SDMX Standards: Section 3A PaRT IV

SDMX-ML:

Schema and Documentation

Data and Metadata

Namespaces

(Version 2.1)

April 2011

Contents

[1 Introduction 1](#_Toc290388057)

[2 Schema Documentation 2](#_Toc290388058)

[2.1 Generic Data Namespace 2](#_Toc290388059)

[2.1.1 Summary 2](#_Toc290388060)

[2.1.2 Complex Types 2](#_Toc290388061)

[2.2 Structure Specific Data Namespace 16](#_Toc290388062)

[2.2.1 Summary 16](#_Toc290388063)

[2.2.2 Complex Types 16](#_Toc290388064)

[2.2.3 Simple Types 35](#_Toc290388065)

[2.3 Generic Metadata Namespace 35](#_Toc290388066)

[2.3.1 Summary 35](#_Toc290388067)

[2.3.2 Complex Types 36](#_Toc290388068)

[2.4 Structure Specific Metadata Namespace 42](#_Toc290388069)

[2.4.1 Summary 42](#_Toc290388070)

[2.4.2 Complex Types 43](#_Toc290388071)

[3 Mapping to Structure-Specific Schemas 52](#_Toc290388072)

[3.1 General 52](#_Toc290388073)

[3.1.1 Basic Terminology 52](#_Toc290388074)

[3.2 Namespace Rules 54](#_Toc290388075)

[3.3 General Rules 54](#_Toc290388076)

[3.3.1 Component Name Determination 54](#_Toc290388077)

[3.3.2 Representation Determination 54](#_Toc290388078)

[3.3.3 Simple / Primitive Type Determination 55](#_Toc290388079)

[3.3.4 Representation with Enumeration 55](#_Toc290388080)

[3.3.5 Representation with Text Format 55](#_Toc290388081)

[3.3.6 *Type Names* 58](#_Toc290388082)

[3.3.7 Type Reuse 58](#_Toc290388083)

[3.4 Data Structure Specific Schema 59](#_Toc290388084)

[3.4.1 DataSetType 59](#_Toc290388085)

[3.4.2 GroupType 60](#_Toc290388086)

[3.4.3 SeriesType 61](#_Toc290388087)

[3.4.4 ObsType 61](#_Toc290388088)

[3.5 Metadata Structure Specific Schema 63](#_Toc290388089)

[3.5.1 MetadataSetType 63](#_Toc290388090)

[3.5.2 TargetType 63](#_Toc290388091)

[3.5.3 ReportType 66](#_Toc290388092)

[4 Special data functions 68](#_Toc290388093)

[4.1 Updates 68](#_Toc290388094)

[4.2 Deletes 68](#_Toc290388095)

# Introduction

The first change in the data and metadata message is one of terminology. In order to foster consistency in the standard, the names and namespaces of the data and metadata message have been changed. The namespaces now have a uniform format of /data/format and /metadata/format, where format is either generic or structured (i.e. the structure specific formats). This also applies to the message names as well, where the names follow the pattern of FormatData (e.g. StructuredData and GenericMetadata).

The various data messages which existed in the standard have been harmonised into two general formats; generic and structured data. The generic data is much like it has been previously, only slightly modified to conform with the harmonised view on data. The structured data combined the principles of the previous versions compact and cross sectional formats into one more generalised format.

The major shift from the previous version is now that all data can be exchanged as either an un-grouped collection of observations, each specifying a full key or it can be exchanged as data grouped into series with any single dimension placed at the observation level. This in effect, combined the abilities of the time series and cross sectional formats in the a single data format. This format also allows for multiple measure to be expressed when the observation dimension is the measure dimension defined in the data structured. The key differences from the previous versions is that this all is possible in a single data structure definition specific format, as well as being possible in the generic format as well. The goal was to make the structures as homogenous as possible.

Another shift from the previous version in the manner in which the base data structure specific format is defined. In previous versions there was not structure defined, so one had to understand all of the requirements and read the specification in order to understand what could be expected in any structure specific format. In this version, that issue has been addressed. The base structures now impose a strict format on the data structure specific schemas. This is achieved much like the referencing structure in the common namespace through the use of unqualified elements. By the elements not existing in a namespace, the structure specific schemas can place the necessary restrictions of them while still being forced to adhere to prescribed structure. This means that not only are the structures able to perform the validation that is required, but that the messages will be much simpler to process as the format will always use the same element names.

Finally, in order to allow for systems which wish to not process the new more flexible data format, time series only variations of both the generic and structured data sets exist. It is important to note that these structures are derived via restriction from the more generalised format. This means that a data set in the general format with the time dimension at the observation level will have exactly the same content as a time series only data set. The result of this is that there is no additional burden for processing the time series specific format it a system can process the more generalised format. These time series only data sets allow for time series only data messages to exist. These messages make it simple for system which, in the previous version of the standard, only used the time series formats to continue to do so.

These same principles have been applied to the reference data messages as well. In the previous version of the standard there were major differences between the generic and metadata structure-specific formats; some of the differences caused some metadata sets were incompatible between versions. In this version the structures of the generic and metadata structure specific formats have been harmonised to the point where they are nearly identical. Yet, the structured format still provides the strong validation against the metadata structure that is intended. And, as with the data, the base metadata structure specific format now imposes a stricter structure on the generated schemas, making the structure specific instance simpler to process in a generic manner.

# Schema Documentation

## Generic Data Namespace

**http://www.sdmx.org/resources/sdmxml/schemas/v2\_1/data/generic**

### Summary

Referenced Namespaces:

| **Namespace** | **Prefix** |
| --- | --- |
| http://www.sdmx.org/resources/sdmxml/schemas/v2\_1/common | com |
| http://www.w3.org/2001/XMLSchema | xs |

Contents:

13 Complex Types

### Complex Types

**DataSetType:**DataSetType defines the structure of the generic data set. Data is organised into either a collection of series (grouped observations) or a collection of un-grouped observations. The organisation used is dependent on the structure specification in the header of the data message (which is referenced via the structureRef attribute). The structure specification states which data occurs at the observation level. If this dimension is "AllDimensions" then the data set must consist of a collection of un-grouped observations; otherwise the data set will contain a collection of series with the observations in the series disambiguated by the specified dimension at the observation level. This data set is capable of containing data (observed values) and/or documentation (attribute values). It is assumed that each series or un-grouped observation will be distinct in its purpose. For example, if series contains both data and documentation, it assumed that each series will have a unique key. If the series contains only data or only documentation, then it is possible that another series with the same key might exist, but with not with the same purpose (i.e. to provide data or documentation) as the first series.

Derivation:

*com:AnnotableType* (extension)   
   DataSetType



Attributes:

structureRef, setID?, action?, reportingBeginDate?, reportingEndDate?, validFromDate?, validToDate?, publicationYear?, publicationPeriod?

Content:

com:Annotations?, DataProvider?, Attributes?, Group\*, (Series+ | Obs+)?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureRef | xs:IDREF | The structureRef contains a reference to a structural specification in the header of a data or reference metadata message. The structural specification details which structure the data or reference metadata conforms to, as well as providing additional information such as how the data is structure (e.g. which dimension occurs at the observation level for a data set). |
| setID | com:IDType | The setID provides an identification of the data or metadata set. |
| action | com:ActionType | The action attribute indicates whether the file is appending, replacing, or deleting. |
| reportingBeginDate | com: BasicTimePeriodType | The reportingBeginDate indicates the inclusive start time of the data reported in the data or metadata set. |
| reportingEndDate | com: BasicTimePeriodType | The reportingEndDate indicates the inclusive end time of the data reported in the data or metadata set. |
| validFromDate | xs:dateTime | The validFromDate indicates the inclusive start time indicating the validity of the information in the data or metadata set. |
| validToDate | xs:dateTime | The validToDate indicates the inclusive end time indicating the validity of the information in the data or metadata set. |
| publicationYear | xs:gYear | The publicationYear holds the ISO 8601 four-digit year. |
| publicationPeriod | com: ObservationalTimePer iodType | The publicationPeriod specifies the period of publication of the data or metadata in terms of whatever provisioning agreements might be in force (i.e., "Q1 2005" if that is the time of publication for a data set published on a quarterly basis). |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| DataProvider | com: DataProviderReferenc eType | DataProvider contains a reference to the provider for the data set. |
| Attributes | ValuesType | Attributes contains the collection of attribute values for attributes defined in the data structure definition which do not have an attribute relationship with any other data structure definition components. |
| Group | GroupType | Group contains a references to a defined group in the data structure definition along with its key (if necessary) and values for the attributes which are associated with the group. An attribute is associated to a group by either an explicit group relationship or by a group attachment when the attribute has a relationship with a dimension which is a member of this group. |
| Series | SeriesType | Series contains a collection of observations that share a common key (set of dimension values). The key of a series is every dimension defined in the data structure definition, save the dimension which is declared in the structure specification to be at the observation level. In addition to the key and observations, the series contains values for attributes which have a relationship with any dimension that is part of the series key, so long as the attribute does not specify an attachment group or also has a relationship with the dimension declared to be at the observation level. |
| Obs | ObsOnlyType | Obs is an un-grouped observation. This observation has a key which is a set of values for all dimensions declared in the data structure definition. In addition to the key, the value of the observation can be provided along with values for all attributes which have an association with the primary measure or any dimension (so long as it does not specify a group attachment). |

**GroupType:**GroupType defines a structure which is used to communicate attribute values for a group defined in a data structure definition. The group can consist of either a subset of the dimensions defined by the data structure definition, or an association to an attachment constraint, which in turn defines key sets to which attributes can be attached. In the case that the group is based on an attachment constraint, only the identification of group is provided. It is expected that a system which is processing this will relate that identifier to the key sets defined in the constraint and apply the values provided for the attributes appropriately.

Derivation:

*com:AnnotableType* (extension)   
   GroupType



Attributes:

type

Content:

com:Annotations?, GroupKey?, Attributes

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type | com:IDType | The type attribute holds the identifier assigned to the group in the data structure definition for which attribute values are being provided. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| GroupKey | ValuesType | GroupKey contains the values for each dimension defined as being a member of the referenced group in the data structure definition. A value must be provided for every group dimension. This is optional, and not used if the group is defined as an association to an attachment constraint rather than a subset of the data structure definition dimensions. |
| Attributes | ValuesType | Attributes contains the set of attribute values which are reported for group. The attribute values provided here apply to all data matching the partial key defined by the group key or the key sets of the associated attachment constraint. |

**SeriesType:**SeriesType defines a structure which is used to group a collection of observations which have a key in common. The key for a series is every dimension defined in the data structure definition, save the dimension declared to be at the observation level for this data set. In addition to observations, values can be provided for attributes which are associated with the dimensions which make up this series key (so long as the attributes do not specify a group attachment or also have an relationship with the observation dimension). It is possible for the series to contain only observations or only attribute values, or both.

Derivation:

*com:AnnotableType* (extension)   
   SeriesType



Content:

com:Annotations?, SeriesKey, Attributes?, Obs\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| SeriesKey | ValuesType | SeriesKey contains the values for each all dimensions defined in the data structure definition, except for that which is declared to be at the observation level for this data set. This key is required, and every dimension must be provided a value. |
| Attributes | ValuesType | Attributes contains the values for attributes which are associated with the dimensions which make up the series key, so long as the attributes do not also specify an attachment group or have a relationship with the observation dimension. |
| Obs | ObsType | Obs contains an observation which shares the dimensionality of the series key. These observations are disambiguated from one another within this series by a single dimension value provided for each dimension. The dimension which provides this observation key is declared in the structure specification for the data set, which is provided in the header of the data message. |

**ObsOnlyType:**ObsOnlyType defines the structure for an un-grouped observation. Unlike a group observation, an un-grouped must provided a full set of values for every dimension declared in the data structure definition. The observation can contain an observed value and/or a collection of attribute values.

