SDMX Standards: Section 3A PaRT II

SDMX-ML:

Schema and Documentation

Common Namespace

Version 2.1

April 2011

© SDMX 2011

http://www.sdmx.org/

Contents

[1 Introduction 1](#_Toc290375228)

[2 Schema Documentation 1](#_Toc290375229)

[2.1 Common Namespace 1](#_Toc290375230)

[2.1.1 Summary 1](#_Toc290375231)

[2.1.2 Global Elements 2](#_Toc290375232)

[2.1.3 Complex Types 6](#_Toc290375233)

[2.1.4 Simple Types 316](#_Toc290375234)

# 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/v2\_1/common**

### Summary

Referenced Namespaces:

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

Contents:

69 Global Elements  
258 Complex Types  
76 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 my reference it.

**Any (EmptyType):**Any is an empty element that denotes an object of any type.

**Agency (EmptyType):**Agency is an empty element that denotes an agency object.

**AgencyScheme (EmptyType):**AgencyScheme is an empty element that denotes an agency scheme object.

**AttachmentConstraint (EmptyType):**AttachmentConstraint is an empty element that denotes an attachment constraint object.

**Attribute (EmptyType):**Attribute is an empty element that denotes an attribute object.

**AttributeDescriptor (EmptyType):**AttributeDescriptor is an empty element that denotes an attribute descriptor object.

**Categorisation (EmptyType):**Categorisation is an empty element that denotes a categorisation object.

**Category (EmptyType):**Category is an empty element that denotes a category object.

**CategorySchemeMap (EmptyType):**CategorySchemeMap is an empty element that denotes a category scheme map object.

**CategoryScheme (EmptyType):**CategoryScheme is an empty element that denotes a category scheme object.

**Code (EmptyType):**Code is an empty element that denotes a code object.

**CodeMap (EmptyType):**CodeMap is an empty element that denotes a code map object.

**Codelist (EmptyType):**Codelist is an empty element that denotes a code list object.

**CodelistMap (EmptyType):**CodelistMap is an empty element that denotes a code list map object.

**ComponentMap (EmptyType):**ComponentMap is an empty element that denotes a component map object.

**Concept (EmptyType):**Concept is an empty element that denotes a concept object.

**ConceptMap (EmptyType):**ConceptMap is an empty element that denotes a concept map object.

**ConceptScheme (EmptyType):**ConceptScheme is an empty element that denotes a concept scheme object.

**ConceptSchemeMap (EmptyType):**ConceptSchemeMap is an empty element that denotes a concept scheme map object.

**ConstraintTarget (EmptyType):**ConstraintTarget is an empty element that denotes a constraint target object.

**ContentConstraint (EmptyType):**ContentConstraint is an empty element that denotes a content constraint object.

**Dataflow (EmptyType):**Dataflow is an empty element that denotes a data flow object.

**DataConsumer (EmptyType):**DataConsumer is an empty element that denotes a data consumer object.

**DataConsumerScheme (EmptyType):**DataConsumerScheme is an empty element that denotes a data consumer scheme object.

**DataProvider (EmptyType):**DataProvider is an empty element that denotes a data provider object.

**DataProviderScheme (EmptyType):**DataProviderScheme is an empty element that denotes a data provider scheme object.

**DataSetTarget (EmptyType):**DataSetTarget is an empty element that denotes a data set target object.

**DataStructure (EmptyType):**DataStructure is an empty element that denotes a data structure definition object.

**Dimension (EmptyType):**Dimension is an empty element that denotes a dimension object.

**DimensionDescriptor (EmptyType):**DimensionDescriptor is an empty element that denotes a dimension descriptor object.

**DimensionDescriptorValuesTarget (EmptyType):**DimensionDescriptorValuesTarget is an empty element that denotes a dimension descriptor values target object.

**GroupDimensionDescriptor (EmptyType):**GroupDimensionDescriptor is an empty element that denotes a group dimension descriptor object.

**HierarchicalCode (EmptyType):**HierarchicalCode is an empty element that denotes a hierarchical code object.

**HierarchicalCodelist (EmptyType):**HierarchicalCodelist is an empty element that denotes a hierarchical codelist object.

**Hierarchy (EmptyType):**Hierarchy is an empty element that denotes a hierarchy within a hiearcharchical codelist.

**HybridCodelistMap (EmptyType):**HybridCodelistMap is an empty element that denotes a hybrid codelist map object.

**HybridCodeMap (EmptyType):**HybridCodeMap is an empty element that denotes a hybrid code map object.

**IdentifiableObjectTarget (EmptyType):**IdentifiableObjectTarget is an empty element that denotes an identifiable object target object.

**Level (EmptyType):**Level is an empty element that denotes a level object.

**MeasureDescriptor (EmptyType):**MeasureDescriptor is an empty element that denotes a measure descriptor object.

**MeasureDimension (EmptyType):**MeasureDimension is an empty element that denotes a measure dimension object.

**Metadataflow (EmptyType):**Metadataflow is an empty element that denotes a metadata flow object.

**MetadataAttribute (EmptyType):**MetadataAttribute is an empty element that denotes a metadata attribute object.

**MetadataSet (EmptyType):**MetadataSet is an empty element that denotes a metadata set object.

**MetadataStructure (EmptyType):**MetadataStructure is an empty element that denotes a metadata structure definition object.

**MetadataTarget (EmptyType):**MetadataTarget is an empty element that denotes a metadata target object.

**OrganisationMap (EmptyType):**OrganisationMap is an empty element that denotes an organisation map object.

**OrganisationSchemeMap (EmptyType):**OrganisationSchemeMap is an empty element that denotes an organisation scheme map object.

**OrganisationUnit (EmptyType):**OrganisationUnit is an empty element that denotes an organisation unit object.

**OrganisationUnitScheme (EmptyType):**OrganisationUnitScheme is an empty element that denotes an organisation unit scheme object.

**PrimaryMeasure (EmptyType):**PrimaryMeasure is an empty element that denotes a primary measure object.

**Process (EmptyType):**Process is an empty element that denotes a process object.

**ProcessStep (EmptyType):**ProcessStep is an empty element that denotes a process step object.

**ProvisionAgreement (EmptyType):**ProvisionAgreement is an empty element that denotes a provision agreement object.

**ReportingCategory (EmptyType):**ReportingCategory is an empty element that denotes a reporting category object.

**ReportingCategoryMap (EmptyType):**ReportingCategoryMap is an empty element that denotes a reporting category map object.

**ReportingTaxonomy (EmptyType):**ReportingTaxonomy is an empty element that denotes a reporting taxonomy object.

**ReportingTaxonomyMap (EmptyType):**ReportingTaxonomyMap is an empty element that denotes a reporting taxonomy map object.

**ReportPeriodTarget (EmptyType):**ReportPeriodTarget is an empty element that denotes a report period target object.

**ReportStructure (EmptyType):**ReportStructure is an empty element that denotes a report structure object.

**StructureMap (EmptyType):**StructureMap is an empty element that denotes a structure map object.

**StructureSet (EmptyType):**StructureSet is an empty element that denotes a structure set object.

**TimeDimension (EmptyType):**TimeDimension is an empty element that denotes a time dimension object.

**Transition (EmptyType):**Transition is an empty element that denotes a transition object.

### Complex Types

**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. |

**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 my 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\*

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 | xs:anyURI | AnnotationURL is a URI - typically a URL - which points to an external resource which may contain or supplement the annotation. 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. |

**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 |  |

***RegionType*:**RegionType is an abstract type which defines a generic constraint region. This type can be refined to define regions for data or metadata sets. A region is defined by a collection of key values - each of which a collection of values for a component which disambiguates data or metadata (i.e. dimensions or the target objects of a metadata target). For each region, as collection of attribute values can be provided. Taken together, the key values and attributes serve to identify or describe a subset of a data or metadata set. Finally, the region can flagged as being included or excluded, although this flag only makes sense when the region is used in a particular context.

Attributes:

include?

Content:

KeyValue\*, Attribute\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| include (default: true) | xs:boolean | The include attribute indicates that the region is to be included or excluded within the context in which it is defined. For example, if the regions is defined as part of a content constraint, the exclude flag would mean the data identified by the region is not present. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| KeyValue | *ComponentValueSetTyp e* | KeyValue contains a reference to a component which disambiguates the data or metadata (i.e. a dimension or target object) and provides a collection of values for the component. The collection of values can be flagged as being inclusive or exclusive to the region being defined. Any key component that is not included is assumed to be wild carded, which is to say that the cube includes all possible values for the un-referenced key components. Further, this assumption applies to the values of the components as well. The values for any given component can only be sub-setted in the region by explicit inclusion or exclusion. For example, a dimension X which has the possible values of 1, 2, 3 is assumed to have all of these values if a key value is not defined. If a key value is defined with an inclusion attribute of true and the values of 1 and 2, the only the values of 1 and 2 for dimension X are included in the definition of the region. If the key value is defined with an inclusion attribute of false and the value of 1, then the values of 2 and 3 for dimension X are included in the definition of the region. Note that any given key component must only be referenced once in the region. |
| Attribute | *ComponentValueSetTyp e* | Attributes contains a reference to an attribute component (data or metadata) and provides a collection of values for the referenced attribute. This serves to state that for the key which defines the region, the attributes that are specified here have or do not have (depending to the include attribute of the value set) the values provided. It is possible to provide and attribute reference without specifying values, for the purpose of stating the attribute is absent (include = false) or present with an unbounded set of values. As opposed to key components, which are assumed to be wild carded if absent, no assumptions are made about the absence of an attribute. Only attributes which are explicitly stated to be present or absent from the region will be know. All unstated attributes for the set cannot be assumed to absent or present. |

***ComponentValueSetType*:**ComponentValueSetType is an abstract base type which is used to provide a set of value for a referenced component. Implementations of this type will be based on a particular component type and refine the allowed values to reflect the types of values that are possible for that type of component.

Attributes:

id, include?

Content:

(Value+ | DataSet+ | DataKey+ | Object+ | TimeRange)?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedNCNameIDType | The id attribute provides the identifier for the component for which values are being provided. This base type allows for a nested identifier to be provided, for the purpose of referencing a nested component (i.e. a metadata attribute). However, specific implementations will restrict this representation to only allow single level identifiers where appropriate. |
| include (default: true) | xs:boolean | The include attribute indicates whether the values provided for the referenced component are to be included are excluded from the region in which they are defined. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Value | SimpleValueType | Value provides a simple value for the component, such as a coded, numeric, or simple text value. This type of component value is applicable for dimensions and attributes. |
| DataSet | SetReferenceType | DataSet provides a reference to a data set and is used to state a value for the data set target component in a metadata target. |
| DataKey | DataKeyType | DataKey provides a set of dimension references and value, which form a full or partial data key. This is used to state a value for the key descriptor values target component in a metadata target. |
| Object | ObjectReferenceType | Object provides a reference to an Identifiable object in the SDMX Information Model. This is used to state a value for an identifiable target component in a metadata target. |
| TimeRange | TimeRangeValueType | TimeValue provides a value for a component which has a time representation. This is repeatable to allow for a range to be specified, although a single value can also be provided. An operator is available on this to indicate whether the specified value indicates an exact value or the beginning/end of a range (inclusive or exclusive). |

***DistinctKeyType*:**DistinctKeyType is an abstract base type which is a special type of region that only defines a distinct region of data or metadata. For each component defined in the region, only a single values is provided. However, the same rules that apply to the general region about unstated components being wild carded apply here as well. Thus, this type can define a distinct full or partial key for data or metadata.

Derivation:

*RegionType* (restriction)   
   *DistinctKeyType*



Attributes:

include?

Content:

KeyValue+

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| include (fixed: true) | xs:boolean | The include attribute has a fixed value of true for a distinct key, since such a key is always assumed to identify existing data or metadata. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| KeyValue | *DinstinctKeyValueTyp e* | KeyValue contains a reference to a component which disambiguates the data or metadata (i.e. a dimension or target object) and provides a collection of values for the component. The collection of values can be flagged as being inclusive or exclusive to the region being defined. Any key component that is not included is assumed to be wild carded, which is to say that the cube includes all possible values for the un-referenced key components. Further, this assumption applies to the values of the components as well. The values for any given component can only be sub-setted in the region by explicit inclusion or exclusion. For example, a dimension X which has the possible values of 1, 2, 3 is assumed to have all of these values if a key value is not defined. If a key value is defined with an inclusion attribute of true and the values of 1 and 2, the only the values of 1 and 2 for dimension X are included in the definition of the region. If the key value is defined with an inclusion attribute of false and the value of 1, then the values of 2 and 3 for dimension X are included in the definition of the region. Note that any given key component must only be referenced once in the region. |

***DinstinctKeyValueType*:**DinstinctKeyValueType is an abstract base type which defines a singular, required value for a key component.

Derivation:

*ComponentValueSetType* (restriction)   
   *DinstinctKeyValueType*



Attributes:

id, include?

Content:

(Value | DataSet | DataKey | Object)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | SingleNCNameIDType | The id attribute provides the identifier for the component for which values are being provided. This base type allows for a nested identifier to be provided, for the purpose of referencing a nested component (i.e. a metadata attribute). However, specific implementations will restrict this representation to only allow single level identifiers where appropriate. |
| include (fixed: true) | xs:boolean | The include attribute indicates whether the values provided for the referenced component are to be included are excluded from the region in which they are defined. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Value | SimpleKeyValueType | Value provides a simple value for the component, such as a coded, numeric, or simple text value. This type of component value is applicable for dimensions and attributes. |
| DataSet | SetReferenceType | DataSet provides a reference to a data set and is used to state a value for the data set target component in a metadata target. |
| DataKey | DataKeyType | DataKey provides a set of dimension references and value, which form a full or partial data key. This is used to state a value for the key descriptor values target component in a metadata target. |
| Object | ObjectReferenceType | Object provides a reference to an Identifiable object in the SDMX Information Model. This is used to state a value for an identifiable target component in a metadata target. |

**DataKeyType:**DataKeyType is a region which defines a distinct full or partial data key. The key consists of a set of values, each referencing a dimension and providing a single value for that dimension. The purpose of the key is to define a subset of a data set (i.e. the observed value and data attribute) which have the dimension values provided in this definition. Any dimension not stated explicitly in this key is assumed to be wild carded, thus allowing for the definition of partial data keys.

Derivation:

*RegionType* (restriction)   
   *DistinctKeyType* (restriction)   
         DataKeyType



Attributes:

include?

Content:

KeyValue+

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| include (fixed: true) | xs:boolean | The include attribute has a fixed value of true for a distinct key, since such a key is always assumed to identify existing data or metadata. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| KeyValue | DataKeyValueType | KeyValue contains a reference to a component which disambiguates the data or metadata (i.e. a dimension or target object) and provides a collection of values for the component. The collection of values can be flagged as being inclusive or exclusive to the region being defined. Any key component that is not included is assumed to be wild carded, which is to say that the cube includes all possible values for the un-referenced key components. Further, this assumption applies to the values of the components as well. The values for any given component can only be sub-setted in the region by explicit inclusion or exclusion. For example, a dimension X which has the possible values of 1, 2, 3 is assumed to have all of these values if a key value is not defined. If a key value is defined with an inclusion attribute of true and the values of 1 and 2, the only the values of 1 and 2 for dimension X are included in the definition of the region. If the key value is defined with an inclusion attribute of false and the value of 1, then the values of 2 and 3 for dimension X are included in the definition of the region. Note that any given key component must only be referenced once in the region. |

**DataKeyValueType:**DataKeyValueType is a type for providing a dimension value for the purpose of defining a distinct data key. Only a single value can be provided for the dimension.

