SDMX Standards: Section 3A PaRT II

SDMX-ML:

Schema and Documentation

Common Namespace

Version 3.0

September 2021

© SDMX 2021

http://www.sdmx.org/

Contents

[1 Introduction 1](#_Toc82185849)

[2 Schema Documentation 1](#_Toc82185850)

[2.1 Common Namespace 1](#_Toc82185851)

[2.1.1 Summary 1](#_Toc82185852)

[2.1.2 Global Elements 2](#_Toc82185853)

[2.1.3 Complex Types 2](#_Toc82185854)

[2.1.4 Simple Types 26](#_Toc82185855)

# Introduction

The common namespace defines a collection of constructs that are reused across the various components of SDMX. Most important of these are the referencing mechanism. The goal of the reference construct was to define a generic structure that could be processed uniformly regardless of the context where the reference was used. But it was also important that references be required to be complete whenever possible.

Any object can be referenced either explicitly with a URN or by a set of complete reference fields. To meet the previously stated requirements, and very general mechanism was created based on the URN structure of SDMX objects for these reference fields.

There was also a requirement that the references be able to be refined to meet particular needs outside of the common namespace. An example of this is in the metadata structure specific schemas. It is a requirement that if an target object is specified as having to come from a particular scheme, that a type based on the reference structure be created that enforced the requirement.

Typically, this would not have been an issues as all of the components which make up the references are of atomic types, and therefore can be expressed as XML attributes which are easily refined and restricted since the XML Schema design principles in SDMX always treats attributes as unqualified.

However, the requirement to allow both a URN and/or a complete set of reference field necessitate that these properties be contained in elements. The fact that they are elements typically would mean that the only way a refinement outside of the namespace could happen was if the element were global and allowed for substitutions. This however would mean that every distinct type of referenced object would have a unique set of elements. This moved away from the requirement that the structure be easy to process regardless of context.

The solution to this problem was to deviate from the normal SDMX XML Schema design principle of always using qualified elements and allowing for these to be unqualified. Doing so allows other namespace to derive from these types and place further restrictions on what can be referenced. The deviation from this principle was justified in that it met the all of the requirements and was not deemed to major of a shift since these properties normally would have been expressed as unqualified attributes if it weren't for the complete reference requirement.

# Schema Documentation

## Common Namespace

**http://www.sdmx.org/resources/sdmxml/schemas/v3\_0/common**

### Summary

Referenced Namespaces:

| **Namespace** | **Prefix** |
| --- | --- |
| http://www.w3.org/1999/xhtml | xhtml |
| http://www.w3.org/2001/XMLSchema | xs |

Contents:

6 Global Elements  
30 Complex Types  
206 Simple Types

### Global Elements

**Name (TextType):** Name is a reusable element, used for providing a human-readable name for an object.

**Description (TextType):** Description is a reusable element, used for providing a longer human-readable description of an object.

**Text (TextType):** Text is a reusable element, used for providing a language specific text value for general purposes (i.e. not for a name or description).

**StructuredText (XHTMLType):** StructuredText is a reusable element, used for providing a language specific text value structured as XHTML.

**Annotations (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 may reference it.

**Link (LinkType):** Allows for the linking of other resources to identifiable objects. For example, if there is reference metadata associated with a structure, a link to the meatadata report can be dynamically inserted in the structure metadata.

### Complex Types

***ValueType*:** ValueType is an abstract class that is the basis for any component value that cannot be simply represented as a space-normalized value (e.g. in an XML attribute). Although its content is mixed, it should be restricted so that only character data or the Text or Structured text is used. See StringValueType, IntValueType, ObserverationalTimeValueType, TextValueType, and StructuredTextValueType for details.

Content:

{text} x (Text\* | StructuredText\*)?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Text | TextType | Text is a reusable element, used for providing a language specific text value for general purposes (i.e. not for a name or description). |
| StructuredText | XHTMLType | StructuredText is a reusable element, used for providing a language specific text value structured as XHTML. |

**BooleanValueType:** BooleanValueType is a refinement of SimpleValueType limiting the content to be a boolean.

Derivation:

*ValueType* (restriction)   
   BooleanValueType

Content:

**StringValueType:** StringValueType is a refinement of SimpleValueType limiting the content to be a string. This can be further refined with facets, patterns, etc.

Derivation:

*ValueType* (restriction)   
   StringValueType

Content:

**IntValueType:** IntValueType is a refinement of SimpleValueType limiting the content to be an integer. This can be further refined ranges, etc.

Derivation:

*ValueType* (restriction)   
   IntValueType

Content:

**DoubleValueType:** DoubleValueType is a refinement of SimpleValueType limiting the content to be a double. This can be further refined ranges, etc.

Derivation:

*ValueType* (restriction)   
   DoubleValueType

Content:

**ObservationalTimePeriodValueType:** ObservationalTimePeriodValueType is a refinement of SimpleValueType limiting the content to be an observational time period.

Derivation:

*ValueType* (restriction)   
   ObservationalTimePeriodValueType

Content:

**TextValueType:** TextValueType is a restriction of ValueType that allows mutliple Text elements to expressed a text value in multiple languages. The content of this should be restricted in its use to only allow a langue code (xml:lang) to be used once within an element of this type.

Derivation:

*ValueType* (restriction)   
   TextValueType

Content:

Text\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Text | TextType | Text is a reusable element, used for providing a language specific text value for general purposes (i.e. not for a name or description). |

**StructuredTextValueType:** StructuredTextValueType is a restriction of ValueType that allows mutliple StructuredText (XHTML mixed content) elements to expressed a text value in multiple languages. The content of this should be restricted in its use to only allow a langue code (xml:lang) to be used once within an element of this type.

Derivation:

*ValueType* (restriction)   
   StructuredTextValueType

Content:

StructuredText\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| StructuredText | XHTMLType | StructuredText is a reusable element, used for providing a language specific text value structured as XHTML. |

**TextType:** TextType provides for a set of language-specific alternates to be provided for any human-readable constructs in the instance.

Derivation:

xs:anySimpleType (restriction)   
   xs:string (extension)   
         TextType

Attributes:

xml:lang?

Content:

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| xml:lang (default: en) | xs:language | The xml:lang attribute specifies a language code for the text. If not supplied, the default language is assumed to be English. |

**StatusMessageType:** StatusMessageType describes the structure of an error or warning message. A message contains the text of the message, as well as an optional language indicator and an optional code.

Attributes:

code?

Content:

Text+

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| code | xs:string | The code attribute holds an optional code identifying the underlying error that generated the message. This should be used if parallel language descriptions of the error are supplied, to distinguish which of the multiple error messages are for the same underlying error. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Text | TextType | Text contains the text of the message, in parallel language values. |

**EmptyType:** EmptyType is an empty complex type for elements where the presence of the tag indicates all that is necessary.

Content:

{Empty}

**CodedStatusMessageType:** CodedStatusMessageType describes the structure of an error or warning message which required a code.

Derivation:

StatusMessageType (restriction)   
   CodedStatusMessageType

Attributes:

code

Content:

Text+

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| code | xs:string | The code attribute holds an optional code identifying the underlying error that generated the message. This should be used if parallel language descriptions of the error are supplied, to distinguish which of the multiple error messages are for the same underlying error. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Text | TextType | Text contains the text of the message, in parallel language values. |

***AnnotableType*:** AnnotableType is an abstract base type used for all annotable artefacts. Any type that provides for annotations should extend this type.

Content:

Annotations?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotations | 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 may reference it. |

**AnnotationsType:** AnnotationsType provides for a list of annotations to be attached to data and structure messages.

Content:

Annotation+

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationType |  |

**AnnotationType:** AnnotationType provides for non-documentation notes and annotations to be embedded in data and structure messages. It provides optional fields for providing a title, a type description, a URI, and the text of the annotation.

Attributes:

id?

Content:

AnnotationTitle?, AnnotationType?, AnnotationURL\*, AnnotationText\*, AnnotationValue?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | xs:string | The id attribute provides a non-standard identification of an annotation. It can be used to disambiguate annotations. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| AnnotationTitle | xs:string | AnnotationTitle provides a title for the annotation. |
| AnnotationType | xs:string | AnnotationType is used to distinguish between annotations designed to support various uses. The types are not enumerated, as these can be specified by the user or creator of the annotations. The definitions and use of annotation types should be documented by their creator. |
| AnnotationURL | AnnotationURLType | AnnotationURL is a URI - typically a URL - which points to an external resource which may contain or supplement the annotation. These can be localised by indicating a language for the resource. If a language is not specified, the resource is assumed to not be localised. If a specific behavior is desired, an annotation type should be defined which specifies the use of this field more exactly. |
| AnnotationText | TextType | AnnotationText holds a language-specific string containing the text of the annotation. |
| AnnotationValue | xs:string | AnnotationValue holds a non-localised value for the annotation. |

**AnnotationURLType:** AnnotationURLType defines an external resource. These can indicate localisation by specifying a language for the resource.

Derivation:

xs:anySimpleType (restriction)   
   xs:anyURI (extension)   
         AnnotationURLType

Attributes:

xml:lang?

Content:

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| xml:lang | xs:language | Indicates the language of the resources at the URL, if it is localised. If this does not exist, the resource is not localised. |

**LinkType:**

Attributes:

rel, url, urn?, type?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| rel | xs:string | The type of object that is being linked to. |
| url | xs:anyURI | The url of the object being linked. |
| urn | xs:anyURI | A SDMX registry urn of the object being linked (if applicable). |
| type | xs:string | The type of link (e.g. PDF, text, HTML, reference metadata). |

***IdentifiableType*:** IdentifiableType is an abstract base type for all identifiable objects.

Derivation:

*AnnotableType* (extension)   
   *IdentifiableType*

Attributes:

id?, urn?, uri?

Content:

Annotations?, Link\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id is the identifier for the object. |
| urn | UrnType | The urn attribute holds a valid SDMX Registry URN (see SDMX Registry Specification for details). |
| uri | xs:anyURI | The uri attribute holds a URI that contains a link to a resource with additional information about the object, such as a web page. This uri is not a SDMX message. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotations | 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 may reference it. |
| Link | LinkType | Allows for the linking of other resources to identifiable objects. For example, if there is reference metadata associated with a structure, a link to the meatadata report can be dynamically inserted in the structure metadata. |

***NameableType*:** NameableType is an abstract base type for all nameable objects.

Derivation:

*AnnotableType* (extension)   
   *IdentifiableType* (extension)   
         *NameableType*

Attributes:

id?, urn?, uri?

Content:

Annotations?, Link\*, Name+, Description\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id is the identifier for the object. |
| urn | UrnType | The urn attribute holds a valid SDMX Registry URN (see SDMX Registry Specification for details). |
| uri | xs:anyURI | The uri attribute holds a URI that contains a link to a resource with additional information about the object, such as a web page. This uri is not a SDMX message. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotations | 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 may reference it. |
| Link | LinkType | Allows for the linking of other resources to identifiable objects. For example, if there is reference metadata associated with a structure, a link to the meatadata report can be dynamically inserted in the structure metadata. |
| Name | TextType | Name provides for a human-readable name for the object. This may be provided in multiple, parallel language-equivalent forms. |
| Description | TextType | Description provides for a longer human-readable description of the object. This may be provided in multiple, parallel language-equivalent forms. |

***VersionableType*:** VersionableType is an abstract base type for all versionable objects.

Derivation:

*AnnotableType* (extension)   
   *IdentifiableType* (extension)   
         *NameableType* (extension)   
               *VersionableType*

Attributes:

id?, urn?, uri?, version?, validFrom?, validTo?

Content:

Annotations?, Link\*, Name+, Description\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id is the identifier for the object. |
| urn | UrnType | The urn attribute holds a valid SDMX Registry URN (see SDMX Registry Specification for details). |
| uri | xs:anyURI | The uri attribute holds a URI that contains a link to a resource with additional information about the object, such as a web page. This uri is not a SDMX message. |
| version | VersionType | This version attribute holds a version number (see common:VersionType definition for details). If not supplied, artefact is considered to be un-versioned. |
| validFrom | xs:dateTime | The validFrom attribute provides the inclusive start date for providing supplemental validity information about the version. |
| validTo | xs:dateTime | The validTo attribute provides the inclusive end date for providing supplemental validity information about the version. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotations | 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 may reference it. |
| Link | LinkType | Allows for the linking of other resources to identifiable objects. For example, if there is reference metadata associated with a structure, a link to the meatadata report can be dynamically inserted in the structure metadata. |
| Name | TextType | Name provides for a human-readable name for the object. This may be provided in multiple, parallel language-equivalent forms. |
| Description | TextType | Description provides for a longer human-readable description of the object. This may be provided in multiple, parallel language-equivalent forms. |

***MaintainableBaseType*:** MaintainableBaseType is an abstract type that only serves the purpose of forming the base for the actual MaintainableType. The purpose of this type is to restrict the VersionableType to require the id attribute.

