines the linear distance from the specimen to the detector.

A.15.4 Examples:

The example below is a WDS detector functioning as a monochromator, such as when performing WDS elemental mapping.

```
<Detector Class="WDS">
   <DetectorName>Ch1</DetectorName>
   <SignalType>WDS</SignalType>
   <DispersionElement>LIF</DispersionElement>
   <Crystal-2d Unit="Å">4.0267</Crystal-2d>
   <Elevation Unit="degrees">40.</Elevation>
   <RowlandCircleDiameter Unit="mm">
      140.
   </RowlandCircleDiameter>
   <PulseHeightAnalyzer>
      <Bias Unit="V">1623.</Bias>
      <Gain>16.</Gain>
      <BaseLevel Unit="V">2.</BaseLevel>
      <Window Unit="V">4.</Window>
      <Mode>Differential</Mode>
   </PulseHeightAnalyzer>
   <WDSPosition Unit="mm">178.255997</WDSPosition>
   <Counter>
      <Type>Flow P10</Type>
   </Counter>
</Detector>
```

A.16 `<Detector Class="XEDS">`

A.16.1 General

The `<Detector Class="XEDS">` condition template describes the type and configuration of an energy dispersive x-ray spectrometer.

A.16.2 Base templates:

— `<Detector>`

A.16.3 Optional elements:

A.16.3.1 The `<Technology>` element

The XEDS detector technology type may be defined using the `<Technology>` element, like so:

```
<Technology>SDD</Technology>
```

Recognized values of the `<Technology>` element include:

— "Ge", for germanium-based detectors

— "SiLi", for lithium-drifted silicon detectors

— "SDD", for silicon drift detectors

— "Microcalorimeter"

A.16.3.2 The `<NominalThroughput>` element

The nominal maximum throughput of the XEDS spectrometer, in terms of kilocounts per second, may be defined using the `<NominalThroughput>` element like so:

```
<NominalThroughput Unit="kcounts/s">
   180.
</NominalThroughput>
```

A.16.3.3 The `<TimeConstant>` element

The time constant of the XEDS detector preamplifier may be defined using the `<TimeConstant>` element like so:

```
<TimeConstant Unit="us">12.5</TimeConstant>
```

A.16.3.4 The `<StrobeRate>` element

The nominal input count rate of the zero-energy noise strobe of the detector, if known, should be defined using the `<StrobeRate>` element like so:

```
<StrobeRate Unit="Hz">2000</StrobeRate>
```

A.16.3.5 The `<Window>` element

The composition and thickness of the window in front of the XEDS detector may be defined using the `<Window>` element, which contains one or more `<Material>` elements (see `<Material>` condition template), likes so:

```
<Window>
   <Material>
      <MaterialName>Al</MaterialName>
      <Thickness Unit="nm">40</Thickness>
   </Material>
```

```
    <Material>
        <MaterialName>PET</MaterialName>
        <Thickness Unit="nm">200</Thickness>
    </Material>
    [ multiple layers are supported ]
</Window>
```

The order of the window layer materials defined in the `<Window>` list shall be given on the order from the specimen side to the detector side.

The composition of the window materials may be defined exhaustively using the `<Composition>` optional element in the `<Material>` condition template, or by providing the material name using the `<MaterialName>` element. Recognized window materials are:

— "Al", for aluminum windows

— "Be", for beryllium windows

— "BN", for boron nitride windows

— "Diamond", for diamond windows

— "PET", for polyethylene terephthalate windows

— "PP", for polypropylene windows

Vendor-specific values of `<MaterialName>` may be used when customary to do so.

A.16.3.6 The `<GoldLayer>` element

The thickness of the gold layer / electrical contact on the surface of the XEDS detector may be defined using the `<GoldLayer>` element like so:

```
<GoldLayer Unit="nm">20.0</GoldLayer>
```

A.16.3.7 The `<DeadLayer>` element

The thickness of the dead layer of the XEDS detector may be defined using the `<DeadLayer>` element like so:

```
<DeadLayer Unit="nm">100.0</DeadLayer>
```

A.16.3.8 The `<ActiveLayer>` element

The thickness of the active layer of the XEDS detector may be defined using the `<ActiveLayer>` element like so:

```
<ActiveLayer Unit="mm">3.0</ActiveLayer>
```

A.16.4 Examples:

The following example is a typical SDD XEDS detector:

```
<Detector Class="XEDS">
    <Manufacturer>ARCN Detector Co.</Manufacturer>
    <Model>ARCN Model 1</Model>
    <SignalType>EDS</SignalType>
```

```
    <MeasurementUnit>counts</MeasurementUnit>
    <CollectionMode>Parallel</CollectionMode>
    <Technology>SDD</Technology>
    <Temperature Unit="degreesC">-25.0</Temperature>
    <NominalThroughput Unit="kcounts/s">
        400.
    </NominalThroughput>
    <StrobeRate Unit="Hz">2000.</StrobeRate>
    <Area Unit="mm2">10.</Area>
    <Elevation Unit="degrees">45.</Elevation>
    <Window>
        <Material>
            <MaterialName>Al</MaterialName>
            <Thickness Unit="nm">40</Thickness>
        </Material>
        <Material>
            <MaterialName>PET</MaterialName>
            <Thickness Unit="nm">200</Thickness>
        </Material>
    </Window>
</Detector>
```

Note that the energy calibration of the detector, if defined, is stored as a separate top-level condition in the `<Conditions>` list, for example:

```
<Calibration Class="LinearDispersion" ID="XEDS calibration">
    <Quantity>Energy</Quantity>
    <Unit>eV</Unit>
    <Gradient>5.0</Gradient>
    <Intercept>480</Intercept>
</Calibration>
```

A.17 `<Acquisition>`

A.17.1 General

The `<Acquisition>` condition template is a generic object that describes the duration and position on the specimen of the measurements in a dataset.

A.17.2 The `<DateTime>` element

The date and time of the start of the acquisition should be recorded in a `<DateTime>` element (see `<DateTime>` condition template in Annex A), like so:

```
<DateTime>
    <Date>1985-10-26</Date>
    <Time>20:04:00</Time>
    <Timezone>UTC+10</Timezone>
</DateTime>
```

A.17.3 The `<SpecimenPosition>` element

The `<SpecimenPosition>` condition template defines a physical location on, or in, the specimen from which the data represents.

The default coordinate system for the `<SpecimenPosition>` element is defined in terms of the probe geometry of the instrument. The Z axis is defined by the probe direction, such that when the analysis surface of the specimen is placed at normal incidence to the incident probe, the positive Z axis is pointing into the specimen. The X and Y axes are orthogonal to the Z axis and to each other and shall be oriented such that they form a 'right-handed' coordinate system. For vertically oriented probes, with the positive probe axis pointing downwards, the positive Y axis should point towards the front of the instrument, and the positive X axis should point to the right of the instrument. The rotation of the X and Y axes around the Z axis relative to the instrument is not defined for other probe orientations, and may be assigned arbitrarily, providing 'right handedness' is preserved. This convention corresponds to the coordinate system used in some microscopes and is consistent with the customary coordinate system used in computer graphics, where X is to the right, Y is down the screen, and Z is into the screen (also a 'right-handed' system).

Note that the specimen coordinate system may be arbitrarily tilted and/or rotated from this default orientation relative to the probe using the `<EulerRotation>` (for tilt) and `<R>` (for rotation) elements as described below.

The order of the translation (`<X>`, `<Y>` and `<Z>`), tilt (`<EulerRotation>`), and Z-axis rotation (`<R>`) operations that convert from default probe coordinates to specimen coordinates is determined by the order of the `<X>`, `<Y>`, `<Z>`, `<EulerRotation>` and `<R>` elements as they are listed in the `<SpecimenPosition>` condition. See example below.