Derivation:

*com:AnnotableType* (extension)   
   ObsOnlyType



Content:

com:Annotations?, ObsKey, ObsValue?, Attributes?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| ObsKey | ValuesType | ObsKey contains the values for every dimension defined in the data structure definition. A value must be provided for each dimension. This key serves to disambiguate the un-grouped observation within the data set. |
| ObsValue | ObsValueType | ObsValue type contains the value for the observation. |
| Attributes | ValuesType | Attributes contains the set of values reported for the attributes which have an association with the primary measure or any dimension in the data structure definition (so long as an attachment group is not also specified). |

**ObsType:**ObsType defines the structure of a grouped observation. The observation must be provided a value for the dimension which is declared to be at the observation level for this data set. This dimension value should disambiguate the observation within the series in which it is defined (i.e. there should not be another observation with the same dimension value). The observation can contain an observed value and/or attribute values.

Derivation:

*com:AnnotableType* (extension)   
   ObsType



Content:

com:Annotations?, ObsDimension, ObsValue?, Attributes?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| ObsDimension | BaseValueType | ObsDimension holds the key for the grouped observation, i.e. the value of the observation dimension. Note that in this element, the reference to the dimension is optional, since it can be inferred from the structure specification for the data set. This saves having to repeat the value unnecessarily. It is assumed that any application processing the data set will have the information from the structure specification available, so that if a dimension identifier is not supplied here, the proper reference can be applied. |
| ObsValue | ObsValueType | ObsValue type contains the value for the observation. |
| Attributes | ValuesType | Attributes contains the set of values reported for the attributes which have an association with the primary measure or the observations dimension (so long as an attachment group is not also specified). |

**ValuesType:**ValuesType is a general structure which contains a collection of data structure definition component values. This type is used to provide both key and attribute collection values.

Content:

Value+

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Value | ComponentValueType | Value contains a component value and a reference to the component for which a value is being provided. |

**BaseValueType:**BaseValueType is a general structure which contains a reference to a data structure definition component and a value for that component. In this structure the reference to the component is optional to allow for usages where the actual reference might be provided in another context.

Attributes:

id?, value

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | com:NCNameIDType | The id attribute contains the identifier for the component for which a value is being provided. |
| value | xs:anySimpleType | The value attribute contains the provided component value. |

**ObsValueType:**ObsValueType is a derivation of the BaseValueType which is used to provide an observation value. Since an observation value is always associated with the data structure definition primary measure, and the identifier for the primary measure is fixed, the component reference for this structure is fixed. Note that this means that it is not necessary to provide a value in an instance as the fixed value will be provided in the post validation information set.

Derivation:

BaseValueType (restriction)   
   ObsValueType



Attributes:

id?, value

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id (fixed: OBS\_VALUE) | com:NCNameIDType | The id attribute contains a fixed reference to the primary measure component of the data structure definition. |
| value | xs:string | The value attribute contains the provided component value. |

**ComponentValueType:**ComponentValueType is a derivation of the BaseValueType which requires that the component reference be provided. This is used when the identification of the component cannot be inferred from another context.

Derivation:

BaseValueType (restriction)   
   ComponentValueType



Attributes:

id, value

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | com:NCNameIDType | The id attribute contains the identifier for the component for which a value is being provided. |
| value | xs:string | The value attribute contains the provided component value. |

**TimeSeriesDataSetType:**TimeSeriesDataSetType is a derivation of the base DataSetType of the generic format the restricts the data set to only allow for grouped observations where the dimension at the observation level is the time dimension of the data structure definition. This means that unlike the base data set structure, there can be no un-grouped observations. Because this derivation is achieved using restriction, data sets conforming to this type will inherently conform to the base data set structure as well. In fact, data structured here will be identical to data in the base data set when the time dimension is the observation dimension. This means that the data contained in this structure can be processed in exactly the same manner as the base structure.

Derivation:

*com:AnnotableType* (extension)   
   DataSetType (restriction)   
         TimeSeriesDataSetType



Attributes:

structureRef, setID?, action?, reportingBeginDate?, reportingEndDate?, validFromDate?, validToDate?, publicationYear?, publicationPeriod?

Content:

com:Annotations?, DataProvider?, Attributes?, Group\*, Series\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureRef | xs:IDREF | The structureRef contains a reference to a structural specification in the header of a data or reference metadata message. The structural specification details which structure the data or reference metadata conforms to, as well as providing additional information such as how the data is structure (e.g. which dimension occurs at the observation level for a data set). |
| setID | com:IDType | The setID provides an identification of the data or metadata set. |
| action | com:ActionType | The action attribute indicates whether the file is appending, replacing, or deleting. |
| reportingBeginDate | com: BasicTimePeriodType | The reportingBeginDate indicates the inclusive start time of the data reported in the data or metadata set. |
| reportingEndDate | com: BasicTimePeriodType | The reportingEndDate indicates the inclusive end time of the data reported in the data or metadata set. |
| validFromDate | xs:dateTime | The validFromDate indicates the inclusive start time indicating the validity of the information in the data or metadata set. |
| validToDate | xs:dateTime | The validToDate indicates the inclusive end time indicating the validity of the information in the data or metadata set. |
| publicationYear | xs:gYear | The publicationYear holds the ISO 8601 four-digit year. |
| publicationPeriod | com: ObservationalTimePer iodType | The publicationPeriod specifies the period of publication of the data or metadata in terms of whatever provisioning agreements might be in force (i.e., "Q1 2005" if that is the time of publication for a data set published on a quarterly basis). |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| DataProvider | com: DataProviderReferenc eType | DataProvider contains a reference to the provider for the data set. |
| Attributes | ValuesType | Attributes contains the collection of attribute values for attributes defined in the data structure definition which do not have an attribute relationship with any other data structure definition components. |
| Group | GroupType | Group contains a references to a defined group in the data structure definition along with its key (if necessary) and values for the attributes which are associated with the group. An attribute is associated to a group by either an explicit group relationship or by a group attachment when the attribute has a relationship with a dimension which is a member of this group. |
| Series | TimeSeriesType | Series contains a collection of observations that share a common key (set of dimension values). The key of a series is every dimension defined in the data structure definition, save the time dimension. In addition to the key and observations, the series contains values for attributes which have a relationship with any dimension that is part of the series key, so long as the attribute does not specify an attachment group or also has a relationship with the time dimension. |

**TimeSeriesType:**TimeSeriesType defines a structure which is used to group a collection of observations which have a key in common, organised by time. The key for a series is every dimension defined in the data structure definition, save the time dimension. In addition to observations, values can be provided for attributes which are associated with the dimensions which make up this series key (so long as the attributes do not specify a group attachment or also have an relationship with the time dimension). It is possible for the series to contain only observations or only attribute values, or both.

Derivation:

*com:AnnotableType* (extension)   
   SeriesType (restriction)   
         TimeSeriesType



Content:

com:Annotations?, SeriesKey, Attributes?, Obs\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| SeriesKey | ValuesType | SeriesKey contains the values for each all dimensions defined in the data structure definition, except for that which is declared to be at the observation level for this data set. This key is required, and every dimension must be provided a value. |
| Attributes | ValuesType | Attributes contains the values for attributes which are associated with the dimensions which make up the series key, so long as the attributes do not also specify an attachment group or have a relationship with the observation dimension. |
| Obs | TimeSeriesObsType | Obs contains an observation which shares the dimensionality of the series key. These observations are disambiguated from one another within this series by a time value. |

**TimeSeriesObsType:**TimeSeriesObsType defines the structure of a time series observation. The observation must be provided a value for the time dimension. This time value should disambiguate the observation within the series in which it is defined (i.e. there should not be another observation with the same time value). The observation can contain an observed value and/or attribute values.

Derivation:

*com:AnnotableType* (extension)   
   ObsType (restriction)   
         TimeSeriesObsType



Content:

com:Annotations?, ObsDimension, ObsValue?, Attributes?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| ObsDimension | TimeValueType | ObsDimension holds the key for the grouped observation, i.e. the value of the observation dimension. Note that in this element, the reference to the dimension is optional, since it can be inferred from the structure specification for the data set. This saves having to repeat the value unnecessarily. It is assumed that any application processing the data set will have the information from the structure specification available, so that if a dimension identifier is not supplied here, the proper reference can be applied. |
| ObsValue | ObsValueType | ObsValue type contains the value for the observation. |
| Attributes | ValuesType | Attributes contains the set of values reported for the attributes which have an association with the primary measure or the time dimension (so long as an attachment group is not also specified). |

**TimeValueType:**TimeValueType is a derivation of the BaseValueType which is used to provide a value for the time dimension. Since the identifier for the time dimension is fixed, the component reference for this structure is fixed. Note that this means that it is not necessary to provide a value in an instance as the fixed value will be provided in the post validation information set.

Derivation:

BaseValueType (restriction)   
   TimeValueType



Attributes:

id?, value

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id (fixed: TIME\_PERIOD) | com:NCNameIDType | The id attribute contains the identifier for the component for which a value is being provided. |
| value | com: ObservationalTimePer iodType | The value attribute contains the provided component value. |

## Structure Specific Data Namespace

**http://www.sdmx.org/resources/sdmxml/schemas/v2\_1/data/structurespecific**

### Summary

Referenced Namespaces:

| **Namespace** | **Prefix** |
| --- | --- |
|  |  |
| http://www.sdmx.org/resources/sdmxml/schemas/v2\_1/common | com |
| http://www.w3.org/2001/XMLSchema | xs |

Contents:

7 Complex Types  
1 Simple Type

### Complex Types

***DataSetType*:**DataSetType is the abstract type which defines the base structure for any data structure definition specific data set. A derived data set type will be created that is specific to a data structure definition and the details of the organisation of the data (i.e. which dimension is the observation dimension and whether explicit measures should be used). Data is organised into either a collection of series (grouped observations) or a collection of un-grouped observations. The derived data set type will restrict this choice to be either grouped or un-grouped observations. If this dimension is "AllDimensions" then the derived data set type must consist of a collection of un-grouped observations; otherwise the data set will contain a collection of series with the observations in the series disambiguated by the specified dimension at the observation level. This data set is capable of containing data (observed values) and/or documentation (attribute values) and can be used for incremental updates and deletions (i.e. only the relevant updates or deletes are exchanged). It is assumed that each series or un-grouped observation will be distinct in its purpose. For example, if series contains both data and documentation, it assumed that each series will have a unique key. If the series contains only data or only documentation, then it is possible that another series with the same key might exist, but with not with the same purpose (i.e. to provide data or documentation) as the first series. This base type is designed such that derived types can be processed in a generic manner; it assures that data structure definition specific data will have a consistent structure. The group, series, and observation elements are unqualified, meaning that they are not qualified with a namespace in an instance. This means that in the derived data set types, the elements will always be the same, regardless of the target namespace of the schemas which defines these derived types. This allows for consistent processing of the structure without regard to what the namespace might be for the data structure definition specific schema. The data set can contain values for attributes which do not have an attribute relationship with any data structure definition components. These attribute values will exist in XML attributes in this element based on this type (DataSet). This is specified in the content model with the declaration of anyAttributes in the "local" namespace. The derived data set type will refine this structure so that the attributes are explicit. The XML attributes will be given a name based on the attribute's identifier. These XML attributes will be unqualified (meaning they do not have a namespace associated with them). To allow for generic processing, it is required that the only unqualified XML attributes in the derived data set type (outside of the standard data set attributes) be for attributes declared in the data structure definition. If additional attributes are required, these should be qualified with a namespace so that a generic application can easily distinguish them as not being meant to represent a data structure definition attribute.

Derivation:

*com:AnnotableType* (extension)   
   *DataSetType*



Attributes:

structureRef, setID?, action?, reportingBeginDate?, reportingEndDate?, validFromDate?, validToDate?, publicationYear?, publicationPeriod?, dataScope, REPORTING\_YEAR\_START\_DAY?