Derivation:

*AnnotableType* (extension)   
   *IdentifiableType* (extension)   
         *NameableType* (extension)   
               *VersionableType* (restriction)   
                     *MaintainableBaseType*

Attributes:

id, urn?, uri?, version?, validFrom?, validTo?

Content:

Annotations?, Link\*, Name+, Description\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id is the identifier for the object. |
| urn | MaintainableUrnType | The urn attribute holds a valid SDMX Registry URN (see SDMX Registry Specification for details). |
| uri | xs:anyURI | The uri attribute holds a URI that contains a link to a resource with additional information about the object, such as a web page. This uri is not a SDMX message. |
| version | VersionType | This version attribute holds a version number (see common:VersionType definition for details). If not supplied, artefact is considered to be un-versioned. |
| validFrom | xs:dateTime | The validFrom attribute provides the inclusive start date for providing supplemental validity information about the version. |
| validTo | xs:dateTime | The validTo attribute provides the inclusive end date for providing supplemental validity information about the version. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotations | 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 may reference it. |
| Link | LinkType | Allows for the linking of other resources to identifiable objects. For example, if there is reference metadata associated with a structure, a link to the meatadata report can be dynamically inserted in the structure metadata. |
| Name | TextType | Name provides for a human-readable name for the object. This may be provided in multiple, parallel language-equivalent forms. |
| Description | TextType | Description provides for a longer human-readable description of the object. This may be provided in multiple, parallel language-equivalent forms. |

***MaintainableType*:** MaintainableType is an abstract base type for all maintainable objects.

Derivation:

*AnnotableType* (extension)   
   *IdentifiableType* (extension)   
         *NameableType* (extension)   
               *VersionableType* (restriction)   
                     *MaintainableBaseType* (extension)   
                           *MaintainableType*

Attributes:

id, urn?, uri?, version?, validFrom?, validTo?, agencyID, isExternalReference?, serviceURL?, structureURL?

Content:

Annotations?, Link\*, Name+, Description\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id is the identifier for the object. |
| urn | MaintainableUrnType | The urn attribute holds a valid SDMX Registry URN (see SDMX Registry Specification for details). |
| uri | xs:anyURI | The uri attribute holds a URI that contains a link to a resource with additional information about the object, such as a web page. This uri is not a SDMX message. |
| version | VersionType | This version attribute holds a version number (see common:VersionType definition for details). If not supplied, artefact is considered to be un-versioned. |
| validFrom | xs:dateTime | The validFrom attribute provides the inclusive start date for providing supplemental validity information about the version. |
| validTo | xs:dateTime | The validTo attribute provides the inclusive end date for providing supplemental validity information about the version. |
| agencyID | NestedNCNameIDType | The agencyID must be provided, and identifies the maintenance agency of the object. |
| isExternalReference (default: false) | xs:boolean | The isExternalReference attribute, if true, indicates that the actual object is not defined the corresponding element, rather its full details are defined elsewhere - indicated by either the registryURL, the repositoryURL, or the structureURL. The purpose of this is so that each structure message does not have to redefine object that are already defined elsewhere. If the isExternalReference attribute is not set, then it is assumed to be false, and the object should contain the full definition of its contents. If more than one of the registryURL, the repositoryURL, and the structureURL are supplied, then the application processing the object can choose the method it finds best suited to retrieve the details of the object. |
| serviceURL | xs:anyURI | The serviceURL attribute indicates the URL of an SDMX SOAP web service from which the details of the object can be retrieved. Note that this can be a registry or and SDMX structural metadata repository, as they both implement that same web service interface. |
| structureURL | xs:anyURI | The structureURL attribute indicates the URL of a SDMX-ML structure message (in the same version as the source document) in which the externally referenced object is contained. Note that this may be a URL of an SDMX RESTful web service which will return the referenced object. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotations | 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 may reference it. |
| Link | LinkType | Allows for the linking of other resources to identifiable objects. For example, if there is reference metadata associated with a structure, a link to the meatadata report can be dynamically inserted in the structure metadata. |
| Name | TextType | Name provides for a human-readable name for the object. This may be provided in multiple, parallel language-equivalent forms. |
| Description | TextType | Description provides for a longer human-readable description of the object. This may be provided in multiple, parallel language-equivalent forms. |

**ReferencePeriodType:** Specifies the inclusive start and end times.

Attributes:

startTime, endTime

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| startTime | xs:dateTime | The startTime attributes contains the inclusive start date for the reference period. |
| endTime | xs:dateTime | The endTime attributes contains the inclusive end date for the reference period. |

**QueryableDataSourceType:** QueryableDataSourceType describes a data source which is accepts an standard SDMX Query message and responds appropriately.

Attributes:

isRESTDatasource, isWebServiceDatasource

Content:

DataURL, WSDLURL?, WADLURL?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| isRESTDatasource | xs:boolean | The isRESTDatasource attribute indicates, if true, that the queryable data source is accessible via the REST protocol. |
| isWebServiceDatasource | xs:boolean | The isWebServiceDatasource attribute indicates, if true, that the queryable data source is accessible via Web Services protocols. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| DataURL | xs:anyURI | DataURL contains the URL of the data source. |
| WSDLURL | xs:anyURI | WSDLURL provides the location of a WSDL instance on the internet which describes the queryable data source. |
| WADLURL | xs:anyURI | WADLURL provides the location of a WADL instance on the internet which describes the REST protocol of the queryable data source. |

**XHTMLType:** XHTMLType allows for mixed content of text and XHTML tags. When using this type, one will have to provide a reference to the XHTML schema, since the processing of the tags within this type is strict, meaning that they are validated against the XHTML schema provided.

Attributes:

xml:lang?

Content:

{text} x {any element with namespace of http://www.w3.org/1999/xhtml}\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| xml:lang (default: en) | xs:language |  |

***PayloadStructureType*:** PayloadStructureType is an abstract base type used to define the structural information for data or metadata sets. A reference to the structure is provided (either explicitly or through a reference to a structure usage).

Attributes:

structureID, schemaURL?, namespace?, dimensionAtObservation?, explicitMeasures?, serviceURL?, structureURL?

Content:

(ProvisionAgreement | StructureUsage | Structure)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureID | xs:ID | The structureID attribute uniquely identifies the structure for the purpose of referencing it from the payload. This is only used in structure specific formats. Although it is required, it is only useful when more than one data set is present in the message. |
| schemaURL | xs:anyURI | The schemaURL attribute provides a location from which the structure specific schema can be located. |
| namespace | xs:anyURI | The namespace attribute is used to provide the namespace for structure-specific formats. By communicating this information in the header, it is possible to generate the structure specific schema while processing the message. |
| dimensionAtObservation | ObservationDimension Type | The dimensionAtObservation is used to reference the dimension at the observation level for data messages. This can also be given the explicit value of "AllDimensions" which denotes that the cross sectional data is in the flat format. |
| explicitMeasures | xs:boolean | The explicitMeasures indicates whether explicit measures are used in the cross sectional format. This is only applicable for the measure dimension as the dimension at the observation level or the flat structure. |
| serviceURL | xs:anyURI | The serviceURL attribute indicates the URL of an SDMX SOAP web service from which the details of the object can be retrieved. Note that this can be a registry or and SDMX structural metadata repository, as they both implement that same web service interface. |
| structureURL | xs:anyURI | The structureURL attribute indicates the URL of a SDMX-ML structure message (in the same version as the source document) in which the externally referenced object is contained. Note that this may be a URL of an SDMX RESTful web service which will return the referenced object. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ProvisionAgreement | ProvisionAgreementRe ferenceType | ProvisionAgreement references a provision agreement which the data or metadata is reported against. |
| StructureUsage | StructureUsageRefere nceType | StructureUsage references a flow which the data or metadata is reported against. |
| Structure | StructureReferenceTy pe | Structure references the structure which defines the structure of the data or metadata set. |

***DataStructureType*:** DataStructureType is an abstract base type the forms the basis for the structural information for a data set.

Derivation:

*PayloadStructureType* (restriction)   
   *DataStructureType*

Attributes:

structureID, schemaURL?, namespace?, dimensionAtObservation?, explicitMeasures?, serviceURL?, structureURL?

Content:

(ProvisionAgreement | StructureUsage | Structure)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureID | xs:ID | The structureID attribute uniquely identifies the structure for the purpose of referencing it from the payload. This is only used in structure specific formats. Although it is required, it is only useful when more than one data set is present in the message. |
| schemaURL | xs:anyURI | The schemaURL attribute provides a location from which the structure specific schema can be located. |
| namespace | xs:anyURI | The namespace attribute is used to provide the namespace for structure-specific formats. By communicating this information in the header, it is possible to generate the structure specific schema while processing the message. |
| dimensionAtObservation | ObservationDimension Type | The dimensionAtObservation is used to reference the dimension at the observation level for data messages. This can also be given the explicit value of "AllDimensions" which denotes that the cross sectional data is in the flat format. |
| explicitMeasures | xs:boolean | The explicitMeasures indicates whether explicit measures are used in the cross sectional format. This is only applicable for the measure dimension as the dimension at the observation level or the flat structure. |
| serviceURL | xs:anyURI | The serviceURL attribute indicates the URL of an SDMX SOAP web service from which the details of the object can be retrieved. Note that this can be a registry or and SDMX structural metadata repository, as they both implement that same web service interface. |
| structureURL | xs:anyURI | The structureURL attribute indicates the URL of a SDMX-ML structure message (in the same version as the source document) in which the externally referenced object is contained. Note that this may be a URL of an SDMX RESTful web service which will return the referenced object. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ProvisionAgreement | ProvisionAgreementRe ferenceType | ProvisionAgreement references a provision agreement which the data is reported against. |
| StructureUsage | DataflowReferenceTyp e | StructureUsage references a dataflow which the data is reported against. |
| Structure | DataStructureReferen ceType | Structure references the data structure definition which defines the structure of the data. |

**StructureSpecificDataStructureType:** StructureSpecificDataStructureType defines the structural information for a structured data set. In addition to referencing the data structure or dataflow which defines the structure of the data, the namespace for the data structure specific schema as well as which dimension is used at the observation level must be provided. It is also necessary to state whether the format uses explicit measures, although this is technically only applicable is the dimension at the observation level is the measure dimension or the flat data format is used.

Derivation:

*PayloadStructureType* (restriction)   
   *DataStructureType* (restriction)   
         StructureSpecificDataStructureType

Attributes:

structureID, schemaURL?, namespace, dimensionAtObservation, explicitMeasures?, serviceURL?, structureURL?

Content:

(ProvisionAgreement | StructureUsage | Structure)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureID | xs:ID | The structureID attribute uniquely identifies the structure for the purpose of referencing it from the payload. This is only used in structure specific formats. Although it is required, it is only useful when more than one data set is present in the message. |
| schemaURL | xs:anyURI | The schemaURL attribute provides a location from which the structure specific schema can be located. |
| namespace | xs:anyURI | The namespace attribute is used to provide the namespace for structure-specific formats. By communicating this information in the header, it is possible to generate the structure specific schema while processing the message. |
| dimensionAtObservation | ObservationDimension Type | The dimensionAtObservation is used to reference the dimension at the observation level for data messages. This can also be given the explicit value of "AllDimensions" which denotes that the cross sectional data is in the flat format. |
| explicitMeasures (default: false) | xs:boolean | The explicitMeasures indicates whether explicit measures are used in the cross sectional format. This is only applicable for the measure dimension as the dimension at the observation level or the flat structure. |
| serviceURL | xs:anyURI | The serviceURL attribute indicates the URL of an SDMX SOAP web service from which the details of the object can be retrieved. Note that this can be a registry or and SDMX structural metadata repository, as they both implement that same web service interface. |
| structureURL | xs:anyURI | The structureURL attribute indicates the URL of a SDMX-ML structure message (in the same version as the source document) in which the externally referenced object is contained. Note that this may be a URL of an SDMX RESTful web service which will return the referenced object. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ProvisionAgreement | ProvisionAgreementRe ferenceType | ProvisionAgreement references a provision agreement which the data is reported against. |
| StructureUsage | DataflowReferenceTyp e | StructureUsage references a dataflow which the data is reported against. |
| Structure | DataStructureReferen ceType | Structure references the data structure definition which defines the structure of the data. |

***MetadataStructureType*:** MetadataStructureType is an abstract base type the forms the basis of the structural information for any metadata message. A reference to the metadata structure definition or a metadataflow must be provided. This can be used to determine the structure of the message.