A.17.4 Position elements:

A.17.4.1 The `<X>`, `<Y>` and `<Z>` elements

The coordinate in the X, Y and Z Cartesian axes may be defined using the `<X>`, `<Y>` and `<Z>` elements, like so:

```
<X Unit="mm">1.01</X>
<Y Unit="mm">-5.0</Y>
<Z Unit="mm">10.0</Z>
```

The origins of the X, Y and Z axes in the specimen coordinate system are arbitrary, and may vary from instrument to instrument depending on the implementation of the probe deflection and focusing system, and/or the specimen stage.

A.17.4.2 The `<EulerRotation>` element

To support a range of specimen tilting apparatus, such as double-tiling TEM holders, eucentric goniometer stages, as well as SEM tilting stages with different tilt axes, the tilt of the specimen may be defined as a rotation by up to three Euler angles, using a `<EulerRotation>` element, like so:

```
<EulerRotation Sequence="z-x-z" Unit="degrees" ArrayType="float"
Count="3">
   30, 10, 0
</EulerRotation>
```

The `Sequence` attribute defines the extrinsic Euler rotation operation, and may include *up to* three of the x, y and z axes in any combination. In the above example using the "z-x-z" sequence, the first rotation is about the Z axis (clockwise from above), the second rotation is about the transformed X axis (clockwise from the left), and the third rotation is about the transformed Z axis (clockwise from above). In all cases, the rotations are 'right-handed'. The number of angles in the `<EulerRotation>` element may be one, two or three, and shall be equal to the number of transformations in the `Sequence` attribute. For example, if two Euler angles are defined, the `Sequence` shall contain two transformations (e.g., "x-y").

Note that for single-axis rotations, the `<EulerRotation>` value is a single value, not an array.

A.17.4.3 The `<R>` element

The rotation coordinate around the normal to the specimen surface may be defined using the `<R>` element, like so:

```
<R Unit="degrees">90.0</R>
```

The rotation operation rotates the physical specimen around the Z axis of the specimen coordinate system, in a clockwise direction when viewed from above (i.e., along the positive Z direction).

A.17.4.4 The `<TotalTime>` element

If the dataset includes multiple measurements (such as a linescan or map), the `<TotalTime>` element may be used to define the total *real time* taken to collect all measurements in the acquisition set. If provided, the `<TotalTime>` element shall be of the form:

```
<TotalTime Unit="s">14400.0</TotalTime>
```

A.17.4.5 The `<FrameCount>` element

If the acquisition is performed by integrating multiple measurements of the same points, the number of passes should be defined using the `<FrameCount>` element, as below.

```
<FrameCount>40</FrameCount>
```

Note that if the number of frames is greater than 1, the `<DwellTime>` condition, if defined, refers to the total integrated time for each point, not the time for each point in each pass.

A.17.4.6 The `<FrameTime>` element

If the dataset includes one or more measurements of an image frame (such as a CCD micrograph), the `<FrameTime>` element may be used to define the total real time taken to collect each frame. If provided, the `<FrameTime>` element shall be of the form:

```
<FrameTime Unit="s">25.0</FrameTime>
```

A.17.4.7 The `<DwellTime>` element

The `<DwellTime>` element may be used to define the uniform real time taken for each individual datum measurement, such as a point spectrum acquisition, a single point in a linescan, or a pixel in a map. If provided, the `<DwellTime>` element shall be of the form:

```
<DwellTime Unit="s">35.0</DwellTime>
```

A.17.4.8 The `<DwellTime_Live>` element

The `<DwellTime_Live>` element may be used to define the detector live time for each individual measurement, if known. If provided, the `<DwellTime_Live>` element shall be of the form:

```
<DwellTime_Live Unit="s">35.0</DwellTime_Live>
```

A.18 `<Sequence>`

A.18.1 General

The `<Sequence>` condition template defines a sequence of measurements in which one type of condition object varies between each measurement in the sequence. Typically, the `<Sequence>` condition is used to define the set of `<Acquisition>` conditions corresponding to a `<Measurement>` dataset dimension (see `<Measurement>` in Annex E), such as may be used with a series of spectrum or image acquisitions.

The `<Sequence>` condition shall contain one or more conditions of the same type, such as `<Acquisition>` conditions (see `<Acquisition>` template in Annex A). Multiple parameters within each condition may vary from measurement to measurement, such as both `<DateTime>` and `<SpecimenPosition>` within `<Acquisition>`. Parameters that are invariant between each measurement condition may be omitted. For example:

— The `<Date>` and `<Timezone>` elements may be omitted from the `<DateTime>` objects in the second and subsequent `<Acquisition>` conditions if unchanged from the preceding entry.

— Dimensions may be omitted from the `<SpecimenPosition>` objects in the second and subsequent `<Acquisition>` conditions if the coordinate values are unchanged from preceding entry. Note that the ordering of dimensions in the `<SpecimenPosition>` objects within each `<Acquisition>` condition shall be consistent with one another, and with the order of dimensions in the `<SpecimenPosition>` object in the standalone `<Acquisition>` condition, if provided.

If the measurements were collected sequentially, the order of measurements in the dataset dimension and in the associated `<Sequence>` condition shall be given in order of increasing time.

Note that member conditions within a `<Sequence>` condition shall not contain an `ID` attribute, as `ID` attributes are reserved for top-level condition objects in the `<Conditions>` list (See 7.4: Condition Identifiers).

A.18.2 The `<Control>` element

The `<Control>` element defined previously can be used within the Acquisition conditions to document other parameters.

A.18.3 Example:

The following example shows the use of the `<Acquisition>` condition to define the starting time, duration, and position of a measurement such as a spectrum, image, or map.

```
<Acquisition>
   <DateTime>
      <Date>1985-10-26</Date>
      <Time>20:04:00</Time>
      <Timezone>UTC+10</Timezone>
   </DateTime>
   <SpecimenPosition Name="Origin">
      <X Unit="mm">1.01</X>
      <Y Unit="mm">-5.0</Y>
      <Z Unit="mm">10.0</Z>
      <R Unit="degrees">90.0</R>
      <EulerRotation Sequence="x" Unit="degrees">
           -70
      </EulerRotation>
   </SpecimenPosition>
   <FrameCount>1</FrameCount>
   <TotalTime Unit="s">14400.0</TotalTime>
```

```
    <DwellTime Unit="ms">35.0</DwellTime>
</Acquisition>
```

A.19 `<Calibration>`

A.19.1 General

The `<Calibration>` condition template describes the calibration of a set of measurement ordinals with respect to a physical quantity, such as converting channels in any spectrum to energy, or steps in a WDS peak scan to position, angle, wavelength or energy. This template should not be used directly. Instead, use a sub-class appropriate for the calibration relationship, such as one of the following:

— `<Calibration Class="Constant">`

— `<Calibration Class="LinearDispersion">`

— `<Calibration Class="Intensity">`

A.19.2 The `<Quantity>` element

The quantity of the calibration object shall be defined using the `<Quantity>` element, like so:

`<Quantity>Energy</Quantity>`

This specification does not enumerate or restrict the values of the `<Quantity>` element. However, the following examples values have been identified:

— "Energy", "Energy loss", "Binding energy", "Energy Dispersion"

— "Wavelength", "Wavenumber", "Wavelength Dispersion"

— "Position", "Depth"

— "Time", "Sputter time"

— "WDS position", "PHA gain"

— "Intensity", "Signal", "Counts"

A.19.3 The `<Unit>` element

If the value of the `<Quantity>` element is a physical quantity (i.e., not a dimensionless number, or set of labels), the calibration unit shall be defined by the `<Unit>` element, like so:

`<Unit>eV</Unit>`

The list of permitted values of the `<Unit>` element is defined in Annex B: Units and prefixes.

A.19.4 `<Calibration Class="Constant">`

A.19.4.1 General

The `<Calibration Class="Constant">` element defines the energy/wavelength (i.e., dispersion) calibration of a spectrometer or other measurement device operating at a fixed position, such as a CL monochromator.

A.19.4.2 The `<Value>` element

The single calibration value shall be defined using the `<Value>` element, like so:

```
<Value>-237.098251</Value>
```

The `<Value>` element shall not include a declaration of physical units using a `Unit` attribute. Instead, the physical unit shall be declared in the separate `<Unit>` element of the base `<Calibration>` template.