Content:

com:Annotations?, DataProvider?, Group\*, (Series+ | Obs+)?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureRef | xs:IDREF | The structureRef contains a reference to a structural specification in the header of a data or reference metadata message. The structural specification details which structure the data or reference metadata conforms to, as well as providing additional information such as how the data is structure (e.g. which dimension occurs at the observation level for a data set). |
| setID | com:IDType | The setID provides an identification of the data or metadata set. |
| action | com:ActionType | The action attribute indicates whether the file is appending, replacing, or deleting. |
| reportingBeginDate | com: BasicTimePeriodType | The reportingBeginDate indicates the inclusive start time of the data reported in the data or metadata set. |
| reportingEndDate | com: BasicTimePeriodType | The reportingEndDate indicates the inclusive end time of the data reported in the data or metadata set. |
| validFromDate | xs:dateTime | The validFromDate indicates the inclusive start time indicating the validity of the information in the data or metadata set. |
| validToDate | xs:dateTime | The validToDate indicates the inclusive end time indicating the validity of the information in the data or metadata set. |
| publicationYear | xs:gYear | The publicationYear holds the ISO 8601 four-digit year. |
| publicationPeriod | com: ObservationalTimePer iodType | The publicationPeriod specifies the period of publication of the data or metadata in terms of whatever provisioning agreements might be in force (i.e., "Q1 2005" if that is the time of publication for a data set published on a quarterly basis). |
| dataScope | DataScopeType | The dataScope attribute indicates the scope at which the data is meant to be validated. These scopes are hierarchical and are (from the top down); DataStructure, ConstrainedDataStructure, Dataflow, and ProvisionAgreement. the hierarchy of these scopes represent the cascading level of constraints, which can restrict the valid values for components. For example, a data structure defines a dimension with a coded representation. A data flow might have a constraint associated with it which further restricts the values allowed from the referenced code list to a subset of the values allowed by the data structure definition. A provision agreement that is based on the dataflow might also have a constraint, which further restricts the subset of the codelist from the dataflow. Therefore, the allowed content becomes stricter lower in the hierarchy. Data that is given a scope of one value is stated to be valid at that level and all levels below it. Therefore, this scope serves to state that data that is meant to be structured simply against the data structure definition is not meant to be validated against the a dataflow, where constraints might be applied. |
| REPORTING\_YEAR\_START\_DAY | xs:gMonthDay | The REPORTING\_YEAR\_START\_DAY attribute is an explict attribute for the reporting year start day, which provides context to the time dimension when its value contains a reporting period (e.g. 2010-Q1). This attribute is used to state the month and day that the reporting year begins (e.g. --07-01 for July 1st). In the absence of an explicit value provided in this attribute, all reporting period values will be assumed to be based on a reporting year start day of January 1. This is declared in the base schema since it has a fixed identifier and representation. The derived data set type may either require or prohibit this attribute, depending on whether the data structure declared the reporting year start day attribute and if so, the attribute relationship and assignment status assigned to it. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| DataProvider | com: DataProviderReferenc eType | DataProvider contains a reference to the provider for the data set. |
| Group | *GroupType* | Group contains a references to a defined group in the data structure definition along with its key (if necessary) and values for the attributes which are associated with the group. An attribute is associated to a group by either an explicit group relationship or by a group attachment when the attribute has a relationship with a dimension which is a member of this group. |
| Series | *SeriesType* | Series contains a collection of observations that share a common key (set of dimension values). The key of a series is every dimension defined in the data structure definition, save the dimension at the observation level. In addition to the key and observations, the series contains values for attributes which have a relationship with any dimension that is part of the series key, so long as the attribute does not specify an attachment group or also has a relationship with the dimension declared to be at the observation level. |
| Obs | *ObsType* | Obs is an un-grouped observation. This observation has a key which is a set of values for all dimensions declared in the data structure definition. In addition to the key, the value of the observation can be provided along with values for all attributes which have an association with the primary measure or any dimension (so long as it does not specify a group attachment). |

***GroupType*:**GroupType is the abstract type which defines a structure which is used to communicate attribute values for a group defined in a data structure definition. The group can consist of either a subset of the dimensions defined by the data structure definition, or an association to an attachment constraint, which in turn defines key sets to which attributes can be attached. In the case that the group is based on an attachment constraint, only the identification of group is provided. It is expected that a system which is processing this will relate that identifier to the key sets defined in the constraint and apply the values provided for the attributes appropriately. Data structure definition schemas will drive types based on this for each group defined in the data structure definition. Both the dimension values which make up the key (if applicable) and the attribute values associated with the group will be represented with XML attributes. This is specified in the content model with the declaration of anyAttributes in the "local" namespace. The derived group type will refine this structure so that the attributes are explicit. The XML attributes will be given a name based on the attribute's identifier. These XML attributes will be unqualified (meaning they do not have a namespace associated with them). The dimension XML attributes will be required while the attribute XML attributes will be optional. To allow for generic processing, it is required that the only unqualified XML attributes in the derived group type be for the group dimensions and attributes declared in the data structure definition. If additional attributes are required, these should be qualified with a namespace so that a generic application can easily distinguish them as not being meant to represent a data structure definition dimension or attribute.

Derivation:

*com:AnnotableType* (extension)   
   *GroupType*



Attributes:

type?, REPORTING\_YEAR\_START\_DAY?

Content:

com:Annotations?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type | com:IDType | The type attribute reference the identifier of the group as defined in the data structure definition. This is optional, but derived group types will provide a fixed value for this so that it always available in the post validation information set. This allows the group to be processed in a generic manner. |
| REPORTING\_YEAR\_START\_DAY | xs:gMonthDay | The REPORTING\_YEAR\_START\_DAY attribute is an explict attribute for the reporting year start day, which provides context to the time dimension when its value contains a reporting period (e.g. 2010-Q1). This attribute is used to state the month and day that the reporting year begins (e.g. --07-01 for July 1st). In the absence of an explicit value provided in this attribute, all reporting period values will be assumed to be based on a reporting year start day of January 1. This is declared in the base schema since it has a fixed identifier and representation. The derived group types may either require or prohibit this attribute, depending on whether the data structure declared the reporting year start day attribute and if so, the attribute relationship and assignment status assigned to it. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |

***SeriesType*:**SeriesType is the abstract type which defines a structure which is used to group a collection of observations which have a key in common. The key for a series is every dimension defined in the data structure definition, save the dimension declared to be at the observation level for this data set. In addition to observations, values can be provided for attributes which are associated with the dimensions which make up this series key (so long as the attributes do not specify a group attachment or also have an relationship with the observation dimension). It is possible for the series to contain only observations or only attribute values, or both. Data structure definition schemas will drive a type based on this that is specific to the data structure definition and the variation of the format being expressed in the schema. Both the dimension values which make up the key and the attribute values associated with the key dimensions will be represented with XML attributes. This is specified in the content model with the declaration of anyAttributes in the "local" namespace. The derived series type will refine this structure so that the attributes are explicit. The XML attributes will be given a name based on the attribute's identifier. These XML attributes will be unqualified (meaning they do not have a namespace associated with them). The dimension XML attributes will be required while the attribute XML attributes will be optional. To allow for generic processing, it is required that the only unqualified XML attributes in the derived group type be for the series dimensions and attributes declared in the data structure definition. If additional attributes are required, these should be qualified with a namespace so that a generic application can easily distinguish them as not being meant to represent a data structure definition dimension or attribute.

Derivation:

*com:AnnotableType* (extension)   
   *SeriesType*



Attributes:

TIME\_PERIOD?, REPORTING\_YEAR\_START\_DAY?

Content:

com:Annotations?, Obs\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| TIME\_PERIOD | com: ObservationalTimePer iodType | The TIME\_PERIOD attribute is an explict attribute for the time dimension. This is declared in the base schema since it has a fixed identifier and representation. The derived series type will either require or prohibit this attribute, depending on whether time is the observation dimension. If the time dimension specifies a more specific representation of time the derived type will restrict the type definition to the appropriate type. |
| REPORTING\_YEAR\_START\_DAY | xs:gMonthDay | The REPORTING\_YEAR\_START\_DAY attribute is an explict attribute for the reporting year start day, which provides context to the time dimension when its value contains a reporting period (e.g. 2010-Q1). This attribute is used to state the month and day that the reporting year begins (e.g. --07-01 for July 1st). In the absence of an explicit value provided in this attribute, all reporting period values will be assumed to be based on a reporting year start day of January 1. This is declared in the base schema since it has a fixed identifier and representation. The derived series type may either require or prohibit this attribute, depending on whether the data structure declared the reporting year start day attribute and if so, the attribute relationship and assignment status assigned to it. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| Obs | *ObsType* |  |

***ObsType*:**ObsType is the abstract type which defines the structure of a grouped or un-grouped observation. The observation must be provided a key, which is either a value for the dimension which is declared to be at the observation level if the observation is grouped, or a full set of values for all dimensions in the data structure definition if the observation is un-grouped. This key should disambiguate the observation within the context in which it is defined (e.g. there should not be another observation with the same dimension value in a series). The observation can contain an observed value and/or attribute values. Data structure definition schemas will drive a type or types based on this that is specific to the data structure definition and the variation of the format being expressed in the schema. The dimension value(s) which make up the key and the attribute values associated with the key dimension(s) or the primary measure will be represented with XML attributes. This is specified in the content model with the declaration of anyAttributes in the "local" namespace. The derived observation type will refine this structure so that the attributes are explicit. The XML attributes will be given a name based on the attribute's identifier. These XML attributes will be unqualified (meaning they do not have a namespace associated with them). The dimension XML attribute(s) will be required while the attribute XML attributes will be optional. To allow for generic processing, it is required that the only unqualified XML attributes in the derived observation type be for the observation dimension(s) and attributes declared in the data structure definition. If additional attributes are required, these should be qualified with a namespace so that a generic application can easily distinguish them as not being meant to represent a data structure definition dimension or attribute. If the data structure definition specific schema requires that explicit measures be used (only possible when the measure dimension is specified at the observation), then there will be types derived for each measure defined by the measure dimension. In this case, the types will be specific to each measure, which is to say that the representation of the primary measure (i.e. the observed value) will be restricted to that which is specified by the specific measure.

Derivation:

*com:AnnotableType* (extension)   
   *ObsType*



Attributes:

type?, TIME\_PERIOD?, REPORTING\_YEAR\_START\_DAY?, OBS\_VALUE?

Content:

com:Annotations?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type | com:IDType | The type attribute is used when the derived format requires that explicit measure be used. In this case, the derived type based on the measure will fix this value to be the identification of the measure concept. This will not be required, but since it is fixed it will be available in the post validation information set which will allow for generic processing of the data. If explicit measures are not used, then the derived type will prohibit the use of this attribute. |
| TIME\_PERIOD | com: ObservationalTimePer iodType | The TIME\_PERIOD attribute is an explicit attribute for the time dimension. This is declared in the base schema since it has a fixed identifier and representation. The derived series type will either require or prohibit this attribute, depending on whether time is the observation dimension. If the time dimension specifies a more specific representation of time the derived type will restrict the type definition to the appropriate type. |
| REPORTING\_YEAR\_START\_DAY | xs:gMonthDay | The REPORTING\_YEAR\_START\_DAY attribute is an explict attribute for the reporting year start day, which provides context to the time dimension when its value contains a reporting period (e.g. 2010-Q1). This attribute is used to state the month and day that the reporting year begins (e.g. --07-01 for July 1st). In the absence of an explicit value provided in this attribute, all reporting period values will be assumed to be based on a reporting year start day of January 1. This is declared in the base schema since it has a fixed identifier and representation. The derived observation type may either require or prohibit this attribute, depending on whether the data structure declared the reporting year start day attribute and if so, the attribute relationship and assignment status assigned to it. |
| OBS\_VALUE | xs:anySimpleType | The OBS\_VALUE attribute is an explicit attribute for the primary measure, which is intended to hold the value for the observation. This is declared in the base schema since it has a fixed identifier. This attribute is un-typed, since the representation of the observed value can vary widely. Derived types will restrict this to be a type based on the representation of the primary measure. In the case that an explicit measure is used, the derived type for a given measure might further restrict the type of the primary measure to be more specific to the core representation for the measure concept. Note that it is required that in the case of multiple measures being used, that the representation of the primary measure is broad enough to handle the various representations of the measure concepts. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |

***TimeSeriesDataSetType*:**TimeSeriesDataSetType is the abstract type which defines the base structure for any data structure definition specific time series based data set. A derived data set type will be created that is specific to a data structure definition. Unlike the base format, only one variation of this is allowed for a data structure definition. This variation is the time dimension as the observation dimension. Data is organised into a collection of time series. Because this derivation is achieved using restriction, data sets conforming to this type will inherently conform to the base data set structure as well. In fact, data structure specific here will be identical to data in the base data set when the time dimension is the observation dimension, even for the derived data set types. This means that the data contained in this structure can be processed in exactly the same manner as the base structure. The same rules for derivation as the base data set type apply to this specialized data set.