Derivation:

*PayloadStructureType* (restriction)   
   *MetadataStructureType*

Attributes:

structureID, schemaURL?, namespace?, serviceURL?, structureURL?

Content:

(ProvisionAgreement | StructureUsage | Structure)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureID | xs:ID | The structureID attribute uniquely identifies the structure for the purpose of referencing it from the payload. This is only used in structure specific formats. Although it is required, it is only useful when more than one data set is present in the message. |
| schemaURL | xs:anyURI | The schemaURL attribute provides a location from which the structure specific schema can be located. |
| namespace | xs:anyURI | The namespace attribute is used to provide the namespace for structure-specific formats. By communicating this information in the header, it is possible to generate the structure specific schema while processing the message. |
| serviceURL | xs:anyURI | The serviceURL attribute indicates the URL of an SDMX SOAP web service from which the details of the object can be retrieved. Note that this can be a registry or and SDMX structural metadata repository, as they both implement that same web service interface. |
| structureURL | xs:anyURI | The structureURL attribute indicates the URL of a SDMX-ML structure message (in the same version as the source document) in which the externally referenced object is contained. Note that this may be a URL of an SDMX RESTful web service which will return the referenced object. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ProvisionAgreement | ProvisionAgreementRe ferenceType | ProvisionAgreement references a provision agreement which the metadata is reported against. |
| StructureUsage | MetadataflowReferenc eType | StructureUsage references a metadataflow which the metadata is reported against. |
| Structure | MetadataStructureRef erenceType | Structure references the metadata structure definition which defines the structure of the metadata. |

**GenericMetadataStructureType:** GenericMetadataStructureType defines the structural information for a generic metadata message.

Derivation:

*PayloadStructureType* (restriction)   
   *MetadataStructureType* (restriction)   
         GenericMetadataStructureType

Attributes:

structureID, schemaURL?, serviceURL?, structureURL?

Content:

(ProvisionAgreement | StructureUsage | Structure)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| structureID | xs:ID | The structureID attribute uniquely identifies the structure for the purpose of referencing it from the payload. This is only used in structure specific formats. Although it is required, it is only useful when more than one data set is present in the message. |
| schemaURL | xs:anyURI | The schemaURL attribute provides a location from which the structure specific schema can be located. |
| serviceURL | xs:anyURI | The serviceURL attribute indicates the URL of an SDMX SOAP web service from which the details of the object can be retrieved. Note that this can be a registry or and SDMX structural metadata repository, as they both implement that same web service interface. |
| structureURL | xs:anyURI | The structureURL attribute indicates the URL of a SDMX-ML structure message (in the same version as the source document) in which the externally referenced object is contained. Note that this may be a URL of an SDMX RESTful web service which will return the referenced object. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ProvisionAgreement | ProvisionAgreementRe ferenceType | ProvisionAgreement references a provision agreement which the metadata is reported against. |
| StructureUsage | MetadataflowReferenc eType | StructureUsage references a metadataflow which the metadata is reported against. |
| Structure | MetadataStructureRef erenceType | Structure references the metadata structure definition which defines the structure of the metadata. |

### Simple Types

**AlphaNumericType:** AlphaNumericType is a reusable simple type that allows for only mixed-case alphabetical and numeric characters.

Derived by restriction of xs:string .  
Regular Expression Pattern: [A-Za-z0-9]+

**AlphaType:** AlphaType is a reusable simple type that allows for only mixed-case alphabetical characters. This is derived from the AlphaNumericType.

Derived by restriction of AlphaNumericType .  
Regular Expression Pattern: [A-Za-z]+

**NumericType:** NumericType is a reusable simple type that allows for only numeric characters. This is not to be confused with an integer, as this may be used to numeric strings which have leading zeros. These leading zeros are not ignored. This is derived from the AlphaNumericType.

Derived by restriction of AlphaNumericType .  
Regular Expression Pattern: [0-9]+

**ObservationalTimePeriodType:** ObservationalTimePeriodType specifies a distinct time period or point in time in SDMX. The time period can either be a Gregorian calendar period, a standard reporting period, a distinct point in time, or a time range with a specific date and duration.

Union of:

xs:gYear, xs:gYearMonth, xs:date, xs:dateTime, ReportingYearType, ReportingSemesterType, ReportingTrimesterType, ReportingQuarterType, ReportingMonthType, ReportingWeekType, ReportingDayType, TimeRangeType.

**StandardTimePeriodType:** StandardTimePeriodType defines the set of standard time periods in SDMX. This includes the reporting time periods and the basic date type (i.e. the calendar time periods and the dateTime format).

Union of:

xs:gYear, xs:gYearMonth, xs:date, xs:dateTime, ReportingYearType, ReportingSemesterType, ReportingTrimesterType, ReportingQuarterType, ReportingMonthType, ReportingWeekType, ReportingDayType.

**BasicTimePeriodType:** BasicTimePeriodType contains the basic dates and calendar periods. It is a combination of the Gregorian time periods and the date time type..

Union of:

xs:gYear, xs:gYearMonth, xs:date, xs:dateTime.

**GregorianTimePeriodType:** GregorianTimePeriodType defines the set of standard calendar periods in SDMX.

Union of:

xs:gYear, xs:gYearMonth, xs:date.

**ReportingTimePeriodType:** ReportingTimePeriodType defines standard reporting periods in SDMX, which are all in relation to the start day (day-month) of a reporting year which is specified in the specialized reporting year start day attribute. If the reporting year start day is not defined, a day of January 1 is assumed. The reporting year must be epxressed as the year at the beginning of the period. Therfore, if the reproting year runs from April to March, any given reporting year is expressed as the year for April. The general format of a report period can be described as [year]-[period][time zone]?, where the type of period is designated with a single character followed by a number representing the period. Note that all periods allow for an optional time zone offset. See the details of each member type for the specifics of its format.

Union of:

ReportingYearType, ReportingSemesterType, ReportingTrimesterType, ReportingQuarterType, ReportingMonthType, ReportingWeekType, ReportingDayType.

**BaseReportPeriodType:** BaseReportPeriodType is a simple type which frames the general pattern of a reporting period for validation purposes. This regular expression is only a general validation which is meant to validate the following structure [year]-[period][time zone]?. This type is meant to be derived from for further validation.

Derived by restriction of xs:string .  
Regular Expression Pattern: \d{4}\-([ASTQ]\d{1}|[MW]\d{2}|[D]\d{3})(Z|((\+|\-)\d{2}:\d{2}))?

**ReportPeriodValidTimeZoneType:** ReportPeriodValidTimeZoneType is a derivation of the BaseReportPeriodType which validates that the time zone provided in the base type is valid. The base type will have provided basic validation already. The patterns below validate that the time zone is "Z" or that it is between -14:00 and +14:00, or that there is no time zone provided. This type is meant to be derived from for further validation.

Derived by restriction of BaseReportPeriodType .  
Regular Expression Pattern: .+Z.{5}.\*(\+|\-)(14:00|((0[0-9]|1[0-3]):[0-5][0-9])).{5}[^\+\-Z]+

**ReportingYearType:** ReportingYearType defines a time period of 1 year (P1Y) in relation to a reporting year which has a start day (day-month) specified in the specialized reporting year start day attribute. In the absence of a start day for the reporting year, a day of January 1 is assumed. In this case a reporting year will coincide with a calendar year. The format of a reporting year is YYYY-A1 (e.g. 2000-A1). Note that the period value of 1 is fixed.

Derived by restriction of ReportPeriodValidTimeZoneType .  
Regular Expression Pattern: .{5}A1.\*

**ReportingSemesterType:** ReportingSemesterType defines a time period of 6 months (P6M) in relation to a reporting year which has a start day (day-month) specified in the specialized reporting year start day attribute. In the absence of a start day for the reporting year, a day of January 1 is assumed. The format of a reporting semester is YYYY-Ss (e.g. 2000-S1), where s is either 1 or 2.

Derived by restriction of ReportPeriodValidTimeZoneType .  
Regular Expression Pattern: .{5}S[1-2].\*

**ReportingTrimesterType:** ReportingTrimesterType defines a time period of 4 months (P4M) in relation to a reporting year which has a start day (day-month) specified in the specialized reporting year start day attribute. In the absence of a start day for the reporting year, a day of January 1 is assumed. The format of a reporting trimester is YYYY-Tt (e.g. 2000-T1), where s is either 1, 2, or 3.

Derived by restriction of ReportPeriodValidTimeZoneType .  
Regular Expression Pattern: .{5}T[1-3].\*

**ReportingQuarterType:** ReportingQuarterType defines a time period of 3 months (P3M) in relation to a reporting year which has a start day (day-month) specified in the specialized reporting year start day attribute. In the absence of a start day for the reporting year, a day of January 1 is assumed. The format of a reporting quarter is YYYY-Qq (e.g. 2000-Q1), where q is a value between 1 and 4.

Derived by restriction of ReportPeriodValidTimeZoneType .  
Regular Expression Pattern: .{5}Q[1-4].\*

**ReportingMonthType:** ReportingMonthType defines a time period of 1 month (P1M) in relation to a reporting year which has a start day (day-month) specified in the specialized reporting year start day attribute. In the absence of a start day for the reporting year, a day of January 1 is assumed. In this case a reporting month will coincide with a calendar month. The format of a reporting month is YYYY-Mmm (e.g. 2000-M01), where mm is a two digit month (i.e. 01-12).

Derived by restriction of ReportPeriodValidTimeZoneType .  
Regular Expression Pattern: .{5}M(0[1-9]|1[0-2]).\*

**ReportingWeekType:** ReportingWeekType defines a time period of 7 days (P7D) in relation to a reporting year which has a start day (day-month) specified in the specialized reporting year start day attribute. A standard reporting week is based on the ISO 8601 defintion of a week date, in relation to the reporting period start day. The first week is defined as the week with the first Thursday on or after the reporting year start day. An equivalent definition is the week starting with the Monday nearest in time to the reporting year start day. There are other equivalent defintions, all of which should be adjusted based on the reporting year start day. In the absence of a start day for the reporting year, a day of January 1 is assumed. The format of a reporting week is YYYY-Www (e.g. 2000-W01), where mm is a two digit week (i.e. 01-53).

Derived by restriction of ReportPeriodValidTimeZoneType .  
Regular Expression Pattern: .{5}W(0[1-9]|[1-4][0-9]|5[0-3]).\*

**ReportingDayType:** ReportingDayType defines a time period of 1 day (P1D) in relation to a reporting year which has a start day (day-month) specified in the specialized reporting year start day attribute. In the absence of a start day for the reporting year, a day of January 1 is assumed. The format of a reporting day is YYYY-Dddd (e.g. 2000-D001), where ddd is a three digit day (i.e. 001-366).

Derived by restriction of ReportPeriodValidTimeZoneType .  
Regular Expression Pattern: .{5}D(0[0-9][1-9]|[1-2][0-9][0-9]|3[0-5][0-9]|36[0-6]).\*

**BaseTimeRangeType:** BaseTimeRangeType is a simple type which frames the general pattern for a time range in SDMX. A time range pattern is generally described as [xs:date or xs:dateTime]\[xs:duration], where the referenced types are defined by XML Schema. This type is meant to be derived from for further validation.

Derived by restriction of xs:string .  
Regular Expression Pattern: \d{4}\-\d{2}\-\d{2}(T\d{2}:\d{2}:\d{2}(\.\d+)?)?(Z|((\+|\-)\d{2}:\d{2}))?/P.+

**RangeValidMonthDayType:** RangeValidMonthDayType is a derivation of the BaseTimeRangeType which validates that the day provided is valid for the month, without regard to leap years. The base type will have provided basic validation already. The patterns below validate that there are up to 29 days in February, up to 30 days in April, June, September, and November and up to 31 days in January, March, May, July, August, October, and December. This type is meant to be derived from for further validation.