A.19.4.3 Example:

```
<Calibration Class="Constant">
    <Quantity>Energy</Quantity>
    <Unit>eV</Unit>
    <Value>-237.098251</Value>
</Calibration>
```

A.19.5 `<Calibration Class="LinearDispersion">`

A.19.5.1 General

The `<Calibration Class="LinearDispersion">` element defines the energy/wavelength (i.e., dispersion) calibration of a spectrometer or other measurement device, for which the measurement ordinals (e.g., channel numbers) have a linear relationship to the physical quantity (e.g., eV), with a constant gradient and intercept (i.e., gain and offset.)

A.19.5.2 The `<Gradient>` element

The `<Gradient>` element shall define the increment in calibration values from one ordinal to the next. For example, with an X-ray Energy Dispersive Spectrometer detector (`<Detector Class="XEDS">`), the value of the `<Gradient>` element can be the number of electron volts per channel.

```
<Gradient>2.49985</Gradient>
```

A.19.5.3 The `<Intercept>` element

The `<Intercept>` element shall define the calibration value (energy, wavelength, position, etc.) corresponding to the first measurement ordinal. For example, with an Xray Energy Dispersive Spectrometer detector (see `<Detector Class="XEDS">`), the value of the `<Intercept>` element is the energy of channel 0.

```
<Intercept>-237.098251</Intercept>
```

The `<Gradient>` and `<Intercept>` elements shall not include declarations of physical units using a `Unit` attribute. Instead, the physical unit is declared for both parameters in the separate `<Unit>` element, as required by the base `<Calibration>` template.

A.19.6 `<Calibration Class="PolynomialDispersion">`

A.19.6.1 General

The `<Calibration Class="PolynomialDispersion">` element defines the energy/wavelength (i.e., dispersion) calibration of a spectrometer or other measurement device, for which the measurement ordinals (e.g., channel numbers) have a relationship to the physical quantity (e.g., nm) that may be modeled by an nth order polynomial, where n > 1.

A.19.6.2 The `<Coefficients>` element

The calibration coefficients shall be defined using the `<Coefficients>` element, as an array of floating-point values (5.5.3: Arrays of values).

```
<Coefficients ArrayType="float" Count="4">
   -2.225, 0.677, 0.134, -0.018
</Coefficients>
```

The `<Coefficients>` element shall not include declarations of physical units using a `Unit` attribute. Instead, the physical unit is declared for all in the separate `<Unit>` element, as required by the base `<Calibration>` template.

A.19.7 `<Calibration Class="Explicit">`

A.19.7.1 General

The `<Calibration Class="Explicit">` element defines the energy/wavelength (i.e., dispersion) calibration of a spectrometer or other measurement device, for which relationship between the measurement ordinals (e.g., channel numbers) and physical quantity (e.g., nm) cannot be adequately modeled by linear or polynomial functions, and therefore shall be declared explicitly for each ordinal.

A.19.7.2 The `<Values>` element

The explicit range of physical values for the calibration object shall be defined using the `<Values>` element, as an array of floating-point values (5.5.3: Arrays of values), like so:

```
<Values ArrayType="float" Count="1024">
   198.557114, 199.364639, 200.172089, 200.979446, 201.786743,
202.593948,
...
</Values>
```

The `<Values>` element shall not include declarations of physical units using a `Unit` attribute. Instead, the physical unit is declared in the separate `<Unit>` element, as required by the base `<Calibration>` template.

A.19.8 `<Calibration Class="Intensity">`

A.19.8.1 General

The `<Calibration Class="Intensity">` element defines the signal intensity calibration of a spectrometer detector or other measurement device, for which relationship between the measurement ordinals and physical quantity needs to be specified. The `<MeasurementUnits>` element of the `<Detector>` can be employed if calibration of the intensity is not required and relative/dimensionless units are sufficient during any measurement.

A.19.8.2 The `<Quantity>` element

The quantity of the calibration object shall be defined using the `<Quantity>` element, like so:

```
<Quantity>Counts</Quantity>
```

This specification does not enumerate or restrict the values of the `<Quantity>` element. However, the following example is provided as a guide:

— "Counts"

— "Signal"

— "Electrons"

A.19.8.3 The `<Unit>` element

If the value of the `<Quantity>` element is a physical quantity (i.e., not a dimensionless number, or set of labels), the calibration unit shall be defined by the `<Unit>` element, like so:

```
<Unit>coulombs</Unit>
```

The list of permitted values of the `<Unit>` element is defined in Annex B: Units and prefixes.

A.19.8.4 Example:

```
<Calibration Class="LinearDispersion">
   <Quantity>Energy Dispersion</Quantity>
   <Unit>eV</Unit>
   <Gradient>2.49985</Gradient>
   <Intercept>-237.098251</Intercept>
</Calibration>
```

Annex B
(normative)

Units and prefixes

B.1 General

Parameters in HMSA XML files shall use SI units, SI derived units, or a limited set of non-SI units defined below. Except where noted below, all SI magnitude prefixes are permitted for all units (e.g., mm, keV).

Units and prefixes are case sensitive and shall be written with appropriate capitalization as given below.

B.2 SI units

The 7 base SI units are given in Table 5.

Table 5 — SI Units

Symbol	Unit	Quantity
M	Metre	Length
Kg	Kilogram	Mass. Prefixes may be used for values smaller than one kilogram (e.g., mg, ng), but shall not for values larger than one kilogram (no Gg, etc.)
s	Second	Time. Prefixes may be used for values smaller than one second (e.g., ns, ms), but shall not for values larger than one second (no ks, Ms etc.)
A	Ampere	Current
K	Kelvin	Temperature
mol	Mole	Amount of substance
Cd	Candela	Luminous intensity

B.3 SI-derived units

SI units derived from the 7 base units are given in Table 6.

Table 6 — SI derived units

Symbol	Unit	Quantity
Å	Angstrom	Length, equivalent to 10^{-10}m. Prefixes are not permitted (e.g., no kÅ). Note the character used is the Latin letter A with ring above (U+00C5). If 'Å' is untypeable, use appropriate conversions to 'nm' or 'pm', but never 'A'.
Bq	Becquerel	Radioactivity, equivalent to s^{-1}.
C	Coulomb	Electrical charge, equivalent to A.s.
Da	Dalton	Atomic mass, equivalent to 1,660 538 86 x 10^{-27} kg.

degreesC	Degree Celcius	Temperature, equivalent to K + 273,15. For compatibility reasons, 'degreesC' should be used in place of the Unicode degree symbol (U+00B0), as in '°C'.
F	Farad	Capacitance, equivalent to C/V.
Gy	Gray	Absorbed dose, equivalent to J/kg.
H	Henry	Inductance, equivalent to Wb/A.
Hz	Hertz	Frequency, equivalent to s^{-1}.
J	Joule	Energy, equivalent to $kg.m^2/s^2$.
L	Litre	Volume, equivalent to $10^{-3}m^3$.
lm	Lumen	Luminous flux, equivalent to W/m^2.
lx	Lux	Illuminance, equivalent to lm/m^2.
N	Newton	Force, equivalent to $kg.m/s^2$.
Ohm	Ohm	Electrical resistance, equivalent to V/A. For compatibility reasons, 'Ohm' should be used in place of the Unicode Greek capital letter omega 'Ω' (U+03A9)
Pa	Pascal	Pressure, equivalent to N/m^2.
rad	Radian	Angle.
S	Siemens	Electrical conductance, equivalent to Ohm^{-1}.
Sv	Sievert	Equivalent dose.
sr	Steradian	Solid angle.
T	Tesla	Magnetic flux density, equivalent to Wb/m^2.
V	Volt	Electrical potential, equivalent to J/C.
W	Watt	Power, equivalent to J/s.
Wb	Weber	Magnetic flux, equivalent to J/A.

B.4 Non-SI units

The permitted set of non-SI units are given in Table 7.