Derivation:

*ComponentValueSetType* (restriction)   
   *DinstinctKeyValueType* (restriction)   
         DataKeyValueType



Attributes:

id, include?

Content:

Value

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | SingleNCNameIDType | The id attribute provides the identifier for the component for which values are being provided. This base type allows for a nested identifier to be provided, for the purpose of referencing a nested component (i.e. a metadata attribute). However, specific implementations will restrict this representation to only allow single level identifiers where appropriate. |
| include (fixed: true) | xs:boolean | The include attribute indicates whether the values provided for the referenced component are to be included are excluded from the region in which they are defined. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Value | SimpleKeyValueType | Value provides a simple value for the component, such as a coded, numeric, or simple text value. This type of component value is applicable for dimensions and attributes. |

**MetadataKeyType:**MetadataKeyType is a region which defines a distinct full or partial metadata key. The key consists of a set of values, each referencing a target object for the metadata target referenced in the metadataTarget attribute, which must be defined in the report structure referenced in the report attribute. Each target object can be assigned a single value. If an target object from the reference metadata target is not included in this key, the value of that is assumed to be all known objects for a reference target object, all possible keys for a key descriptor values target object, or all dates for report period target object. The purpose of this key reference a metadata conforming to a particular report structure for given object or set of objects.

Derivation:

*RegionType* (restriction)   
   *DistinctKeyType* (restriction)   
         MetadataKeyType



Attributes:

include?, report, metadataTarget

Content:

KeyValue+

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| include (fixed: true) | xs:boolean | The include attribute has a fixed value of true for a distinct key, since such a key is always assumed to identify existing data or metadata. |
| report | IDType | The report attribute is required and holds the identifier of the report structure which the reference metadata being defined by this key is based on. |
| metadataTarget | IDType | The metadataTarget attribute is required and identifies the metadata target for the report structure which this key is based upon. Note that a report structure can have multiple metadata targets, so to properly determine the object or objects for which the key applies, the proper metadata target must be identified. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| KeyValue | MetadataKeyValueType | KeyValue contains a reference to a component which disambiguates the data or metadata (i.e. a dimension or target object) and provides a collection of values for the component. The collection of values can be flagged as being inclusive or exclusive to the region being defined. Any key component that is not included is assumed to be wild carded, which is to say that the cube includes all possible values for the un-referenced key components. Further, this assumption applies to the values of the components as well. The values for any given component can only be sub-setted in the region by explicit inclusion or exclusion. For example, a dimension X which has the possible values of 1, 2, 3 is assumed to have all of these values if a key value is not defined. If a key value is defined with an inclusion attribute of true and the values of 1 and 2, the only the values of 1 and 2 for dimension X are included in the definition of the region. If the key value is defined with an inclusion attribute of false and the value of 1, then the values of 2 and 3 for dimension X are included in the definition of the region. Note that any given key component must only be referenced once in the region. |

**MetadataKeyValueType:**MetadataKeyValueType is a type for providing a target object value for the purpose of defining a distinct metadata key. Only a single value can be provided for the target object.

Derivation:

*ComponentValueSetType* (restriction)   
   *DinstinctKeyValueType* (restriction)   
         MetadataKeyValueType



Attributes:

id, include?

Content:

(Value | DataSet | DataKey | Object)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | SingleNCNameIDType | The id attribute provides the identifier for the component for which values are being provided. This base type allows for a nested identifier to be provided, for the purpose of referencing a nested component (i.e. a metadata attribute). However, specific implementations will restrict this representation to only allow single level identifiers where appropriate. |
| include (fixed: true) | xs:boolean | The include attribute indicates whether the values provided for the referenced component are to be included are excluded from the region in which they are defined. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Value | SimpleKeyValueType | Value provides a simple value for the component, such as a coded, numeric, or simple text value. This type of component value is applicable for dimensions and attributes. |
| DataSet | SetReferenceType | DataSet provides a reference to a data set and is used to state a value for the data set target component in a metadata target. |
| DataKey | DataKeyType | DataKey provides a set of dimension references and value, which form a full or partial data key. This is used to state a value for the key descriptor values target component in a metadata target. |
| Object | ObjectReferenceType | Object provides a reference to an Identifiable object in the SDMX Information Model. This is used to state a value for an identifiable target component in a metadata target. |

**CubeRegionType:**CubeRegionType defines the structure of a data cube region. This is based on the abstract RegionType and simply refines the key and attribute values to conform with what is applicable for dimensions and attributes, respectively. See the documentation of the base type for more details on how a region is defined.

Derivation:

*RegionType* (restriction)   
   CubeRegionType



Attributes:

include?

Content:

KeyValue\*, Attribute\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| include (default: true) | xs:boolean | The include attribute indicates that the region is to be included or excluded within the context in which it is defined. For example, if the regions is defined as part of a content constraint, the exclude flag would mean the data identified by the region is not present. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| KeyValue | CubeRegionKeyType | KeyValue contains a reference to a component which disambiguates the data or metadata (i.e. a dimension or target object) and provides a collection of values for the component. The collection of values can be flagged as being inclusive or exclusive to the region being defined. Any key component that is not included is assumed to be wild carded, which is to say that the cube includes all possible values for the un-referenced key components. Further, this assumption applies to the values of the components as well. The values for any given component can only be sub-setted in the region by explicit inclusion or exclusion. For example, a dimension X which has the possible values of 1, 2, 3 is assumed to have all of these values if a key value is not defined. If a key value is defined with an inclusion attribute of true and the values of 1 and 2, the only the values of 1 and 2 for dimension X are included in the definition of the region. If the key value is defined with an inclusion attribute of false and the value of 1, then the values of 2 and 3 for dimension X are included in the definition of the region. Note that any given key component must only be referenced once in the region. |
| Attribute | AttributeValueSetTyp e | Attributes contains a reference to an attribute component (data or metadata) and provides a collection of values for the referenced attribute. This serves to state that for the key which defines the region, the attributes that are specified here have or do not have (depending to the include attribute of the value set) the values provided. It is possible to provide and attribute reference without specifying values, for the purpose of stating the attribute is absent (include = false) or present with an unbounded set of values. As opposed to key components, which are assumed to be wild carded if absent, no assumptions are made about the absence of an attribute. Only attributes which are explicitly stated to be present or absent from the region will be know. All unstated attributes for the set cannot be assumed to absent or present. |

**MetadataTargetRegionType:**MetadataTargetRegionType defines the structure of a metadata target region. A metadata target region must define the report structure and the metadata target from that structure on which the region is based. This type is based on the abstract RegionType and simply refines the key and attribute values to conform with what is applicable for target objects and metadata attributes, respectively. See the documentation of the base type for more details on how a region is defined.

Derivation:

*RegionType* (restriction)   
   MetadataTargetRegionType



Attributes:

include?, report, metadataTarget

Content:

KeyValue\*, Attribute\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| include (default: true) | xs:boolean | The include attribute indicates that the region is to be included or excluded within the context in which it is defined. For example, if the regions is defined as part of a content constraint, the exclude flag would mean the data identified by the region is not present. |
| report | IDType | The report attribute is required and holds the identifier of the report structure which the reference metadata being defined by this region is based on. |
| metadataTarget | IDType | The metadataTarget attribute is required and identifies the metadata target for the report structure which this region is based upon. Note that a report structure can have multiple metadata targets, so to properly determine the object or objects for which the region applies, the proper metadata target must be identified. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| KeyValue | MetadataTargetRegion KeyType | KeyValue contains a reference to a component which disambiguates the data or metadata (i.e. a dimension or target object) and provides a collection of values for the component. The collection of values can be flagged as being inclusive or exclusive to the region being defined. Any key component that is not included is assumed to be wild carded, which is to say that the cube includes all possible values for the un-referenced key components. Further, this assumption applies to the values of the components as well. The values for any given component can only be sub-setted in the region by explicit inclusion or exclusion. For example, a dimension X which has the possible values of 1, 2, 3 is assumed to have all of these values if a key value is not defined. If a key value is defined with an inclusion attribute of true and the values of 1 and 2, the only the values of 1 and 2 for dimension X are included in the definition of the region. If the key value is defined with an inclusion attribute of false and the value of 1, then the values of 2 and 3 for dimension X are included in the definition of the region. Note that any given key component must only be referenced once in the region. |
| Attribute | MetadataAttributeVal ueSetType | Attributes contains a reference to an attribute component (data or metadata) and provides a collection of values for the referenced attribute. This serves to state that for the key which defines the region, the attributes that are specified here have or do not have (depending to the include attribute of the value set) the values provided. It is possible to provide and attribute reference without specifying values, for the purpose of stating the attribute is absent (include = false) or present with an unbounded set of values. As opposed to key components, which are assumed to be wild carded if absent, no assumptions are made about the absence of an attribute. Only attributes which are explicitly stated to be present or absent from the region will be know. All unstated attributes for the set cannot be assumed to absent or present. |

**CubeRegionKeyType:**CubeRegionKeyType is a type for providing a set of values for a dimension for the purpose of defining a data cube region. A set of distinct value can be provided, or if this dimension is represented as time, and time range can be specified.

Derivation:

*ComponentValueSetType* (restriction)   
   CubeRegionKeyType



Attributes:

id, include?

Content:

(Value+ | TimeRange)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | SingleNCNameIDType | The id attribute provides the identifier for the component for which values are being provided. This base type allows for a nested identifier to be provided, for the purpose of referencing a nested component (i.e. a metadata attribute). However, specific implementations will restrict this representation to only allow single level identifiers where appropriate. |
| include (default: true) | xs:boolean | The include attribute indicates whether the values provided for the referenced component are to be included are excluded from the region in which they are defined. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Value | SimpleValueType | Value provides a simple value for the component, such as a coded, numeric, or simple text value. This type of component value is applicable for dimensions and attributes. |
| TimeRange | TimeRangeValueType | TimeValue provides a value for a component which has a time representation. This is repeatable to allow for a range to be specified, although a single value can also be provided. An operator is available on this to indicate whether the specified value indicates an exact value or the beginning/end of a range (inclusive or exclusive). |

**MetadataTargetRegionKeyType:**MetadataTargetRegionKeyType is a type for providing a set of values for a target object in a metadata target of a re fence metadata report. A set of values or a time range can be provided for a report period target object. A collection of the respective types of references can be provided for data set reference and identifiable object reference target objects. For a key descriptor values target object, a collection of data keys can be provided.

Derivation:

*ComponentValueSetType* (restriction)   
   MetadataTargetRegionKeyType



Attributes:

id, include?

Content:

(Value+ | DataSet+ | DataKey+ | Object+ | TimeRange)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | SingleNCNameIDType | The id attribute provides the identifier for the component for which values are being provided. This base type allows for a nested identifier to be provided, for the purpose of referencing a nested component (i.e. a metadata attribute). However, specific implementations will restrict this representation to only allow single level identifiers where appropriate. |
| include (default: true) | xs:boolean | The include attribute indicates whether the values provided for the referenced component are to be included are excluded from the region in which they are defined. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Value | SimpleKeyValueType | Value provides a simple value for the component, such as a coded, numeric, or simple text value. This type of component value is applicable for dimensions and attributes. |
| DataSet | SetReferenceType | DataSet provides a reference to a data set and is used to state a value for the data set target component in a metadata target. |
| DataKey | DataKeyType | DataKey provides a set of dimension references and value, which form a full or partial data key. This is used to state a value for the key descriptor values target component in a metadata target. |
| Object | ObjectReferenceType | Object provides a reference to an Identifiable object in the SDMX Information Model. This is used to state a value for an identifiable target component in a metadata target. |
| TimeRange | TimeRangeValueType | TimeValue provides a value for a component which has a time representation. This is repeatable to allow for a range to be specified, although a single value can also be provided. An operator is available on this to indicate whether the specified value indicates an exact value or the beginning/end of a range (inclusive or exclusive). |

**AttributeValueSetType:**AttributeValueSetType defines the structure for providing values for a data attribute. If no values are provided, the attribute is implied to include/excluded from the region in which it is defined, with no regard to the value of the data attribute. Note that for metadata attributes which occur within other metadata attributes, a nested identifier can be provided. For example, a value of CONTACT.ADDRESS.STREET refers to the metadata attribute with the identifier STREET which exists in the ADDRESS metadata attribute in the CONTACT metadata attribute, which is defined at the root of the report structure.

Derivation:

*ComponentValueSetType* (restriction)   
   AttributeValueSetType



Attributes:

id, include?

Content:

(Value+ | TimeRange)?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | SingleNCNameIDType | The id attribute provides the identifier for the component for which values are being provided. This base type allows for a nested identifier to be provided, for the purpose of referencing a nested component (i.e. a metadata attribute). However, specific implementations will restrict this representation to only allow single level identifiers where appropriate. |
| include (default: true) | xs:boolean | The include attribute indicates whether the values provided for the referenced component are to be included are excluded from the region in which they are defined. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Value | SimpleValueType | Value provides a simple value for the component, such as a coded, numeric, or simple text value. This type of component value is applicable for dimensions and attributes. |
| TimeRange | TimeRangeValueType | TimeValue provides a value for a component which has a time representation. This is repeatable to allow for a range to be specified, although a single value can also be provided. An operator is available on this to indicate whether the specified value indicates an exact value or the beginning/end of a range (inclusive or exclusive). |

**MetadataAttributeValueSetType:**MetadataAttributeValueSetType defines the structure for providing values for a metadata attribute. If no values are provided, the attribute is implied to include/excluded from the region in which it is defined, with no regard to the value of the metadata attribute.

Derivation:

*ComponentValueSetType* (restriction)   
   MetadataAttributeValueSetType



Attributes:

id, include?

Content:

(Value+ | TimeRange)?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedNCNameIDType | The id attribute provides the identifier for the component for which values are being provided. This base type allows for a nested identifier to be provided, for the purpose of referencing a nested component (i.e. a metadata attribute). However, specific implementations will restrict this representation to only allow single level identifiers where appropriate. |
| include (default: true) | xs:boolean | The include attribute indicates whether the values provided for the referenced component are to be included are excluded from the region in which they are defined. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Value | SimpleValueType | Value provides a simple value for the component, such as a coded, numeric, or simple text value. This type of component value is applicable for dimensions and attributes. |
| TimeRange | TimeRangeValueType | TimeValue provides a value for a component which has a time representation. This is repeatable to allow for a range to be specified, although a single value can also be provided. An operator is available on this to indicate whether the specified value indicates an exact value or the beginning/end of a range (inclusive or exclusive). |

**SimpleValueType:**SimpleValueType contains a simple value for a component, and if that value is from a code list, the ability to indicate that child codes in a simple hierarchy are part of the value set of the component for the region.

Derivation:

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



Attributes:

cascadeValues?