Derived by restriction of BaseTimeRangeType .  
Regular Expression Pattern: .{5}02\-(0[1-9]|[1-2][0-9]).+.{5}(04|06|09|11)\-(0[1-9]|[1-2][0-9]|30).+.{5}(01|03|05|07|08|10|12)\-(0[1-9]|[1-2][0-9]|3[0-1]).+

**RangeValidLeapYearType:** RangeValidLeapYearType is a derivation of the RangeValidMonthDayType which validates that a date of February 29 occurs in a valid leap year (i.e. if the year is divisible 4 and not by 100, unless it is also divisible by 400). This type is meant to be derived from for further validation.

Derived by restriction of RangeValidMonthDayType .  
Regular Expression Pattern: ((\d{2}(04|08|12|16|20|24|28|32|36|40|44|48|52|56|60|64|68|72|76|80|84|88|92|96))|((00|04|08|12|16|20|24|28|32|36|40|44|48|52|56|60|64|68|72|76|80|84|88|92|96)00))\-02\-29.+.{5}02\-(([0-1][0-9])|(2[^9])).+.{5}((0[1,3-9])|1[0-2]).+

**RangeValidTimeType:** RangeValidTimeType is a derivation of the RangeValidLeapYearType which validates that the time (if provided) is validly formatted. The base type will have provided basic validation already. The patterns below validate that the time falls between 00:00:00 and 24:00:00. Note that as the XML dateTime type does, seconds are required. It is also permissible to have fractions of seconds, but only within the boundaries of the range specified. This type is meant to be derived from for further validation.

Derived by restriction of RangeValidLeapYearType .  
Regular Expression Pattern: .{10}T(24:00:00(\.[0]+)?|((([0-1][0-9])|(2[0-3])):[0-5][0-9]:[0-5][0-9](\.\d+)?))(/|Z|\+|\-).+[^T]+/.+

**RangeValidTimeZoneType:** RangeValidMonthDayType is a derivation of the RangeValidTimeType which validates that the time zone provided in the base type is valid. The base type will have provided basic validation already. The patterns below validate that the time zone is "Z" or that it is between -14:00 and +14:00, or that there is no time zone provided. This type is meant to be derived from for further validation.

Derived by restriction of RangeValidTimeType .  
Regular Expression Pattern: .+Z/.+.{10}.\*(\+|\-)(14:00|((0[0-9]|1[0-3]):[0-5][0-9]))/.+.{10}[^\+\-Z]+

**TimeRangeValidDateDurationType:** TimeRangeValidDateDurationType is an abstract derivation of the RangeValidTimeType which validates that duration provided is generally valid, up to the time component.

Derived by restriction of RangeValidTimeZoneType .  
Regular Expression Pattern: .+/P(\d+Y)?(\d+M)?(\d+D)?(T.+)?

**TimeRangeType:** TimeRangeType defines the structure of a time range in SDMX. The pattern of a time range can be generally described as [start date]\[duration], where start date is an date or dateTime type as defined in XML Schema and duration is a time duration as defined in XML Schema. Note that it is permissible for a time zone offset to be provided on the date or date time.

Derived by restriction of TimeRangeValidDateDurationType .  
Regular Expression Pattern: .+/P.\*T(\d+H)?(\d+M)?(\d+(.\d+)?S)?.+/P[^T]+

**TimezoneType:** TimezoneType defines the pattern for a time zone. An offset of -14:00 to +14:00 or Z can be specified.

Derived by restriction of xs:string .  
Regular Expression Pattern: Z(\+|\-)(14:00|((0[0-9]|1[0-3]):[0-5][0-9]))

**OccurenceType:** OccurenceType is used to express the maximum occurrence of an object. It combines an integer, equal or greater than 1, and the literal text, "unbounded", for objects which have no upper limit on its occurrence.

Union of:

MaxOccursNumberType, UnboundedCodeType.

**MaxOccursNumberType:** MaxOccursNumberType is a base type used to restrict an integer to be greater than 1, for the purpose of expressing the maximum number of occurrences of an object.

Derived by restriction of xs:nonNegativeInteger .  
Minimum (inclusive): 1  
Fraction Digits: 0

**UnboundedCodeType:** UnboundedCodeType provides single textual value of "unbounded", for use in OccurentType.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| unbounded | Object has no upper limit on occurrences. |

**ActionType:** ActionType provides a list of actions, describing the intention of the data transmission from the sender's side. Each action provided at the data or metadata set level applies to the entire data set for which it is given. Note that the actions indicated in the Message Header are optional, and used to summarize specific actions indicated with this data type for all registry interactions. The "Informational" value is used when the message contains information in response to a query, rather than being used to invoke a maintenance activity.

Derived by restriction of xs:NMTOKEN .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Append | Append - this is an incremental update for an existing data/metadata set or the provision of new data or documentation (attribute values) formerly absent. If any of the supplied data or metadata is already present, it will not replace that data or metadata. This corresponds to the "Update" value found in version 1.0 of the SDMX Technical Standards. |
| Replace | Replace - data/metadata is to be replaced, and may also include additional data/metadata to be appended. The replacement occurs at the level of the observation - that is, it is not possible to replace an entire series. |
| Delete | Delete - data/metadata is to be deleted. Deletion occurs at the lowest level object. For instance, if a delete data message contains a series with no observations, then the entire series will be deleted. If the series contains observations, then only those observations specified will be deleted. The same basic concept applies for attributes. If a series or observation in a delete message contains attributes, then only those attributes will be deleted. |
| Information | Informational - data/metadata is being exchanged for informational purposes only, and not meant to update a system. |

**WildCardValueType:** WildCardValueType is a single value code list, used to include the '%' character - indicating that an entire field is wild carded.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| % | Indicates a wild card value. |

**CascadeSelectionType:**

Union of:

xs:boolean, ExcludeRootType.

**ExcludeRootType:** ExcludeRootType is a single enumerated value that indciates that child values should be included, but that the actual root should not.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| excluderoot |  |

**ObservationDimensionType:** ObservationDimensionType allows for the dimension at the observation level to be specified by either providing the dimension identifier or using the explicit value "AllDimensions".

Union of:

NCNameIDType, ObsDimensionsCodeType.

**ObsDimensionsCodeType:** ObsDimensionsCodeType is an enumeration containing the values "TimeDimension" and "AllDimensions"

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| AllDimensions | AllDimensions notes that the cross sectional format shall be flat; that is all dimensions should be contained at the observation level. |
| TIME\_PERIOD | TIME\_PERIOD refers to the fixed identifier for the time dimension. |

**DataType:** DataTypeType provides an enumerated list of the types of data formats allowed as the for the representation of an object.

Derived by restriction of xs:NMTOKEN .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| String | A string datatype corresponding to W3C XML Schema's xs:string datatype. |
| Alpha | A string datatype which only allows for the simple aplhabetic charcter set of A-Z, a-z. |
| AlphaNumeric | A string datatype which only allows for the simple alphabetic character set of A-Z, a-z plus the simple numeric character set of 0-9. |
| Numeric | A string datatype which only allows for the simple numeric character set of 0-9. This format is not treated as an integer, and therefore can having leading zeros. |
| BigInteger | An integer datatype corresponding to W3C XML Schema's xs:integer datatype. |
| Integer | An integer datatype corresponding to W3C XML Schema's xs:int datatype. |
| Long | A numeric datatype corresponding to W3C XML Schema's xs:long datatype. |
| Short | A numeric datatype corresponding to W3C XML Schema's xs:short datatype. |
| Decimal | A numeric datatype corresponding to W3C XML Schema's xs:decimal datatype. |
| Float | A numeric datatype corresponding to W3C XML Schema's xs:float datatype. |
| Double | A numeric datatype corresponding to W3C XML Schema's xs:double datatype. |
| Boolean | A datatype corresponding to W3C XML Schema's xs:boolean datatype. |
| URI | A datatype corresponding to W3C XML Schema's xs:anyURI datatype. |
| Count | A simple incrementing Integer type. The isSequence facet must be set to true, and the interval facet must be set to "1". |
| InclusiveValueRange | This value indicates that the startValue and endValue attributes provide the inclusive boundaries of a numeric range of type xs:decimal. |
| ExclusiveValueRange | This value indicates that the startValue and endValue attributes provide the exclusive boundaries of a numeric range, of type xs:decimal. |
| Incremental | This value indicates that the value increments according to the value provided in the interval facet, and has a true value for the isSequence facet. |
| ObservationalTimePeriod | Observational time periods are the superset of all time periods in SDMX. It is the union of the standard time periods (i.e. Gregorian time periods, the reporting time periods, and date time) and a time range. |
| StandardTimePeriod | Standard time periods is a superset of distinct time period in SDMX. It is the union of the basic time periods (i.e. the Gregorian time periods and date time) and the reporting time periods. |
| BasicTimePeriod | BasicTimePeriod time periods is a superset of the Gregorian time periods and a date time. |
| GregorianTimePeriod | Gregorian time periods correspond to calendar periods and are represented in ISO-8601 formats. This is the union of the year, year month, and date formats. |
| GregorianYear | A Gregorian time period corresponding to W3C XML Schema's xs:gYear datatype, which is based on ISO-8601. |
| GregorianYearMonth | A time datatype corresponding to W3C XML Schema's xs:gYearMonth datatype, which is based on ISO-8601. |
| GregorianDay | A time datatype corresponding to W3C XML Schema's xs:date datatype, which is based on ISO-8601. |
| ReportingTimePeriod | Reporting time periods represent periods of a standard length within a reporting year, where to start of the year (defined as a month and day) must be defined elsewhere or it is assumed to be January 1. This is the union of the reporting year, semester, trimester, quarter, month, week, and day. |
| ReportingYear | A reporting year represents a period of 1 year (P1Y) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingYearType. |
| ReportingSemester | A reporting semester represents a period of 6 months (P6M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingSemesterType. |
| ReportingTrimester | A reporting trimester represents a period of 4 months (P4M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingTrimesterType. |
| ReportingQuarter | A reporting quarter represents a period of 3 months (P3M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingQuarterType. |
| ReportingMonth | A reporting month represents a period of 1 month (P1M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingMonthType. |
| ReportingWeek | A reporting week represents a period of 7 days (P7D) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingWeekType. |
| ReportingDay | A reporting day represents a period of 1 day (P1D) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingDayType. |
| DateTime | A time datatype corresponding to W3C XML Schema's xs:dateTime datatype. |
| TimeRange | TimeRange defines a time period by providing a distinct start (date or date time) and a duration. |
| Month | A time datatype corresponding to W3C XML Schema's xs:gMonth datatype. |
| MonthDay | A time datatype corresponding to W3C XML Schema's xs:gMonthDay datatype. |
| Day | A time datatype corresponding to W3C XML Schema's xs:gDay datatype. |
| Time | A time datatype corresponding to W3C XML Schema's xs:time datatype. |
| Duration | A time datatype corresponding to W3C XML Schema's xs:duration datatype. |
| GeospatialInformation | A string used to describe geographical features like points (e.g., locations of places, landmarks, buildings, etc.), lines (e.g., rivers, roads, streets, etc.), or areas (e.g., geographical regions, countries, islands, land lots, etc.). A mix of different features is possible too, e.g., combining polygons and geographical points to describe a country and the location of its capital. |
| XHTML | This value indicates that the content of the component can contain XHTML markup. |
| KeyValues | This value indicates that the content of the component will be data key (a set of dimension references and values for the dimensions). |
| IdentifiableReference | This value indicates that the content of the component will be complete reference (either URN or full set of reference fields) to an Identifiable object in the SDMX Information Model. |
| DataSetReference | This value indicates that the content of the component will be reference to a data provider, which is actually a formal reference to a data provider and a data set identifier value. |

**BasicComponentDataType:** BasicComponentDataType provides an enumerated list of the types of characters allowed in the textType attribute for all non-target object components.