Derivation:

*com:AnnotableType* (extension)   
   *DataSetType* (restriction)   
         *TimeSeriesDataSetType*



Attributes:

structureRef, setID?, action?, reportingBeginDate?, reportingEndDate?, validFromDate?, validToDate?, publicationYear?, publicationPeriod?, dataScope, REPORTING\_YEAR\_START\_DAY?

Content:

com:Annotations?, DataProvider?, Group\*, Series\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureRef | xs:IDREF | The structureRef contains a reference to a structural specification in the header of a data or reference metadata message. The structural specification details which structure the data or reference metadata conforms to, as well as providing additional information such as how the data is structure (e.g. which dimension occurs at the observation level for a data set). |
| setID | com:IDType | The setID provides an identification of the data or metadata set. |
| action | com:ActionType | The action attribute indicates whether the file is appending, replacing, or deleting. |
| reportingBeginDate | com: BasicTimePeriodType | The reportingBeginDate indicates the inclusive start time of the data reported in the data or metadata set. |
| reportingEndDate | com: BasicTimePeriodType | The reportingEndDate indicates the inclusive end time of the data reported in the data or metadata set. |
| validFromDate | xs:dateTime | The validFromDate indicates the inclusive start time indicating the validity of the information in the data or metadata set. |
| validToDate | xs:dateTime | The validToDate indicates the inclusive end time indicating the validity of the information in the data or metadata set. |
| publicationYear | xs:gYear | The publicationYear holds the ISO 8601 four-digit year. |
| publicationPeriod | com: ObservationalTimePer iodType | The publicationPeriod specifies the period of publication of the data or metadata in terms of whatever provisioning agreements might be in force (i.e., "Q1 2005" if that is the time of publication for a data set published on a quarterly basis). |
| dataScope | DataScopeType | The dataScope attribute indicates the scope at which the data is meant to be validated. These scopes are hierarchical and are (from the top down); DataStructure, ConstrainedDataStructure, Dataflow, and ProvisionAgreement. the hierarchy of these scopes represent the cascading level of constraints, which can restrict the valid values for components. For example, a data structure defines a dimension with a coded representation. A data flow might have a constraint associated with it which further restricts the values allowed from the referenced code list to a subset of the values allowed by the data structure definition. A provision agreement that is based on the dataflow might also have a constraint, which further restricts the subset of the codelist from the dataflow. Therefore, the allowed content becomes stricter lower in the hierarchy. Data that is given a scope of one value is stated to be valid at that level and all levels below it. Therefore, this scope serves to state that data that is meant to be structured simply against the data structure definition is not meant to be validated against the a dataflow, where constraints might be applied. |
| REPORTING\_YEAR\_START\_DAY | xs:gMonthDay | The REPORTING\_YEAR\_START\_DAY attribute is an explict attribute for the reporting year start day, which provides context to the time dimension when its value contains a reporting period (e.g. 2010-Q1). This attribute is used to state the month and day that the reporting year begins (e.g. --07-01 for July 1st). In the absence of an explicit value provided in this attribute, all reporting period values will be assumed to be based on a reporting year start day of January 1. This is declared in the base schema since it has a fixed identifier and representation. The derived data set type may either require or prohibit this attribute, depending on whether the data structure declared the reporting year start day attribute and if so, the attribute relationship and assignment status assigned to it. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| DataProvider | com: DataProviderReferenc eType | DataProvider contains a reference to the provider for the data set. |
| Group | *GroupType* | Group contains a references to a defined group in the data structure definition along with its key (if necessary) and values for the attributes which are associated with the group. An attribute is associated to a group by either an explicit group relationship or by a group attachment when the attribute has a relationship with a dimension which is a member of this group. |
| Series | TimeSeriesType | Series contains a collection of observations that share a common key (set of dimension values). The key of a series is every dimension defined in the data structure definition, save the dimension at the observation level. In addition to the key and observations, the series contains values for attributes which have a relationship with any dimension that is part of the series key, so long as the attribute does not specify an attachment group or also has a relationship with the dimension declared to be at the observation level. |

**TimeSeriesType:**TimeSeriesType defines an abstract structure which is used to group a collection of observations which have a key in common, organised by time. The key for a series is every dimension defined in the data structure definition, save the time dimension. In addition to observations, values can be provided for attributes which are associated with the dimensions which make up this series key (so long as the attributes do not specify a group attachment or also have an relationship with the time dimension). It is possible for the series to contain only observations or only attribute values, or both. The same rules for derivation as the base series type apply to this specialized series.

Derivation:

*com:AnnotableType* (extension)   
   *SeriesType* (restriction)   
         TimeSeriesType



Attributes:

REPORTING\_YEAR\_START\_DAY?

Content:

com:Annotations?, Obs\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| REPORTING\_YEAR\_START\_DAY | xs:gMonthDay | The REPORTING\_YEAR\_START\_DAY attribute is an explict attribute for the reporting year start day, which provides context to the time dimension when its value contains a reporting period (e.g. 2010-Q1). This attribute is used to state the month and day that the reporting year begins (e.g. --07-01 for July 1st). In the absence of an explicit value provided in this attribute, all reporting period values will be assumed to be based on a reporting year start day of January 1. This is declared in the base schema since it has a fixed identifier and representation. The derived series type may either require or prohibit this attribute, depending on whether the data structure declared the reporting year start day attribute and if so, the attribute relationship and assignment status assigned to it. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| Obs | *TimeSeriesObsType* |  |

***TimeSeriesObsType*:**TimeSeriesObsType defines the abstract structure of a time series observation. The observation must be provided a value for the time dimension. This time value should disambiguate the observation within the series in which it is defined (i.e. there should not be another observation with the same time value). The observation can contain an observed value and/or attribute values. The same rules for derivation as the base observation type apply to this specialized observation.

Derivation:

*com:AnnotableType* (extension)   
   *ObsType* (restriction)   
         *TimeSeriesObsType*



Attributes:

TIME\_PERIOD, REPORTING\_YEAR\_START\_DAY?, OBS\_VALUE?

Content:

com:Annotations?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| TIME\_PERIOD | com: ObservationalTimePer iodType | The TIME\_PERIOD attribute is an explicit attribute for the time dimension. This is declared in the base schema since it has a fixed identifier and representation. Since this data is structured to be time series only, this attribute is always required. If the time dimension specifies a more specific representation of time the derived type will restrict the type definition to the appropriate type. |
| REPORTING\_YEAR\_START\_DAY | xs:gMonthDay | The REPORTING\_YEAR\_START\_DAY attribute is an explict attribute for the reporting year start day, which provides context to the time dimension when its value contains a reporting period (e.g. 2010-Q1). This attribute is used to state the month and day that the reporting year begins (e.g. --07-01 for July 1st). In the absence of an explicit value provided in this attribute, all reporting period values will be assumed to be based on a reporting year start day of January 1. This is declared in the base schema since it has a fixed identifier and representation. The derived observation type may either require or prohibit this attribute, depending on whether the data structure declared the reporting year start day attribute and if so, the attribute relationship and assignment status assigned to it. |
| OBS\_VALUE | xs:anySimpleType | The OBS\_VALUE attribute is an explicit attribute for the primary measure, which is intended to hold the value for the observation. This is declared in the base schema since it has a fixed identifier. This attribute is un-typed, since the representation of the observed value can vary widely. Derived types will restrict this to be a type based on the representation of the primary measure. In the case that an explicit measure is used, the derived type for a given measure might further restrict the type of the primary measure to be more specific to the core representation for the measure concept. Note that it is required that in the case of multiple measures being used, that the representation of the primary measure is broad enough to handle the various representations of the measure concepts. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |

### Simple Types

**DataScopeType:**DataScopeType is an enumeration of the possible validity scopes for a data set. These scopes indicate the level at which the data is stated to be valid.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| DataStructure | The data set conforms simply to the data structure definition as it is defined, without regard to any constraints. |
| ConstrainedDataStructure | The data set conforms to the known allowable content constraints applied to the data structure definition. |
| Dataflow | The data set conforms to the known allowable content constraints applied to the dataflow. |
| ProvisionAgreement | The data set conforms to the known allowable content constraints applied to the provision agreement. |

## Generic Metadata Namespace

**http://www.sdmx.org/resources/sdmxml/schemas/v2\_1/metadata/generic**

### Summary

Referenced Namespaces:

| **Namespace** | **Prefix** |
| --- | --- |
| http://www.sdmx.org/resources/sdmxml/schemas/v2\_1/common | com |
| http://www.w3.org/2001/XMLSchema | xs |

Contents:

6 Complex Types

### Complex Types

**MetadataSetType:**MetadataSetType describes the structure for a metadata set, which contains a collection of reported metadata against a set of values for a given full or partial target identifier, as described in a metadata structure definition. The metadata set may contain reported metadata for multiple report structures defined in a metadata structure definition.

Derivation:

*com:AnnotableType* (extension)   
   MetadataSetType



Attributes:

structureRef, setID?, action?, reportingBeginDate?, reportingEndDate?, validFromDate?, validToDate?, publicationYear?, publicationPeriod?

Content:

com:Annotations?, com:Name\*, DataProvider?, Report+

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureRef | xs:IDREF | The structureRef contains a reference to a structural specification in the header of a data or reference metadata message. The structural specification details which structure the data or reference metadata conforms to, as well as providing additional information such as how the data is structure (e.g. which dimension occurs at the observation level for a data set). |
| setID | com:IDType | The setID provides an identification of the data or metadata set. |
| action | com:ActionType | The action attribute indicates whether the file is appending, replacing, or deleting. |
| reportingBeginDate | com: BasicTimePeriodType | The reportingBeginDate indicates the inclusive start time of the data reported in the data or metadata set. |
| reportingEndDate | com: BasicTimePeriodType | The reportingEndDate indicates the inclusive end time of the data reported in the data or metadata set. |
| validFromDate | xs:dateTime | The validFromDate indicates the inclusive start time indicating the validity of the information in the data or metadata set. |
| validToDate | xs:dateTime | The validToDate indicates the inclusive end time indicating the validity of the information in the data or metadata set. |
| publicationYear | xs:gYear | The publicationYear holds the ISO 8601 four-digit year. |
| publicationPeriod | com: ObservationalTimePer iodType | The publicationPeriod specifies the period of publication of the data or metadata in terms of whatever provisioning agreements might be in force (i.e., "Q1 2005" if that is the time of publication for a data set published on a quarterly basis). |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| com:Name | com:TextType | Name is a reusable element, used for providing a human-readable name for an object. |
| DataProvider | com: DataProviderReferenc eType | DataProviderReference provides a references to an organisation with the role of data provider that is providing this metadata set. |
| Report | ReportType | Report contains the details of a the reported metadata, including the identification of the target and the report attributes. |

**ReportType:**ReportType contains a set of report attributes and identifies a target objects] to which they apply.

Derivation:

*com:AnnotableType* (extension)   
   ReportType



Attributes:

id

Content:

com:Annotations?, Target, AttributeSet

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | com:IDType | The id attribute holds the identifier of the report structure as defined in the metadata structure definition. This identifies the report structure which defines the structure of metadata that is being reported. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| Target | TargetType | Target contains a set of target reference values which when taken together, identify the object or objects to which the reported metadata apply. |
| AttributeSet | AttributeSetType | AttributeSet contains the reported metadata attribute values for the reported metadata. |

**TargetType:**TargetType defines the structure of a target. It contains a set of target reference values which when taken together, identify the object or objects to which the reported metadata apply.

Attributes:

id

Content:

ReferenceValue+

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | com:IDType | The id attribute holds the identifier of the metadata target as defined in the metadata structure definition. This identifies the metadata target of the report structure that identifies the target object(s) of the reported metadata. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReferenceValue | ReferenceValueType | ReferenceValue contains a value for a target reference object reference. When this is taken with its sibling elements, they identify the object or objects to which the reported metadata apply. The content of this will either be a reference to an identifiable object, a data key, a reference to a data set, or a report period. |

**ReferenceValueType:**ReferenceValueType defines the structure of a target object reference value. A target reference value will either be a reference to an identifiable object, a data key, a reference to a data set, or a report period.

