SDMX Standards: Section 3A PaRT V

SDMX-ML:

Schema and Documentation

Query Namespace

Version 2.1

April 2011

© SDMX 2011

http://www.sdmx.org/

Contents

[1 Introduction 1](#_Toc290376885)

[2 Schema Documentation 1](#_Toc290376886)

[2.1 Query Namespace 1](#_Toc290376887)

[2.1.1 Summary 1](#_Toc290376888)

[2.1.2 Global Elements 1](#_Toc290376889)

[2.1.3 Complex Types 8](#_Toc290376890)

[2.1.4 Simple Types 178](#_Toc290376891)

# Introduction

The query constructs in SDMX have been modified to more closely follow the information model. The intention is to create a consistent structure for all queries so that they can be more easily processed and so that there is no ambiguity as to what the purpose of the request is. This means that for structural metadata, the queries derive from types as they do in the information model. Because a data flow and a metadata flow both are derived from the structure usage, there query constructs are nearly identical. The same basic principle is applied to the data and metadata queries. Where possible, they are consistent with each other.

To allow for web services to be explicit in the inputs to their functions, specific messages have been created for the various structural objects. This allows a function for querying codelists to be explicit as to the fact that the only acceptable input to this query is a codelist query.

Finally, a mechanism for requesting structure specific schemas has been introduced. The intentions is that one could request the schemas from an organisation so that they will have them readily available for processing of data retrieved from the same organisation.

# Schema Documentation

## Query Namespace

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

### Summary

Referenced Namespaces:

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

Contents:

40 Global Elements  
124 Complex Types  
11 Simple Types

### Global Elements

***StructuralMetadataWhere* (*MaintainableWhereType*):**StructuralMetadataWhere is an abstract substitution head that forms the basis of any structural metadata query's details. This contains the actual parameters to be matched. These parameters are implicitly joined by an "and" connector (i.e. each of the parameters must be matched in order to satisfy the query). If it is necessary to supply "or" conditions for a parameter, this should be done by supplying multiple queries.

Substitutions: CategorisationWhere, CategorySchemeWhere, CodelistWhere,ConceptSchemeWhere, ConstraintWhere, DataflowWhere,DataStructureWhere, HierarchicalCodelistWhere, MetadataflowWhere,MetadataStructureWhere, OrganisationSchemeWhere, ProcessWhere,ProvisionAgreementWhere, ReportingTaxonomyWhere, StructureSetWhere,StructuresWhere

***ItemWhere* (*ItemWhereType*):**ItemWhere is an abstract substitution head to query for an item within a parent item where or an item scheme query. An item where query is implicitly an and-query - that is the conditions within the item where must all be met in order to return a match. If this is nested within another item where, the meaning is that the conditions specified must be matched within the hierarchy provided.

Substitutions: CategoryWhere, CodeWhere, ConceptWhere, OrganisationWhere,ReportingCategoryWhere

***ComponentListWhere* (*ComponentListWhereType*):**ComponentListWhere is an abstract substitution head to query for a component list within a structure. A component list where query is implicitly an and-query - that is the conditions within the component list where must all be met in order to return a match.

Substitutions: GroupWhere, MetadataTargetWhere, ReportStructureWhere

***ComponentWhere* (*ComponentWhereType*):**ComponentWhere is an abstract substitution head to query for a component within a component list or a structure. A component where query is implicitly an and-query - that is the conditions within the component where must all be met in order to return a match.

Substitutions: GroupDimensionWhere, AttributeWhere, DimensionWhere,TimeDimensionWhere, PrimaryMeasureWhere, MeasureDimensionWhere,TargetObjectWhere, MetadataAttributeWhere

**NumericValue (NumericValueType):**NumericValue is used to query for a the value of a concept or component based on a numeric search. This is typically used when the value needs to be searched explicitly as a number, such as when data is sought with an observed value greater than some threshold. If only a simple match is required (i.e. the numeric value equals 'x') then the Value element can be used.

**TextValue (QueryTextType):**TextValue is used to query for the value of a concept or component based on textual parameters. The text value can be language specific (where parallel multi-lingual values are available) and is qualified with an operator indicating how the supplied text should be matched against the sought components. If only a simple equality check is necessary, regardless of language, the Value element can be used.

**TimeValue (TimePeriodValueType):**TimeValue is used to query for the value of a concept or component based on time parameters. This is typically used when the value needs to be treated explicitly as a time, for example when searching for data after a particular point in time. If only a simple equality check is necessary, the Value element can be used.