Derived by restriction of DataType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| String | A string datatype corresponding to W3C XML Schema's xs:string datatype. |
| Alpha | A string datatype which only allows for the simple aplhabetic charcter set of A-Z, a-z. |
| AlphaNumeric | A string datatype which only allows for the simple alphabetic character set of A-Z, a-z plus the simple numeric character set of 0-9. |
| Numeric | A string datatype which only allows for the simple numeric character set of 0-9. This format is not treated as an integer, and therefore can having leading zeros. |
| BigInteger | An integer datatype corresponding to W3C XML Schema's xs:integer datatype. |
| Integer | An integer datatype corresponding to W3C XML Schema's xs:int datatype. |
| Long | A numeric datatype corresponding to W3C XML Schema's xs:long datatype. |
| Short | A numeric datatype corresponding to W3C XML Schema's xs:short datatype. |
| Decimal | A numeric datatype corresponding to W3C XML Schema's xs:decimal datatype. |
| Float | A numeric datatype corresponding to W3C XML Schema's xs:float datatype. |
| Double | A numeric datatype corresponding to W3C XML Schema's xs:double datatype. |
| Boolean | A datatype corresponding to W3C XML Schema's xs:boolean datatype. |
| URI | A datatype corresponding to W3C XML Schema's xs:anyURI datatype. |
| Count | A simple incrementing Integer type. The isSequence facet must be set to true, and the interval facet must be set to "1". |
| InclusiveValueRange | This value indicates that the startValue and endValue attributes provide the inclusive boundaries of a numeric range of type xs:decimal. |
| ExclusiveValueRange | This value indicates that the startValue and endValue attributes provide the exclusive boundaries of a numeric range, of type xs:decimal. |
| Incremental | This value indicates that the value increments according to the value provided in the interval facet, and has a true value for the isSequence facet. |
| ObservationalTimePeriod | Observational time periods are the superset of all time periods in SDMX. It is the union of the standard time periods (i.e. Gregorian time periods, the reporting time periods, and date time) and a time range. |
| StandardTimePeriod | Standard time periods is a superset of distinct time period in SDMX. It is the union of the basic time periods (i.e. the Gregorian time periods and date time) and the reporting time periods. |
| BasicTimePeriod | BasicTimePeriod time periods is a superset of the Gregorian time periods and a date time. |
| GregorianTimePeriod | Gregorian time periods correspond to calendar periods and are represented in ISO-8601 formats. This is the union of the year, year month, and date formats. |
| GregorianYear | A Gregorian time period corresponding to W3C XML Schema's xs:gYear datatype, which is based on ISO-8601. |
| GregorianYearMonth | A time datatype corresponding to W3C XML Schema's xs:gYearMonth datatype, which is based on ISO-8601. |
| GregorianDay | A time datatype corresponding to W3C XML Schema's xs:date datatype, which is based on ISO-8601. |
| ReportingTimePeriod | Reporting time periods represent periods of a standard length within a reporting year, where to start of the year (defined as a month and day) must be defined elsewhere or it is assumed to be January 1. This is the union of the reporting year, semester, trimester, quarter, month, week, and day. |
| ReportingYear | A reporting year represents a period of 1 year (P1Y) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingYearType. |
| ReportingSemester | A reporting semester represents a period of 6 months (P6M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingSemesterType. |
| ReportingTrimester | A reporting trimester represents a period of 4 months (P4M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingTrimesterType. |
| ReportingQuarter | A reporting quarter represents a period of 3 months (P3M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingQuarterType. |
| ReportingMonth | A reporting month represents a period of 1 month (P1M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingMonthType. |
| ReportingWeek | A reporting week represents a period of 7 days (P7D) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingWeekType. |
| ReportingDay | A reporting day represents a period of 1 day (P1D) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingDayType. |
| DateTime | A time datatype corresponding to W3C XML Schema's xs:dateTime datatype. |
| TimeRange | TimeRange defines a time period by providing a distinct start (date or date time) and a duration. |
| Month | A time datatype corresponding to W3C XML Schema's xs:gMonth datatype. |
| MonthDay | A time datatype corresponding to W3C XML Schema's xs:gMonthDay datatype. |
| Day | A time datatype corresponding to W3C XML Schema's xs:gDay datatype. |
| Time | A time datatype corresponding to W3C XML Schema's xs:time datatype. |
| Duration | A time datatype corresponding to W3C XML Schema's xs:duration datatype. |
| GeospatialInformation | A string used to describe geographical features like points (e.g., locations of places, landmarks, buildings, etc.), lines (e.g., rivers, roads, streets, etc.), or areas (e.g., geographical regions, countries, islands, land lots, etc.). A mix of different features is possible too, e.g., combining polygons and geographical points to describe a country and the location of its capital. |
| XHTML | This value indicates that the content of the component can contain XHTML markup. |

**SimpleDataType:** SimpleDataType restricts BasicComponentDataType to specify the allowable data types for a data structure definition component. The XHTML representation is removed as a possible type.

Derived by restriction of BasicComponentDataType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| String | A string datatype corresponding to W3C XML Schema's xs:string datatype. |
| Alpha | A string datatype which only allows for the simple aplhabetic charcter set of A-Z, a-z. |
| AlphaNumeric | A string datatype which only allows for the simple alphabetic character set of A-Z, a-z plus the simple numeric character set of 0-9. |
| Numeric | A string datatype which only allows for the simple numeric character set of 0-9. This format is not treated as an integer, and therefore can having leading zeros. |
| BigInteger | An integer datatype corresponding to W3C XML Schema's xs:integer datatype. |
| Integer | An integer datatype corresponding to W3C XML Schema's xs:int datatype. |
| Long | A numeric datatype corresponding to W3C XML Schema's xs:long datatype. |
| Short | A numeric datatype corresponding to W3C XML Schema's xs:short datatype. |
| Decimal | A numeric datatype corresponding to W3C XML Schema's xs:decimal datatype. |
| Float | A numeric datatype corresponding to W3C XML Schema's xs:float datatype. |
| Double | A numeric datatype corresponding to W3C XML Schema's xs:double datatype. |
| Boolean | A datatype corresponding to W3C XML Schema's xs:boolean datatype. |
| URI | A datatype corresponding to W3C XML Schema's xs:anyURI datatype. |
| Count | A simple incrementing Integer type. The isSequence facet must be set to true, and the interval facet must be set to "1". |
| InclusiveValueRange | This value indicates that the startValue and endValue attributes provide the inclusive boundaries of a numeric range of type xs:decimal. |
| ExclusiveValueRange | This value indicates that the startValue and endValue attributes provide the exclusive boundaries of a numeric range, of type xs:decimal. |
| Incremental | This value indicates that the value increments according to the value provided in the interval facet, and has a true value for the isSequence facet. |
| ObservationalTimePeriod | Observational time periods are the superset of all time periods in SDMX. It is the union of the standard time periods (i.e. Gregorian time periods, the reporting time periods, and date time) and a time range. |
| StandardTimePeriod | Standard time periods is a superset of distinct time period in SDMX. It is the union of the basic time periods (i.e. the Gregorian time periods and date time) and the reporting time periods. |
| BasicTimePeriod | BasicTimePeriod time periods is a superset of the Gregorian time periods and a date time. |
| GregorianTimePeriod | Gregorian time periods correspond to calendar periods and are represented in ISO-8601 formats. This is the union of the year, year month, and date formats. |
| GregorianYear | A Gregorian time period corresponding to W3C XML Schema's xs:gYear datatype, which is based on ISO-8601. |
| GregorianYearMonth | A time datatype corresponding to W3C XML Schema's xs:gYearMonth datatype, which is based on ISO-8601. |
| GregorianDay | A time datatype corresponding to W3C XML Schema's xs:date datatype, which is based on ISO-8601. |
| ReportingTimePeriod | Reporting time periods represent periods of a standard length within a reporting year, where to start of the year (defined as a month and day) must be defined elsewhere or it is assumed to be January 1. This is the union of the reporting year, semester, trimester, quarter, month, week, and day. |
| ReportingYear | A reporting year represents a period of 1 year (P1Y) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingYearType. |
| ReportingSemester | A reporting semester represents a period of 6 months (P6M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingSemesterType. |
| ReportingTrimester | A reporting trimester represents a period of 4 months (P4M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingTrimesterType. |
| ReportingQuarter | A reporting quarter represents a period of 3 months (P3M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingQuarterType. |
| ReportingMonth | A reporting month represents a period of 1 month (P1M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingMonthType. |
| ReportingWeek | A reporting week represents a period of 7 days (P7D) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingWeekType. |
| ReportingDay | A reporting day represents a period of 1 day (P1D) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingDayType. |
| DateTime | A time datatype corresponding to W3C XML Schema's xs:dateTime datatype. |
| TimeRange | TimeRange defines a time period by providing a distinct start (date or date time) and a duration. |
| Month | A time datatype corresponding to W3C XML Schema's xs:gMonth datatype. |
| MonthDay | A time datatype corresponding to W3C XML Schema's xs:gMonthDay datatype. |
| Day | A time datatype corresponding to W3C XML Schema's xs:gDay datatype. |
| Time | A time datatype corresponding to W3C XML Schema's xs:time datatype. |
| Duration | A time datatype corresponding to W3C XML Schema's xs:duration datatype. |
| GeospatialInformation | A string used to describe geographical features like points (e.g., locations of places, landmarks, buildings, etc.), lines (e.g., rivers, roads, streets, etc.), or areas (e.g., geographical regions, countries, islands, land lots, etc.). A mix of different features is possible too, e.g., combining polygons and geographical points to describe a country and the location of its capital. |

**TimeDataType:** TimeDataType restricts SimpleDataType to specify the allowable data types for representing a time value.

Derived by restriction of SimpleDataType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| ObservationalTimePeriod | Observational time periods are the superset of all time periods in SDMX. It is the union of the standard time periods (i.e. Gregorian time periods, the reporting time periods, and date time) and a time range. |
| StandardTimePeriod | Standard time periods is a superset of distinct time period in SDMX. It is the union of the basic time periods (i.e. the Gregorian time periods and date time) and the reporting time periods. |
| BasicTimePeriod | BasicTimePeriod time periods is a superset of the Gregorian time periods and a date time. |
| GregorianTimePeriod | Gregorian time periods correspond to calendar periods and are represented in ISO-8601 formats. This is the union of the year, year month, and date formats. |
| GregorianYear | A Gregorian time period corresponding to W3C XML Schema's xs:gYear datatype, which is based on ISO-8601. |
| GregorianYearMonth | A time datatype corresponding to W3C XML Schema's xs:gYearMonth datatype, which is based on ISO-8601. |
| GregorianDay | A time datatype corresponding to W3C XML Schema's xs:date datatype, which is based on ISO-8601. |
| ReportingTimePeriod | Reporting time periods represent periods of a standard length within a reporting year, where to start of the year (defined as a month and day) must be defined elsewhere or it is assumed to be January 1. This is the union of the reporting year, semester, trimester, quarter, month, week, and day. |
| ReportingYear | A reporting year represents a period of 1 year (P1Y) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingYearType. |
| ReportingSemester | A reporting semester represents a period of 6 months (P6M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingSemesterType. |
| ReportingTrimester | A reporting trimester represents a period of 4 months (P4M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingTrimesterType. |
| ReportingQuarter | A reporting quarter represents a period of 3 months (P3M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingQuarterType. |
| ReportingMonth | A reporting month represents a period of 1 month (P1M) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingMonthType. |
| ReportingWeek | A reporting week represents a period of 7 days (P7D) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingWeekType. |
| ReportingDay | A reporting day represents a period of 1 day (P1D) from the start date of the reporting year. This is expressed as using the SDMX specific ReportingDayType. |
| DateTime | A time datatype corresponding to W3C XML Schema's xs:dateTime datatype. |
| TimeRange | TimeRange defines a time period by providing a distinct start (date or date time) and a duration. |

**UrnPrefixPart:** This is validates the first part of the URN (urn:sdmx.org.infomodel.<package>.<class=). It is intended to be further restricted.

Derived by restriction of xs:string .  
Regular Expression Pattern: urn:sdmx:org\.sdmx\.infomodel\.[a-z]+\.[A-Za-z]+=[^=]+

**UrnClassesPart:** This refines the prefix to make specific restrictions of package and class values. Note that only one of the patterns must match. It is intended to be further restricted.