Table 7 — Non-SI units

Symbol	Unit	Quantity
degrees	Degree	Angle. For compatibility reasons, 'degrees' should be used in place of the Unicode degree symbol '°' (U+00B0).
atoms	Number of atoms	
counts	Counts	Dimensionless number of events.
counts/s	Counts per second	Number of events per second, equivalent to Hz. Customary equivalents "cps" or "c/s" shall not be used.
electrons	Electrons	Equivalent to 1,602 176 46 x 10^{-19} coulombs of charge.
eV	Electron volt	Energy, equivalent to 1,602 176 46 x 10^{-19}J.
%	Percent	Dimensionless ratio. *Not to be used* for concentrations (use mol%, vol%, wt%, etc.)

mol%	Molar/atomic percent	
vol%	Volumetric percent	
wt%	Weight percent	
mol_ppm	Molar parts per million	
vol_ppm	Volumetric parts per million	
wt_ppm	Weight parts per million	
mol_ppb	Molar parts per billion[*]	*Parts per billion shall refer to short billions (10^9), *never* long billions (10^{12}).
vol_ppb	Volumetric parts per billion[*]	
wt_ppb	Weight parts per billion[*]	

B.5 SI prefixes

SI magnitude prefixes, for the range 10^{24} to 10^{-24}, are given in Table 8.

Table 8 — SI magnitude prefixes

Symbol	Magnitude	Note
Y	10^{24}	"yotta"
Z	10^{21}	"zetta"
E	10^{18}	"exa"
P	10^{15}	"peca"
T	10^{12}	"tera"
G	10^{9}	"giga"
M	10^{6}	"mega"
k	10^{3}	"kilo"
m	10^{-3}	"milli"
u	10^{-6}	"micro". For compatibility reasons, the Latin character 'u' (U+0075) should be used in place of the Unicode micro sign 'µ' (U+00B5)
n	10^{-9}	"nano"
p	10^{-12}	"pico"
f	10^{-15}	"femto"
a	10^{-18}	"atto"
z	10^{-21}	"zepto"
y	10^{-24}	"yocto"

Annex C
(normative)

Unicode character substitutions

In the Unicode character set, there are several code points that produce visually indistinguishable glyphs. Consequently, to avoid confusion and maximize compatibility, the lowest code point shall be used in these cases. A non-exhaustive list of the required character substitutions is provided below:

— For the Ångström symbol, the Latin capital A with ring above 'Å' (U+00C5) shall be used, and not the Unicode Ångström symbol 'Å' (U+212B).

— For Kelvin, the Latin capital letter 'K' (U+004B) shall be used, and not the Unicode Kelvin symbol 'K' (U+212A).

— For 'Ω', the Greek capital letter omega 'Ω' (U+03A9) shall be used, and not the Unicode Ohm symbol 'Ω' (U+2126).

— For 'μ', the Unicode micro sign 'µ' (U+00B5) shall be used in place of the Unicode Greek small letter mu 'μ' (U+03BC).

Annex D
(informative)

Example files

D.1 Optical micrograph

The following example demonstrates a typical optical color image. In this example, the image contains 2048 x 2048 pixels, with each pixel consisting of three bytes for red, green and blue.

```xml
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="688761798AB6CF44"
xml:lang="en-US">
   <Header>
      <Title>Large field image</Title>
      <Date>2016-09-28</Date>
      <Time>23:42</Time>
      <Timezone>UTC+10 AUS Eastern Standard Time</Timezone>
      <Author>
            Microbeam Laboratory Team; ARCN Mineralogy Labs Inc.
      </Author>
      <Owner>ARCN Mineralogy Labs Inc.</Owner>
   </Header>
   <Conditions>
      <Instrument>
            <Manufacturer>ARCN Optical Microscopes Pty.
Ltd.</Manufacturer>
            <Model>9000</Model>
            <SerialNumber>0774-X11</SerialNumber>
      </Instrument>
      <Detector Class="Camera">
            <SignalType>OPR</SignalType>
            <Elevation Unit="degrees">90.</Elevation>
            <FocalLength Unit="mm">10.</FocalLength>
            <ExposureTime Unit="ms">40.</ExposureTime>
            <FrameIntegration>1</FrameIntegration>
            <Magnification>10.</Magnification>
            <NumericalAperture>0.3</NumericalAperture>
      </Detector>
      <Calibration Class="LinearDispersion" ID="X">
            <Unit>mm</Unit>
            <Gradient>0.01953125</Gradient>
      </Calibration>
      <Calibration Class="LinearDispersion" ID="Y">
            <Unit>mm</Unit>
            <Gradient>0.01953125</Gradient>
      </Calibration>
   </Conditions>
   <Dataset>
      <DataLength>12582912</DataLength>
```

```
        <DatumType>byte</DatumType>
        <Dimensions>
                <Color>3</Color>
                <X>2048</X>
                <Y>2048</Y>
        </Dimensions>
    </Dataset>
</MSAHyperDimensionalDataFile>
```

The same file, stripped of all conditions and header metadata, produces the following baseline file.

```
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="688761798AB6CF44"
xml:lang="en-US">
    <Header />
    <Conditions />
    <Dataset>
        <DataLength>12582912</DataLength>
        <DatumType>byte</DatumType>
        <Dimensions>
                <Color>3</Color>
                <X>2048</X>
                <Y>2048</Y>
        </Dimensions>
    </Dataset>
</MSAHyperDimensionalDataFile>
```

D.2 Single XEDS spectrum

The following example demonstrates a single spectrum acquisition using an XEDS detector.

```
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="03FF85CDAB6DC0EE"
xml:lang="en-US">
    <Header>
        <Title>Ca5(PO4)3(F) spectrum</Title>
        <Date>2016-09-28</Date>
        <Time>23:28:27</Time>
        <Timezone>UTC+10 AUS Eastern Standard Time</Timezone>
        <Author>
                Microbeam Laboratory Team; ARCN Mineralogy Labs Inc.
        </Author>
        <Owner>ARCN Mineralogy Labs Inc.</Owner>
    </Header>
    <Conditions>
        <Instrument>
                <Manufacturer>Vandelay Industries</Manufacturer>
                <Model>Model 200</Model>
        </Instrument>
        <Probe Class="EM">
                <BeamVoltage Unit="kV">5.</BeamVoltage>
                <BeamCurrent Unit="nA">10.</BeamCurrent>
                <ProbeDiameter Unit="um">10.</ProbeDiameter>
                <WorkingDistance Unit="mm">10</WorkingDistance>
```

```
        </Probe>
        <Acquisition>
                <DwellTime Unit="s">30</DwellTime>
                <DwellTime_Live Unit="s">25.2</DwellTime_Live>
                <SpecimenPosition>
                    <X Unit="mm">16.2</X>
                    <Y Unit="mm">43.8</Y>
                </SpecimenPosition>
        </Acquisition>
        <Detector Class="XEDS">
                <Manufacturer>ARCN Detector Co.</Manufacturer>
                <Model>ARCN Model 1</Model>
                <MeasurementUnit>counts</MeasurementUnit>
                <Technology>SDD</Technology>
                <NominalThroughput Unit="kcounts/s">
                        20.
                </NominalThroughput>
                <StrobeRate Unit="Hz">1000.</StrobeRate>
                <Area Unit="mm2">10.</Area>
                <Elevation Unit="degrees">45.</Elevation>
        </Detector>
        <Calibration Class="LinearDispersion" ID="XEDS calibration">
                <Quantity>Energy</Quantity>
                <Unit>eV</Unit>
                <Gradient>1.25</Gradient>
                <Intercept>-120.</Intercept>
        </Calibration>
    </Conditions>
    <Dataset>
        <DataLength>8192</DataLength>
        <DatumType>uint16</DatumType>
        <Dimensions>
                <Channel ConditionID="XEDS calibration">4096</Channel>
        </Dimensions>
    </Dataset>
</MSAHyperDimensionalDataFile>
```