**Value (SimpleValueType):**Value is used to query for the value of a component. This should be used for concepts or components based on a simple value (e.g. a code or a simple string). This should be used when a simple equal/not equal operator is all that is necessary for matching the sought value.

**CategorisationWhere (CategorisationWhereType):**CategorisationWhere defines the parameters for a categorisation query. All parameters must be matched for an object to satisfy the query.In addition to querying based on the basic maintainable properties, it is also possible to search based on the source object being categorized and target category the object is categorized against.

Substitution For: *StructuralMetadataWhere*

**CategorySchemeWhere (CategorySchemeWhereType):**CategorySchemeWhere contains the parameters for a category scheme query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a category scheme based on the details of its categories. In any case, the category scheme will be returned according the indicated return detail.

Substitution For: *StructuralMetadataWhere*

**CategoryWhere (CategoryWhereType):**CategoryWhere is used to query for categories matching the parameters supplied. It allows for nested category queries so that hierarchical categories can be queried explicitly by their nested level, although a top level category will always result in a search for categories at any level. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match.

Substitution For: *ItemWhere*

**CodelistWhere (CodelistWhereType):**CodelistWhere defines the parameters for a codelist query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a codlist based on the details of its codes. In any case, the codelist will be returned according the indicated return detail.

Substitution For: *StructuralMetadataWhere*

**CodeWhere (CodeWhereType):**CodeWhere is used to query for codes matching the parameters supplied. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match.

Substitution For: *ItemWhere*

**ConceptSchemeWhere (ConceptSchemeWhereType):**ConceptSchemeWhere defines the parameters for a concept scheme query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a concept scheme based on the details of its concepts. In any case, the concept scheme will be returned according the indicated return detail.

Substitution For: *StructuralMetadataWhere*

**ConceptWhere (ConceptWhereType):**ConceptWhere is used to query for concepts matching the parameters supplied. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match.

Substitution For: *ItemWhere*

**ConstraintWhere (ConstraintWhereType):**AttachmentConstraintWhere contains the parameters for a constraint query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a constraint based on the objects it applies to.

Substitution For: *StructuralMetadataWhere*

**DataflowWhere (DataflowWhereType):**DataflowWhere defines the parameters for a dataflow query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a dataflow based on the key family it defines the usage of.

Substitution For: *StructuralMetadataWhere*

**DataStructureWhere (DataStructureWhereType):**DataStructureWhere contains the parameters for a data structure definition query. All parameters must be matched for an object to satisfy the query. The query is simply a refinement of the base structure query to make the parameters specific to the data structure definition.

Substitution For: *StructuralMetadataWhere*

**GroupWhere (GroupWhereType):**GroupWhere is used to query for a data structure definition that contains a group meeting the conditions detailed in this container. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentListWhere*

**GroupDimensionWhere (DimensionWhereType):**GroupDimensionWhere is used to query a group based on the details of the dimensions it groups. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentWhere*

**AttributeWhere (AttributeWhereType):**AttributeWhere is used to query for a data structure definition that contains an attribute meeting the conditions contained in this structure. The attribute can be queried based on its identification, the concept from which it takes its semantic, its attachment level, the role it plays, and the code list it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentWhere*

**DimensionWhere (DimensionWhereType):**DimensionWhere is used to query for a data structure definition that contains a dimension meeting the conditions contained in this structure. The dimension can be queried based on its identification, the concept from which it takes its semantic, the role it plays, and the code list it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentWhere*

**TimeDimensionWhere (TimeDimensionWhereType):**TimeDimensionWhere is used to query for a data structure definition that contains a time dimension meeting the conditions contained in this structure. The time dimension can be queried based on its identification and the concept from which it takes its semantic. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentWhere*

**PrimaryMeasureWhere (PrimaryMeasureWhereType):**PrimaryMeasureWhere is used to query for a data structure definition that contains a primary measure meeting the conditions contained in this structure. The primary measure can be queried based on its identification, the concept from which it takes its semantic, and the code list it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentWhere*

**MeasureDimensionWhere (MeasureDimensionWhereType):**MeasureDimensionWhere is used to query for a data structure definition that contains a measure dimension meeting the conditions contained in this structure. The cross-sectional measure can be queried based on its identification, the concept from which it takes its semantic, and the concept scheme it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentWhere*

**HierarchicalCodelistWhere (HierarchicalCodelistWhereType):**HierarchicalCodelistWhere defines the parameters for a hierarchical codelist query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a hierarchical codelist based on the codelists it arranges into hierarchies.

Substitution For: *StructuralMetadataWhere*

**MetadataflowWhere (MetadataflowWhereType):**MetadataflowWhere contains the parameters for a metadataflow query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a metadataflow based on the metadata structure definition it defines the usage of.