Content:

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| cascadeValues (default: false) | xs:boolean | The cascadeValues attribute, if true, indicates that if the value is taken from a code all child codes in a simple hierarchy are understood be included in the region. |

**SimpleKeyValueType:**SimpleKeyValueType derives from the SimpleValueType, but does not allow for the cascading of value in the hierarchy, as keys are meant to describe a distinct full or partial key.

Derivation:

xs:anySimpleType (restriction)   
   xs:string (extension)   
         SimpleValueType (restriction)   
               SimpleKeyValueType



Content:

**TimeRangeValueType:**TimeRangeValueType allows a time period value to be expressed as a range. It can be expressed as the period before a period, after a period, or between two periods. Each of these properties can specify their inclusion in regards to the range.

Content:

(BeforePeriod | AfterPeriod | (StartPeriod, EndPeriod))

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| BeforePeriod | TimePeriodRangeType | BeforePeriod is the period before which the period is meant to cover. This date may be inclusive or exclusive in the range. |
| AfterPeriod | TimePeriodRangeType | AfterPeriod is the period after which the period is meant to cover. This date may be inclusive or exclusive in the range. |
| StartPeriod | TimePeriodRangeType | StartPeriod is the start date or the range that the queried date must occur within. This date may be inclusive or exclusive in the range. |
| EndPeriod | TimePeriodRangeType | EndPeriod is the end period of the range. This date may be inclusive or exclusive in the range. |

**TimePeriodRangeType:**TimePeriodRangeType defines a time period, and indicates whether it is inclusive in a range.

Derivation:

xs:anySimpleType (restriction)   
   ObservationalTimePeriodType (extension)   
         TimePeriodRangeType



Attributes:

isInclusive?

Content:

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| isInclusive (default: true) | xs:boolean | The isInclusive attribute, when true, indicates that the time period specified is included in the range. |

***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:

(ProvisionAgrement | 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** |
| --- | --- | --- |
| ProvisionAgrement | ProvisionAgreementRe ferenceType | ProvisionAgreement references a provision agreement which the data or metadata is reported against. |
| StructureUsage | *StructureUsageRefere nceBaseType* | StructureUsage references a flow which the data or metadata is reported against. |
| Structure | *StructureReferenceBa seType* | 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:

(ProvisionAgrement | 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** |
| --- | --- | --- |
| ProvisionAgrement | 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. |

**DataStructureRequestType:**DataStructureRequestType is a variation of a the DataStructureType for querying purposes. Only the observation dimension and the explicit measures flag are allowed.

Derivation:

*PayloadStructureType* (restriction)   
   *DataStructureType* (restriction)   
         DataStructureRequestType



Attributes:

structureID, dimensionAtObservation, explicitMeasures?, serviceURL?, structureURL?

Content:

(ProvisionAgrement | 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. |
| 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** |
| --- | --- | --- |
| ProvisionAgrement | 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. |

**GenericDataStructureRequestType:**GenericDataStructureRequestType is a variation of a the DataStructureRequestType for querying purposes. The explicit measure flag in not allowed.

Derivation:

*PayloadStructureType* (restriction)   
   *DataStructureType* (restriction)   
         DataStructureRequestType (restriction)   
               GenericDataStructureRequestType



Attributes:

structureID, dimensionAtObservation, serviceURL?, structureURL?

Content:

(ProvisionAgrement | 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. |
| 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. |
| 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** |
| --- | --- | --- |
| ProvisionAgrement | 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. |

**TimeSeriesDataStructureRequestType:**TimeSeriesDataStructureRequestType is a variation of a the DataStructureRequestType for querying purposes. The observation dimension is fixed to TIME\_PERIOD

Derivation:

*PayloadStructureType* (restriction)   
   *DataStructureType* (restriction)   
         DataStructureRequestType (restriction)   
               TimeSeriesDataStructureRequestType



Attributes:

structureID, dimensionAtObservation, explicitMeasures?, serviceURL?, structureURL?

Content:

(ProvisionAgrement | 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. |
| dimensionAtObservation (fixed: TIME\_PERIOD) | 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** |
| --- | --- | --- |
| ProvisionAgrement | 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. |

**TimeSeriesGenericDataStructureRequestType:**TimeSeriesGenericDataStructureRequestType is a variation of a the GenericDataStructureRequestType for querying purposes. The observation dimension is fixed to TIME\_PERIOD.

Derivation:

*PayloadStructureType* (restriction)   
   *DataStructureType* (restriction)   
         DataStructureRequestType (restriction)   
               GenericDataStructureRequestType (restriction)   
                     TimeSeriesGenericDataStructureRequestType



Attributes:

structureID, dimensionAtObservation, serviceURL?, structureURL?

Content:

(ProvisionAgrement | 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. |
| dimensionAtObservation (fixed: TIME\_PERIOD) | 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. |
| 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** |
| --- | --- | --- |
| ProvisionAgrement | 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. |

**GenericDataStructureType:**GenericDataStructureType defines the structural information for a generic data set. A reference to the structure, either explicitly or through a dataflow or provision agreement is required as well as the dimension which occurs at the observation level.

Derivation:

*PayloadStructureType* (restriction)   
   *DataStructureType* (restriction)   
         GenericDataStructureType



Attributes:

structureID, dimensionAtObservation, serviceURL?, structureURL?

Content:

(ProvisionAgrement | 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. |
| 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. |
| 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** |
| --- | --- | --- |
| ProvisionAgrement | 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. |

**GenericTimeSeriesDataStructureType:**GenericTimeSeriesDataStructureType defines the structural information for a generic time series based data set. The dimension at the observation level is fixed to be TIME\_PERIOD.

Derivation:

*PayloadStructureType* (restriction)   
   *DataStructureType* (restriction)   
         GenericDataStructureType (restriction)   
               GenericTimeSeriesDataStructureType



Attributes:

structureID, dimensionAtObservation, serviceURL?, structureURL?

Content:

(ProvisionAgrement | 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. |
| dimensionAtObservation (fixed: TIME\_PERIOD) | 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. |
| 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** |
| --- | --- | --- |
| ProvisionAgrement | 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:

(ProvisionAgrement | 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** |
| --- | --- | --- |
| ProvisionAgrement | 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. |

**StructureSpecificDataTimeSeriesStructureType:**StructureSpecificDataTimeSeriesStructureType defines the structural information for a structure definition specific time series data set. The dimension at the observation level is fixed to be TIME\_PERIOD.

Derivation:

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



Attributes:

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

Content:

(ProvisionAgrement | 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 (fixed: TIME\_PERIOD) | 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. |
| 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** |
| --- | --- | --- |
| ProvisionAgrement | 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:

(ProvisionAgrement | 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** |
| --- | --- | --- |
| ProvisionAgrement | 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. |

**StructureSpecificMetadataStructureType:**StructureSpecificMetadataStructureType defines the structural information for a structured metadata message.

Derivation:

*PayloadStructureType* (restriction)   
   *MetadataStructureType* (restriction)   
         StructureSpecificMetadataStructureType



Attributes:

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

Content:

(ProvisionAgrement | 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** |
| --- | --- | --- |
| ProvisionAgrement | 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:

(ProvisionAgrement | 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** |
| --- | --- | --- |
| ProvisionAgrement | 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. |

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

Content:

{Empty}

***ReferenceType*:**ReferenceType is an abstract base type. It is used as the basis for all references, to all for a top level generic object reference that can be substituted with an explicit reference to any object. Any reference can consist of a Ref (which contains all required reference fields separately) and/or a URN. These must result in the identification of the same object. Note that the Ref and URN elements are local and unqualified in order to allow for refinement of this structure outside of the namespace. This allows any reference to further refined by a different namespace. For example, a metadata structure definition specific metadata set might wish to restrict the URN to only allow for a value from an enumerated list. The general URN structure, for the purpose of mapping the reference fields is as follows: urn:sdmx:org.package-name.class-name=agency-id:(maintainable-parent-object-id[maintainable-parent-object-version].)?(container-object-id.)?object-id([object-version])?.

Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *RefBaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. |

***RefBaseType*:**RefBaseType is an abstract base type the defines the basis for any set of complete reference fields. This should be refined by derived types so that only the necessary fields are available and required as necessary. This can be used for both full and local references (when some of the values are implied from another context). A local reference is indicated with the local attribute. The values in this type correspond directly to the components of the URN structure, and thus can be used to compose a URN when the local attribute value is false. As this is the case, any reference components which are not part of the URN structure should not be present in the derived types.

Attributes:

agencyID?, maintainableParentID?, maintainableParentVersion?, containerID?, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| containerID | NestedIDType | The containerID attribute identifies the object within a maintainable object in which the referenced object is defined (container-object-id in the URN structure). This is only used in references where the referenced object is not contained directly within a maintainable object (e.g. a Component within a ComponentList, within a maintainable Structure). If the container has a fixed identifier, this attribute will not be present. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ObjectReferenceType:**ObjectReferenceType is a generic reference which allows for any object to be referenced. The type of object actually referenced can be determined from the URN or from the class attribute of the full set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   ObjectReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ObjectRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ObjectRefType:**ObjectRefType contains a set of reference fields for the purpose of referencing any object. This cannot be a local reference, therefore the agency identifier is required. It is also required that the class and package be supplied for the referenced object such that a complete URN reference can be built from the values provided. Note that this is not capable of fully validating that all necessary fields are supplied for a given object type.

Derivation:

*RefBaseType* (restriction)   
   ObjectRefType



Attributes:

agencyID, maintainableParentID?, maintainableParentVersion?, containerID?, id, version?, local?, class, package

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| containerID | NestedIDType | The containerID attribute identifies the object within a maintainable object in which the referenced object is defined (container-object-id in the URN structure). This is only used in references where the referenced object is not contained directly within a maintainable object (e.g. a Component within a ComponentList, within a maintainable Structure). If the container has a fixed identifier, this attribute will not be present. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***MaintainableRefBaseType*:**MaintainableRefBaseType is an abstract base type for referencing a maintainable object.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType*



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | MaintainableTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**MaintainableRefType:**MaintainableRefType contains a complete set of reference fields for referencing any maintainable object.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         MaintainableRefType



Attributes:

agencyID, id, version?, local?, class, package

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ConcreteMaintainable TypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***ItemSchemeRefBaseType*:**ItemSchemeRefBaseType is an abstract base type for referencing an item scheme.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType*



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ItemSchemeTypeCodeli stType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ItemSchemeRefType:**ItemSchemeRefType contains a complete set of reference fields for referencing any item scheme. The class and package a required so that the reference is explicit as to the exact object being referenced.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               ItemSchemeRefType



Attributes:

agencyID, id, version?, local?, class, package

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ItemSchemeTypeCodeli stType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***StructureOrUsageRefBaseType*:**StructureOrUsageRefBaseType is an abstract base type for referencing a structure or structure usage.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *StructureOrUsageRefBaseType*



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | StructureOrUsageType CodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***StructureRefBaseType*:**StructureRefBaseType is an abstract base type for referencing a structure.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *StructureOrUsageRefBaseType* (restriction)   
               *StructureRefBaseType*



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | StructureTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**StructureRefType:**StructureRefType contains a set of reference fields for referencing any structure.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *StructureOrUsageRefBaseType* (restriction)   
               *StructureRefBaseType* (restriction)   
                     StructureRefType



Attributes:

agencyID, id, version?, local?, class, package

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | StructureTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***StructureUsageRefBaseType*:**StructureUsageRefBaseType is an abstract base type for referencing a structure usage.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *StructureOrUsageRefBaseType* (restriction)   
               *StructureUsageRefBaseType*



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | StructureUsageTypeCo delistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**StructureUsageRefType:**StructureUsageRefType contains a set of reference fields for referencing any structure usage.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *StructureOrUsageRefBaseType* (restriction)   
               *StructureUsageRefBaseType* (restriction)   
                     StructureUsageRefType



Attributes:

agencyID, id, version?, local?, class, package

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | StructureUsageTypeCo delistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***ChildObjectRefBaseType*:**ChildObjectRefBaseType is an abstract base type for referencing any child object defined directly within a maintainable object.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType*



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***ContainerChildObjectRefBaseType*:**ContainerChildObjectRefBaseType is an abstract base type for referencing any child object within container defined in a maintainable object.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType*



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, containerID?, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| containerID | NestedIDType | The containerID attribute identifies the object within a maintainable object in which the referenced object is defined (container-object-id in the URN structure). This is only used in references where the referenced object is not contained directly within a maintainable object (e.g. a Component within a ComponentList, within a maintainable Structure). If the container has a fixed identifier, this attribute will not be present. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***ItemRefBaseType*:**ItemRefBaseType is an abstract base type for referencing an item within an item scheme.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType*



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the item scheme in which the item being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the item scheme in which the item being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ItemTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***ComponentListRefBaseType*:**ComponentListRefBaseType is an abstract base type for referencing a component list within a structure.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ComponentListRefBaseType*



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component list being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component list being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***ComponentRefBaseType*:**ComponentRefBaseType is an abstract base type for referencing a component contained in a component list within a structure.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType*



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, containerID?, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| containerID | NestedIDType | The containerID attribute references the component list of that contains the component being referenced. It is optional for the cases where the component list has a fixed identifier. Specific implementations of this will prohibit or require this accordingly. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**AnyCodelistRefType:**AnyCodelistRefType is a type for referencing any codelist object (either a codelist or a hierarchical codelist).

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         AnyCodelistRefType



Attributes:

agencyID, id, version?, local?, class, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | CodelistTypeCodelist Type | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: codelist) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**StructureOrUsageRefType:**StructureOrUsageRefType is a type for referencing a structure or structure usage.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *StructureOrUsageRefBaseType* (restriction)   
               StructureOrUsageRefType



Attributes:

agencyID, id, version?, local?, class, package

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | StructureOrUsageType CodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***LocalIdentifiableRefBaseType*:**LocalIdentifiableRefBaseType is an abstract base type which provides a local reference to any identifiable object.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType*



Attributes:

containerID?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| containerID | NestedIDType | The containerID attribute identifies the object within a maintainable object in which the referenced object is defined (container-object-id in the URN structure). This is only used in references where the referenced object is not contained directly within a maintainable object (e.g. a Component within a ComponentList, within a maintainable Structure). If the container has a fixed identifier, this attribute will not be present. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***LocalItemRefBaseType*:**LocalItemRefBaseType is an abstract base type which provides a local reference to a item object.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType*



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ItemTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***LocalComponentListRefBaseType*:**LocalComponentListRefBaseType is an abstract base type which provides a local reference to a component list object.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListRefBaseType*



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***LocalComponentListComponentRefBaseType*:**LocalComponentRefBaseType is an abstract base type which provides a local reference to a component object.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListComponentRefBaseType*



Attributes:

containerID?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| containerID | IDType | The containerID attribute identifies the object within a maintainable object in which the referenced object is defined (container-object-id in the URN structure). This is only used in references where the referenced object is not contained directly within a maintainable object (e.g. a Component within a ComponentList, within a maintainable Structure). If the container has a fixed identifier, this attribute will not be present. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalComponentListComponentRefType:**LocalComponentListComponentRefType provides a local reference to any component object within a specific component list. References for both of these are required as well as an indication of which type of type of component is being referenced via the class attribute.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListComponentRefBaseType* (restriction)   
               LocalComponentListComponentRefType



Attributes:

containerID, id, local?, class, package

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| containerID | IDType | The containerID attribute identifies the object within a maintainable object in which the referenced object is defined (container-object-id in the URN structure). This is only used in references where the referenced object is not contained directly within a maintainable object (e.g. a Component within a ComponentList, within a maintainable Structure). If the container has a fixed identifier, this attribute will not be present. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***LocalComponentRefBaseType*:**LocalComponentRefBaseType is an abstract base type which provides a local reference to a component object.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListComponentRefBaseType* (restriction)   
               *LocalComponentRefBaseType*



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalComponentRefType:**LocalComponentRefType provides a local reference to any type component object.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListComponentRefBaseType* (restriction)   
               *LocalComponentRefBaseType* (restriction)   
                     LocalComponentRefType



Attributes:

id, local?, class, package

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**AnyLocalCodeRefType:**AnyLocalCodeRefType provides a local reference to any code object.