The same file, stripped of all conditions and header metadata, produces the following baseline file.

```
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="03FF85CDAB6DC0EE"
xml:lang="en-US">
    <Header />
    <Conditions />
    <Dataset>
        <DataLength>8192</DataLength>
        <DatumType>uint</DatumType>
        <Dimensions>
                <Channel>4096</Channel>
        </Dimensions>
    </Dataset>
</MSAHyperDimensionalDataFile>
```

D.3 SEM backscattered electron image

This example represents a typical SEM backscattered electron image.

```xml
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="9904CC6205EF159F"
xml:lang="en-US">
   <Header>
      <Title>Example BEI image</Title>
      <Date>2016-09-28</Date>
      <Time>23:49</Time>
      <Timezone>UTC+10 AUS Eastern Standard Time</Timezone>
      <Author>
            Microbeam Laboratory Team; ARCN Mineralogy Labs Inc.
      </Author>
      <Owner>ARCN Mineralogy Labs Inc.</Owner>
   </Header>
   <Conditions>
      <Instrument>
            <Manufacturer>Vandelay Industries</Manufacturer>
            <Model>Model 200</Model>
      </Instrument>
      <Probe Class="EM">
            <BeamVoltage Unit="kV">15.</BeamVoltage>
      </Probe>
      <MeasurementMode Class="Microscope">
            <OperatingMode>SEM</OperatingMode>
            <NominalMagnification>2000</NominalMagnification>
      </MeasurementMode>
      <Detector>
            <SignalType>BEI</SignalType>
      </Detector>
      <Acquisition>
            <TotalTime Unit="s">60.</TotalTime>
      </Acquisition>
      <Calibration Class="LinearDispersion" ID="X">
            <Unit>um</Unit>
            <Gradient>0.09765625</Gradient>
      </Calibration>
      <Calibration Class="LinearDispersion" ID="Y">
         <Unit>um</Unit>
         <Gradient>0.09765625</Gradient>
      </Calibration>
   </Conditions>
   <Dataset>
      <DataLength>6553600</DataLength>
      <DatumType>uint16</DatumType>
      <Dimensions>
            <X>2048</X>
            <Y>1600</Y>
      </Dimensions>
   </Dataset>
</MSAHyperDimensionalDataFile>
```

The same file, stripped of all conditions and header metadata, produces the following baseline file.

```
 <?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="9904CC6205EF159F"
xml:lang="en-US">
    <Header />
    <Conditions />
    <Dataset>
        <DataLength>6553600</DataLength>
        <DatumType>uint16</DatumType>
        <Dimensions>
            <X>2048</X>
            <Y>1600</Y>
        </Dimensions>
    </Dataset>
</MSAHyperDimensionalDataFile>
```

D.4 Conventional TEM image

This example represents a typical conventional TEM image.

```
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="35611D89B3188257"
xml:lang="en-US">
    <Header>
        <Title>Biopsy 579344</Title>
        <DateTime>
            <Date>2015-06-03</Date>
            <Time>9:09:34</Time>
            <Timezone>UTC-08:00</Timezone>
        </DateTime>
        <Author>Jim User</Author>
        <Owner>Central Pathology Lab</Owner>
    </Header>
    <Conditions>
        <Instrument>
            <Manufacturer>TEMCO</Manufacturer>
            <Model>ThruView 120I</Model>
        </Instrument>
        <Probe Class="EM/TEM">
            <GunType>LaB6</GunType>
            <GunPressure Unit="Pa">5.2E-8</GunPressure>
            <BeamVoltage Unit="kV">120</BeamVoltage>
            <FilamentCurrent Unit="A">2.4</FilamentCurrent>
            <EmissionCurrent Unit="uA">5.5</EmissionCurrent>
            <OperatingMode>TEM</OperatingMode>
            <BeamDiameter Unit="um">12.4</BeamDiameter>
            <BeamCurrent Unit="nA">530</BeamCurrent>
            <ConvergenceAngle Unit="mrad">1.5</ConvergenceAngle>
            <Aperture>
                <Name>Condenser</Name>
                <Index>2</Index>
                <Diameter Unit="um">100</Diameter>
            </Aperture>
            <Control Label="SpotSize">2</Control>
```

　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　61

```
            <Control Label="Brightness">8A53</Control>
            <LensCurrent Label="C2" Unit="A">0.87</LensCurrent>
    </Probe>
    <Specimen>
            <Name>Biopsy 579344</Name>
            <Description>Liver tissue section</Description>
            <Owner>Central Pathology Lab</Owner>
    </Specimen>
    <MeasurementMode Class="TEM/Imaging">
            <OperatingMode>MAG</OperatingMode>
            <Defocus Unit="nm">420</Defocus>
            <ApertureDiameter Label="Objective"
Unit="um">20</ApertureDiameter>
            <Aperture>
                 <Name>Objective</Name>
                 <Index>2</Index>
                 <Diameter Unit="um">20</Diameter>
            </Aperture>
            <CollectionAngle Unit="mrad">4.5</CollectionAngle>
            <NominalMagnification>10000</NominalMagnification>
            <Control Label="Focus coarse">9C53</Control>
            <Control Label="Focus fine">46B2</Control>
            <Control Label="Mag">4</Control>
            <LensCurrent Label="Obj" Unit="A">2.89</LensCurrent>
    </MeasurementMode>
    <Detector Class="Camera/EM">
            <Manufacturer>EMCAM</Manufacturer>
            <Model>SharpEye 2000</Model>
            <SignalType>TEI</SignalType>
            <ExposureTime Unit="s">1.5</ExposureTime>
            <Binning>1</Binning>
            <ReadoutMode>High quality</ReadoutMode>
            <CollectionMode>Parallel</CollectionMode>
            <PixelSize Unit="um">7.4</PixelSize>
            <PixelColumns>2048</PixelColumns>
            <PixelRows>2048</PixelRows>
            <SaturationLevel>14500</SaturationLevel>
            <DarkCorrection>on</DarkCorrection>
            <GainCorrection>on</GainCorrection>
            <Calibration Class="LinearDispersion">
                 <Quantity>Primary electrons</Quantity>
                 <Unit>counts</Unit>
                 <Intercept>-209.15</Intercept>
                 <Gradient>0.235</Gradient>
            </Calibration>
            <Magnification>2235</Magnification>
    </Detector>
    <Acquisition>
            <DateTime>
                 <Date>2015-06-03</Date>
                 <Time>9:02:27</Time>
                 <Timezone>UTC-08:00</Timezone>
            </DateTime>
    </Acquisition>
```

```
        <Calibration Class="LinearDispersion" ID="X position">
                <Quantity>X position</Quantity>
                <Unit>nm</Unit>
                <Intercept>0</Intercept>
                <Gradient>3.31</Gradient>
        </Calibration>
        <Calibration Class="LinearDispersion" ID="Y position">
                <Quantity>Y position</Quantity>
                <Unit>nm</Unit>
                <Intercept>0</Intercept>
                <Gradient>3.31</Gradient>
        </Calibration>
    </Conditions>
    <Dataset>
        <DataLength>8388608</DataLength>
        <DatumType>int16</DatumType>
        <X CondtionID="X position">2048</X>
        <Y CondtionID="Y position">2048</Y>
    </Dataset>
</MSAHyperDimensionalDataFile>
```

The same file, stripped of all conditions and header metadata, produces the following baseline file.

```
 <?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="35611D89B3188257"
xml:lang="en-US">
    <Header />
    <Conditions />
    <Dataset>
        <DataLength>8388608</DataLength>
        <DatumType>int16</DatumType>
        <X CondtionID="X position">2048</X>
        <Y CondtionID="Y position">2048</Y>
    </Dataset>
</MSAHyperDimensionalDataFile>
```