Substitution For: *StructuralMetadataWhere*

**MetadataStructureWhere (MetadataStructureWhereType):**MetadataStructureWhere contains the parameters for a metadata structure definition query. All parameters must be matched for an object to satisfy the query. The query is simply a refinement of the base structure query to make the parameters specific to the metadata structure definition.

Substitution For: *StructuralMetadataWhere*

**MetadataTargetWhere (MetadataTargetWhereType):**MetadataTargetWhere is used to query for a metadata structure definition that contains a metadata target meeting the conditions contained in this structure. The metadata target can be queried based on its identification and/or the details of its target objects. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentListWhere*

**TargetObjectWhere (TargetObjectWhereType):**IdentifierComponentWhere is used to query for specific target identifiers or metadata structure definitions where a contained identifier component meets the conditions detailed. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentWhere*

**ReportStructureWhere (ReportStructureWhereType):**ReportStructureWhere is used to query for metadata structure definitions where a given report structure meets the conditions specified. A report structure can be queried based on identification or details about its metadata attributes. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentListWhere*

**MetadataAttributeWhere (MetadataAttributeWhereType):**MetadataAttributeWhere is a parameter which is used in a report structure parameter or to query metadata structure definitions where a contained metadata attribute meets the conditions specified. A metadata attribute can be queried based on its identification, the concept from which it takes its semantic, and an item scheme it uses as its representation. Nested metadata attributes allow for the querying of metadata attributes explicitly at nested level, although a top level metadata attribute query will be processed by querying metadata attributes at any level. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Substitution For: *ComponentWhere*

**OrganisationSchemeWhere (OrganisationSchemeWhereType):**OrganisationSchemeWhere defines the parameters for an organisation scheme query, regardless of the specific type of organisation scheme being sought. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for an organisation scheme based on the details of its organisations. In any case, the organisation scheme will be returned according the indicated return detail.

Substitution For: *StructuralMetadataWhere*

**OrganisationWhere (OrganisationWhereType):**OrganisationWhere is used to query for organisations matching the parameters supplied. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match.

Substitution For: *ItemWhere*

**ProcessWhere (ProcessWhereType):**ProcessWhere contains the parameters for a process query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to query based on the details of the process steps defined within the process. In any case, the entire process will be returned according the indicated return detail.

Substitution For: *StructuralMetadataWhere*

**ProvisionAgreementWhere (ProvisionAgreementWhereType):**ProvisionAgreementWhere contains the parameters for a provision agreement query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a provision agreement based on the data provider and the structure usage it pairs.

Substitution For: *StructuralMetadataWhere*

**ReportingTaxonomyWhere (ReportingTaxonomyWhereType):**ReportingTaxonomyWhere contains the parameters for a reporting taxonomy query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a reporting taxonomy based on the details of its reporting categories. In any case, the reporting taxonomy will be returned according the indicated return detail.

Substitution For: *StructuralMetadataWhere*

**ReportingCategoryWhere (ReportingCategoryWhereType):**ReportingCategoryWhere is used to query for reporting categories matching the parameters supplied. It allows for nested reporting category queries so that hierarchical reporting categories can be queried explicitly by their nested level, although a top level reporting category will always result in a search for reporting categories at any level. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match.

Substitution For: *ItemWhere*

**StructureSetWhere (StructureSetWhereType):**StructureSetWhere contains the parameters for a structure query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search based on the structures that are related by the set or the objects which are mapped by the set's maps. In any case, the structure set will be returned according the indicated return detail.

Substitution For: *StructuralMetadataWhere*

**StructuresWhere (StructuresWhereType):**StructuresWhere defines the parameters for a structures query. All parameters must be matched for an object to satisfy the query. Only the basic maintainable parameters are available.

Substitution For: *StructuralMetadataWhere*

### Complex Types

***ReturnDetailsBaseType*:**ReturnDetailsBaseType is an abstract type that forms the basis for any query return details.

Attributes:

defaultLimit?, detail?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| defaultLimit | xs:integer | The defaultLimit attribute is the suggested maximum response size in kilobytes. |
| detail | xs:string | >The detail attribute is used to indicate how much of the matched object should be returned. |

***StructuralMetadataQueryType*:**StructureWhereQueryType is an abstract base type that serves as the basis for any structural metadata query. Concrete instances of this type are implied to be an and-query. A structural object will be returned for any object where all of the conditions are met.