Derived by restriction of UrnPrefixPart .  
Regular Expression Pattern: .+\.base\.Agency=.+.+\.base\.AgencyScheme=.+.+\.base\.Any=.+.+\.base\.DataConsumer=.+.+\.base\.DataConsumerScheme=.+.+\.base\.DataProvider=.+.+\.base\.DataProviderScheme=.+.+\.base\.MetadataProvider=.+.+\.base\.MetadataProviderScheme=.+.+\.base\.OrganisationUnit=.+.+\.base\.OrganisationUnitScheme=.+.+\.categoryscheme\.Categorisation=.+.+\.categoryscheme\.Category=.+.+\.categoryscheme\.CategoryScheme=.+.+\.categoryscheme\.ReportingCategory=.+.+\.categoryscheme\.ReportingTaxonomy=.+.+\.codelist\.Code=.+.+\.codelist\.Codelist=.+.+\.codelist\.HierarchicalCode=.+.+\.codelist\.Hierarchy=.+.+\.codelist\.HierarchyAssociation=.+.+\.codelist\.Level=.+.+\.codelist\.ValueList=.+.+\.conceptscheme\.Concept=.+.+\.conceptscheme\.ConceptScheme=.+.+\.datastructure\.AttributeDescriptor=.+.+\.datastructure\.DataAttribute=.+.+\.datastructure\.Dataflow=.+.+\.datastructure\.DataStructure=.+.+\.datastructure\.Dimension=.+.+\.datastructure\.DimensionDescriptor=.+.+\.datastructure\.GroupDimensionDescriptor=.+.+\.datastructure\.Measure=.+.+\.datastructure\.MeasureDescriptor=.+.+\.datastructure\.TimeDimension=.+.+\.metadatastructure\.MetadataAttribute=.+.+\.metadatastructure\.Metadataflow=.+.+\.metadatastructure\.MetadataSet=.+.+\.metadatastructure\.MetadataStructure=.+.+\.process\.Process=.+.+\.process\.ProcessStep=.+.+\.process\.Transition=.+.+\.registry\.DataConstraint=.+.+\.registry\.MetadataConstraint=.+.+\.registry\.MetadataProvisionAgreement=.+.+\.registry\.ProvisionAgreement=.+.+\.structuremapping\.CategorySchemeMap=.+.+\.structuremapping\.ConceptSchemeMap=.+.+\.structuremapping\.DatePatternMap=.+.+\.structuremapping\.EpochMap=.+.+\.structuremapping\.FrequencyFormatMapping=.+.+\.structuremapping\.OrganisationSchemeMap=.+.+\.structuremapping\.ReportingTaxonomyMap=.+.+\.structuremapping\.RepresentationMap=.+.+\.structuremapping\.StructureMap=.+.+\.transformation\.CustomType=.+.+\.transformation\.CustomTypeScheme=.+.+\.transformation\.NamePersonalisation=.+.+\.transformation\.NamePersonalisationScheme=.+.+\.transformation\.Ruleset=.+.+\.transformation\.RulesetScheme=.+.+\.transformation\.Transformation=.+.+\.transformation\.TransformationScheme=.+.+\.transformation\.UserDefinedOperator=.+.+\.transformation\.UserDefinedOperatorScheme=.+.+\.transformation\.VtlCodelistMapping=.+.+\.transformation\.VtlConceptMapping=.+.+\.transformation\.VtlDataflowMapping=.+.+\.transformation\.VtlMappingScheme=.+

**UrnAgencyPart:** This restricts the prefix and classes patterns to validate the agency part of the URN (=<agency\_id>:).

Derived by restriction of UrnClassesPart .  
Regular Expression Pattern: .+=([A-Za-z][A-Za-z0-9\_\-]\*(\.[A-Za-z][A-Za-z0-9\_\-]\*)\*):[^:]+

**WildcardUrnAgencyPart:** This restricts the prefix and classes patterns to validate the agency part of a wildcarded URN reference (=<agency\_id>:).

Derived by restriction of UrnClassesPart .  
Regular Expression Pattern: .+=([A-Za-z][A-Za-z0-9\_\-]\*(\.[A-Za-z][A-Za-z0-9\_\-]\*)\*):[^:]+.+=\\*:[^:]+

**UrnMaintainableIdPart:** This refines the prefix, classes, and agnecy patterns to validate the maintainable ID part of the URN (:<maintainable\_id(<version\_number>)). Note that it does not restrict the version part as it is intended to be further restricted.

Derived by restriction of UrnAgencyPart .  
Regular Expression Pattern: .+:([A-Za-z0-9\_@$\-]+)\([0-9A-Za-z\-\.\+]+\)[^(\(\))]\*

**WildcardUrnMaintainableIdPart:** This refines the prefix, classes, and agnecy patterns to validate the maintainable ID part of a wildcarded URN reference (:<maintainable\_id(<version\_number>)). Note that it does not restrict the version part as it is intended to be further restricted.

Derived by restriction of WildcardUrnAgencyPart .  
Regular Expression Pattern: .+:([A-Za-z0-9\_@$\-]+)\([0-9A-Za-z\-\.\+\\*]+\)[^(\(\))]\*.+:\\*\([0-9A-Za-z\-\.\+\\*]+\)[^(\(\))]\*

**UrnVersionPart:** This refines the prefix, classes, agency, and maintainable id patterns to validate the version number part of the URN ((<version\_number)). It accounts for both legacy and semantic versioning, but not wildarding (for referencing). It is meant to be further refined, although all parts after this are optional.

Derived by restriction of UrnMaintainableIdPart .  
Regular Expression Pattern: .+\((0|[1-9]\d\*)(\.(0|[1-9]\d\*))?\).\*.+\((0|[1-9]\d\*)(\.(0|[1-9]\d\*)){2}(\-(([A-Za-z\-]|([A-Za-z\-][A-Za-z0-9\-]+)|([A-Za-z0-9\-]+[A-Za-z\-][A-Za-z0-9\-]\*))|(0|[1-9][0-9]\*))(\.(([A-Za-z\-]|([A-Za-z\-][A-Za-z0-9\-]+)|([A-Za-z0-9\-]+[A-Za-z\-][A-Za-z0-9\-]\*))|(0|[1-9][0-9]\*)))\*)?\).\*

**UrnReferenceVersionPart:** This refines the prefix, classes, agency, and maintainable id patterns to validate the version number part of a URN reference ((<version\_number)). It accounts for both legacy and semantic versioning (including late binding). It is meant to be further refined, although all parts after this are optional.

Derived by restriction of UrnMaintainableIdPart .  
Regular Expression Pattern: .+\((0|[1-9]\d\*)(\.(0|[1-9]\d\*))?\).\*.+\((0|[1-9]\d\*)(\.(0|[1-9]\d\*)){2}(\-(([A-Za-z\-]|([A-Za-z\-][A-Za-z0-9\-]+)|([A-Za-z0-9\-]+[A-Za-z\-][A-Za-z0-9\-]\*))|(0|[1-9][0-9]\*))(\.(([A-Za-z\-]|([A-Za-z\-][A-Za-z0-9\-]+)|([A-Za-z0-9\-]+[A-Za-z\-][A-Za-z0-9\-]\*))|(0|[1-9][0-9]\*)))\*)?\).\*.+\(((0|[1-9]\d\*)\+?)(\.((0|[1-9]\d\*))){2}\).\*.+\(((0|[1-9]\d\*))(\.((0|[1-9]\d\*)\+?))(\.((0|[1-9]\d\*)))\).\*.+\(((0|[1-9]\d\*)\+?)(\.((0|[1-9]\d\*)))(\.((0|[1-9]\d\*)\+?))\).\*

**WildcardUrnVersionPart:** This refines the prefix, classes, agency, and maintainable id patterns to validate the version number part of a wildcarded URN reference ((<version\_number)). It accounts for both legacy and semantic versioning (including late binding). It is meant to be further refined, although all parts after this are optional.

Derived by restriction of WildcardUrnMaintainableIdPart .  
Regular Expression Pattern: .+\((0|[1-9]\d\*)(\.(0|[1-9]\d\*))?\).\*.+\((0|[1-9]\d\*)(\.(0|[1-9]\d\*)){2}(\-(([A-Za-z\-]|([A-Za-z\-][A-Za-z0-9\-]+)|([A-Za-z0-9\-]+[A-Za-z\-][A-Za-z0-9\-]\*))|(0|[1-9][0-9]\*))(\.(([A-Za-z\-]|([A-Za-z\-][A-Za-z0-9\-]+)|([A-Za-z0-9\-]+[A-Za-z\-][A-Za-z0-9\-]\*))|(0|[1-9][0-9]\*)))\*)?\).\*.+\(((0|[1-9]\d\*)\+?)(\.((0|[1-9]\d\*))){2}\).\*.+\(((0|[1-9]\d\*))(\.((0|[1-9]\d\*)\+?))(\.((0|[1-9]\d\*)))\).\*.+\(((0|[1-9]\d\*)\+?)(\.((0|[1-9]\d\*)))(\.((0|[1-9]\d\*)\+?))\).\*.+\(\\*\).\*

**UrnType:** The completes the refinement of the prefix, classes, agency, maintainable id, and version number patterns to validate the last, and optional, nested component part of the URN (e.g. (<version\_number>)<containerobject-id>.<object-id>\*). The nested patterns provide a complete validation of the URN pattern.

Derived by restriction of UrnVersionPart .  
Regular Expression Pattern: .+\)(\.[A-Za-z0-9\_@$\-]+(\.[A-Za-z0-9\_@$\-]+)\*)?

**UrnReferenceType:** The completes the refinement of the prefix, classes, agency, maintainable id, and version number patterns to validate the last, and optional, nested component part of a URN reference (e.g. (<version\_number>)<containerobject-id>.<object-id>\*). The nested patterns provide a complete validation of the URN pattern.

Derived by restriction of UrnReferenceVersionPart .  
Regular Expression Pattern: .+\)(\.[A-Za-z0-9\_@$\-]+(\.[A-Za-z0-9\_@$\-]+)\*)?

**WildcardUrnType:** The completes the refinement of the prefix, classes, agency, maintainable id, and version number patterns to validate the last, and optional, nested component part of a wildcarded URN reference (e.g. (<version\_number>)<containerobject-id>.<object-id>\*). The nested patterns provide a complete validation of the URN pattern.

Derived by restriction of WildcardUrnVersionPart .  
Regular Expression Pattern: .+\)(\.[A-Za-z0-9\_@$\-]+(\.[A-Za-z0-9\_@$\-]+)\*)?.+\)(\.\\*(\.\\*)\*)?

**MaintainableUrnType:** Restricts the URN so that the pattern ends after the version part.

Derived by restriction of UrnType .  
Regular Expression Pattern: .+\)

**MaintainableUrnReferenceType:** Restricts the URN reference so that the pattern ends after the version part.

Derived by restriction of UrnReferenceType .  
Regular Expression Pattern: .+\)

**ComponentUrnType:** Restricts the URN so that the pattern ends after the first component part.

Derived by restriction of UrnType .  
Regular Expression Pattern: .+\)\.[A-Za-z0-9\_@$\-]+

**ComponentUrnReferenceType:** Restricts the URN reference so that the pattern ends after the first component part.

Derived by restriction of UrnReferenceType .  
Regular Expression Pattern: .+\)\.[A-Za-z0-9\_@$\-]+

**AgencyUrnType:** Urn type for an agency.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.base\.Agency=.+:AGENCIES\(1\.0\).+

**AgencySchemeUrnType:** Urn type for an agency scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.base\.AgencyScheme=.+:AGENCIES\(1\.0\)

**DataConsumerUrnType:** Urn type for a data consumer.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.base\.DataConsumer=.+:DATA\_CONSUMERS\(1\.0\).+

**DataConsumerSchemeUrnType:** Urn type for a data consumer scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.base\.DataConsumerScheme=.+:DATA\_CONSUMERS\(1\.0\)

**DataProviderUrnType:** Urn type for a data provider.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.base\.DataProvider=.+:DATA\_PROVIDERS\(1\.0\).+

**DataProviderSchemeUrnType:** Urn type for a data provider scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.base\.DataProviderScheme=.+:DATA\_PROVIDERS\(1\.0\)

**MetadataProviderUrnType:** Urn type for a metadata provider.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.base\.MetadataProvider=.+:METADATA\_PROVIDERS\(1\.0\).+

**MetadataProviderSchemeUrnType:** Urn type for a metadata provider scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.base\.MetadataProviderScheme=.+:METADATA\_PROVIDERS\(1\.0\)