D.5 Conventional electron diffraction pattern

This example represents a typical conventional TEM electron diffraction pattern.

```
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="35611D89B3188257"
xml:lang="en-US">
    <Header>
        <Title>Alloy 6-18 batch 12</Title>
        <DateTime>
                <Date>2016-04-18</Date>
                <Time>15:42:17</Time>
                <Timezone>UTC-06:00</Timezone>
        </DateTime>
        <Author>Molly Forger</Author>
        <Owner>Metals Inc.</Owner>
    </Header>
    <Conditions>
```

```
<Instrument>
        <Manufacturer>TEMCO</Manufacturer>
        <Model>ThruView 200A</Model>
</Instrument>
<Probe Class="EM/TEM">
        <GunType>Schottky FEG</GunType>
        <GunPressure Unit="Pa">1.4E-9</GunPressure>
        <BeamVoltage Unit="kV">200</BeamVoltage>
        <ExtractorBias Unit="V">3800</ExtractorBias>
        <EmissionCurrent Unit="uA">7.2</EmissionCurrent>
        <OperatingMode>TEM</OperatingMode>
        <BeamDiameter Unit="um">2.3</BeamDiameter>
        <BeamCurrent Unit="nA">418</BeamCurrent>
        <ConvergenceAngle Unit="mrad">1.7</ConvergenceAngle>
        <Aperture>
             <Name>Condenser</Name>
             <Index>3</Index>
             <Diameter Unit="um">50</Diameter>
        </Aperture>
        <Control Label="SpotSize">2</Control>
        <Control Label="Intensity">0.798</Control>
</Probe>
<Specimen>
        <Name>6-18 Foil 160416-09</Name>
        <Description>Alloy 6-18 24 min @ 550K</Description>
        <Owner>Metals Inc.</Owner>
</Specimen>
<MeasurementMode Class="TEM/Diffraction">
        <OperatingMode>DIFF</OperatingMode>
        <Aperture>
             <Name>Selected-area</Name>
             <Index>4</Index>
             <Diameter Unit="um">10</Diameter>
        </Aperture>
        <AreaDiameter Unit="nm">434</AreaDiameter>s
        <NominalCameraLength Unit="cm">80</NominalCameraLength>
        <Control Label="Diff focus">0.643</Control>
</MeasurementMode>
<Detector Class="Camera/EM">
        <Manufacturer>EMCAM</Manufacturer>
        <Model>DeepEye 4000</Model>
        <SignalType>TEI</SignalType>
        <ExposureTime Unit="s">0.1</ExposureTime>
        <Binning>2</Binning>
        <ReadoutMode>HDR</ReadoutMode>
        <CollectionMode>Parallel</CollectionMode>
        <PixelSize Unit="um">9</PixelSize>
        <PixelColumns>4096</PixelColumns>
        <PixelRows>4096</PixelRows>
        <SaturationLevel>58000</SaturationLevel>
        <DarkCorrection>on</DarkCorrection>
        <GainCorrection>on</GainCorrection>
        <Calibration Class="LinearDispersion">
              <Quantity>Primary electrons</Quantity>
```

```
                        <Unit>counts</Unit>
                        <Intercept>-587.8</Intercept>
                        <Gradient>0.437</Gradient>
                </Calibration>
                <CameraLength Unit="cm">113.4</CameraLength>
        </Detector>
        <Acquisition>
                <DateTime>
                        <Date>2016-04-18</Date>
                        <Time>15:30:19</Time>
                        <Timezone>UTC-06:00</Timezone>
                </DateTime>
        </Acquisition>
        <Calibration Class="LinearDispersion" ID="U component">
                <Quantity>U component</Quantity>
                <Unit>1/nm</Unit>
                <Intercept>-6.287</Intercept>
                <Gradient>0.006325</Gradient>
        </Calibration>
        <Calibration Class="LinearDispersion" ID="V component">
                <Quantity>V component</Quantity>
                <Unit>1/nm</Unit>
                <Intercept>6.524</Intercept>
                <Gradient>-0.006325</Gradient>
        </Calibration>
    </Conditions>
    <Dataset>
        <DataLength>8388608</DataLength>
        <DatumType>int16</DatumType>
        <U CondtionID="U component">2048</U>
        <V CondtionID="V component">2048</V>
    </Dataset>
</MSAHyperDimensionalDataFile>
```

D.6 SEM-XEDS hyper-spectral map

This example represents a typical XEDS spectral map, as captured on an SEM. A baseline example of the same map, excluding all optional conditions and metadata, is provided thereafter.

```
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="7FE6B4B91EB3B81E"
xml:lang="en-US">
    <Header>
        <Title>Gneiss</Title>
        <Date>2012-08-15</Date>
        <Time>16:15:16</Time>
        <Timezone>UTC+10 AUS Eastern Standard Time</Timezone>
        <Author>
                Microbeam Laboratory Team; ARCN Mineralogy Labs Inc.
        </Author>
        <Owner>ARCN Mineralogy Labs Inc.</Owner>
        <Checksum Algorithm="SHA-1">
                79C5C30510A4F515E62F9F8BC9762BB8F59CF6ED
```

```
        </Checksum>
    </Header>
    <Conditions>
        <Instrument>
            <Manufacturer>Vandelay Industries</Manufacturer>
            <Model>Model 400 FEG-ESEM</Model>
        </Instrument>
        <Probe Class="EM">
            <BeamVoltage Unit="kV">15.</BeamVoltage>
        </Probe>
        <Detector Class="XEDS">
            <Manufacturer>ARCN Detector Co.</Manufacturer>
            <Model>ARCN Model 1</Model>
            <MeasurementUnit>counts</MeasurementUnit>
            <Technology>SDD</Technology>
            <NominalThroughput Unit="kcounts/s">
                60.
            </NominalThroughput>
            <TimeConstant Unit="us">
                16.700001
            </TimeConstant>
            <StrobeRate Unit="Hz">1000.</StrobeRate>
            <Area Unit="mm2">10.</Area>
            <Elevation Unit="degrees">45.</Elevation>
        </Detector>
        <Acquisition>
            <DwellTime Unit="ms">272.</DwellTime>
        </Acquisition>
        <Calibration Class="LinearDispersion" ID="X">
            <Unit>um</Unit>
            <Gradient>3.36</Gradient>
        </Calibration>
        <Calibration Class="LinearDispersion" ID="Y">
            <Unit>um</Unit>
            <Gradient>3.36</Gradient>
        </Calibration>
        <Calibration Class="LinearDispersion" ID="XEDS calibration">
            <Quantity>Energy</Quantity>
            <Unit>eV</Unit>
            <Gradient>10.</Gradient>
            <Intercept>-475.</Intercept>
        </Calibration>
    </Conditions>
    <Dataset>
        <DataLength>419225600</DataLength>
        <DatumType>byte</DatumType>
        <Dimensions>
            <Channel ConditionID="XEDS calibration">
                2047
            </Channel>
            <X>512</X>
            <Y>400</Y>
        </Dimensions>
    </Dataset>
```

```
</MSAHyperDimensionalDataFile>
```

The same file, stripped of all conditions and header metadata, produces the following baseline file. Note that to use this file, the user shall manually keep track of EDS gain & offset, beam current & voltage, etc.

```
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="1801E95BD3570275"
xml:lang="en-US">
    <Header />
    <Conditions />
    <Dataset>
        <DataLength>419225600</DataLength>
        <DatumType>byte</DatumType>
        <Dimensions>
              <Channel>2047</Channel>
              <X>512</X>
              <Y>400</Y>
        </Dimensions>
    </Dataset>
</MSAHyperDimensionalDataFile>
```