Attributes:

id

Content:

(ObjectReference | DataKey | DataSetReference | ConstraintContentReference | ReportPeriod)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | com:IDType | The id attribute holds the identifier of the target reference object as defined in the metadata structure definition. This identifies the target reference of the metadata target that identifes one of the target references, that when taken together, identify the target of the reported metadata. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ObjectReference | com: ObjectReferenceType | ObjectReference provides a reference to an identifiable object in the SDMX information model. An identifiable object target will utilize this option as the representation of the target reference value. |
| DataKey | com:DataKeyType | ObjectReference provides a set of dimension references and values for those dimension for the purpose of reporting metadata against a set of data. A key descriptor values target will utilize this option as the representation of the target reference value. |
| DataSetReference | com:SetReferenceType | DataSetReference provides a reference to a data set for the purpose of reporting metadata against the data. A data set target will utilize this option as the representation of the target reference value. |
| ConstraintContentRef erence | com: AttachmentConstraint ReferenceType | ConstraintContentReference provides a reference to an attachment constraint for the purpose of reporting metadata against the data identified in the key sets and/or cube regions identified by the constraint. A constraint target will utilize this option as the representation of the target reference value. |
| ReportPeriod | com: ObservationalTimePer iodType | ReportPeriod provides a report period for the purpose of qualifying the target reporting period of reported metadata. A report period target will utilize this option as the representation of the target reference value. |

**AttributeSetType:**AttributeSetType defines the structure for a collection of reported metadata attributes.

Content:

ReportedAttribute+

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReportedAttribute | ReportedAttributeTyp e | ReportedAttribute provides the details of a reported attribute, including a value and/or child reported attributes. |

**ReportedAttributeType:**ReportedAttributeType defines the structure for a reported metadata attribute. A value for the attribute can be supplied as either a single value, or multi-lingual text values (either structured or unstructured). An optional set of child metadata attributes is also available if the metadata attribute definition defines nested metadata attributes.

Derivation:

*com:AnnotableType* (extension)   
   ReportedAttributeType



Attributes:

id, value?

Content:

com:Annotations?, (com:Text+ | com:StructuredText+)?, AttributeSet?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | com:IDType | The id attribute identifies the metadata attribute that the value is being reported for. |
| value | xs:string | The value attribute holds any simple value for the metadata attribute. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| com:Text | com:TextType | Text is used to supply parallel multi-lingual textual values for the reported metadata attribute. This will be used if the text format of the metadata attribute has a type of string and the multi-lingual value is set to true. |
| com:StructuredText | com:XHTMLType | StructuredText is used to supply parallel multi-lingual structured (as XHTML) textual values for the reported metadata attribute. This will be used if the text format of the metadata attribute has a type of XHTML and the multi-lingual value is set to true. |
| AttributeSet | AttributeSetType | AttributeSet contains the reported metadata attribute values for the child metadata attributes. |

## Structure Specific Metadata Namespace

**http://www.sdmx.org/resources/sdmxml/schemas/v2\_1/metadata/structurespecific**

### Summary

Referenced Namespaces:

| **Namespace** | **Prefix** |
| --- | --- |
|  |  |
| http://www.sdmx.org/resources/sdmxml/schemas/v2\_1/common | com |
| http://www.w3.org/2001/XMLSchema | xs |

Contents:

5 Complex Types

### Complex Types

***MetadataSetType*:**MetadataSetType is an abstract base type the forms the basis for a metadata structure specific metadata set. It is restricted by the metadata structure definition specific schema to meet its needs.

Derivation:

*com:AnnotableType* (extension)   
   *MetadataSetType*



Attributes:

structureRef, setID?, action?, reportingBeginDate?, reportingEndDate?, validFromDate?, validToDate?, publicationYear?, publicationPeriod?

Content:

com:Annotations?, com:Name\*, DataProvider?, Report+

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureRef | xs:IDREF | The structureRef contains a reference to a structural specification in the header of a data or reference metadata message. The structural specification details which structure the data or reference metadata conforms to, as well as providing additional information such as how the data is structure (e.g. which dimension occurs at the observation level for a data set). |
| setID | com:IDType | The setID provides an identification of the data or metadata set. |
| action | com:ActionType | The action attribute indicates whether the file is appending, replacing, or deleting. |
| reportingBeginDate | com: BasicTimePeriodType | The reportingBeginDate indicates the inclusive start time of the data reported in the data or metadata set. |
| reportingEndDate | com: BasicTimePeriodType | The reportingEndDate indicates the inclusive end time of the data reported in the data or metadata set. |
| validFromDate | xs:dateTime | The validFromDate indicates the inclusive start time indicating the validity of the information in the data or metadata set. |
| validToDate | xs:dateTime | The validToDate indicates the inclusive end time indicating the validity of the information in the data or metadata set. |
| publicationYear | xs:gYear | The publicationYear holds the ISO 8601 four-digit year. |
| publicationPeriod | com: ObservationalTimePer iodType | The publicationPeriod specifies the period of publication of the data or metadata in terms of whatever provisioning agreements might be in force (i.e., "Q1 2005" if that is the time of publication for a data set published on a quarterly basis). |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| com:Name | com:TextType | Name is a reusable element, used for providing a human-readable name for an object. |
| DataProvider | com: DataProviderReferenc eType | >DataProviderReference provides a references to an organisation with the role of data provider that is providing this metadata set. |
| Report | *ReportType* | Report contains the details of a the reported metadata, including the identification of the target and the report attributes. This element is unqualified so that the metadata structure definition specific schema can refine the type of the element such that it requires types built according to the metadata structure definition. This allows the metadata structure definition to validate the structure of the reported metadata against the metadata structure definition while still allowing the content to be processed in a generic manner. |

***ReportType*:**ReportType is an abstract base type the forms the basis for a metadata structure definition specific report, based on the defined report structures. This type is restricted in the metadata structure definition specific schema so that the Target and AttributeSet conform to the prescribed report structure.

Derivation:

*com:AnnotableType* (extension)   
   *ReportType*



Attributes:

id?

Content:

com:Annotations?, Target, AttributeSet

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | com:IDType | The id attribute holds the identifier of the report structure as defined in the metadata structure definition. This identifies the report structure which defines the structure of metadata that is being reported. This is optional and not expected to be supplied as the metadata structure definition specific schema will specify a fixed value such that the reference to the report structure will always be available if required for processing. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| Target | *TargetType* | Target contains a set of target reference values which when taken together, identify the object or objects to which the reported metadata apply. This element is unqualified so that the metadata structure definition specific schema can refine the type of the element such that the references values can be validated against those defined in the metadata targets for the report structure. |
| AttributeSet | xs:anyType | AttributeSet contains the reported metadata attribute values for the reported metadata. This element is unqualified and un-typed so that it can refined by the metadata structure definition specific schema to validate that the reported metadata attributes conform to those prescribed by the report structure. The content of this must be element only, and these elements must always represent a reported attribute. Since this can not be strictly enforced in XML Schema, additional steps have been added to make generic processing of this element simpler. When processing this element, any element found with the attribute isMetadataAttribute in this target namespace is assumed to be a reported metadata attribute and can be processed as such. |

***TargetType*:**TargetType is an abstract base type that forms the basis of a the metadata report's metadata target value. This type is restricted in the metadata structure definition specific schema so that the ReferenceValue elements conform to the targets specified in the metadata target defined in the metadata structure definition.

Attributes:

id?

Content:

ReferenceValue+

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | com:IDType | The id attribute holds the identifier of the metadata target as defined in the metadata structure definition. This identifies the metadata target of the report structure that identifies the target object(s) of the reported metadata. This is optional and not expected to be supplied as the metadata structure definition specific schema will specify a fixed value such that the reference to the metadata target will always be available if required for processing. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReferenceValue | *ReferenceValueType* | ReferenceValue contains a value for a target reference. When this is taken with its sibling elements, they identify the object or objects to which the reported metadata apply. The content of this will either be a reference to an identifiable object, a data key, a reference to a data set, or a report period. This element is unqualified so that the metadata structure definition specific schema can refine the type of the element such that value can be validated against the format defined in the metadata structure definition. |

***ReferenceValueType*:**ReferenceValueType is an abstract base type that forms the basis of a target reference value. A target reference value will either be a reference to an identifiable object, a data key, a reference to a data set, or a report period. The choice of these options will be refined to only one according to the definition of the target in the metadata structure definition.

Attributes:

id?

Content:

(ObjectReference | DataKey | DataSetReference | ConstraintContentReference | ReportPeriod)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | com:IDType | The id attribute holds the identifier of the target reference object reference as defined in the metadata structure definition. This identifies the target reference of the metadata target that identifes one of the target references, that when taken together, identify the target of the reported metadata. This is optional and not expected to be supplied as the metadata structure definition specific schema will specify a fixed value such that the reference to the target object definition will always be available if required for processing. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ObjectReference | *com:ReferenceType* | ObjectReference provides a reference to an identifiable object in the SDMX information model. An identifiable object target will utilize this option as the representation of the target reference value. This element is unqualified so that the metadata structure definition specific schema can refine the type of the element such that the type of reference is valid according to the object type specified in the identifiable object target in the metadata structure definition. At the very least, the reference will be specific to the type of object being referenced so that a complete set of reference fields must be provided. In cases where an item object is restricted to be from a particular scheme, this type will be further restricted so that only a valid item reference can be supplied. The structure of this reference is such that it can be generically processed without needing to know what the intended target object type is prior to processing, as this information is part of the reference. |
| DataKey | com:DataKeyType | ObjectReference provides a set of dimension references and values for those dimension for the purpose of reporting metadata against a set of data. A key descriptor values target will utilize this option as the representation of the target reference value. It is not expect that the metadata structure definition specific schema would refine this, but none the less, it is an unqualified element. |
| DataSetReference | com:SetReferenceType | DataSetReference provides a reference to a data set for the purpose of reporting metadata against the data. A data set target will utilize this option as the representation of the target reference value. |
| ConstraintContentRef erence | com: AttachmentConstraint ReferenceType | ConstraintContentReference provides a reference to an attachment constraint for the purpose of reporting metadata against the data identified in the key sets and/or cube regions identified by the constraint. A constraint target will utilize this option as the representation of the target reference value. |
| ReportPeriod | com: ObservationalTimePer iodType | ReportPeriod provides a report period for the purpose of qualifying the target reporting period of reported metadata. A report period target will utilize this option as the representation of the target reference value. It is not expect that the metadata structure definition specific schema would refine this, but none the less, it is an unqualified element. This element is unqualified so that the metadata structure definition specific schema can refine the type of the element such that the specific type of time value prescribed in the metadata structure definition can be validated. |

***ReportedAttributeType*:**ReportedAttributeType is an abstract base type that forms the basis for a metadata structure specific metadata attribute. A value for the attribute can be supplied as either a single value, or multi-lingual text values (either structured or unstructured). An optional set of child metadata attributes is also available if the metadata attribute definition defines nested metadata attributes. The metadata structure definition specific schema will refine this type for each metadata attribute such that the content can be validation against what is defined in the metadata structure definition.

Derivation:

*com:AnnotableType* (extension)   
   *ReportedAttributeType*



Attributes:

id?, value?, isMetadataAttribute?