Derivation:

*RefBaseType* (restriction)   
   AnyLocalCodeRefType



Attributes:

containerID?, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| containerID | IDType | The containerID attribute references the hierarchy which defines the hierarchical code in the case that this reference is for a hierarchical code. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | CodeTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: codelist) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**URNReferenceType:**URNReferenceType is a type referencing any object via its URN. The exact type of object is not specified, although it can be determined from the URN value.

Derivation:

*ReferenceType* (restriction)   
   URNReferenceType



Content:

URN

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***MaintainableReferenceBaseType*:**MaintainableReferenceBaseType is an abstract base type for referencing a maintainable object. It consists of a URN and/or a complete set of reference fields; agency, id, and version.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *MaintainableRefBaseT ype* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**MaintainableReferenceType:**MaintainableReferenceType is a type for referencing any maintainable object. It consists of a URN and/or a complete set of reference fields; agency, id, and version.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         MaintainableReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | MaintainableRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***LocalIdentifiableReferenceType*:**LocalIdentifiableReferenceType is an abstract base type for referencing an identifiable object locally, where the maintainable object in which it is defined is referenced in another context..

Derivation:

*ReferenceType* (restriction)   
   *LocalIdentifiableReferenceType*



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *LocalIdentifiableRef BaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

***StructureReferenceBaseType*:**StructureReferneceBaseType is a specific type of MaintainableReference that is used for referencing structure definitions. It consists of a URN and/or a complete set of reference fields; agency, id, and version.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *StructureReferenceBaseType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *StructureRefBaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**StructureReferenceType:**StructureReferenceType is a specific type of MaintainableReference that is used for referencing any structure. It consists of a URN and/or a complete set of reference fields; agency, id, and version.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *StructureReferenceBaseType* (restriction)   
               StructureReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | StructureRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***StructureUsageReferenceBaseType*:**StructureUsageReferenceBaseType is a specific type of MaintainableReference that is used for referencing structure usages. It consists of a URN and/or a complete set of reference fields; agency, id, and version.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *StructureUsageReferenceBaseType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *StructureUsageRefBas eType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**StructureUsageReferenceType:**StructureUsageReferenceType is a specific type of MaintainableReference that is used for referencing any structure usages. It consists of a URN and/or a complete set of reference fields; agency, id, and version.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *StructureUsageReferenceBaseType* (restriction)   
               StructureUsageReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | StructureUsageRefTyp e | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***ItemSchemeReferenceBaseType*:**ItemSchemeReferenceBaseType is a specific type of MaintainableReference that is used for referencing item schemes. It consists of a URN and/or a complete set of reference fields; agency, id, and version.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *ItemSchemeRefBaseTyp e* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ItemSchemeReferenceType:**ItemSchemeReferenceType is a reference that is used for referencing any type of item scheme. It consists of a URN and/or a complete set of reference fields; agency, id, and version.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               ItemSchemeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ItemSchemeRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***ChildObjectReferenceType*:**ChildObjectReferenceType is an abstract base type used for referencing a particular object defined directly within a maintainable object. It consists of a URN and/or a complete set of reference fields; agency, maintainable id (maintainableParentID), maintainable version (maintainableParentVersion), the object id (which can be nested), and optionally the object version (if applicable).

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *ChildObjectRefBaseTy pe* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***ContainerChildObjectReferenceType*:**ContainerChildObjectReferenceType is an abstract base type used for referencing a particular object defined in a container object within a maintainable object. It consists of a URN and/or a complete set of reference fields; agency, maintainable id (maintainableParentID), maintainable version (maintainableParentVersion), container id (which is optional in order to allow for containers with fixed values to be omitted), container version (if applicable), the object id (which can be nested), and optionally the object version (if applicable).

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *ContainerChildObject RefBaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***ItemReferenceType*:**ItemReferenceType is an abstract base type used for referencing a particular item within an item scheme. Note that this reference also has the ability to reference items contained within other items inside of the item scheme. It consists of a URN and/or a complete set of reference fields; agency, scheme id (maintainableParentID), scheme version (maintainableParentVersion), and item id (which can be nested).

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *ItemRefBaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***ComponentListReferenceType*:**ComponentListReferenceType is an abstract base type used for referencing component lists within a structure. It consists of a URN and/or a complete set of reference fields (structure agency, structure id, structure version, and component list id).

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ComponentListReferenceType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *ComponentListRefBase Type* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***ComponentReferenceType*:**ComponentReferenceType is an abstract base type used for referencing components within a structure definition. It consists of a URN and/or a complete set of reference fields (structure agency, structure id, structure version, component list id, and component id).

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *ComponentRefBaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***LocalItemReferenceType*:**LocalItemReferenceType is an abstract base type which provides a simple reference to an item where the reference to the item scheme which defines it are provided in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType*



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *LocalItemRefBaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

***LocalComponentListReferenceType*:**LocalComponentListReferenceType is an abstract base type which provides a simple reference to a component list where the reference to the structure which defines it is provided in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListReferenceType*



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *LocalComponentListRe fBaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

***LocalComponentListComponentReferenceBaseType*:**LocalComponentListComponentReferenceBaseType is an abstract base type which provides a simple reference to any type of component in a specific component list where the reference to the structure which defines it are provided in another context, and the component list may or may not be defined in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListComponentReferenceBaseType*



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *LocalComponentListCo mponentRefBaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalComponentListComponentReferenceType:**LocalComponentListComponentReferenceType provides a simple reference to any type of component in a specific component list where the reference to the structure which defines it are provided in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListComponentReferenceBaseType*(restriction)   
         LocalComponentListComponentReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalComponentListCo mponentRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

***LocalComponentReferenceBaseType*:**LocalComponentReferenceBaseType is an abstract base type which provides a simple reference to a component where the references to the component list which contains it and the structure which defines it are provided in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListComponentReferenceBaseType*(restriction)   
         *LocalComponentReferenceBaseType*



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *LocalComponentRefBas eType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalComponentReferenceType:**LocalComponentReferenceType provides a simple reference to any type of component in a component list where the references to the component list and the structure which defines them are provided in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListComponentReferenceBaseType*(restriction)   
         LocalComponentReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalComponentRefTyp e | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**StructureOrUsageReferenceType:**StructureOrUsageReferenceType is a specific type of a reference for referencing either a structure or a structure usage. It consists of a URN and/or a complete set of reference fields; agency, id and version. If the complete set of reference fields is used, it is required that a class and package be provided so that the type of object referenced is clear.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         StructureOrUsageReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | StructureOrUsageRefT ype | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**CategorisationReferenceType:**CategorisationReferenceType is a type for referencing a categorisation object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         CategorisationReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | CategorisationRefTyp e | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**CategorisationRefType:**CategorisationRefType provides a reference to a categorisation via a complete set of reference fields.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         CategorisationRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Categorisation) | MaintainableTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: categoryscheme) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**CategorySchemeReferenceType:**CategorySchemeReferenceType is a type for referencing a category scheme object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               CategorySchemeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | CategorySchemeRefTyp e | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**CategorySchemeRefType:**CategorySchemeRefType provides a reference to a category scheme via a complete set of reference fields.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               CategorySchemeRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: CategoryScheme) | ItemSchemeTypeCodeli stType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: categoryscheme) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**CategoryReferenceType:**CategoryReferenceType is a type for referencing a category object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType* (restriction)   
               CategoryReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | CategoryRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**CategoryRefType:**CategoryRefType references a category from within a category scheme. Reference fields are required for both the scheme and the item.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType* (restriction)   
               CategoryRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the category scheme in which the category being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the category scheme in which the category being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Category) | ItemTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: categoryscheme) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalCategoryReferenceType:**LocalCategoryReferenceType provides a simple references to a category where the identification of the category scheme which defines it is contained in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType* (restriction)   
         LocalCategoryReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalCategoryRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalCategoryRefType:**LocalCategoryRefType references a category locally where the references to the category scheme which defines it is provided elsewhere.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType* (restriction)   
               LocalCategoryRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Category) | ItemTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: categoryscheme) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**CodelistReferenceType:**CodelistReferenceType is a type for referencing a codelist object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               CodelistReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | CodelistRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**CodelistRefType:**CodelistRefType provides a reference to a codelist via a complete set of reference fields.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               CodelistRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Codelist) | ItemSchemeTypeCodeli stType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: codelist) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**CodeReferenceType:**CodeReferenceType is a type for referencing a code object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType* (restriction)   
               CodeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | CodeRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**CodeRefType:**CodeRefType references a code from within a codelist. Reference fields are required for both the scheme and the item.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType* (restriction)   
               CodeRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the codelist in which the code being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the codelist in which the code being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Code) | ItemTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: codelist) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalCodeReferenceType:**LocalCodeReferenceType provides a simple references to a code where the identification of the codelist which defines it is contained in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType* (restriction)   
         LocalCodeReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalCodeRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalCodeRefType:**LocalCodeRefType references a code locally where the references to the codelist which defines it is provided elsewhere.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType* (restriction)   
               LocalCodeRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Code) | ItemTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: codelist) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**AnyCodelistReferenceType:**AnyCodelistReferenceType is a specific type of a reference for referencing either a codelist or a hierarchical codelist usage. It consists of a URN and/or a complete set of reference fields; agency, id and version. If the complete set of reference fields is used, it is required that a class be provided so that the type of object referenced is clear.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         AnyCodelistReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | AnyCodelistRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**AnyLocalCodeReferenceType:**AnyLocalCodeReferenceType provides a simple references to any code or hierarchical code where the identification of the codelist or hierarchical codelist which defines it is contained in another context.

Derivation:

*ReferenceType* (restriction)   
   AnyLocalCodeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | AnyLocalCodeRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ConceptSchemeReferenceType:**ConceptSchemeReferenceType is a type for referencing a concept scheme object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               ConceptSchemeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ConceptSchemeRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ConceptSchemeRefType:**ConceptSchemeRefType provides a reference to a concept scheme via a complete set of reference fields.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               ConceptSchemeRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ConceptScheme) | ItemSchemeTypeCodeli stType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: conceptscheme) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ConceptReferenceType:**ConceptReferenceType is a type for referencing a concept object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType* (restriction)   
               ConceptReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ConceptRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ConceptRefType:**ConceptRefType references a concept from within a concept scheme. Reference fields are required for both the scheme and the item.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType* (restriction)   
               ConceptRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the concept scheme in which the concept being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the concept scheme in which the concept being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Concept) | ItemTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: conceptscheme) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalConceptReferenceType:**LocalConceptReferenceType provides a simple references to a concept where the identification of the concept scheme which defines it is contained in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType* (restriction)   
         LocalConceptReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalConceptRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalConceptRefType:**LocalConceptRefType references a concept locally where the references to the concept scheme which defines it is provided elsewhere.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType* (restriction)   
               LocalConceptRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Concept) | ItemTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: conceptscheme) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***OrganisationSchemeReferenceBaseType*:**OrganisationSchemeReferenceBaseType is a type for referencing a organisation scheme object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               *OrganisationSchemeReferenceBaseType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *OrganisationSchemeRe fBaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***OrganisationSchemeRefBaseType*:**OrganisationSchemeRefBaseType contains a set of reference fields for an organisation scheme.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               *OrganisationSchemeRefBaseType*



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | OrganisationSchemeTy peCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**OrganisationSchemeReferenceType:**OrganisationSchemeReferenceType references an organisation scheme regardless of the specific type. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               *OrganisationSchemeReferenceBaseType* (restriction)   
                     OrganisationSchemeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | OrganisationSchemeRe fType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**OrganisationSchemeRefType:**OrganisationSchemeRefType provides a reference to an organisation scheme via a complete set of reference fields. It is required that the class (i.e. the type) of organisation scheme being referenced be specified.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               *OrganisationSchemeRefBaseType* (restriction)   
                     OrganisationSchemeRefType



Attributes:

agencyID, id, version?, local?, class, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | OrganisationSchemeTy peCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***OrganisationReferenceBaseType*:**OrganisationReferenceBaseType is a type for referencing any organisation object, regardless of its type. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType* (restriction)   
               *OrganisationReferenceBaseType*



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *OrganisationRefBaseT ype* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***OrganisationRefBaseType*:**OrganisationRefBaseType is an abstract base type which references an organisation from within a organisation scheme. Reference fields are required for both the scheme and the organisation.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType* (restriction)   
               *OrganisationRefBaseType*



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the organisation scheme in which the organisation being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the organisation scheme in which the organisation being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**OrganisationReferenceType:**OrganisationReferenceType references an organisation regardless of the specific type. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType* (restriction)   
               *OrganisationReferenceBaseType* (restriction)   
                     OrganisationReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *OrganisationRefBaseT ype* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**OrganisationRefType:**OrganisationRefType provides a reference to any organisation via a complete set of reference fields. It is required that the class (i.e. the type) of organisation being referenced be specified.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType* (restriction)   
               *OrganisationRefBaseType* (restriction)   
                     OrganisationRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the organisation scheme in which the organisation being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the organisation scheme in which the organisation being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

***LocalOrganisationReferenceBaseType*:**LocalOrganisationReferenceBaseType is an abstract base type which provides a simple references to an organisation, regardless of type, where the identification of the organisation scheme which defines it is contained in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType* (restriction)   
         *LocalOrganisationReferenceBaseType*



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *LocalOrganisationRef BaseType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

***LocalOrganisationRefBaseType*:**LocalOrganisationRefBaseType is an abstract base type that references an organisation locally where the reference to the organisation scheme which defines it is provided elsewhere.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType* (restriction)   
               *LocalOrganisationRefBaseType*