D.7 EPMA+XEDS+CL+BSE map

This is an example dataset definition for a multi-dataset EPMA map with simultaneous data from two WDS monochromators, an XEDS spectrometer, a CL spectrometer, and a BSE detector.

```
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="6EDDBFC5A78F0940"
xml:lang="en-US">
    <Header>
        <Title>Multi-dataset example </Title>
        <Date>2016-09-29</Date>
        <Time>01:27</Time>
        <Timezone>UTC+10 AUS Eastern Standard Time</Timezone>
        <Author>
              Microbeam Laboratory Team; ARCN Mineralogy Labs Inc.
        </Author>
        <Owner>ARCN Mineralogy Labs Inc.</Owner>
    </Header>
    <Conditions>
        <Instrument>
              <Manufacturer>Vandelay Industries</Manufacturer>
              <Model>Model 5000 FEG-EPMA</Model>
        </Instrument>
        <Probe Class="EM">
              <BeamVoltage Unit="kV">10.</BeamVoltage>
              <BeamCurrent Unit="nA">15.</BeamCurrent>
              <ProbeDiameter Unit="um">1.</ProbeDiameter>
        </Probe>
        <Detector Class="XEDS" ID="XEDS detector">
              <Manufacturer>ARCN Detector Co.</Manufacturer>
              <Model>ARCN Model 1</Model>
              <SignalType>EDS</SignalType>
              <MeasurementUnit>counts</MeasurementUnit>
```

```
                <Technology>SDD</Technology>
                <NominalThroughput Unit="kcounts/s">
                        90.
                </NominalThroughput>
                <StrobeRate Unit="Hz">1000.</StrobeRate>
                <Area Unit="mm2">10.</Area>
                <Elevation Unit="degrees">45.</Elevation>
        </Detector>
        <Detector Class="CL" ID="CL detector">
                <SignalType>CLS</SignalType>
                <CollectionMode>Parallel</CollectionMode>
                <Manufacturer>ARCN Optical Instruments, GmbH</Manufacturer>
                <Model>MX800</Model>
                <DispersionElement>H50</DispersionElement>
                <Temperature Unit="degreesC">-10</Temperature>
                <EntranceSlit Unit="um">200.</EntranceSlit>
        </Detector>
        <Detector Class="WDS" ID="WDS ch1">
                <DetectorName>Ch1</DetectorName>
                <SignalType>WDS</SignalType>
                <DispersionElement>LDEB</DispersionElement>
                <WDSPosition Unit="mm">126</WDSPosition>
        </Detector>
        <ElementalID Class="X-ray" ID="B Ka">
                <Element>B</Element>
                <Line>Ka</Line>
        </ElementalID>
        <Detector Class="WDS" ID="WDS ch2">
                <DetectorName>Ch2</DetectorName>
                <SignalType>WDS</SignalType>
                <DispersionElement>TAP</DispersionElement>
                <WDSPosition Unit="mm">90.66</WDSPosition>
        </Detector>
        <ElementalID Class="X-ray" ID="Al Ka">
                <Element>Al</Element>
                <Line>Ka</Line>
        </ElementalID>
        <Detector ID="BSE detector">
                <SignalType>BEI</SignalType>
        </Detector>
        <Acquisition>
                <DwellTime Unit="ms">20</DwellTime>
        </Acquisition>
        <Calibration Class="LinearDispersion" ID="X">
                <Unit>um</Unit>
                <Gradient>1</Gradient>
        </Calibration>
        <Calibration Class="LinearDispersion" ID="Y">
                <Unit>um</Unit>
                <Gradient>1</Gradient>
        </Calibration>
        <Calibration Class="LinearDispersion" ID="XEDS calibration">
                <Quantity>Energy</Quantity>
                <Unit>eV</Unit>
```

```
        <Gradient>5.</Gradient>
        <Intercept>-480.</Intercept>
    </Calibration>
    <Calibration Class="PolynomialDispersion" ID="CL calibration">
        <Quantity>Wavelength</Quantity>
        <Unit>nm</Unit>
        <Coefficients ArrayType="float" Count="3">
            199.945602, 0.79385, -0.00003
        </Coefficients>
    </Calibration>
</Conditions>
<Dataset Name="XEDS">
    <DataLength>8589934592</DataLength>
    <DatumType>uint16</DatumType>
    <Dimensions>
        <Channel ConditionID="XEDS calibration">4096</Channel>
        <X>1024</X>
        <Y>1024</Y>
    </Dimensions>
    <IncludeConditions>
        <Detector>XEDS detector</Detector>
    </IncludeConditions>
</Dataset>
<Dataset Name="CL">
    <DataOffset>12884901896</DataOffset>
    <DataLength>2147483648</DataLength>
    <DatumType>uint16</DatumType>
    <Dimensions>
        <Channel ConditionID="CL calibration">1024</Channel>
        <X>1024</X>
        <Y>1024</Y>
    </Dimensions>
    <IncludeConditions>
        <Detector>CL detector</Detector>
    </IncludeConditions>
</Dataset>
<Dataset Name="WDS_ch1_LDEB">
    <DataOffset>15032385544</DataOffset>
    <DataLength>2097152</DataLength>
    <DatumType>uint16</DatumType>
    <Dimensions>
        <X>1024</X>
        <Y>1024</Y>
    </Dimensions>
    <IncludeConditions>
        <Detector>WDS ch1</Detector>
        <ElementalID>B Ka</ElementalID>
    </IncludeConditions>
</Dataset>
<Dataset Name="WDS_ch2_TAP">
    <DataOffset>15034482696</DataOffset>
    <DataLength>2097152</DataLength>
    <DatumType>uint16</DatumType>
    <Dimensions>
```

```
            <X>1024</X>
            <Y>1024</Y>
        </Dimensions>
        <IncludeConditions>
            <Detector>WDS ch2</Detector>
            <ElementalID>Al Ka</ElementalID>
        </IncludeConditions>
    </Dataset>
    <Dataset Name="BSE">
        <DataOffset>15035531272</DataOffset>
        <DataLength>1048576</DataLength>
        <DatumType>byte</DatumType>
        <Dimensions>
            <X>1024</X>
            <Y>1024</Y>
        </Dimensions>
        <IncludeConditions>
            <Detector>BSE detector</Detector>
        </IncludeConditions>
    </Dataset>
</MSAHyperDimensionalDataFile>
```

The same file, stripped of all conditions and header metadata, produces the following baseline file.

```
<?xml version="1.02" encoding="UTF-8" standalone="yes" ?>
<MSAHyperDimensionalDataFile Version="1.01" UID="6EDDBFC5A78F0940"
xml:lang="en-US">
    <Header />
    <Conditions />
    <Dataset Name="XEDS">
        <DataLength>8589934592</DataLength>
        <DatumType>uint16</DatumType>
        <Dimensions>
            <Channel>4096</Channel>
            <X>1024</X>
            <Y>1024</Y>
        </Dimensions>
    </Dataset>
    <Dataset Name="CL">
        <DataOffset>12884901896</DataOffset>
        <DataLength>2147483648</DataLength>
        <DatumType>uint16</DatumType>
        <Dimensions>
            <Channel>1024</Channel>
            <X>1024</X>
            <Y>1024</Y>
        </Dimensions>
    </Dataset>
    <Dataset Name="WDS_ch1_LDEB">
        <DataOffset>15032385544</DataOffset>
        <DataLength>2097152</DataLength>
        <DatumType>uint16</DatumType>
        <Dimensions>
            <X>1024</X>
            <Y>1024</Y>
```

```
            </Dimensions>
        </Dataset>
        <Dataset Name="WDS_ch2_TAP">
            <DataOffset>15034482696</DataOffset>
            <DataLength>2097152</DataLength>
            <DatumType>uint16</DatumType>
            <Dimensions>
                <X>1024</X>
                <Y>1024</Y>
            </Dimensions>
        </Dataset>
        <Dataset Name="BSE">
            <DataOffset>15035531272</DataOffset>
            <DataLength>1048576</DataLength>
            <DatumType>byte</DatumType>
            <Dimensions>
                <X>1024</X>
                <Y>1024</Y>
            </Dimensions>
        </Dataset>
</MSAHyperDimensionalDataFile>
```

<div align="center">

Annex E
(Informative)

Common dataset dimensions

</div>

E.1 General

This annex describes a set of common names for dataset dimensions, including examples of appropriate dimension calibration conditions for a range of experimental configurations (see 8.4.4: Identity and calibration of dimensions). The HMSA specification does not rigidly specify the names of dimensions, and implementations may use any dimension name that best describes their dataset (and is valid XML). However, in the interest of compatibility, implementers are strongly encouraged to select dimension names from the examples in this Annex where sensible to do so.

To suggest additional dataset dimension names, please refer to 3: Terms and definitions.