Content:

com:Annotations?, (com:Text+ | com:StructuredText+)?, AttributeSet?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | com:IDType | The id attribute identifies the metadata attribute that the value is being reported for. This is optional and not expected to be supplied as the metadata structure definition specific schema will specify a fixed value such that the reference to the metadata attribute will always be available if required for processing. |
| value | xs:anySimpleType | The value attribute holds any simple value for the metadata attribute. This attribute is un-typed such that the metadata structure definition specific schema can specify any simple type according the text format / local representation defined by the metadata structure definition. |
| isMetadataAttribute (fixed: true) | xs:boolean | The isMetadataAttribute attribute is a fixed boolean (true) and is intended to designate to processing applications that a given element represents a reported attribute. This attribute is qualified (meaning that it will be qualified in an instance with the target namespace) to ensure that it can be properly identified by applications. The purpose of this is to allow applications to identify elements with unknown names as reported attributes so that they may process a metadata structure definition specific instance without knowledge of the underlying metadata structure definition. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| com:Annotations | com:AnnotationsType | Annotations is a reusable element the provides for a collection of annotations. It has been made global so that restrictions of types that extend AnnotatableType my reference it. |
| com:Text | com:TextType | Text is used to supply parallel multi-lingual textual values for the reported metadata attribute. This will be used if the text format of the metadata attribute has a type of string and the multi-lingual value is set to true. |
| com:StructuredText | com:XHTMLType | StructuredText is used to supply parallel multi-lingual structured (as XHTML) textual values for the reported metadata attribute. This will be used if the text format of the metadata attribute has a type of XHTML and the multi-lingual value is set to true. If the multi-lingual flag is not set to true, it is expected that the maximum occurrence of this will be refined to be 1 in the metadata structure definition specific schema. |
| AttributeSet | xs:anyType | AttributeSet contains the reported metadata attribute values for the child metadata attributes. This element is unqualified and un-typed so that it can refined by the metadata structure definition specific schema to validate that the reported metadata attributes conform to those prescribed by the metadata attribute definition. The content of this must be element only, and these elements must always represent a reported attribute. Since this can not be strictly enforced in XML Schema, additional steps have been added to make generic processing of this element simpler. When processing this element, any element found with the attribute isMetadataAttribute in this target namespace is assumed to be a reported metadata attribute and can be processed as such. |

# Mapping to Structure-Specific Schemas

## General

Data structure and metadata structure-specific schemas are each based on one single core construct found in their respective structure-specific namespaces;

Data - http://www.SDMX.org/resources/SDMXML/schemas/v2\_1/data/structurespecific

Metadata - http://www.SDMX.org/resources/SDMXML/schemas/v2\_1/data/structurespecific

For a metadata structure, only a single schema will be created for the entire structure definition. For data structures, many variations of the base schema are possible, each specific as to how the data is organised. This will be discussed in more details in the specific rules for the data structure schemas.

### Basic Terminology

In the subsequent sections, the following namespace prefixes are used:

|  |  |
| --- | --- |
| **Namespace** | **Prefix** |
| http://www.w3.org/2001/XMLSchema | xs |
| http://www.sdmx.org/resources/sdmxml/schemas/v2\_1/common | common |
| http://www.sdmx.org/resources/sdmxml/schemas/ v2\_1/data/structurespecific | dsd |
| http://www.sdmx.org/resources/sdmxml/schemas/ v2\_1/metadata/structurespecific | msd |

It is assumed that in order to use this guide, the reader is familiar with schema terminology. However, for convenience the following is list of the terminology used here:

**Schema:** Refers to the format specific schema in general, and in particular the root xs:schema element of that schema file.

**Global Element:** Refers to an element definition at the top level of the schema (i.e. an xs:element element in the root xs:schema element). It will define a name and type (name and type attributes) and possibly a substitution group (substitutionGroup attribute).

**Local Element:** Refers to an element definition within a complex type (i.e. an xs:element element contained within a xs:sequence element that is contained in a xs:complexType element). A local element must define a name and type (name and type attributes) and may also specify a minimum and maximum occurrence (minOccurs and maxOccurs attribute).

**Qualified/Unqualified Element:** A qualified element is an element that must be referred to by the namespace in which it was defined. An unqualified element does not have a namespace associated with it. The structure specific schemas make use of unqualified elements to that the structure specific schemas can restrict the base content to meet the specific needs of the structure, while maintaining as much of the original document structure as possible.

**Element Reference:** Refers to an element definition within a complex type that is a reference to a global element (i.e. an xs:element element contained within a xs:sequence element that is contained in a xs:complexType element). An element reference must reference a global element (via its ref attribute) and may also specify a minimum and maximum occurrence (minOccurs and maxOccurs attribute).

**Complex Type:** Refers to a complex type definition. In this context, all complex type definitions occur at the top level of the schema (i.e. an xs:complexType element in the root xs:schema element). A complex type must define a name (name attribute) and may be made abstract (via the abstract attribute’s boolean value).

**Simple Type:** Refers to a simple type definition. In this context, all simple type definitions occur at the top level of the schema (i.e. an xs:simpleType element in the root xs:schema element). In this context, a simple type will always be defined via a restriction (an xs:restriction element in the xs:simpleType element). The restriction will reference a base type.

**Anonymous Type:** A complex or simple type definition which occurs within an element definition. The method is sometimes referred to a the "Russian-doll" technique as it creates nested constructs. Anonymous types are not given names and cannot be abstract. The can however, be derived from other types.

**Content Group:** A group which defines a content model for reuse. This is contained in the xs:group element, and is defined at the root of the schema. It allows for a common sequence or choice of elements to be reused across multiple types without having to redefine the sequence or choice in each type.

**Uniqueness Constraint:** A uniqueness constraint is defined within an element and is used to force descendent elements to be unique based on some criteria of it fields (elements or attributes). This is defined in an <xs:unique> element, and has content of an <xs:selector> and multiple <xs:field> elements. The selector designates the descendants that must be unique (with an xpath attribute) and the field specifies which property of the selected element must be unique (also with an xpath attribute)

**Extension:** Refers to the definition of a complex type that is an extension of another complex type. The extension will always make a reference to a base. In the schema, this is defined within the xs:complexType element as a child xs:complexContent element containing an xs:extension element (with a base attribute).

**Restriction:** Refers to the definition of a simple or complex type that is a restriction of another type of the same variety. The restriction will always make a reference to a base. In the schema, this is defined with an xs:restriction element (with a base attribute).

**Sequence:** Refers to a sequence of elements that may be defined as the root of a complex type content model, or as part of the content of a choice or another sequence. This is defined as an xs:sequence element. The sequence may specify a minimum and maximum occurrence (minOccurs and maxOccurs attribute).

**Choice:** Refers to a choice of elements that may be defined as the root of a complex type content model, or as part of the content of a sequence or another choice. This is defined as an xs:choice element. The sequence may specify a minimum and maximum occurrence (minOccurs and maxOccurs attribute).

**Facet:** Refers to a single detail of a simple type restriction. This is represented by elements such as xs:minInclusive, xs:totalDigits, xs:minLength, and is contained in the xs:restriction element of a simple type definition. The value of the facet is contained in a value attribute of the particular element.

**Enumeration:** Refers to an enumerated value of a simple type definition. It is represented by an xs:enumeration element contained within an xs:restriction element of a simple type definition. An enumeration defines a value (in the value attribute) and documentation (in xs:documentation elements contained in an xs:annotation element).

**XML Attribute:** Refers to the definition of an XML attribute for a complex type (i.e. and xs:attribute element in a xs:complexType element). An attribute must define a name and type (name and type attributes) and may also specify a usage (use attribute).

## Namespace Rules

Each format specific schema will specify its namespace in the target namespace of the schema (the targetNamespace attribute of the schema). This document also assumes that the root namespace (that which is defined by the xmlns attribute) of the schema will be the same as the target namespace. Therefore any types or global elements referenced in these descriptions without a namespace prefix are assumed to be in the format specific namespace.

The format specific schemas will incorporate the core format namespace and the common namespace by importing the schemas (via the xs:import element). If necessary, additional namespaces may be imported and referenced.

For the purpose of the descriptions here, the default element form for the schema (as specified in the elementFormDefault attribute of the schema) is “qualified", and the default attribute form (as specified in the attributeFormDefault attribute of the schema) is "unqualified".

## General Rules

The following section details the general rules which apply to all structure specific schema creation.

### Component Name Determination

When required to create an XML element or attribute, the name for a component is always its identifier. However, the identifier may be inherited. Therefore, the general rules is as follows:

1. If the component defines an identifier, the element or attribute name is the value of that identifier
2. Otherwise, the element or attribute name is the identifier of the concept from which it takes its semantic (Note that this is technically the component identifer).

### Representation Determination

Every component has a representation associated with it, whether it is defined as a local representation in the component definition, or it is inherited from the concept from which the component takes it semantic (as defined in the concept identity of the component).

The representation of a component is determined by the following precedence:

1. The local representation defined by the component
2. The core representation defined in the concept from which the component takes its semantic
3. A default representation of an un-faceted text format with a data type of String.

The representation will either define a text format, or an enumeration with an enumeration format.

A text format consists of a data type and an optional collection of facets. It is the combination of these which determine the exact nature of the component representation. An enumeration consists of a reference to an item scheme, for which an enumerated list of possible values can be created.

### Simple / Primitive Type Determination

For any given representation, there exist rules for determining the simple or primitive type which should be used to validate the value. There are no specific requirements to how a simple type is named or if it is referenced or used as an anonymous type. This section simply serves to state the requirements of the type for a component based on its [determined representation](#_Representation_Determination:).

For example, a dimension may inherit its representation for a concept, and the data type of a representation data type may be a String. The simplest solution would be to use the xs:string primitive type. However, an implementer may have chosen to generate simple types for all concepts to avoid having to look up the concept core representation for very component. In this case, the type may be given a name based on the concept and be a simple derivation from the xs:string type that places no further restrictions. The result would be that the type that is actually used for the dimension, although named after the concept, is effectively the required xs:string. These rules are meant to allow such flexibility in how types are created. The only requirement is that the type meet the requirements stated here.

### Representation with Enumeration

A representation which defines an enumeration will result in a simple type that is a restriction of the common:IDType. The simple type will define enumerations for each item in the item scheme. The value for these enumerations will be identifier of the item. If desired, the names of the item may be placed in the documentation of the enumeration, but this is not required. Example:

<xs:simpleType name="ESTAT.CL\_COUNTRY.1.0">

<xs:restriction base="common:IDType">

<xs:enumeration value="BE">

<xs:annotation>

<xs:documentation xml:lang="en">Belgium</xs:documentation>

</xs:annotation>

</xs:enumeration>

### Representation with Text Format

A representation which defines a text format will result in a simple type or primitive type. The first step is to determine the base type from the text format data type:

|  |  |
| --- | --- |
| **SDMX Data Type** | **XML Schema Data Type** |
| String | xs:string |
| AlphaNumeric | common:AlphaNumericType |
| Alpha | common:AlphaType |
| Numeric | common:NumericType |
| BigInteger | xs:integer |
| Integer | xs:int |
| Long | xs:long |
| Short | xs:short |
| Decimal | xs:decimal |
| Float | xs:float |
| Double | xs:double |
| Boolean | xs:boolean |
| URI | xs:anyURI |
| Count | xs:integer |
| InclusiveValueRange | xs:decimal |
| ExclusiveValueRange | xs: decimal |
| Incremental | xs: decimal |
| ObservationalTimePeriod | common:ObservationalTimePeriodType |
| StandardTimePeriod | common:StandardTimePeriodType |
| BasicTimePeriod | common:BasicTimePeriodType |
| GregorianTimePeriod | common:GregorianTimePeriodType |
| GregorianYear | xs:gYear |
| GregorianYearMonth | xs:gYearMonth |
| GregorianDay | xs:date |
| ReportingTimePeriod | common:ReportingTimePeriodType |
| ReportingYear | common:ReportingYearType |
| ReportingSemester | common:ReportingSemesterType |
| ReportingTrimester | common:ReportingTrimesterType |
| ReportingQuarter | common:ReportingQuarterType |
| ReportingMonth | common:ReportingMonthType |
| ReportingWeek | common:ReportingWeekType |
| ReportingDay | common:ReportingDayType |
| DateTime | xs:dateTime |
| TimeRange | common:TimeRangeType |
| Month | xs:gMonth |
| MonthDay | xs:gMonthDay |
| Day | xs:gDay |
| Time | xs:time |
| Duration | xs:duration |
| XHTML | N/A[[1]](#footnote-1) |
| KeyValues | N/A1 |
| IdentifiableReference | N/A1 |
| DataSetReference | N/A1 |
| AttachmentConstraintReference | N/A1 |

If the text format does not specify any further facets, then the determined type is the listed type or a type which derives from the listed type without placing any addition restrictions on it. However, if one or more facets are specified, then a simple type based on the listed type is necessary. The simple type derives via restriction from the listed type and adds facets according to the following table (the values are mapped as is):

|  |  |
| --- | --- |
| **SDMX Facet** | **XML Schema Facet** |
| minLength | xs:minLength |
| maxLength | xs:maxLength |
| minValue[[2]](#footnote-2) | if ExclusiveValueRange: xs:minExclusives, else: xs:minInclusive |
| maxValue2 | if ExclusiveValueRange: xs:maxExclusives, else: xs:maxInclusive |
| decimals2 | xs:fractionDigits |
| pattern | xs:pattern |

Any other facets are informational only, and will not affect the determined type.