**OrganisationUnitUrnType:** Urn type for an organisation unit.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.base\.OrganisationUnit=.+\(1\.0\).+

**OrganisationUnitSchemeUrnType:** Urn type for an organisation unit scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.base\.OrganisationUnitScheme=.+\(1\.0\)

**CategorisationUrnType:** Urn type for a categorisation.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.categoryscheme\.Categorisation=.+

**CategoryUrnType:** Urn type for a category.

Derived by restriction of UrnType .  
Regular Expression Pattern: .+\.categoryscheme\.Category=.+

**CategorySchemeUrnType:** Urn type for a category scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.categoryscheme\.CategoryScheme=.+

**ReportingCategoryUrnType:** Urn type for a reporting category.

Derived by restriction of UrnType .  
Regular Expression Pattern: .+\.categoryscheme\.ReportingCategory=.+

**ReportingTaxonomyUrnType:** Urn type for a reporting taxonomy.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.categoryscheme\.ReportingTaxonomy=.+

**CodeUrnType:** Urn type for a code.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.codelist\.Code=.+

**CodelistUrnType:** Urn type for a codelist.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.codelist\.Codelist=.+

**HierarchicalCodeUrnType:** Urn type for a hierarchical code.

Derived by restriction of UrnType .  
Regular Expression Pattern: .+\.codelist\.HierarchicalCode=.+

**HierarchyUrnType:** Urn type for a hierarchy.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.codelist\.Hierarchy=.+

**HierarchyAssociationUrnType:** Urn type for a hierarchy association.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.codelist\.HierarchyAssociation=.+

**LevelUrnType:** Urn type for a level.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.codelist\.Level=.+

**ValueListUrnType:** Urn type for a value list.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.codelist\.ValueList=.+

**ConceptUrnType:** Urn type for a concept.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.conceptscheme\.Concept=.+

**ConceptSchemeUrnType:** Urn type for a concept scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.conceptscheme\.ConceptScheme=.+

**AttributeDescriptorUrnType:** Urn type for an attribute descriptor.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.datastructure\.AttributeDescriptor=.+

**DataAttributeUrnType:** Urn type for a data attribute.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.datastructure\.DataAttribute=.+

**DataflowUrnType:** Urn type for a dataflow.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.datastructure\.Dataflow=.+

**DataStructureUrnType:** Urn type for a data structure.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.datastructure\.DataStructure=.+

**DimensionUrnType:** Urn type for a dimension.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.datastructure\.Dimension=.+

**DimensionDescriptorUrnType:** Urn type for a dimension descriptor.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.datastructure\.DimensionDescriptor=.+

**GroupDimensionDescriptorUrnType:** Urn type for a group dimension descriptor.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.datastructure\.GroupDimensionDescriptor=.+

**MeasureUrnType:** Urn type for a measure.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.datastructure\.Measure=.+

**MeasureDescriptorUrnType:** Urn type for a measure descriptor.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.datastructure\.MeasureDescriptor=.+

**TimeDimensionUrnType:** Urn type for a time dimension.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.datastructure\.TimeDimension=.+

**MetadataAttributeUrnType:** Urn type for a metadata attibute.

Derived by restriction of UrnType .  
Regular Expression Pattern: .+\.metadatastructure\.MetadataAttribute=.+

**MetadataflowUrnType:** Urn type for a metadataflow.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.metadatastructure\.Metadataflow=.+

**MetadataSetUrnType:** Urn type for a metadata set.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.metadatastructure\.MetadataSet=.+

**MetadataStructureUrnType:** Urn type for a metadata structure.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.metadatastructure\.MetadataStructure=.+

**ProcessUrnType:** Urn type for a process.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.process\.Process=.+

**ProcessStepUrnType:** Urn type for a process step.

Derived by restriction of UrnType .  
Regular Expression Pattern: .+\.process\.ProcessStep=.+

**TransitionUrnType:** Urn type for a transition.

Derived by restriction of UrnType .  
Regular Expression Pattern: .+\.process\.Transition=.+

**DataConstraintUrnType:** Urn type for a data constraint.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.registry\.DataConstraint=.+

**MetadataConstraintUrnType:** Urn type for a metadata constraint.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.registry\.MetadataConstraint=.+

**MetadataProvisionAgreementUrnType:** Urn type for a metadata provision agreement.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.registry\.MetadataProvisionAgreement=.+

**ProvisionAgreementUrnType:** Urn type for a provision agreement.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.registry\.ProvisionAgreement=.+

**CategorySchemeMapUrnType:** Urn type for a category scheme map.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.structuremapping\.CategorySchemeMap=.+

**ConceptSchemeMapUrnType:** Urn type for a concept scheme map.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.structuremapping\.ConceptSchemeMap=.+

**DatePatternMapUrnType:** Urn type for a date pattern map.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.structuremapping\.DatePatternMap=.+

**EpochMapUrnType:** Urn type for a epoch map.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.structuremapping\.EpochMap=.+

**FrequencyFormatMappingUrnType:** Urn type for a frequency format mapping.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.structuremapping\.FrequencyFormatMapping=.+

**OrganisationSchemeMapUrnType:** Urn type for a organisation scheme map.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.structuremapping\.OrganisationSchemeMap=.+

**ReportingTaxonomyMapUrnType:** Urn type for a reporting taxonomy map.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.structuremapping\.ReportingTaxonomyMap=.+

**RepresentationMapUrnType:** Urn type for a representation map.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.structuremapping\.RepresentationMap=.+

**StructureMapUrnType:** Urn type for a structure map.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.structuremapping\.StructureMap=.+

**CustomTypeUrnType:** Urn type for a custom type.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.transformation\.CustomType=.+

**CustomTypeSchemeUrnType:** Urn type for a custom type scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.transformation\.CustomTypeScheme=.+

**NamePersonalisationUrnType:** Urn type for a name personalisation.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.transformation\.NamePersonalisation=.+

**NamePersonalisationSchemeUrnType:** Urn type for a name personalisation scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.transformation\.NamePersonalisationScheme=.+

**RulesetUrnType:** Urn type for a ruleset.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.transformation\.Ruleset=.+

**RulesetSchemeUrnType:** Urn type for a ruleste scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.transformation\.RulesetScheme=.+

**TransformationUrnType:** Urn type for a transformation.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.transformation\.Transformation=.+

**TransformationSchemeUrnType:** Urn type for a transformation scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.transformation\.TransformationScheme=.+

**UserDefinedOperatorUrnType:** Urn type for a user defined operator.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.transformation\.UserDefinedOperator=.+

**UserDefinedOperatorSchemeUrnType:** Urn type for a user defined operator scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.transformation\.UserDefinedOperatorScheme=.+

**VtlMappingUrnType:** Urn type for a VTL mapping.

Derived by restriction of ComponentUrnType .  
Regular Expression Pattern: .+\.transformation\.VtlCodelistMapping=.+.+\.transformation\.VtlConceptMapping=.+.+\.transformation\.VtlDataflowMapping=.+

**VtlMappingSchemeUrnType:** Urn type for a VTL mapping scheme.

Derived by restriction of MaintainableUrnType .  
Regular Expression Pattern: .+\.transformation\.VtlMappingScheme=.+

**StructureOrUsageReferenceType:** A reference type for a structure or structure usage.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.datastructure\.DataStructure=.+.+\.datastructure\.Dataflow=.+.+\.metadatastructure\.MetadataStructure=.+.+\.metadatastructure\.Metadataflow=.+

**StructureUsageReferenceType:** A reference type for structure usage.

Derived by restriction of StructureOrUsageReferenceType .  
Regular Expression Pattern: .+\.datastructure\.Dataflow=.+.+\.metadatastructure\.Metadataflow=.+

**StructureReferenceType:** A reference type for a structure.

Derived by restriction of StructureOrUsageReferenceType .  
Regular Expression Pattern: .+\.datastructure\.DataStructure=.+.+\.metadatastructure\.MetadataStructure=.+

**AnyCodelistReferenceType:** A reference type for a codelist or value list.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.codelist\.Codelist=.+.+\.codelist\.ValueList=.+

**OrganisationSchemeReferenceType:** A reference type for any type of organisation scheme.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.base\.AgencyScheme=.+:AGENCIES\(.+\).+\.base\.DataConsumerScheme=.+:DATA\_CONSUMERS\(.+\).+\.base\.DataProviderScheme=.+:DATA\_PROVIDERS\(.+\).+\.base\.MetadataProviderScheme=.+:METADATA\_PROVIDERS\(.+\).+\.base\.OrganisationUnitScheme=.+

**OrganisationReferenceType:** A reference type for any type of organisation.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.base\.Agency=.+:AGENCIES\(.+\).+.+\.base\.DataConsumer=.+:DATA\_CONSUMERS\(.+\).+.+\.base\.DataProvider=.+:DATA\_PROVIDERS\(.+\).+.+\.base\.MetadataProvider=.+:METADATA\_PROVIDERS\(.+\).+.+\.base\.OrganisationUnit=.+

**ConstraintReferenceType:** A reference for any type of constraint.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.registry\.DataConstraint=.+.+\.registry\.MetadataConstraint=.+

**AgencyReferenceType:** A reference type for an agency.

Derived by restriction of OrganisationReferenceType .  
Regular Expression Pattern: .+\.base\.Agency=.+:AGENCIES\(.+\).+

**AgencySchemeReferenceType:** A reference type for an agency scheme.

Derived by restriction of OrganisationSchemeReferenceType .  
Regular Expression Pattern: .+\.base\.AgencyScheme=.+:AGENCIES\(.+\)

**DataConsumerReferenceType:** A reference type for a data consumer.

Derived by restriction of OrganisationReferenceType .  
Regular Expression Pattern: .+\.base\.DataConsumer=.+:DATA\_CONSUMERS\(.+\).+

**DataConsumerSchemeReferenceType:** A reference type for a data consumer scheme.

Derived by restriction of OrganisationSchemeReferenceType .  
Regular Expression Pattern: .+\.base\.DataConsumerScheme=.+:DATA\_CONSUMERS\(.+\)

**DataProviderReferenceType:** A reference type for a data provider.

Derived by restriction of OrganisationReferenceType .  
Regular Expression Pattern: .+\.base\.DataProvider=.+:DATA\_PROVIDERS\(.+\).+

**DataProviderSchemeReferenceType:** A reference type for a data provider scheme.

Derived by restriction of OrganisationSchemeReferenceType .  
Regular Expression Pattern: .+\.base\.DataProviderScheme=.+:DATA\_PROVIDERS\(.+\)

**MetadataProviderReferenceType:** A reference type for a metadata provider.

Derived by restriction of OrganisationReferenceType .  
Regular Expression Pattern: .+\.base\.MetadataProvider=.+:METADATA\_PROVIDERS\(.+\).+

**MetadataProviderSchemeReferenceType:** A reference type for a metadata provider scheme.

Derived by restriction of OrganisationSchemeReferenceType .  
Regular Expression Pattern: .+\.base\.MetadataProviderScheme=.+:METADATA\_PROVIDERS\(.+\)

**OrganisationUnitReferenceType:** A reference type for an organisation unit.

Derived by restriction of OrganisationReferenceType .  
Regular Expression Pattern: .+\.base\.OrganisationUnit=.+

**OrganisationUnitSchemeReferenceType:** A reference type for an organisation unit scheme.

Derived by restriction of OrganisationSchemeReferenceType .  
Regular Expression Pattern: .+\.base\.OrganisationUnitScheme=.+

**CategorisationReferenceType:** A reference type for a categorisation.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.categoryscheme\.Categorisation=.+

**CategoryReferenceType:** A reference type for a category.

Derived by restriction of UrnReferenceType .  
Regular Expression Pattern: .+\.categoryscheme\.Category=.+

**CategorySchemeReferenceType:** A reference type for a cateogry scheme.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.categoryscheme\.CategoryScheme=.+

**ReportingCategoryReferenceType:** A reference type for a reporting category.

Derived by restriction of UrnReferenceType .  
Regular Expression Pattern: .+\.categoryscheme\.ReportingCategory=.+

**ReportingTaxonomyReferenceType:** A reference type for a reporting taxonomy.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.categoryscheme\.ReportingTaxonomy=.+

**CodeReferenceType:** A reference type for a code.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.codelist\.Code=.+