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalOrganisationReferenceType:**LocalOrganisationReferenceType provides a simple reference to an organisation, regardless of type, where the identification of the organisation scheme which defines it is contained in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType* (restriction)   
         *LocalOrganisationReferenceBaseType* (restriction)   
               LocalOrganisationReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalOrganisationRef Type | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalOrganisationRefType:**LocalOrganisationRefType references an organisation locally where the reference to the organisation scheme which defines it is provided elsewhere. The reference requires that the class (i.e. the type) or the organisation being reference be provided.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType* (restriction)   
               *LocalOrganisationRefBaseType* (restriction)   
                     LocalOrganisationRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**OrganisationUnitSchemeReferenceType:**OrganisationUnitSchemeReferenceType is a type for referencing an organisation unit scheme object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               *OrganisationSchemeReferenceBaseType* (restriction)   
                     OrganisationUnitSchemeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | OrganisationUnitSche meRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**OrganisationUnitSchemeRefType:**OrganisationUnitSchemeRefType contains a set of reference fields for an organisation unit scheme.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               *OrganisationSchemeRefBaseType* (restriction)   
                     OrganisationUnitSchemeRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: OrganisationUnitScheme) | OrganisationSchemeTy peCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**OrganisationUnitReferenceType:**OrganisationUnitReferenceType is a type for referencing an organisation unit. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType* (restriction)   
               *OrganisationReferenceBaseType* (restriction)   
                     OrganisationUnitReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | OrganisationUnitRefT ype | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**OrganisationUnitRefType:**OrganisationUnitRefType contains a set of reference fields for referencing an organisation unit within an organisation unit scheme.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType* (restriction)   
               *OrganisationRefBaseType* (restriction)   
                     OrganisationUnitRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the organisation scheme in which the organisation being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the organisation scheme in which the organisation being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: OrganisationUnit) | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalOrganisationUnitReferenceType:**LocalOrganisationUnitReferenceType provides a simple reference to an organisation unit, where the reference to the organisation unit scheme which defines it is provided in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType* (restriction)   
         *LocalOrganisationReferenceBaseType* (restriction)   
               LocalOrganisationUnitReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalOrganisationUni tRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalOrganisationUnitRefType:**LocalOrganisationUnitRefType references an organisation unit locally where the reference to the organisation unit scheme which defines it is provided elsewhere.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType* (restriction)   
               *LocalOrganisationRefBaseType* (restriction)   
                     LocalOrganisationUnitRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: OrganisationUnit) | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**AgencySchemeReferenceType:**AgencySchemeReferenceType is a type for referencing an agency scheme object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               *OrganisationSchemeReferenceBaseType* (restriction)   
                     AgencySchemeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | AgencySchemeRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**AgencySchemeRefType:**AgencySchemeRefType contains a set of reference fields for an agency scheme.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               *OrganisationSchemeRefBaseType* (restriction)   
                     AgencySchemeRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: AgencyScheme) | OrganisationSchemeTy peCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**AgencyReferenceType:**AgencyReferenceType is a type for referencing an agency. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType* (restriction)   
               *OrganisationReferenceBaseType* (restriction)   
                     AgencyReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | AgencyRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**AgencyRefType:**AgencyRefType contains a set of reference fields for referencing an agency within an agency scheme.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType* (restriction)   
               *OrganisationRefBaseType* (restriction)   
                     AgencyRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the organisation scheme in which the organisation being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the organisation scheme in which the organisation being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Agency) | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalAgencyReferenceType:**LocalAgencyReferenceType provides a simple reference to an agency, where the reference to the agency scheme which defines it is provided in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType* (restriction)   
         *LocalOrganisationReferenceBaseType* (restriction)   
               LocalAgencyReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalAgencyRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalAgencyRefType:**LocalAgencyRefType references an agency locally where the reference to the agency scheme which defines it is provided elsewhere.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType* (restriction)   
               *LocalOrganisationRefBaseType* (restriction)   
                     LocalAgencyRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Agency) | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**DataConsumerSchemeReferenceType:**DataConsumerSchemeReferenceType is a type for referencing a data consumer scheme object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               *OrganisationSchemeReferenceBaseType* (restriction)   
                     DataConsumerSchemeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | DataConsumerSchemeRe fType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**DataConsumerSchemeRefType:**DataConsumerSchemeRefType contains a set of reference fields for a data consumer scheme.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               *OrganisationSchemeRefBaseType* (restriction)   
                     DataConsumerSchemeRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: DataConsumerScheme) | OrganisationSchemeTy peCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**DataConsumerReferenceType:**DataConsumerReferenceType is a type for referencing a data consumer. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType* (restriction)   
               *OrganisationReferenceBaseType* (restriction)   
                     DataConsumerReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | DataConsumerRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**DataConsumerRefType:**DataConsumerRefType contains a set of reference fields for referencing a data consumer within a data consumer scheme.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType* (restriction)   
               *OrganisationRefBaseType* (restriction)   
                     DataConsumerRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the organisation scheme in which the organisation being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the organisation scheme in which the organisation being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: DataConsumer) | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalDataConsumerReferenceType:**LocalDataConsumerReferenceType provides a simple reference to a data consumer, where the reference to the data consumer scheme which defines it is provided in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType* (restriction)   
         *LocalOrganisationReferenceBaseType* (restriction)   
               LocalDataConsumerReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalDataConsumerRef Type | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalDataConsumerRefType:**LocalDataConsumerRefType references a data consumer locally where the reference to the data consumer scheme which defines it is provided elsewhere.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType* (restriction)   
               *LocalOrganisationRefBaseType* (restriction)   
                     LocalDataConsumerRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: DataConsumer) | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**DataProviderSchemeReferenceType:**DataProviderSchemeReferenceType is a type for referencing a data provider scheme object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               *OrganisationSchemeReferenceBaseType* (restriction)   
                     DataProviderSchemeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | DataProviderSchemeRe fType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**DataProviderSchemeRefType:**DataProviderSchemeRefType contains a set of reference fields for a data provider scheme.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               *OrganisationSchemeRefBaseType* (restriction)   
                     DataProviderSchemeRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: DataProviderScheme) | OrganisationSchemeTy peCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**DataProviderReferenceType:**DataProviderReferenceType is a type for referencing a data provider. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType* (restriction)   
               *OrganisationReferenceBaseType* (restriction)   
                     DataProviderReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | DataProviderRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**DataProviderRefType:**DataProviderRefType contains a set of reference fields for referencing a data provider within a data provider scheme.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType* (restriction)   
               *OrganisationRefBaseType* (restriction)   
                     DataProviderRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the organisation scheme in which the organisation being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the organisation scheme in which the organisation being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: DataProvider) | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalDataProviderReferenceType:**LocalDataProviderReferenceType provides a simple reference to a data provider, where the reference to the data provider scheme which defines it is provided in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType* (restriction)   
         *LocalOrganisationReferenceBaseType* (restriction)   
               LocalDataProviderReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalDataProviderRef Type | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalDataProviderRefType:**LocalDataProviderRefType references a data provider locally where the reference to the data provider scheme which defines it is provided elsewhere.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType* (restriction)   
               *LocalOrganisationRefBaseType* (restriction)   
                     LocalDataProviderRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: DataProvider) | OrganisationTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: base) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ReportingTaxonomyReferenceType:**ReportingTaxonomyReferenceType is a type for referencing a reporting taxonomy object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               ReportingTaxonomyReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ReportingTaxonomyRef Type | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ReportingTaxonomyRefType:**ReportingTaxonomyRefType provides a reference to a reporting taxonomy via a complete set of reference fields.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               ReportingTaxonomyRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ReportingTaxonomy) | ItemSchemeTypeCodeli stType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: categoryscheme) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ReportingCategoryReferenceType:**ReportingCategoryReferenceType is a type for referencing a reporting category object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ItemReferenceType* (restriction)   
               ReportingCategoryReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ReportCategoryRefTyp e | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ReportCategoryRefType:**ReportCategoryRefType contains a set of fields for referencing a reporting category within a reporting taxonomy.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ItemRefBaseType* (restriction)   
               ReportCategoryRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the reporting taxonomy in which the reporting category being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the reporting taxonomy in which the reporting category being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ReportingCategory) | ItemTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: categoryscheme) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalReportingCategoryReferenceType:**LocalReportingCategoryReferenceType provides a simple references to a reporting category where the identification of the reporting taxonomy which defines it is contained in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalItemReferenceType* (restriction)   
         LocalReportingCategoryReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalReportingCatego ryRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalReportingCategoryRefType:**LocalReportingCategoryRefType references a reporting category locally where the references to the reporting taxonomy which defines it is provided elsewhere.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalItemRefBaseType* (restriction)   
               LocalReportingCategoryRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ReportingCategory) | ItemTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: categoryscheme) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**HierarchicalCodelistReferenceType:**HierarchicalCodelistReferenceType is a type for referencing a hierarchical codelist object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         HierarchicalCodelistReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | HierarchicalCodelist RefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**HierarchicalCodelistRefType:**HierarchicalCodelistRefType contains a set of reference fields for a hierarchical codelist.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         HierarchicalCodelistRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: HierarchicalCodelist) | MaintainableTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: codelist) | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**HierarchyReferenceType:**HierarchyReferenceType is a type for referencing a hierarchy within a hierarchical codelist.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         HierarchyReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | HierarchyRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**HierarchyRefType:**HierarchyRefType is type which references a hierarchy from within a hierarchical codelist. Reference fields are required for both the hierarchy and the codelist.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         HierarchyRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Hierarchy) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: codelist) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LevelReferenceType:**LevelReferenceType is a type for referencing a level object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         LevelReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LevelRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**LevelRefType:**LevelRefType references a level from within a hierarchical codelist. Reference fields are required for both the level and the codelist.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         LevelRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Level) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: codelist) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalLevelReferenceType:**LocalLevelReferenceType is a type for referencing a level object where the reference to the hierarchical codelist and the hierarchy in which it is defined is provided in another context (e.g. is inferred from the hierarchy in which the reference is defined).

Derivation:

*ReferenceType* (restriction)   
   LocalLevelReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalLevelRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalLevelRefType:**LocalLevelRefType references a level object where the reference to the hierarchy in which it is contained and the hierarchical codelist which define it are provided in another context.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         LocalLevelRefType



Attributes:

containerID?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| containerID | NestedIDType | The containerID attribute identifies the object within a maintainable object in which the referenced object is defined (container-object-id in the URN structure). This is only used in references where the referenced object is not contained directly within a maintainable object (e.g. a Component within a ComponentList, within a maintainable Structure). If the container has a fixed identifier, this attribute will not be present. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Level) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**HierarchicalCodeReferenceType:**HierarchicalCodeReferenceType is a type for referencing a hierarchical code object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         HierarchicalCodeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | HierarchicalCodeRefT ype | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**HierarchicalCodeRefType:**HierarchicalCodeRefType references a code from within a hierarchical codelist. Reference fields are required for both the code and the codelist.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         HierarchicalCodeRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, containerID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the hierarchical codelist in which the code being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the hierarchical codelist in which the code being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| containerID | IDType | The containerID references the hierarchy which contains the code being referenced is defined. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: HierarchicalCode) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: codelist) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ConstraintReferenceType:**ConstraintReferenceType is a type for referencing a constraint object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         ConstraintReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | *ConstraintRefType* | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

***ConstraintRefType*:**ConstraintRefType contains a set of reference fields for a constraint.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ConstraintRefType*



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | ConstraintTypeCodeli stType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: registry) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**AttachmentConstraintReferenceType:**AttachmentConstraintReferenceType is a type for referencing a attachment constraint object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         ConstraintReferenceType (restriction)   
               AttachmentConstraintReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | AttachmentConstraint RefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**AttachmentConstraintRefType:**AttachmentConstraintRefType contains a set of reference fields for an attachment constraint.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ConstraintRefType* (restriction)   
               AttachmentConstraintRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: AttachmentConstraint) | ConstraintTypeCodeli stType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: registry) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ContentConstraintReferenceType:**ContentConstraintReferenceType is a type for referencing a content constraint object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         ConstraintReferenceType (restriction)   
               ContentConstraintReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ContentConstraintRef Type | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ContentConstraintRefType:**ContentConstraintRefType contains a set of reference fields for a content constraint.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ConstraintRefType* (restriction)   
               ContentConstraintRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ContentConstraint) | ConstraintTypeCodeli stType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: registry) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**DataflowReferenceType:**DataflowReferenceType is a type for referencing a dataflow object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *StructureUsageReferenceBaseType* (restriction)   
               DataflowReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | DataflowRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**DataflowRefType:**DataflowRefType contains a set of reference fields for a data flow.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *StructureOrUsageRefBaseType* (restriction)   
               *StructureUsageRefBaseType* (restriction)   
                     DataflowRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Dataflow) | StructureUsageTypeCo delistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**MetadataflowReferenceType:**MetadataflowReferenceType is a type for referencing a metadata flow object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *StructureUsageReferenceBaseType* (restriction)   
               MetadataflowReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | MetadataflowRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**MetadataflowRefType:**MetadataflowRefType contains a set of reference fields for a metadata flow.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *StructureOrUsageRefBaseType* (restriction)   
               *StructureUsageRefBaseType* (restriction)   
                     MetadataflowRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Metadataflow) | StructureUsageTypeCo delistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**DataStructureReferenceType:**DataStructureReferenceType is a type for referencing a data structure definition object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *StructureReferenceBaseType* (restriction)   
               DataStructureReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | DataStructureRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**DataStructureRefType:**DataStructureRefType contains a set of reference fields for a data structure definition.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *StructureOrUsageRefBaseType* (restriction)   
               *StructureRefBaseType* (restriction)   
                     DataStructureRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: DataStructure) | StructureTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**KeyDescriptorReferenceType:**KeyDescriptorReferenceType is a type for referencing a key descriptor object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ComponentListReferenceType* (restriction)   
               KeyDescriptorReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | KeyDescriptorRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**KeyDescriptorRefType:**KeyDescriptorRefType contains a reference to the key descriptor within a data structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ComponentListRefBaseType* (restriction)   
               KeyDescriptorRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component list being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component list being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id (fixed: DIMENSION\_DESCRIPTOR) | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: DimensionDescriptor) | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**AttributeDescriptorReferenceType:**AttributeDescriptorReferenceType is a type for referencing an attribute descriptor object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ComponentListReferenceType* (restriction)   
               AttributeDescriptorReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | AttributeDescriptorR efType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**AttributeDescriptorRefType:**AttributeDescriptorRefType contains a reference to the attribute descriptor within a data structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ComponentListRefBaseType* (restriction)   
               AttributeDescriptorRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component list being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component list being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id (fixed: ATTRIBUTE\_DESCRIPTOR) | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: AttributeDescriptor) | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**MeasureDescriptorReferenceType:**MeasureDescriptorReferenceType is a type for referencing a measure descriptor object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ComponentListReferenceType* (restriction)   
               MeasureDescriptorReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | MeasureDescriptorRef Type | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**MeasureDescriptorRefType:**MeasureDescriptorRefType contains a reference to the measure descriptor within a data structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ComponentListRefBaseType* (restriction)   
               MeasureDescriptorRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component list being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component list being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id (fixed: MEASURE\_DESCRIPTOR) | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: MeasureDescriptor) | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**GroupKeyDescriptorReferenceType:**GroupKeyDescriptorReferenceType is a type for referencing a group key descriptor object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ComponentListReferenceType* (restriction)   
               GroupKeyDescriptorReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | GroupKeyDescriptorRe fType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**GroupKeyDescriptorRefType:**GroupKeyDescriptorRefType contains a reference to a group key descriptor within a data structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ComponentListRefBaseType* (restriction)   
               GroupKeyDescriptorRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component list being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component list being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: GroupDimensionDescriptor) | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalGroupKeyDescriptorReferenceType:**LocalGroupKeyDescriptorReferenceType is a type for referencing a group key descriptor locally, where the reference to the data structure definition which defines it is provided in another context (for example the data structure definition in which the reference occurs).