### *Type Names*

These rules will only dictate type names where absolutely necessary. In all other cases, it is the decision of the implementer as to how to name or use the type. It is also the implementer's requirement to ensure that any type name is properly unique within its scope. To assist in this, the following recommendations are offered for naming types such that they are unique.

* It the type is an enumeration from an item scheme, the recommended name is [Item Scheme Class].[Maintenance Agency].[Item Scheme ID].[Item Scheme Version]
* If the type is based on a text format of a concept core representation, the recommended name is Concept.[Maintenance Agency].[Concept Scheme ID].[Concept Scheme Version].[Concept ID]
* If the type is based on a text format of a component local representation, and;
  + The component id is required to be unique for all components within the scope of the structure which defines it (e.g. a dimension), the recommended name is [Component ID]
  + The component id is only required to be unique within the component list and which defines it (e.g. a metadata attribute), the recommend name is [Component List ID].[Parent Component ID]\*.[Component ID]

### Type Reuse

It is possible that organisations that manage a large number of structure specific schemas my wish to take advantage of the reuse of previously defined type in order to simply the structure specific schema creation and lessen the number of schema elements which are created. The structure specific formats are designed in such a way that this would be allowed without any adverse affects.

For example, an organisation my create predefined types for all of codelists and concept schemes which their structures utilize. These could be contained in a common schema with any namespace deemed appropriate. This would allow the structure specific schemas generation process to recognize the reused components and not be concerned with regenerating types. The logical flow for setting the representation of a component might be as follows:

Does the component define a local type?

Yes: Is that type enumerated?

Yes: Type is the qualified type name for the item scheme

No: Generate simple type for text format

No: Type is the qualified name for the concept from which the component takes its semantic.

Only the constructs that will be detailed in the data and metadata structure-specific rules below are required to be in the specified target namespace of the structure-specific schema. So long as any other generated type conforms to the rules specified, it may exist in any namespace.

## Data Structure Specific Schema

Separate schemas will be created for the data structure depending on which dimension occurs at the observation level, and whether explicit measures are used in the case that the observation dimension is the measure dimension. The recommended target namespace of the data structured specific schema is: [Data Structure URN]:ObsLevelDim:[Observation Dimensions](:explicit)?. Note that the explicit trailer is only used when the measure dimension is used at the observation level and the explicit measure concepts are to be used.

The rules for generating the data structure specific-schema are broken into sections based on the level within the structure (i.e. data set, group, series, observation). Each section will state the rules for each variation of the structure specific format.

### DataSetType

A complex type named DataSetType must be created. Its content model will be derived via restriction. If the dimension at the observation level is the time dimension (TIME\_PERIOD) the base type of the restriction is dsd:TimeSeriesDataSetType. Otherwise, the base type of the restriction is dsd:DataSetType. The complex type content model will be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
   2. A local element named DataProvider with the type common:DataProviderReferenceType, a form of unqualified and a minimum occurrence of 0
   3. If the data structure defines groups, a local element named Group with a form of unqualified, a minimum occurrence of 0, and a maximum occurrence of unbounded. The type of this element should be the type that is described in the GroupType section which follows.
   4. A choice with a minimum occurrence of 0 consisting of:
      1. If the dimension at the observation level is not AllDimensions, a local element named Series with a form of unqualified, a maximum occurrence of unbounded, and a type of SeriesType (as defined in the SeriesType section which follows)
      2. If the dimension at the observation level is AllDimensions, a local element named Obs with a form of unqualified, a maximum occurrence of unbounded, and a type of ObsType (as defined in the ObsType section which follows)
2. If the reporting year start day attribute is not declared in the data structure definition or if it is declared but does not declare an attribute relationship of None, an attribute named REPORTING\_YEAR\_START\_DAY with a type of xs:gMonthDay and a usage of prohibited
3. An attribute for each attribute defined in the data structured definition that declares an attribute relationship of None. The XML attribute [name](#_Component_Name_Determination) and [type](#_Simple_/_Primitive) are defined according to the general rules defined in the previous section, and the usage is optional

### GroupType

If the data structure definition defines only one group, a complex type with its name taken from the identifier of the lone group must be defined. This type is used for the Group element in the DataSetType. Its content model will be derived via restriction of the dsd:GroupType. The complex type content model will be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
2. An attribute for each dimension referenced by the group. The XML attribute [name](#_Component_Name_Determination) and [type](#_Simple_/_Primitive) are defined according to the general rules defined in the previous section, and the usage is required
3. If the reporting year start day attribute is not declared in the data structure definition or if it is declared but does not declare an attribute relationship with the group and does not specify the group as an attachment group, an attribute named REPORTING\_YEAR\_START\_DAY with a type of xs:gMonthDay and a usage of prohibited
4. An attribute for each attribute defined in the data structure that declares an attribute relationship with the group or specifies the group as an attachment group. The XML attribute [name](#_Component_Name_Determination) and [type](#_Simple_/_Primitive) are defined according to the general rules defined in the previous section, and the usage is optional
5. An attribute named type with a type of common:IDType, usage of optional, and a fixed value of the identifier of the group

If the data structure definitions defines more than one group, an abstract complex type with name GroupType must be created. This type is used for the Group element in the DataSetType. Its content model will be derived via restriction of the dsd:GroupType. The complex type content model will be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
2. An attribute named type with a type of Group.ID, and a usage of optional
3. An anyAttribute declaration with a namespace of ##local

A simple type named Group.ID must be created. This should restrict the common:IDType. For each group defined by the data structure definition, an enumeration will be created within the restriction with a value of the group identifier.

For each group defined in the data structure definition, a complex type with its name taken from the group identifier is defined. Its content model will be derived via restriction of the previously defined GroupType. The complex type content model will be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
2. An attribute for each dimension referenced by the group. The XML attribute [name](#_Component_Name_Determination) and [type](#_Simple_/_Primitive) are defined according to the general rules defined in the previous section, and the usage is required
3. If the reporting year start day attribute is not declared in the data structure definition or if it is declared but does not declare an attribute relationship with the group and does not specify the group as an attachment group, an attribute named REPORTING\_YEAR\_START\_DAY with a type of xs:gMonthDay and a usage of prohibited
4. An attribute for each attribute defined in the data structure that declares an attribute relationship with the group or specifies the group as an attachment group. The XML attribute [name](#_Component_Name_Determination) and [type](#_Simple_/_Primitive) are defined according to the general rules defined in the previous section, and the usage is optional
5. An attribute named type with a type of Group.ID, usage of optional, and a fixed value of the identifier of the group

### SeriesType

If the dimension at the observation is not AllDimensions, a complex type name SeriesType must be created. Its content model will be derived via restriction. If the dimension at the observation level is the time dimension (TIME\_PERIOD) the base type of the restriction is dsd:TimeSeriesType. Otherwise, the base type of the restriction is dsd:SeriesType. The complex type content model will be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
   2. A local element named Obs with a form of unqualified, a minimum occurrence of 0, a maximum occurrence of unbounded, and a type of ObsType (as defined in the ObsType section which follows)
2. An attribute for each dimension defined by the data structure definition, except for the dimension at the observation level. The XML attribute [name](#_Component_Name_Determination) and [type](#_Simple_/_Primitive) are defined according to the general rules defined in the previous section, and the usage is required
3. If the reporting year start day attribute is not declared in the data structure definition or if it is declared and declares an attribute relationship of None, or with a group, or the dimension at the observation level, or specifies a group as an attachment group, an attribute named REPORTING\_YEAR\_START\_DAY with a type of xs:gMonthDay and a usage of prohibited
4. An attribute for each attribute defined in the data structure that declares an attribute relationship with any dimension outside of the dimension at the observation level (so long as it does not also declare an attachment group). The XML attribute [name](#_Component_Name_Determination) and [type](#_Simple_/_Primitive) are defined according to the general rules defined in the previous section, and the usage is optional

### ObsType

A complex type name ObsType must be created. Its content model will be derived via restriction. If the dimension at the observation level is the time dimension (TIME\_PERIOD) the base type of the restriction is dsd:TimeSeriesObsType. Otherwise, the base type of the restriction is dsd:ObsType. If the explicit measure option is used, this complex type must be abstract. The complex type content model will be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
2. If the dimension at the observation level is not the time dimension (TIME\_PERIOD) an attribute named TIME\_PERIOD with a type of common:TimePeriodType and a usage of prohibited
3. If the dimension at the observation level is not the time dimension (TIME\_PERIOD) an attribute for the dimension at the observation level. The XML attribute [name](#_Component_Name_Determination) and [type](#_Simple_/_Primitive) are defined according to the general rules defined in the previous section, and the usage is required
4. An attribute for the primary measure (OBS\_VALUE) defined by the data structure definition. The XML attribute [name](#_Component_Name_Determination) and [type](#_Simple_/_Primitive) is defined according to the general rules defined in the previous section, and the usage is optional
5. An attribute for each attribute defined in the data structure that declares an attribute relationship with the dimension at the observation level or the primary measure (OBS\_VALUE). The XML attribute [name](#_Component_Name_Determination) and [type](#_Simple_/_Primitive) are defined according to the general rules defined in the previous section, and the usage is optional
6. An attribute named type. If the explicit measure option is not used, this attribute must have a type of common:IDType and a usage of prohibited. If the explicit measure option is used, this attribute must have a type of the simple type generated for the representation of the measure dimension that is the dimension at the observation level (this will be an simple type with enumerations with values of the concept identifiers which make up the concept scheme that is the representation of the measure dimension) a and a usage of optional

If the explicit measure option is used, then complex type must be created for every concept which make up the representation of the measure dimension which is the dimension at the observation level. The name of this complex type will be taken from the identifier of the concept. Its content model will be derived via restriction of the ObsType. The complex type content model will be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
2. If the core representation of the Concept differs from that of the primary measure, an attribute for the primary measure (OBS\_VALUE) defined by the data structure definition, except for the dimension at the observation level. The XML [name](#_Component_Name_Determination) is defined according to the general rules defined in the previous section, and the usage is optional. The type of the attribute is the type that is generated for the core representation of the measure concept. Note that this representation type must have an explicit derivation from the type resulting from the primary measure. For example, if the primary measure type is xyz:CodeType, then the simple type which is defined for the core representation of the concept must restrict xyz:CodeType or one another type which restricts it
3. An attribute named type with a type of the simple type generated for the representation of the measure dimension that is the dimension at the observation level, a usage of optional, and a fixed value of the concept identifier.

## Metadata Structure Specific Schema

One schema will be created for each metadata structure. This schema will define the contents of all report structures defined by the metadata structure. The recommended target namespace of the data structured specific schema is the URN of the metadata structure.

The rules for generating the data structure specific-schema are broken into sections based on the level within the structure (i.e. metadata set, metadata targets, reports, metadata attributes). These rules will recommend names for generated types, and will refer back to these names throughout the description. These names are simply recommendations that should produce a unique name, but there is no requirement to use them. When a name is required, it will be made clear this is the case.

### MetadataSetType

A complex type that must be named MetadataSetType must be created. Its content model will be derived via restriction of the base type, msd:MetadataSetType. The complex type content model will be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
   2. An element reference to common:Name, with a minimum occurrence of 0 and a maximum occurrence of unbounded
   3. A local element named DataProvider with the type common:DataProviderReferenceType, a form of unqualified and a minimum occurrence of 0
   4. A local element named Report with a type ReportType, a form of unqualified, a minimum occurrence of 0, and a maximum occurrence of unbounded.

### TargetType

An abstract complex type with a recommended name of TargetType must be created. Its content model will be derived via restriction of the msd:TargetType. The complex type content model will be as follows:

1. A sequence consisting of:
   1. A local element named ReferenceValue with a type of msd:ReferenceValueType, a form of unqualified, and a maximum occurrence of unbounded
2. An attribute named id with a type of Target.ID, and a usage of optional

A simple type with a recommended name of Target.ID must be created. This should restrict the common:IDType. For each report structure defined by the metadata structure definition, an enumeration will be created within the restriction with a value of the report structure identifier.