**CodelistReferenceType:** A reference type for a codelist.

Derived by restriction of AnyCodelistReferenceType .  
Regular Expression Pattern: .+\.codelist\.Codelist=.+

**HierarchicalCodeReferenceType:** A reference type for a hierarchical code.

Derived by restriction of UrnReferenceType .  
Regular Expression Pattern: .+\.codelist\.HierarchicalCode=.+

**HierarchyReferenceType:** A reference type for a hierarchy.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.codelist\.Hierarchy=.+

**HierarchyAssociationReferenceType:** A reference type for a hierarchy association.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.codelist\.HierarchyAssociation=.+

**LevelReferenceType:** A reference type for a level.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.codelist\.Level=.+

**ValueListReferenceType:** A reference type for a value list.

Derived by restriction of AnyCodelistReferenceType .  
Regular Expression Pattern: .+\.codelist\.ValueList=.+

**ConceptReferenceType:** A reference type for a concept.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.conceptscheme\.Concept=.+

**ConceptSchemeReferenceType:** A reference type for a concept scheme.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.conceptscheme\.ConceptScheme=.+

**AttributeDescriptorReferenceType:** A reference type for an attribute descriptor.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.datastructure\.AttributeDescriptor=.+

**DataAttributeReferenceType:** A reference type for a data attribute.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.datastructure\.DataAttribute=.+

**DataflowReferenceType:** A reference type for a dataflow.

Derived by restriction of StructureUsageReferenceType .  
Regular Expression Pattern: .+\.datastructure\.Dataflow=.+

**DataStructureReferenceType:** A reference type for a data structure.

Derived by restriction of StructureReferenceType .  
Regular Expression Pattern: .+\.datastructure\.DataStructure=.+

**DimensionReferenceType:** A reference type for a dimension.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.datastructure\.Dimension=.+

**DimensionDescriptorReferenceType:** A reference type for a dimension descriptor.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.datastructure\.DimensionDescriptor=.+

**GroupDimensionDescriptorReferenceType:** A reference type for a group dimension descriptor.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.datastructure\.GroupDimensionDescriptor=.+

**MeasureReferenceType:** A reference type for a measure.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.datastructure\.Measure=.+

**MeasureDescriptorReferenceType:** A reference type for a measure descriptor.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.datastructure\.MeasureDescriptor=.+

**TimeDimensionReferenceType:** A reference type for a time dimension.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.datastructure\.TimeDimension=.+

**MetadataAttributeReferenceType:** A reference type for a metadata attribute.

Derived by restriction of UrnReferenceType .  
Regular Expression Pattern: .+\.metadatastructure\.MetadataAttribute=.+

**MetadataflowReferenceType:** A reference type for a metadataflow.

Derived by restriction of StructureUsageReferenceType .  
Regular Expression Pattern: .+\.metadatastructure\.Metadataflow=.+

**MetadataSetReferenceType:** A reference type for a metadata set.

Derived by restriction of StructureReferenceType .  
Regular Expression Pattern: .+\.metadatastructure\.MetadataSet=.+

**MetadataStructureReferenceType:** A reference type for a metadata structure.

Derived by restriction of StructureReferenceType .  
Regular Expression Pattern: .+\.metadatastructure\.MetadataStructure=.+

**ProcessReferenceType:** A reference type for a process.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.process\.Process=.+

**ProcessStepReferenceType:** A reference type for a process step.

Derived by restriction of UrnReferenceType .  
Regular Expression Pattern: .+\.process\.ProcessStep=.+

**TransitionReferenceType:** A reference type for a transition.

Derived by restriction of UrnReferenceType .  
Regular Expression Pattern: .+\.process\.Transition=.+

**DataConstraintReferenceType:** A reference type for a data constraint.

Derived by restriction of ConstraintReferenceType .  
Regular Expression Pattern: .+\.registry\.DataConstraint=.+

**MetadataConstraintReferenceType:** A reference type for a metadata constraint.

Derived by restriction of ConstraintReferenceType .  
Regular Expression Pattern: .+\.registry\.MetadataConstraint=.+

**MetadataProvisionAgreementReferenceType:** A reference type for a metadata provision agreement.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.registry\.MetadataProvisionAgreement=.+

**ProvisionAgreementReferenceType:** A reference type for a provision agreement.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.registry\.ProvisionAgreement=.+

**CategorySchemeMapReferenceType:** A reference type for a category scheme map.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.structuremapping\.CategorySchemeMap=.+

**ConceptSchemeMapReferenceType:** A reference type for a concept scheme map.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.structuremapping\.ConceptSchemeMap=.+

**DatePatternMapReferenceType:** A reference type for a date pattern map.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.structuremapping\.DatePatternMap=.+

**EpochMapReferenceType:** A reference type for an epoch map.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.structuremapping\.EpochMap=.+

**FrequencyFormatMappingReferenceType:** A reference type for a frequnecy format mapping.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.structuremapping\.FrequencyFormatMapping=.+

**OrganisationSchemeMapReferenceType:** A reference type for a organisation scheme map.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.structuremapping\.OrganisationSchemeMap=.+

**ReportingTaxonomyMapReferenceType:** A reference type for a reporting taxonomy map.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.structuremapping\.ReportingTaxonomyMap=.+

**RepresentationMapReferenceType:** A reference type for a representation map.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.structuremapping\.RepresentationMap=.+

**StructureMapReferenceType:** A reference type for a structure map.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.structuremapping\.StructureMap=.+

**CustomTypeReferenceType:** A reference type for a custom type.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.CustomType=.+

**CustomTypeSchemeReferenceType:** A reference type for a custom type scheme.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.CustomTypeScheme=.+

**NamePersonalisationReferenceType:** A reference type for a name personalisation.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.NamePersonalisation=.+

**NamePersonalisationSchemeReferenceType:** A reference type for a name personalisation scheme.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.NamePersonalisationScheme=.+

**RulesetReferenceType:** A reference type for a ruleset.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.Ruleset=.+

**RulesetSchemeReferenceType:** A reference type for a ruleset scheme.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.RulesetScheme=.+

**TransformationReferenceType:** A reference type for a transformation.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.Transformation=.+

**TransformationSchemeReferenceType:** A reference type for a transformation scheme.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.TransformationScheme=.+

**UserDefinedOperatorReferenceType:** A reference type for a user defined operator.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.UserDefinedOperator=.+

**UserDefinedOperatorSchemeReferenceType:** A reference type for a user defined operator scheme.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.UserDefinedOperatorScheme=.+

**VtlMappingReferenceType:** A reference type for a VTL mapping.

Derived by restriction of ComponentUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.VtlCodelistMapping=.+.+\.transformation\.VtlConceptMapping=.+.+\.transformation\.VtlDataflowMapping=.+

**VtlMappingSchemeReferenceType:** A reference type for a VTL mapping scheme.

Derived by restriction of MaintainableUrnReferenceType .  
Regular Expression Pattern: .+\.transformation\.VtlMappingScheme=.+

**VersionReferenceType:** VersionReferenceType defines the structure of version number in a reference. When semantic versioning is used, the major, minor, or patch version parts can be wildcarded using "+" as an extension. For example, 2+.3.1 means the currently latest available version >= 2.3.1 (even if not backwards compatible). Note that wildcarded semantic version references cannot be combined with version extended reference (e.g. 2.3+.1-draft is not permissable). Version-extended (e.g. 1.3.1-draft) and legacy version numbers (e.g. 1 or 1.0) are also supported.

Union of:

LegacyVersionNumberType, SemanticVersionNumberType, SemanticVersionReferenceType.

**SemanticVersionReferenceType:** SemanticVersionReferenceType is a simple type for referencing semantic version numbers. It allows for the wildcarding of only one the major, minor, or patch version parts using "+".

Derived by restriction of xs:string .  
Regular Expression Pattern: ((0|[1-9]\d\*)\+?)(\.((0|[1-9]\d\*))){2}((0|[1-9]\d\*))(\.((0|[1-9]\d\*)\+?))(\.((0|[1-9]\d\*)))((0|[1-9]\d\*)\+?)(\.((0|[1-9]\d\*)))(\.((0|[1-9]\d\*)\+?))

**WildcardVersionType:** WildcardVersionType combines the VersionType and WildcardType to allow a reference to either a specific version of an object, or to wildcard the version in the reference by specifying the '\*' value.

Union of:

LegacyVersionNumberType, SemanticVersionNumberType, SemanticVersionReferenceType, WildcardType.

**WildcardType:** WildcardType is a single value code list, used to include the '\*' character - indicating that the identification component is wildcarded.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| \* | Indicates that any value of the identifier component is allowed. |

**NestedIDType:** NestedIDType is the least restrictive form of an identifier used throughout all SDMX-ML messages. It allows for a hierarchical identifier, with each portion separated by the '.' character. For the identifier portions, valid characters include A-Z, a-z, @, 0-9, \_, -, $.

Derived by restriction of xs:string .  
Regular Expression Pattern: [A-Za-z0-9\_@$\-]+(\.[A-Za-z0-9\_@$\-]+)\*

**TwoLevelIDType:** TwoLevelIDType defines an identifier with exactly two levels.

Derived by restriction of NestedIDType .  
Regular Expression Pattern: [A-Za-z0-9\_@$\-]+\.[A-Za-z0-9\_@$\-]+

**IDType:** IDType provides a type which is used for restricting the characters in codes and IDs throughout all SDMX-ML messages. Valid characters include A-Z, a-z, @, 0-9, \_, -, $.

Derived by restriction of NestedIDType .  
Regular Expression Pattern: [A-Za-z0-9\_@$\-]+

**NCNameIDType:** NCNameIDType restricts the IDType, so that the id may be used to generate valid XML components. IDs created from this type conform to the W3C XML Schema NCNAME type, and therefore can be used as element or attribute names.

Derived by restriction of IDType .  
Regular Expression Pattern: [A-Za-z][A-Za-z0-9\_\-]\*

**NestedNCNameIDType:** NestedNCNameIDType restricts the NestedIDType, so that the id may be used to generate valid XML components. IDs created from this type conform to the W3C XML Schema NCNAME type, and therefore can be used as element or attribute names.

Derived by restriction of NestedIDType .  
Regular Expression Pattern: [A-Za-z][A-Za-z0-9\_\-]\*(\.[A-Za-z][A-Za-z0-9\_\-]\*)\*

**SingleNCNameIDType:** SingleNCNameIDType restricts the NestedNCNameIDType to allow only one level. Note that this is the same pattern as the NCNameIDType, but can be used when the base type to be restricted is a nested NCNameIDType (where as the NCNameIDType could only restrict the IDType).

Derived by restriction of NestedNCNameIDType .  
Regular Expression Pattern: [A-Za-z][A-Za-z0-9\_\-]\*

**VersionType:** VersionType is used to communicate version information. Semantic versioning, based on 3 or 4 version parts (major.minor.patch[-extension]) is supported. The legency SDMX version format is also support.

Union of:

LegacyVersionNumberType, SemanticVersionNumberType.

**SemanticVersionNumberType:** SemanticVersionNumberType is a simple type for validating semantic version in the format (major.minor.patch[-extension]).

Derived by restriction of xs:string .  
Regular Expression Pattern: (0|[1-9]\d\*)(\.(0|[1-9]\d\*)){2}(\-(([A-Za-z\-]|([A-Za-z\-][A-Za-z0-9\-]+)|([A-Za-z0-9\-]+[A-Za-z\-][A-Za-z0-9\-]\*))|(0|[1-9][0-9]\*))(\.(([A-Za-z\-]|([A-Za-z\-][A-Za-z0-9\-]+)|([A-Za-z0-9\-]+[A-Za-z\-][A-Za-z0-9\-]\*))|(0|[1-9][0-9]\*)))\*)?

**LegacyVersionNumberType:** LegacyVersionNumberType describes the version number format previously supported in SDMX. The format is restricted to allow for simple incrementing and sorting of version number. The version consists of a set of maximum 2 numeric components, separated by the '.' character. When processing version, each numeric component (the number preceding and following any '.' character) should be parsed as an integer. Thus, a version of 1.3 and 1.03 would be equivalent, as both the '3' component and the '03' component would parse to an integer value of 3.

Derived by restriction of xs:string .  
Regular Expression Pattern: (0|[1-9]\d\*)(\.(0|[1-9]\d\*))?