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListReferenceType* (restriction)   
         LocalGroupKeyDescriptorReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalGroupKeyDescrip torRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalGroupKeyDescriptorRefType:**LocalGroupKeyDescriptorRefType contains a local reference to a group key descriptor.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListRefBaseType* (restriction)   
               LocalGroupKeyDescriptorRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: GroupDimensionDescriptor) | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**DimensionReferenceType:**DimensionReferenceType is a type for referencing a dimension object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               DimensionReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | DimensionRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**DimensionRefType:**DimensionRefType contains a reference to a dimension within a data structure definition. Note that since there is only one key descriptor, the container reference fields are prohibited.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               DimensionRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Dimension) | DimensionTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**MeasureDimensionReferenceType:**MeasureDimensionReferenceType is a type for referencing a measure dimension object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               MeasureDimensionReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | MeasureDimensionRefT ype | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**MeasureDimensionRefType:**MeasureDimensionRefType contains a reference to the measure dimension within a data structure definition. Note that since there is only one key descriptor, the container reference fields are prohibited.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               MeasureDimensionRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: MeasureDimension) | DimensionTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**TimeDimensionReferenceType:**TimeDimensionReferenceType is a type for referencing a time dimension object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               TimeDimensionReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | TimeDimensionRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**TimeDimensionRefType:**TimeDimensionRefType contains a reference to the time dimension within a data structure definition. Note that since there is only one key descriptor, the container reference fields are prohibited.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               TimeDimensionRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id (fixed: TIME\_PERIOD) | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: TimeDimension) | DimensionTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalDimensionReferenceType:**LocalDimensionReferenceType is a type for referencing any type of dimension locally, where the reference to the data structure definition which defines the dimension is provided in another context (for example the data structure definition in which the reference occurs).

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListComponentReferenceBaseType*(restriction)   
         *LocalComponentReferenceBaseType* (restriction)   
               LocalDimensionReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalDimensionRefTyp e | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalDimensionRefType:**LocalDimensionRefType contains the reference fields for referencing a dimension locally.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListComponentRefBaseType* (restriction)   
               *LocalComponentRefBaseType* (restriction)   
                     LocalDimensionRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (default: Dimension) | DimensionTypeCodelis tType | The class attribute is optional and provided a default value of Dimension. It is strongly recommended that if the time or measure dimension is referenced, that the proper value be set for this field. However, this is not absolutely necessary since all data structure definition components must have a unique identifier within the scope of the entire data structure. It does, however, allow systems which will treat such a reference as a URN to easily construct the URN without having to verify the object class of the referenced dimension. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**AttributeReferenceType:**AttributeReferenceType is a type for referencing an attribute object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               AttributeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | AttributeRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**AttributeRefType:**AttributeRefType contains a reference to an attribute within a data structure definition. Note that since there is only one attribute descriptor, the container reference fields are prohibited.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               AttributeRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Attribute) | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**PrimaryMeasureReferenceType:**PrimaryMeasureReferenceType is a type for referencing a primary measure object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               PrimaryMeasureReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | PrimaryMeasureRefTyp e | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**PrimaryMeasureRefType:**PrimaryMeasureRefType contains a reference to the primary measure within a data structure definition. Note that since there is only one key descriptor, the container reference fields are prohibited.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               PrimaryMeasureRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id (fixed: OBS\_VALUE) | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: PrimaryMeasure) | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalPrimaryMeasureReferenceType:**LocalPrimaryMeasureReferenceType is a type for referencing a primary measure locally, where the reference to the data structure definition which defines the primary measure is provided in another context (for example the data structure definition in which the reference occurs).

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListComponentReferenceBaseType*(restriction)   
         *LocalComponentReferenceBaseType* (restriction)   
               LocalPrimaryMeasureReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalPrimaryMeasureR efType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalPrimaryMeasureRefType:**LocalPrimaryMeasureRefType contains the reference fields for referencing a primary measure locally.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListComponentRefBaseType* (restriction)   
               *LocalComponentRefBaseType* (restriction)   
                     LocalPrimaryMeasureRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id (fixed: OBS\_VALUE) | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: PrimaryMeasure) | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalDataStructureComponentReferenceType:**LocalDataStructureComponentReferenceType is a type for referencing any type of data structure component locally, where the reference for the data structure definition which defines the components is available in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListComponentReferenceBaseType*(restriction)   
         LocalDataStructureComponentReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalDataStructureCo mponentRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalDataStructureComponentRefType:**LocalDataStructureComponentRefType contains the reference fields for referencing any data structure component locally. This reference must specify the class of the component being referenced.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListComponentRefBaseType* (restriction)   
               LocalDataStructureComponentRefType



Attributes:

id, local?, class, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | DataStructureCompone ntTypeCodelistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: datastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**DataStructureEnumerationSchemeReferenceType:**DataStructureEnumerationSchemeReferenceType is a type for referencing any type of item scheme that is allowable as the enumeration of the representation of a data structure definition component. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *ItemSchemeReferenceBaseType* (restriction)   
               ItemSchemeReferenceType (restriction)   
                     DataStructureEnumerationSchemeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | DataStructureEnumera tionSchemeRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**DataStructureEnumerationSchemeRefType:**DataStructureEnumerationSchemeRefType contains the reference fields for referencing any item scheme that is allowable as the enumeration of the representation of a data structure definition component.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *ItemSchemeRefBaseType* (restriction)   
               ItemSchemeRefType (restriction)   
                     DataStructureEnumerationSchemeRefType



Attributes:

agencyID, id, version?, local?, class, package

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | DimensionEumerationS chemeTypeCodelistTyp e | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package | ItemSchemePackageTyp eCodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**MetadataStructureReferenceType:**MetadataStructureReferenceType is a type for referencing a metadata structure definition object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         *StructureReferenceBaseType* (restriction)   
               MetadataStructureReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | MetadataStructureRef Type | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**MetadataStructureRefType:**MetadataStructureRefType contains a set of reference fields for a metadata structure definition.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         *StructureOrUsageRefBaseType* (restriction)   
               *StructureRefBaseType* (restriction)   
                     MetadataStructureRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: MetadataStructure) | StructureTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**MetadataTargetReferenceType:**MetadataTargetReferenceType is a type for referencing a metadata target object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ComponentListReferenceType* (restriction)   
               MetadataTargetReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | MetadataTargetRefTyp e | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**MetadataTargetRefType:**MetadataTargetRefType contains a reference to a metadata target within a metadata structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ComponentListRefBaseType* (restriction)   
               MetadataTargetRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component list being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component list being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: MetadataTarget) | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalMetadataTargetReferenceType:**LocalMetadataTargetReferenceType is a type for referencing a metadata target locally, where the reference to the metadata structure definition which defines it is provided in another context (for example the metadata structure definition in which the reference occurs).

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListReferenceType* (restriction)   
         LocalMetadataTargetReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalMetadataTargetR efType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalMetadataTargetRefType:**LocalMetadataTargetRefType contains a local reference to a metadata target object.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListRefBaseType* (restriction)   
               LocalMetadataTargetRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: MetadataTarget) | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ConstraintTargetReferenceType:**ConstraintTargetReferenceType is a type for referencing a constraint target object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               ConstraintTargetReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ConstraintTargetRefT ype | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ConstraintTargetRefType:**ConstraintTargetRefType contains a reference to a constraint target within a metadata target of a data structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               ConstraintTargetRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, containerID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| containerID | IDType | The containerID attribute references the component list of that contains the component being referenced. It is optional for the cases where the component list has a fixed identifier. Specific implementations of this will prohibit or require this accordingly. |
| id (fixed: CONSTRAINT\_TARGET) | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ConstraintTarget) | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**DataSetTargetReferenceType:**DataSetTargetReferenceType is a type for referencing a data set target object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               DataSetTargetReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | DataSetTargetRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**DataSetTargetRefType:**DataSetTargetRefType contains a reference to a data set target within a metadata target of a data structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               DataSetTargetRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, containerID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| containerID | IDType | The containerID attribute references the component list of that contains the component being referenced. It is optional for the cases where the component list has a fixed identifier. Specific implementations of this will prohibit or require this accordingly. |
| id (fixed: DATA\_SET\_TARGET) | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: DataSetTarget) | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**KeyDescriptorValuesTargetReferenceType:**KeyDescriptorValuesTargetType is a type for referencing a key descriptor values target object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               KeyDescriptorValuesTargetReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | KeyDescriptorValuesT argetRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**KeyDescriptorValuesTargetRefType:**KeyDescriptorValuesTargetRefType contains a reference to a key descriptor values target within a metadata target of a data structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               KeyDescriptorValuesTargetRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, containerID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| containerID | IDType | The containerID attribute references the component list of that contains the component being referenced. It is optional for the cases where the component list has a fixed identifier. Specific implementations of this will prohibit or require this accordingly. |
| id (fixed: DIMENSION\_DESCRIPTOR\_VALUES\_TARGET) | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: DimensionDescriptorValuesTarget) | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ReportPeriodTargetReferenceType:**ReportPeriodTargetReferenceType is a type for referencing a report period target object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               ReportPeriodTargetReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ReportPeriodTargetRe fType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ReportPeriodTargetRefType:**ReportPeriodTargetRefType contains a reference to a report period target within a metadata target of a data structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               ReportPeriodTargetRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, containerID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| containerID | IDType | The containerID attribute references the component list of that contains the component being referenced. It is optional for the cases where the component list has a fixed identifier. Specific implementations of this will prohibit or require this accordingly. |
| id (fixed: REPORT\_PERIOD\_TARGET) | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ReportPeriodTarget) | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**IdentifiableObjectTargetReferenceType:**IdentifiableObjectTargetReferenceType is a type for referencing an identifiable object target object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               IdentifiableObjectTargetReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | IdentifiableObjectTa rgetRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**IdentifiableObjectTargetRefType:**IdentifiableObjectTargetRefType contains a reference to an identifiable object target within a metadata target of a data structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               IdentifiableObjectTargetRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, containerID?, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| containerID | NestedIDType | The containerID attribute references the component list of that contains the component being referenced. It is optional for the cases where the component list has a fixed identifier. Specific implementations of this will prohibit or require this accordingly. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: IdentifiableObjectTarget) | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalTargetObjectReferenceType:**LocalTargetObjectReferenceType is a type for referencing any type of target object within a metadata target locally, where the references to the metadata target and the metadata structure definition which defines the target reference are provided in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListComponentReferenceBaseType*(restriction)   
         *LocalComponentReferenceBaseType* (restriction)   
               LocalTargetObjectReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalTargetObjectRef Type | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalTargetObjectRefType:**LocalTargetObjectRefType contains the reference fields for referencing a target object locally.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListComponentRefBaseType* (restriction)   
               *LocalComponentRefBaseType* (restriction)   
                     LocalTargetObjectRefType



Attributes:

id, local?, class, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | TargetObjectTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ReportStructureReferenceType:**ReportStructureReferenceType is a type for referencing a report structure object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         *ComponentListReferenceType* (restriction)   
               ReportStructureReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ReportStructureRefTy pe | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ReportStructureRefType:**ReportStructureRefType contains a reference to a report structure within a metadata structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         *ComponentListRefBaseType* (restriction)   
               ReportStructureRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component list being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component list being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ReportStructure) | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalReportStructureReferenceType:**LocalReportStructureReferenceType is a type for referencing a report structure locally, where the reference to the metadata structure definition which defines it is provided in another context (for example the metadata structure definition in which the reference occurs).

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListReferenceType* (restriction)   
         LocalReportStructureReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalReportStructure RefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalReportStructureRefType:**LocalReportStructureRefType contains a local reference to a report structure object.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListRefBaseType* (restriction)   
               LocalReportStructureRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ReportStructure) | ComponentListTypeCod elistType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**MetadataAttributeReferenceType:**MetadataAttributeReferenceType is a type for referencing a metadata attribute object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         *ComponentReferenceType* (restriction)   
               MetadataAttributeReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | MetadataAttributeRef Type | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**MetadataAttributeRefType:**MetadataAttributeRefType contains a reference to a metadata attribute within a report structure in a metadata structure definition.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         *ComponentRefBaseType* (restriction)   
               MetadataAttributeRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, containerID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID references the structure in which the component being referenced is defined. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute references the version of the structure in which the component being referenced is defined. If not supplied, a default value of 1.0 is assumed. |
| containerID | IDType | The containerID attribute references the component list of that contains the component being referenced. It is optional for the cases where the component list has a fixed identifier. Specific implementations of this will prohibit or require this accordingly. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: MetadataAttribute) | ComponentTypeCodelis tType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalMetadataStructureComponentReferenceType:**LocalMetadataStructureComponentReferenceType is a type for referencing any type of metadata structure component locally, where the reference for the metadata structure definition which defines the components is available in another context.

Derivation:

*ReferenceType* (restriction)   
   *LocalComponentListComponentReferenceBaseType*(restriction)   
         LocalMetadataStructureComponentReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalMetadataStructu reComponentRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalMetadataStructureComponentRefType:**LocalMetadataStructureComponentRefType contains the reference fields for referencing any metadata structure component locally. This reference must specify the class of the component being referenced.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         *LocalComponentListComponentRefBaseType* (restriction)   
               LocalMetadataStructureComponentRefType



Attributes:

containerID, id, local?, class, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| containerID | IDType | The containerID attribute identifies the object within a maintainable object in which the referenced object is defined (container-object-id in the URN structure). This is only used in references where the referenced object is not contained directly within a maintainable object (e.g. a Component within a ComponentList, within a maintainable Structure). If the container has a fixed identifier, this attribute will not be present. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class | MetadataStructureCom ponentTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: metadatastructure) | StructurePackageType CodelistType | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ProvisionAgreementReferenceType:**ProvisionAgreementReferenceType is a type for referencing a provision agreement. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         ProvisionAgreementReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ProvisionAgreementRe fType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ProvisionAgreementRefType:**ProvisionAgreementRefType contains a set of reference fields for a provision agreement.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         ProvisionAgreementRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ProvisionAgreement) | MaintainableTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: registry) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ProcessReferenceType:**ProcessReferenceType is a type for referencing a process object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         ProcessReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ProcessRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ProcessRefType:**ProcessRefType contains a set of reference fields for a process.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         ProcessRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Process) | MaintainableTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: process) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ProcessStepReferenceType:**ProcessStepReferenceType is a type for referencing a process step object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         ProcessStepReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ProcessStepRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ProcessStepRefType:**ProcessStepRefType provides for a reference to a process step through its id. Support for referencing nested process steps is provided through a nested identifier.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         ProcessStepRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ProcessStep) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: process) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalProcessStepReferenceType:**LocalProcessStepReferenceType is a type for referencing a process step locally, where the reference to the process which defines it is provided in another context (for example the metadata structure definition in which the reference occurs).