A simple type with a recommended name of [metadata target identifier].TargetObject.ID must be created. This should restrict the Target.ID type. For each target object defined by the metadata target, an enumeration will be created within the restriction with a value of the target object identifier.

An abstract complex type with a recommended name of [metadata target identifier].ReferenceValueType will be created. Its content model will be derived via restriction of the msd:ReferenceValueType. The complex type content model will be as follows:

1. A choice consisting of:
   1. A local element named ObjectReference with a type of common:ReferenceType, and a form of unqualified
   2. A local element named DataKey with a type of common:DataKeyType, and a form of unqualified
   3. A local element named DataSetReference with a type of common:SetReferenceType, and a form of unqualified
   4. A local element named ReportPeriod with a type of common:ObservationalTimePeriodType, and a form of unqualified
2. An attribute named id with the type defined previously as [metadata target identifier].TargetObject.ID, and a usage of optional

For each metadata target defined in the metadata structure definition, a content group with a recommended name of [metadata target identifier] with be defined. Its content model will be as follows:

1. A sequence consisting of:
   1. A local element named ReferenceValue with the type previously defined as [metadata target identifier].ReferenceValueType, a form of unqualified, and a minimum and maximum occurrence of the number of target objects defined within the metadata target

For each target object defined by the metadata target, a complex type that must be named [metadata target identifier].[target object identifier] must be created. Its content model will be derived via restriction of the type previously created as [metadata target identifier].ReferenceValueType type created prior to this. The complex type content model will be as follows:

1. A choice consisting of:
   1. If the target object is an identifiable object target, a local element named ObjectReference with a type determined as defined in Identifiable Object Target Type Determination:
   2. If the target object is an key descriptor values target, a local element named DataKey with a type of common:DataKeyType, and a form of unqualified
   3. If the target object is a data set target, a local element named DataSetReference with a type of common:SetReferenceType, and a form of unqualified
   4. If the target object is a constraint target, a local element named ConstraintReference with a type of common:AttachmentConstraintReferenceType, and a form of unqualified
   5. If the target object is report period target, a local element named ReportPeriod with a form of unqualified, and a [type](#_Simple_/_Primitive) as defined based on the data type of the local representation according to the general rules defined in the previous section
2. An attribute named id with a type previously created as [metadata target identifier].TargetObject.ID, a usage of optional, and fixed value of the target object identifier

The final complex type (which will be the determined type) is derived by restriction from the reference type determined above. The suggested name of this type is [metadata target identifier].[target object identifier].Reference. The content model of this type will be as follows:

1. A choice consisting of:
   1. A sequence consisting of
      1. A local element named Ref, with a form of unqualified and a the type defined above as is [metadata target].[target object].Ref
      2. A local element named URN, with a form of unqualified, a minimum occurrence of 0, and the type defined above as [item scheme agency].[item scheme identifier].[item scheme version].URN
   2. A local element named URN, with a form of unqualified, and the type defined above as [item scheme agency].[item scheme identifier].[item scheme version].URN

#### Identifiable Object Target Type Determination

An identifiable object target is meant to reference an SDMX identifiable object. In the metadata structure specific schema, the complex type which defines the reference is restricted according to the local representation and object type of the identifiable object target. If the identifiable object target does not provide an enumeration in its local representation, then the complex type for the target object is the specific reference type from the common namespace, based on the object type defined by the target object. For example, if the object type is DataProivder, the type will be common:DataProviderReferenceType. The general rule for the name of this type is common:[object type]ReferenceType.

If the target object defines an enumeration in its local representation, then the declared object type must be for the items contained the enumeration's referenced item scheme. If this is not true, the reference type as defined above is the type for the target object.

If the object type does correspond with the enumeration item scheme, then the type is complex type which restricts the reference to the allowable values. In order to construct this type, the following must be created:

A simple type which enumerates with URNs of the items contained in the items scheme referenced from the enumeration. The recommended name is [item scheme agency].[item scheme identifier].[item scheme version].URN. This simple type should restrict the xs:anyURI type an contain enumerations with values of the URN of every item defined by the scheme.

A second simple type which enumerates the IDs of the items contained in the items scheme referenced from the enumeration. The recommended name is [item scheme agency].[item scheme identifier].[item scheme version].ID. This simple type should restrict the either the common:NestedIDType for hierarchical items or the common:IDType for flat items. The restriction should contain enumerations with values of the ID of every item defined by the scheme. For hierarchical items (such as categories), the id should be nested to reflect the full path (e.g. A.C.D.F).

A complex type which restricts the full set of reference fields for the item to be reference must be created. The recommended name is [metadata target].[target object].Ref. This type is derived via restriction of the type which defines the full set of reference fields for the item referenced by the target object. For example, if the object type is Category, the type that is the base of the restriction is common:CategoryRefType. . The general rule for the name of this type is common:[object type]RefType. The content model of this type must be as follows:

1. An attribute named agencyID with a type of common:NestNCNameIDType, a use of required, and a fixed value of the item scheme referenced from the enumeration
2. An attribute named maintinableParentID with a type of common:IDType, a use of required, and a fixed value of the identifier of the item scheme referenced from the enumeration
3. An attribute named maintainableParentVersion with a type of common:VersionType, a use of optional, and a fixed value of the version of the item scheme referenced from the enumeration
4. An attribute named id, with a use of option, and the type defined above as [item scheme agency].[item scheme identifier].[item scheme version].ID.

### ReportType

A simple type with a recommend name of Report.ID must be created. This must restrict the common:IDType. For each report structure defined by the metadata structure definition, an enumeration will be created within the restriction with a value of the report structure identifier.

An abstract complex type with a recommended name of ReportType must be created. Its content model will be derived via restriction of the msd:ReportType. The complex type content model will be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
   2. A local element named Target with the type previously defined as TargetType and a form of unqualified
   3. A local element named AttributeSet with no type assigned and a form of unqualified
2. An attribute named id with a type previously defined as Report.ID, and a usage of optional

For each report structure defined in the metadata structure definition, a complex type with its name taken from the report structure identifier must be defined. In order to construct this type, some other complex type must be defined first.

A simple type with a recommended name of [report structure identifier].Target.ID must be created. This must restrict the type previously created as Target.ID. For each metadata target referenced by the report structure, an enumeration will be created within the restriction with a value of the metadata target identifier.

An abstract complex type must be create, with a recommended name of [report structure identifier].TargetType. Its content model will be derived by restriction of the type previously created as TargetType. The content model of this type must be as follows:

1. A sequence consisting of:
   1. A local element named ReferenceValue with a type of msd:ReferenceValueType, a form of unqualified, and a maximum occurrence of unbounded
2. An attribute named id with the type previously created as [report structure identifier].Target.ID, and a use of optional

For each metadata target referenced from the report structure, a complex type that must be named [report structure identifier].[metadata target identifier] must be created. This type must derived its content by restriction of the type previously defined as [report structure identifier].TargetType. The content model is as follows:

1. A reference to the content group that was previously created as [metadata target identifier] for this metadata target
2. An attribute named id with the type create previously as [report structure identifier].Target.ID, a use of optional, and a fixed value of [metadata target identifier]

For every metadata attribute defined by the report structure a complex type must be created with the recommended name of [report structure identifier].[nested metadata attribute identifier]. Note that the recommend name assumes the metadata attribute identifier is the nested identifier for the full hierarchy in which the metadata attribute was defined. This content model of this type is derived from restriction of the msd:ReportedAttributeType. The content of this must consists be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
   2. If the isPresentational flag is not set to true:
      1. If the [determined representation](#_Representation_Determination:) has data type of XHTML, an element reference to common:StructuredText with a maximum occurrence of unbounded
      2. If the [determined representation](#_Representation_Determination:) has is not XHTML, but has a facet of isMultiLinguale, an element reference to common: Text with a maximum occurrence of unbounded
   3. If the metadata attribute defines which child metadata attributes, a local element named AttributeSet with a form of unqualified, and a type as defined according to the rules defined in Attribute Set Complex Type Creation
2. An attribute named id with a type of common:IDType, a use of optional, and a fixed value of [metadata attribute identifier], where this identifier is not nested
3. If the isPresentational flag is not set to true, and the [determined representation](#_Representation_Determination:) is not XHTML and does not have a facet of isMultiLingual, an attribute named value where the [type](#_Simple_/_Primitive) is defined according to the general rules defined in the previous section, and the use is required

For each report structure defined in the metadata structure, a complex type that must be named [report structure identifier] must be created. Its content model will be derived via restriction of the previously defined ReportType. The complex type content model will be as follows:

1. A sequence consisting of:
   1. An element reference to common:Annotations, with a minimum occurrence of 0
   2. A local element named Target with the type previously defined as [report structure identifier].TargetType, and a form of unqualified. This element must define a uniqueness constraint with a recommended name of [report structure identifier].Target.ReferenceValue.Unique. This uniqueness constraint must contain:
      1. A selector with an xpath of "\*"
      2. A field with an xpath of "@id"
   3. A local element named AttributeSet with a form of unqualified and a type as defined according to the rules defined in Attribute Set Complex Type Creation
2. An attribute named id with the type previously defined as Report.ID, a use of optional, and a fixed value of [report structure identifier]

#### Attribute Set Complex Type Creation

An attribute set is created for a report structure or a metadata attribute which contains child metadata attributes. This type is a complex type with a recommend name of [report structure identifier].[nested metadata attribute identifier].AttributeSet, where the metadata attribute section only applies to attribute sets created for metadata attributes and is assumed to consist of the full hierarchy of the metadata attribute's definition. This complex type must consists of:

1. A sequence consisting of:
   1. A local element for each metadata attribute defined as a direct child of the object for which the attribute set is being created. The recommended name of this element is [metadata attribute identifier] and it has a form of unqualified. If the metadata attribute defines a minOccurs and maxOccurs, these values are translated to this element. The type of this element is the type previously created as [report structure identifier].[nested metadata attribute identifier].

# Special data functions

## Updates

Both the generic and the structure-specific data messages allow for incremental updating of data. This purpose is noted in the action for the data set, which is either inherited from the header of the data message or explicitly stated at the data set level.

A dataset with an action of Append is assumed to be an incremental update. This means that one the information provided explicitly in the message should be altered. Any data attribute or observation value that is to be changed must be provided. However, the absence of an observation value or a data attribute at any level does not imply deletion; instead it is simply implied that the value is to remain unchanged. Therefore, it is valid and acceptable to send a data message with an action of Append which contains only a Series elements with attribute values. In this case, the values for the attributes will be updated. Note that it is not permissible to update data attributes using partial keys (outside of those associated with defined groups). In order to update an attribute, a full key must always be provided even if the message format does not require this.

## Deletes

Both the generic and the structure-specific data messages allow for incremental deletion of data. This purpose is noted in the action for the data set, which is either inherited from the header of the data message or explicitly stated at the data set level.

A dataset with an action of Delete is assumed to be an incremental deletion. The deletion is assumed to take place of the lowest level of detail provided in the message. For example, if a delete message is sent with only a data set element, the entire data set will be deleted. On the other hand, if that data set contains a data attribute, only that data attribute value will be deleted. This same dynamic continues through the data set hierarchy. A data set containing only a series with no data attributes or observations will result in that entire series (all observations and data attributes) being deleted. If the series contains data attributes, only the supplied data attributes for that series will be deleted. Finally, if a series contains observations, then only the specified observations will be deleted. If an entire observation is to be deleted (value and data attributes), only the observation dimension should be provided. If only the observation value or particular data attributes are to be deleted, then these should be specified for the observation. Note that a group can only be used to delete the data attributes associated with it. Although the format might not require it, a full key must be provided to delete a series or observation. It is not permissible to wild card a key in order to delete more than one series or observation. Finally, to delete a data attribute or observation value it is recommended that the value to be deleted be supplied; however, it is only required that any valid value be provided.

1. These types are only used in complex types and will be discussed within their appropriate context. [↑](#footnote-ref-1)
2. Note that these options only apply to numeric representations and should be ignored if the data type is non-numeric [↑](#footnote-ref-2)