E.2 <x>, <y> and <z>

The `<X>`, `<Y>` and `<Z>` dimension names are principally used to describe three Cartesian dimensions relating to directions within the specimen. For a definition of the specimen coordinate system, please see the `<SpecimenPosition>` condition in Annex A.

If the dimensions do not precisely correspond with the axes of the specimen coordinate system, a modifier may be appended to the dimension name, such as `<X-Camera>`, `<ZSputterTime>`, etc. Likewise, modifiers should be used in datasets where there are multiple instances of dimensions, such as in an X/Y mosaic of X/Y images.

A calibration condition should be associated with each of the `<X>`, `<Y>`, and `<Z>` dimensions, as illustrated below:

```
<Calibration Class="LinearDispersion" ID="X">
   <Unit>um</Unit>
   <Gradient>0.05</Gradient>
</Calibration>
```

E.3 <u> and <v>

The `<U>` and `<V>` dimension names are reserved for use with two-dimensional diffraction dataset axes, such as with TEM or x-ray crystallography. A calibration condition should be associated with the `<U>` and `<V>` dimensions, such as below:

```
<Calibration Class="LinearDispersion" ID="U">
   <Quantity>Diffraction angle two-theta</Quantity>
   <Unit>degrees</Unit>
   <Intercept>10</Intercept>
   <Gradient>0.1</Gradient>
</Calibration>
```

The quantities and units of the associated calibration conditions are not restricted by this version of the specification but may for example include scattering angles (in units of "mrad" or "degrees"), distances in reciprocal space ("nm-1"), or simple CCD element coordinates (no units).

Diffraction data with a single dimension, such as from powder x-ray diffraction, should use the `<Channel>` or `<Position>` dimension names instead of `<U>` or `<V>`, for consistency with datasets from other multi-channel or moveable detectors.

E.4 `<Position>`

The `<Position>` dimension name is principally intended for use with spatial dimensions that are not related to the specimen (see `<X>`, `<Y>` and `<Z>` in this annex), such as the position of movable spectrometers, monochromators or diffractometers. Optionally, a modifier suffix may be provided to further specify the type of position movement represented by the dimension, such as `<Position-WDS>` for use with a wavelength dispersive spectrometer wavescan over a range of linear crystal positions *or* two-theta angles.

A calibration condition should be associated with a `<Position>` dimension, such as below:

```
<Calibration Class="LinearDispersion" ID="Position-WDS">
   <Quantity>WDS wavescan position</Quantity>
   <Unit>mm</Unit>
   <Intercept>138</Intercept>
   <Gradient>0.04</Gradient>
</Calibration>
```

Note that the specific physical quantity ("WDS wavescan position") and units ("mm") are provided in the `<Quantity>` element in the calibration condition, and not in the dimension name.

E.5 `<Channel>`

The `<Channel>` dimension name is principally intended for use with the data dimension of a multi-channel detector (e.g., EELS, XEDS, CL), or with other non-spatial dimensions. This dimension is deliberately more generic than names such as `<Energy>` or `<Wavelength>`, and allows spectrum and hyperspectral map data to be structured and interpreted in a common way.

A calibration condition should be associated with a `<Channel>` dimension, such as below:

```
<Calibration Class="LinearDispersion" ID="Channel">
   <Quantity>Photoelectron binding energy</Quantity>
   <Unit>eV</Unit>
   <Intercept>5</Intercept>
   <Gradient>0.1</Gradient>
</Calibration>
```

Note that the specific physical quantity ("Photoelectron binding energy") and units ("eV") are provided in the calibration condition, and not in the dimension name.

E.6 `<Color>`

The `<Color>` dimension name is principally used to describe the color components of a color image. Typically, a `<Color>` dimension will have a size of three, for the red, green and blue components, as shown in the following dataset dimensions list for a 1 024 x 768 pixel color image:

```
<Dimensions>
   <Color>3</Color>
   <X>1024</X>
   <Y>768</Y>
</Dimensions>
```

If the `<Color>` dimension has a size of three, and if no calibration condition is associated with the dimension, then the three color components of the dimension shall represent red, green and blue, in that order. Alternative color orders of red, green, and blue colors may be specified by attaching a modifier suffix to the dimension name, such as `<Color-BGR>` for colors ordered in blue, green, then red. The dimension name `<Color-RGB>` is equivalent to `<Color>`.

For `<Color>` dimensions with a size other than three, or with colors other than red, green, and blue, the identity of the colors should be defined using the `<Labels>` element of a `<Calibration>` condition, like so:

```
<Calibration ID="Color">
   <Quantity>Color</Quantity>
   <Labels Count="3">Yellow, Magenta, Cyan</Labels>
</Calibration>
```

If the color components correspond to specific wavelengths or energies of light, these values may also be defined in a `<Calibration Class="Explicit">` condition, like so:

```
<Calibration Class="Explicit" ID="Color">
   <Quantity>Color wavelength</Quantity>
   <Unit>nm</nm>
   <Values ArrayType="float" Count="3">
      480, 555, 640
   </Values>
</Calibration>
```

For multi-channel color images with larger numbers of color values, such as hyperspectral luminescence maps, the color dimension should instead be labeled `<Channel>`, for consistency with other hyperspectral datasets.

E.7 `<Rotation>` and `<Tilt>`

The `<Rotation>` and `<Tilt>` dimension names are intended to describe dataset dimension where the varied parameter was the physical rotation of the specimen (as in tomography), or of another component of the instrument.

A calibration condition should be associated with each `<Rotation>` or `<Tilt>` dimension, such as below:

```
<Calibration Class="LinearDispersion" ID="Tilt">
   <Quantity>Specimen tilt</Quantity>
   <Unit>degrees</Unit>
   <Intercept>-20</Intercept>
   <Gradient>.5</Gradient>
</Calibration>
```

By convention, the `<Tilt>` name is reserved for rotations around axes that are orthogonal to the Z-axis of the defined coordinate system. Rotations around other arbitrary axes should use the `<Rotation>`

dimension name. If multiple tilt or rotation dimensions are present in a dataset, the dimension names shall be disambiguated by appending a modifier such as `<Tilt-A>` and `<Tilt-B>`.

E.8 `<Focus>`

The `<Focus>` dimension name is principally intended for use with dimensions where the varied parameter was the focal plane of an optical component, such as a through-focus image series in phase contrast imaging. Optionally, a modifier suffix may be provided to identify which optical component was varied, such as `<Focus-Objective>` or `<Focus-Condenser>`.

The `<Focus>` dimension name should not be used where the shift in the focal plane directly relates to a change in the depth of the specimen sampled in the Z axis. In such instances, a dimension name of `<Z>` or `<Z-focus>` should be used.

A calibration condition should be associated with a `<Focus>` dimension, such as below:

```
<Calibration Class="LinearDispersion" ID="Focus-Objective">
   <Quantity>Objective lens current</Quantity>
   <Unit>A</Unit>
   <Intercept>2.2</Intercept>
   <Gradient>0.01</Gradient>
</Calibration>
```

Note that the specific physical quantity ("Objective lens current") and units ("A") are provided in the calibration condition, and not in the dimension name.

E.9 `<Measurement>`

The `<Measurement>` dimension name is intended for use with dataset dimensions where multiple parameters were varied from measurement to measurement, such as in an irregular sequence of spectra where the position and time of acquisition varied between measurements.

A `<Measurement>` should be associated with a `<Sequence>` condition to describe the conditions that vary between each measurement ordinal. The associated `<Sequence>` condition, if provided, shall contain only one type of condition object (e.g., `<Acquisition>`), and the number of objects within the `<Sequence>` condition shall be equal to the size of the `<Measurement>` dimension. Multiple parameters within each condition object may vary from measurement to measurement, such as `<DateTime>` and `<SpecimenPosition>` within `<Acquisition>`.

A dataset may not contain more than one `<Measurement>` dimension. Datasets with more than one irregularly varying dimension shall instead be structured as multiple datasets in either the same HMSA/XML file pair, or in multiple HMSA/XML files.