Derivation:

*ReferenceType* (restriction)   
   *LocalIdentifiableReferenceType* (restriction)   
         LocalProcessStepReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalProcessStepRefT ype | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalProcessStepRefType:**LocalProcessStepRefType contains a local reference to a process step object.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         LocalProcessStepRefType



Attributes:

containerID?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| containerID | NestedIDType | The containerID attribute identifies the object within a maintainable object in which the referenced object is defined (container-object-id in the URN structure). This is only used in references where the referenced object is not contained directly within a maintainable object (e.g. a Component within a ComponentList, within a maintainable Structure). If the container has a fixed identifier, this attribute will not be present. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ProcessStep) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: process) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**TransitionReferenceType:**TransiationReferenceType is a type for referencing a process step object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ContainerChildObjectReferenceType* (restriction)   
         TransitionReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | TransitionRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**TransitionRefType:**TransitionRefType provides for a reference to a transition definition in process step through its id.

Derivation:

*RefBaseType* (restriction)   
   *ContainerChildObjectRefBaseType* (restriction)   
         TransitionRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, containerID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| containerID | NestedIDType | The containerID attribute identifies the object within a maintainable object in which the referenced object is defined (container-object-id in the URN structure). This is only used in references where the referenced object is not contained directly within a maintainable object (e.g. a Component within a ComponentList, within a maintainable Structure). If the container has a fixed identifier, this attribute will not be present. |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: Transition) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: process) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**StructureSetReferenceType:**StructureSetReferenceType is a type for referencing a structure set object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *MaintainableReferenceBaseType* (restriction)   
         StructureSetReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | StructureSetRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**StructureSetRefType:**StructureSetRefType contains a set of reference fields for a structure set.

Derivation:

*RefBaseType* (restriction)   
   *MaintainableRefBaseType* (restriction)   
         StructureSetRefType



Attributes:

agencyID, id, version?, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | VersionType | The version attribute identifies the version of the object being reference, if applicable. If this is available, a default value of 1.0 will always apply. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: StructureSet) | MaintainableTypeCode listType | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: mapping) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**StructureMapReferenceType:**StructureMapReferenceType is a type for referencing a structure map object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         StructureMapReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | StructureMapRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**StructureMapRefType:**StructureMapRefType contains fields for referencing a structure map within a structure set.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         StructureMapRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: StructureMap) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: mapping) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**CategorySchemeMapReferenceType:**CategorySchemeMapReferenceType is a type for referencing a category scheme map object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         CategorySchemeMapReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | CategorySchemeMapRef Type | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**CategorySchemeMapRefType:**CategorySchemeMapRefType contains a set of reference fields for a category scheme map.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         CategorySchemeMapRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: CategorySchemeMap) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: mapping) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**CodelistMapReferenceType:**CodelistMapReferenceType is a type for referencing a codelist map object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         CodelistMapReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | CodelistMapRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**CodelistMapRefType:**CodelistMapRefType contains a set of reference fields for a codelist map.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         CodelistMapRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: CodelistMap) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: mapping) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**LocalCodelistMapReferenceType:**LocalCodelistMapReferenceType is a type for referencing a codelist map object where the reference to the structure set which defines it is provided in another context (e.g. the structure set in which this reference occurs).

Derivation:

*ReferenceType* (restriction)   
   LocalCodelistMapReferenceType



Content:

Ref

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | LocalCodelistMapRefT ype | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |

**LocalCodelistMapRefType:**LocalCodelistMapRefType contains a set of reference fields for a codelist map locally.

Derivation:

*RefBaseType* (restriction)   
   *LocalIdentifiableRefBaseType* (restriction)   
         LocalCodelistMapRefType



Attributes:

id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| id | NestedIDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: true) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: CodelistMap) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: mapping) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**ConceptSchemeMapReferenceType:**ConceptSchemeMapReferenceType is a type for referencing a concept scheme map object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         ConceptSchemeMapReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | ConceptSchemeMapRefT ype | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**ConceptSchemeMapRefType:**ConceptSchemeMapRefType contains a set of reference fields for a concept scheme map.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         ConceptSchemeMapRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: ConceptSchemeMap) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: mapping) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**OrganisationSchemeMapReferenceType:**OrganisationSchemeMapReferenceType is a type for referencing a organisation scheme map object. It consists of a URN and/or a complete set of reference fields.

Derivation:

*ReferenceType* (restriction)   
   *ChildObjectReferenceType* (restriction)   
         OrganisationSchemeMapReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | OrganisationSchemeMa pRefType | Ref is used to provide a complete set of reference fields. Derived reference types will restrict the RefType so that the content of the Ref element requires exactly what is needed for a complete reference. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |
| URN | xs:anyURI | URN is used to hold the URN of the referenced object. This must be the same URN that would be constructed from the individual fields in the Ref element. |

**OrganisationSchemeMapRefType:**OrganisationSchemeMapRefType contains a set of reference fields for an organisation scheme map.

Derivation:

*RefBaseType* (restriction)   
   *ChildObjectRefBaseType* (restriction)   
         OrganisationSchemeMapRefType



Attributes:

agencyID, maintainableParentID, maintainableParentVersion?, id, local?, class?, package?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | NestedNCNameIDType | The agencyID attribute identifies the maintenance agency for the object being referenced (agency-id in the URN structure). This is optional to allow for local references (where the other reference fields are inferred from another context), but all complete references will require this. |
| maintainableParentID | IDType | The maintainableParentID attribute identifies the maintainable object in which the referenced object is defined, if applicable (maintainable-parent-object-id in the URN structure). This is only used in references where the referenced object is not itself maintainable. |
| maintainableParentVersion (default: 1.0) | VersionType | The maintainableParentVersion attribute identifies the version of the maintainable object in which the referenced object is defined (maintainable-parent-object-version in the URN structure). This is only used in references where the referenced object is not itself maintainable. This should only be used when the maintainableParentID is present. If this is available, a default of 1.0 will always apply. |
| id | IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| local (fixed: false) | xs:boolean | The local attribute indicates whether this set of reference fields is meant for local referencing, in which case some of the reference fields will be implied from another context. Concrete instances of this class will always fix this value to either true or false, depending on their intended usage. If the value is fixed to true, then the complete set of reference fields will be required and a URN can be fully composed from the values. |
| class (fixed: OrganisationSchemeMap) | ObjectTypeCodelistTy pe | The class attribute indicates the class name of the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |
| package (fixed: mapping) | PackageTypeCodelistT ype | The package attribute indicates the package name for the object being referenced. This attribute allows any reference to be processed generically from this definition. References derived from this should fix the value of this attribute to indicate the type of object that is being referenced, or in the case that a reference which allows specific types of fields, the representation should be sub-setted to the appropriate values. |

**SetReferenceType:**SetReferenceType defines the structure of a reference to a data/metadata set. A full reference to a data provider and the identifier for the data set must be provided. Note that this is not derived from the base reference structure since data/metadata sets are not technically identifiable.

Content:

DataProvider, ID

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| DataProvider | DataProviderReferenc eType | DataProvider references a the provider of the data/metadata set. A URN and/or a complete set of reference fields must be provided. |
| ID | IDType | ID contains the identifier of the data/metadata set being referenced. |

**ObjectTypeListType:**ObjectTypeListType provides a means for enumerating object types.

Content:

(Any?, Agency?, AgencyScheme?, AttachmentConstraint?, Attribute?, AttributeDescriptor?, Categorisation?, Category?, CategorySchemeMap?, CategoryScheme?, Code?, CodeMap?, Codelist?, CodelistMap?, ComponentMap?, Concept?, ConceptMap?, ConceptScheme?, ConceptSchemeMap?, ContentConstraint?, Dataflow?, DataConsumer?, DataConsumerScheme?, DataProvider?, DataProviderScheme?, DataSetTarget?, DataStructure?, Dimension?, DimensionDescriptor?, DimensionDescriptorValuesTarget?, GroupDimensionDescriptor?, HierarchicalCode?, HierarchicalCodelist?, Hierarchy?, HybridCodelistMap?, HybridCodeMap?, IdentifiableObjectTarget?, Level?, MeasureDescriptor?, MeasureDimension?, Metadataflow?, MetadataAttribute?, MetadataSet?, MetadataStructure?, MetadataTarget?, OrganisationMap?, OrganisationSchemeMap?, OrganisationUnit?, OrganisationUnitScheme?, PrimaryMeasure?, Process?, ProcessStep?, ProvisionAgreement?, ReportingCategory?, ReportingCategoryMap?, ReportingTaxonomy?, ReportingTaxonomyMap?, ReportPeriodTarget?, ReportStructure?, StructureMap?, StructureSet?, TimeDimension?, Transition?)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Any | EmptyType | Any is an empty element that denotes an object of any type. |
| Agency | EmptyType | Agency is an empty element that denotes an agency object. |
| AgencyScheme | EmptyType | AgencyScheme is an empty element that denotes an agency scheme object. |
| AttachmentConstraint | EmptyType | AttachmentConstraint is an empty element that denotes an attachment constraint object. |
| Attribute | EmptyType | Attribute is an empty element that denotes an attribute object. |
| AttributeDescriptor | EmptyType | AttributeDescriptor is an empty element that denotes an attribute descriptor object. |
| Categorisation | EmptyType | Categorisation is an empty element that denotes a categorisation object. |
| Category | EmptyType | Category is an empty element that denotes a category object. |
| CategorySchemeMap | EmptyType | CategorySchemeMap is an empty element that denotes a category scheme map object. |
| CategoryScheme | EmptyType | CategoryScheme is an empty element that denotes a category scheme object. |
| Code | EmptyType | Code is an empty element that denotes a code object. |
| CodeMap | EmptyType | CodeMap is an empty element that denotes a code map object. |
| Codelist | EmptyType | Codelist is an empty element that denotes a code list object. |
| CodelistMap | EmptyType | CodelistMap is an empty element that denotes a code list map object. |
| ComponentMap | EmptyType | ComponentMap is an empty element that denotes a component map object. |
| Concept | EmptyType | Concept is an empty element that denotes a concept object. |
| ConceptMap | EmptyType | ConceptMap is an empty element that denotes a concept map object. |
| ConceptScheme | EmptyType | ConceptScheme is an empty element that denotes a concept scheme object. |
| ConceptSchemeMap | EmptyType | ConceptSchemeMap is an empty element that denotes a concept scheme map object. |
| ContentConstraint | EmptyType | ContentConstraint is an empty element that denotes a content constraint object. |
| Dataflow | EmptyType | Dataflow is an empty element that denotes a data flow object. |
| DataConsumer | EmptyType | DataConsumer is an empty element that denotes a data consumer object. |
| DataConsumerScheme | EmptyType | DataConsumerScheme is an empty element that denotes a data consumer scheme object. |
| DataProvider | EmptyType | DataProvider is an empty element that denotes a data provider object. |
| DataProviderScheme | EmptyType | DataProviderScheme is an empty element that denotes a data provider scheme object. |
| DataSetTarget | EmptyType | DataSetTarget is an empty element that denotes a data set target object. |
| DataStructure | EmptyType | DataStructure is an empty element that denotes a data structure definition object. |
| Dimension | EmptyType | Dimension is an empty element that denotes a dimension object. |
| DimensionDescriptor | EmptyType | DimensionDescriptor is an empty element that denotes a dimension descriptor object. |
| DimensionDescriptorV aluesTarget | EmptyType | DimensionDescriptorValuesTarget is an empty element that denotes a dimension descriptor values target object. |
| GroupDimensionDescri ptor | EmptyType | GroupDimensionDescriptor is an empty element that denotes a group dimension descriptor object. |
| HierarchicalCode | EmptyType | HierarchicalCode is an empty element that denotes a hierarchical code object. |
| HierarchicalCodelist | EmptyType | HierarchicalCodelist is an empty element that denotes a hierarchical codelist object. |
| Hierarchy | EmptyType | Hierarchy is an empty element that denotes a hierarchy within a hiearcharchical codelist. |
| HybridCodelistMap | EmptyType | HybridCodelistMap is an empty element that denotes a hybrid codelist map object. |
| HybridCodeMap | EmptyType | HybridCodeMap is an empty element that denotes a hybrid code map object. |
| IdentifiableObjectTa rget | EmptyType | IdentifiableObjectTarget is an empty element that denotes an identifiable object target object. |
| Level | EmptyType | Level is an empty element that denotes a level object. |
| MeasureDescriptor | EmptyType | MeasureDescriptor is an empty element that denotes a measure descriptor object. |
| MeasureDimension | EmptyType | MeasureDimension is an empty element that denotes a measure dimension object. |
| Metadataflow | EmptyType | Metadataflow is an empty element that denotes a metadata flow object. |
| MetadataAttribute | EmptyType | MetadataAttribute is an empty element that denotes a metadata attribute object. |
| MetadataSet | EmptyType | MetadataSet is an empty element that denotes a metadata set object. |
| MetadataStructure | EmptyType | MetadataStructure is an empty element that denotes a metadata structure definition object. |
| MetadataTarget | EmptyType | MetadataTarget is an empty element that denotes a metadata target object. |
| OrganisationMap | EmptyType | OrganisationMap is an empty element that denotes an organisation map object. |
| OrganisationSchemeMa p | EmptyType | OrganisationSchemeMap is an empty element that denotes an organisation scheme map object. |
| OrganisationUnit | EmptyType | OrganisationUnit is an empty element that denotes an organisation unit object. |
| OrganisationUnitSche me | EmptyType | OrganisationUnitScheme is an empty element that denotes an organisation unit scheme object. |
| PrimaryMeasure | EmptyType | PrimaryMeasure is an empty element that denotes a primary measure object. |
| Process | EmptyType | Process is an empty element that denotes a process object. |
| ProcessStep | EmptyType | ProcessStep is an empty element that denotes a process step object. |
| ProvisionAgreement | EmptyType | ProvisionAgreement is an empty element that denotes a provision agreement object. |
| ReportingCategory | EmptyType | ReportingCategory is an empty element that denotes a reporting category object. |
| ReportingCategoryMap | EmptyType | ReportingCategoryMap is an empty element that denotes a reporting category map object. |
| ReportingTaxonomy | EmptyType | ReportingTaxonomy is an empty element that denotes a reporting taxonomy object. |
| ReportingTaxonomyMap | EmptyType | ReportingTaxonomyMap is an empty element that denotes a reporting taxonomy map object. |
| ReportPeriodTarget | EmptyType | ReportPeriodTarget is an empty element that denotes a report period target object. |
| ReportStructure | EmptyType | ReportStructure is an empty element that denotes a report structure object. |
| StructureMap | EmptyType | StructureMap is an empty element that denotes a structure map object. |
| StructureSet | EmptyType | StructureSet is an empty element that denotes a structure set object. |
| TimeDimension | EmptyType | TimeDimension is an empty element that denotes a time dimension object. |
| Transition | EmptyType | Transition is an empty element that denotes a transition object. |

**MaintainableObjectTypeListType:**MaintainableObjectTypeListType provides a means for enumerating maintainable object types.

Derivation:

ObjectTypeListType (restriction)   
   MaintainableObjectTypeListType



Content:

(AgencyScheme?, AttachmentConstraint?, Categorisation?, CategoryScheme?, Codelist?, ConceptScheme?, ContentConstraint?, Dataflow?, DataConsumerScheme?, DataProviderScheme?, DataStructure?, HierarchicalCodelist?, Metadataflow?, MetadataStructure?, OrganisationUnitScheme?, Process?, ProvisionAgreement?, ReportingTaxonomy?, StructureSet?)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| AgencyScheme | EmptyType | AgencyScheme is an empty element that denotes an agency scheme object. |
| AttachmentConstraint | EmptyType | AttachmentConstraint is an empty element that denotes an attachment constraint object. |
| Categorisation | EmptyType | Categorisation is an empty element that denotes a categorisation object. |
| CategoryScheme | EmptyType | CategoryScheme is an empty element that denotes a category scheme object. |
| Codelist | EmptyType | Codelist is an empty element that denotes a code list object. |
| ConceptScheme | EmptyType | ConceptScheme is an empty element that denotes a concept scheme object. |
| ContentConstraint | EmptyType | ContentConstraint is an empty element that denotes a content constraint object. |
| Dataflow | EmptyType | Dataflow is an empty element that denotes a data flow object. |
| DataConsumerScheme | EmptyType | DataConsumerScheme is an empty element that denotes a data consumer scheme object. |
| DataProviderScheme | EmptyType | DataProviderScheme is an empty element that denotes a data provider scheme object. |
| DataStructure | EmptyType | DataStructure is an empty element that denotes a data structure definition object. |
| HierarchicalCodelist | EmptyType | HierarchicalCodelist is an empty element that denotes a hierarchical codelist object. |
| Metadataflow | EmptyType | Metadataflow is an empty element that denotes a metadata flow object. |
| MetadataStructure | EmptyType | MetadataStructure is an empty element that denotes a metadata structure definition object. |
| OrganisationUnitSche me | EmptyType | OrganisationUnitScheme is an empty element that denotes an organisation unit scheme object. |
| Process | EmptyType | Process is an empty element that denotes a process object. |
| ProvisionAgreement | EmptyType | ProvisionAgreement is an empty element that denotes a provision agreement object. |
| ReportingTaxonomy | EmptyType | ReportingTaxonomy is an empty element that denotes a reporting taxonomy object. |
| StructureSet | EmptyType | StructureSet is an empty element that denotes a structure set object. |

### 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-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-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, 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. |

**DimensionTypeType:**DimensionTypeType enumerates the sub-classes of a dimension.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Dimension | An ordinary dimension. |
| MeasureDimension | A measure dimension. |
| TimeDimension | The time dimension. |

**ContentConstraintTypeCodeType:**ContentConstraintTypeCodeType defines a list of types for a content constraint. A content constraint can state which data is present or which content is allowed for the constraint attachment.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Allowed | The constraint contains the allowed values for attachable object. |
| Actual | The constraints contains the actual data present for the attachable object. |

**SimpleOperatorType:**SimpleOperatorType provides an enumeration of simple operators to be applied to any value.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| notEqual | (!=) - value must not be equal to the value supplied. |
| equal | (=) - value must be exactly equal to the value supplied. |

**RangeOperatorType:**RangeOperatorType provides an enumeration of range operators to be applied to an ordered value.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| greaterThanOrEqual | (>=) - value must be greater than or equal to the value supplied. |
| lessThanOrEqual | (<=) - value must be less than or equal to the value supplied. |
| greaterThan | (>) - value must be greater than the value supplied. |
| lessThan | (<) - value must be less than the value supplied. |

**TextSearchOperatorType:**TextSearchOperatorType provides an enumeration of text search operators.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| contains | The text being searched must contain the supplied text. |
| startsWith | The text being searched must start with the supplied text. |
| endsWith | The text being searched must end with the supplied text. |
| doesNotContain | The text being searched cannot contain the supplied text. |
| doesNotStartWith | The text being searched cannot start with the supplied text. |
| doesNotEndWith | The text being searched cannot end with the supplied text. |

**OrderedOperatorType:**OrderedOperatorType combines the SimpleOperatorType and the RangeOperatorType to provide a full range or operators for any ordered value.

Union of:

SimpleOperatorType, RangeOperatorType.

**TextOperatorType:**

Union of:

SimpleOperatorType, TextSearchOperatorType.

**TimeOperatorType:**TimeOperatorType derives from the OrderedOperatorType to remove the notEqual operator.

Derived by restriction of OrderedOperatorType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| equal |  |
| greaterThanOrEqual |  |
| lessThanOrEqual |  |
| greaterThan |  |
| lessThan |  |

**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. |
| AlphaNumeric | A string datatype which only allows for the simple alphabetic character set of 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. |
| 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. |
| AttachmentConstraintReference | This value indicates that the content of the component will be reference to an attachment constraint, which is actually a combination of a collection of full or partial key values and a reference to a data set or data structure, usage, or provision agreement. |

**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. |
| AlphaNumeric | A string datatype which only allows for the simple alphabetic character set of 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. |
| 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. |
| AlphaNumeric | A string datatype which only allows for the simple alphabetic character set of 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. |

**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. |

**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-z0-9\_@$\-]+(\.[A-z0-9\_@$\-]+)\*

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

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

**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-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-z][A-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-z][A-z0-9\_\-]\*(\.[A-z][A-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-z][A-z0-9\_\-]\*

**VersionType:**VersionType is used to communicate version information. The format is restricted to allow for simple incrementing and sorting of version number. The version consists of an unlimited set of 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: [0-9]+(\.[0-9]+)\*

**VersionQueryType:**VersionQueryType combines the VersionType and LateBoundVersionType to allow one to query for either a specific version of an object, or the latest version by specifying the '\*' value.

Union of:

VersionType, LateBoundVersionType.

**LateBoundVersionType:**LateBoundVersionType is a single value code list, used to include the '\*' character - indicating that the latest version of an object is required.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| \* | Indicates that the latest version of an object is requested. |

**PackageTypeCodelistType:**PackageTypeCodelistType provides an enumeration of all SDMX package names.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| base |  |
| datastructure |  |
| metadatastructure |  |
| process |  |
| registry |  |
| mapping |  |
| codelist |  |
| categoryscheme |  |
| conceptscheme |  |

**ItemSchemePackageTypeCodelistType:**ItemSchemePackageTypeCodelistType provides an enumeration of all SDMX packages which contain item schemes.

Derived by restriction of PackageTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| base |  |
| codelist |  |
| categoryscheme |  |
| conceptscheme |  |

**StructurePackageTypeCodelistType:**StructurePackageTypeCodelistType provides an enumeration of all SDMX packages which contain structure and structure usages.

Derived by restriction of PackageTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| datastructure |  |
| metadatastructure |  |

**ObjectTypeCodelistType:**ObjectTypeCodelistType provides an enumeration of all objects outside of the base infomration model class. This includes some abstract object types such as Organsiation and Constraint.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Any |  |
| Agency |  |
| AgencyScheme |  |
| AttachmentConstraint |  |
| Attribute |  |
| AttributeDescriptor |  |
| Categorisation |  |
| Category |  |
| CategorySchemeMap |  |
| CategoryScheme |  |
| Code |  |
| CodeMap |  |
| Codelist |  |
| CodelistMap |  |
| ComponentMap |  |
| Concept |  |
| ConceptMap |  |
| ConceptScheme |  |
| ConceptSchemeMap |  |
| Constraint |  |
| ConstraintTarget |  |
| ContentConstraint |  |
| Dataflow |  |
| DataConsumer |  |
| DataConsumerScheme |  |
| DataProvider |  |
| DataProviderScheme |  |
| DataSetTarget |  |
| DataStructure |  |
| Dimension |  |
| DimensionDescriptor |  |
| DimensionDescriptorValuesTarget |  |
| GroupDimensionDescriptor |  |
| HierarchicalCode |  |
| HierarchicalCodelist |  |
| Hierarchy |  |
| HybridCodelistMap |  |
| HybridCodeMap |  |
| IdentifiableObjectTarget |  |
| Level |  |
| MeasureDescriptor |  |
| MeasureDimension |  |
| Metadataflow |  |
| MetadataAttribute |  |
| MetadataSet |  |
| MetadataStructure |  |
| MetadataTarget |  |
| Organisation |  |
| OrganisationMap |  |
| OrganisationScheme |  |
| OrganisationSchemeMap |  |
| OrganisationUnit |  |
| OrganisationUnitScheme |  |
| PrimaryMeasure |  |
| Process |  |
| ProcessStep |  |
| ProvisionAgreement |  |
| ReportingCategory |  |
| ReportingCategoryMap |  |
| ReportingTaxonomy |  |
| ReportingTaxonomyMap |  |
| ReportingYearStartDay |  |
| ReportPeriodTarget |  |
| ReportStructure |  |
| StructureMap |  |
| StructureSet |  |
| TimeDimension |  |
| Transition |  |

**MaintainableTypeCodelistType:**MaintainableTypeCodelistType provides an enumeration of all maintainable objects.

Derived by restriction of ObjectTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Any |  |
| AgencyScheme |  |
| AttachmentConstraint |  |
| Categorisation |  |
| CategoryScheme |  |
| Codelist |  |
| ConceptScheme |  |
| Constraint |  |
| ContentConstraint |  |
| Dataflow |  |
| DataConsumerScheme |  |
| DataProviderScheme |  |
| DataStructure |  |
| HierarchicalCodelist |  |
| Metadataflow |  |
| MetadataStructure |  |
| OrganisationScheme |  |
| OrganisationUnitScheme |  |
| Process |  |
| ProvisionAgreement |  |
| ReportingTaxonomy |  |
| StructureSet |  |

**ConcreteMaintainableTypeCodelistType:**ConcreteMaintainableTypeCodelistType provides an enumeration of all concrete maintainable objects.

Derived by restriction of MaintainableTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| AgencyScheme |  |
| AttachmentConstraint |  |
| Categorisation |  |
| CategoryScheme |  |
| Codelist |  |
| ConceptScheme |  |
| ContentConstraint |  |
| Dataflow |  |
| DataConsumerScheme |  |
| DataProviderScheme |  |
| DataStructure |  |
| HierarchicalCodelist |  |
| Metadataflow |  |
| MetadataStructure |  |
| OrganisationUnitScheme |  |
| Process |  |
| ProvisionAgreement |  |
| ReportingTaxonomy |  |
| StructureSet |  |

**CodelistTypeCodelistType:**CodelistTypeCodelistType provides an enumeration of all codelist objects.

Derived by restriction of MaintainableTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Codelist |  |
| HierarchicalCodelist |  |

**CodeTypeCodelistType:**CodeTypeCodelistType provides an enumeration of all code objects.

Derived by restriction of ObjectTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Code |  |
| HierarchicalCode |  |

**ConstraintTypeCodelistType:**ConstraintTypeCodelistType provides an enumeration of all constraint objects.

Derived by restriction of MaintainableTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| AttachmentConstraint |  |
| ContentConstraint |  |

**ItemSchemeTypeCodelistType:**ItemSchemeTypeCodelistType provides an enumeration of all item scheme objects.

Derived by restriction of MaintainableTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| AgencyScheme |  |
| CategoryScheme |  |
| Codelist |  |
| ConceptScheme |  |
| DataConsumerScheme |  |
| DataProviderScheme |  |
| OrganisationUnitScheme |  |
| ReportingTaxonomy |  |

**OrganisationSchemeTypeCodelistType:**OrganisationSchemeTypeCodelistType provides an enumeration of all organisation scheme objects.

Derived by restriction of ItemSchemeTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| AgencyScheme |  |
| DataConsumerScheme |  |
| DataProviderScheme |  |
| OrganisationUnitScheme |  |

**OrganisationTypeCodelistType:**OrganisationTypeCodelistType provides an enumeration of all organisation objects.

Derived by restriction of ItemTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Agency |  |
| DataConsumer |  |
| DataProvider |  |
| OrganisationUnit |  |

**StructureOrUsageTypeCodelistType:**StructureOrUsageTypeCodelistType provides an enumeration all structure and structure usage objects

Derived by restriction of MaintainableTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Dataflow |  |
| DataStructure |  |
| Metadataflow |  |
| MetadataStructure |  |

**StructureTypeCodelistType:**StructureTypeCodelistType provides an enumeration all structure objects

Derived by restriction of StructureOrUsageTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| DataStructure |  |
| MetadataStructure |  |

**StructureUsageTypeCodelistType:**StructureUsageTypeCodelistType provides an enumeration all structure usage objects

Derived by restriction of StructureOrUsageTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Dataflow |  |
| Metadataflow |  |

**ItemTypeCodelistType:**ItemTypeCodelistType provides an enumeration of all item objects.

Derived by restriction of ObjectTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Agency |  |
| Category |  |
| Code |  |
| Concept |  |
| DataConsumer |  |
| DataProvider |  |
| OrganisationUnit |  |
| ReportingCategory |  |

**ComponentListTypeCodelistType:**ComponentListTypeCodelistType provides an enumeration of all component list objects.

Derived by restriction of ObjectTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| AttributeDescriptor |  |
| DimensionDescriptor |  |
| GroupDimensionDescriptor |  |
| MeasureDescriptor |  |
| MetadataTarget |  |
| ReportStructure |  |

**ComponentTypeCodelistType:**ComponentTypeCodelistType provides an enumeration of all component objects.

Derived by restriction of ObjectTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Attribute |  |
| ConstraintTarget |  |
| DataSetTarget |  |
| Dimension |  |
| IdentifiableObjectTarget |  |
| DimensionDescriptorValuesTarget |  |
| MeasureDimension |  |
| MetadataAttribute |  |
| PrimaryMeasure |  |
| ReportingYearStartDay |  |
| ReportPeriodTarget |  |
| TimeDimension |  |

**DataStructureComponentTypeCodelistType:**DataStructureComponentTypeCodelistType provides an enumeration of all data structure component objects, except for the primary measure.

Derived by restriction of ComponentTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Attribute |  |
| Dimension |  |
| MeasureDimension |  |
| PrimaryMeasure |  |
| ReportingYearStartDay |  |
| TimeDimension |  |

**DimensionEumerationSchemeTypeCodelistType:**DimensionEumerationSchemeTypeCodelistType provides an enumeration of all item schemes which are allowable as the representation of a data structure definition component.

Derived by restriction of ItemSchemeTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Codelist |  |
| ConceptScheme |  |

**MetadataStructureComponentTypeCodelistType:**MetadataStructureComponentTypeCodelistType provides an enumeration of all metadata structure component objects.

Derived by restriction of ComponentTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| ConstraintTarget |  |
| DataSetTarget |  |
| IdentifiableObjectTarget |  |
| DimensionDescriptorValuesTarget |  |
| MetadataAttribute |  |
| ReportPeriodTarget |  |

**DimensionTypeCodelistType:**DimensionTypeCodelistType provides an enumeration of all dimension objects.

Derived by restriction of ComponentTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Dimension |  |
| MeasureDimension |  |
| TimeDimension |  |

**TargetObjectTypeCodelistType:**TargetObjectTypeCodelistType provides an enumeration of all target object objects.

Derived by restriction of ComponentTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| ConstraintTarget |  |
| DataSetTarget |  |
| IdentifiableObjectTarget |  |
| DimensionDescriptorValuesTarget |  |
| ReportPeriodTarget |  |