Content:

ReturnDetails, *StructuralMetadataWhere*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | StructureReturnDetai lsType |  |
| *StructuralMetadataWh ere* | *MaintainableWhereTyp e* | StructuralMetadataWhere is an abstract substitution head that forms the basis of any structural metadata query's details. This contains the actual parameters to be matched. These parameters are implicitly joined by an "and" connector (i.e. each of the parameters must be matched in order to satisfy the query). If it is necessary to supply "or" conditions for a parameter, this should be done by supplying multiple queries. |

***StructureReturnDetailsBaseType*:**StructureReturnDetailsBaseType is an abstract base type which forms the basis of StructureReturnDetailsType.

Derivation:

*ReturnDetailsBaseType* (restriction)   
   *StructureReturnDetailsBaseType*



Attributes:

defaultLimit?, detail?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| defaultLimit | xs:integer | The defaultLimit attribute is the suggested maximum response size in kilobytes. |
| detail (default: Full) | StructureReturnDetai lType | The detail attribute is used to indicate whether the response to the query should return the full details of matched objects, or just a subset of the information should be returned. A value of "Full" indicates that the full details of all matched objects should be returned. A value of "CompleteStub" indicates that the identification information, name, description, and annotations for the matched object should be returned. A value of "Stub" indicates that just the identification information and name should be returned for the matched objects. Note that this applies only to the object(s) matched by the query parameters. The References element has a separate field for indicating the level of detail returned for referenced objects. |

**StructureReturnDetailsType:**StructureReturnDetailsType defines the structure of the return details for any structural metadata query.

Derivation:

*ReturnDetailsBaseType* (restriction)   
   *StructureReturnDetailsBaseType* (extension)   
         StructureReturnDetailsType



Attributes:

defaultLimit?, detail?, returnMatchedArtefact?

Content:

References

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| defaultLimit | xs:integer | The defaultLimit attribute is the suggested maximum response size in kilobytes. |
| detail (default: Full) | StructureReturnDetai lType | The detail attribute is used to indicate whether the response to the query should return the full details of matched objects, or just a subset of the information should be returned. A value of "Full" indicates that the full details of all matched objects should be returned. A value of "CompleteStub" indicates that the identification information, name, description, and annotations for the matched object should be returned. A value of "Stub" indicates that just the identification information and name should be returned for the matched objects. Note that this applies only to the object(s) matched by the query parameters. The References element has a separate field for indicating the level of detail returned for referenced objects. |
| returnMatchedArtefact (default: true) | xs:boolean | The returnMatchedArtefact attribute indicates whether the object(s) match by the query should be returned. If this is set to false, only the reference objects from the match object(s) will be returned. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| References | ReferencesType | References is used to communicate how objects that reference or are referenced by the object(s) being queried should be returned. |

**MaintainableReturnDetailsType:**MaintainableReturnDetailsType defines the structure for the return details of a non-item scheme maintainable object. It eliminates the detail options that are specific to searching an item scheme

Derivation:

*ReturnDetailsBaseType* (restriction)   
   *StructureReturnDetailsBaseType* (extension)   
         StructureReturnDetailsType (restriction)   
               MaintainableReturnDetailsType



Attributes:

defaultLimit?, detail?, returnMatchedArtefact?

Content:

References

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| defaultLimit | xs:integer | The defaultLimit attribute is the suggested maximum response size in kilobytes. |
| detail (default: Full) | MaintainableReturnDe tailType | The detail attribute is used to indicate whether the response to the query should return the full details of matched objects, or just a subset of the information should be returned. A value of "Full" indicates that the full details of all matched objects should be returned. A value of "CompleteStub" indicates that the identification information, name, description, and annotations for the matched object should be returned. A value of "Stub" indicates that just the identification information and name should be returned for the matched objects. Note that this applies only to the object(s) matched by the query parameters. The References element has a separate field for indicating the level of detail returned for referenced objects. |
| returnMatchedArtefact (default: true) | xs:boolean | The returnMatchedArtefact attribute indicates whether the object(s) match by the query should be returned. If this is set to false, only the reference objects from the match object(s) will be returned. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| References | ReferencesType | References is used to communicate how objects that reference or are referenced by the object(s) being queried should be returned. |

**ReferencesType:**ReferencesType defines the structure for indicating which referenced objects should be returned in a structural metadata query. It is possible to return both objects which reference the object(s) matched by the query and objects referenced from the match object(s). The type(s) of reference objects to be returned consists of a choice between None, All, Default, or an explicit list of object types.

Attributes:

processConstraints?, detail?

Content:

(None | All | Parents | ParentsAndSiblings | Children | Descendants | SpecificObjects)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| processConstraints (default: false) | xs:boolean | The processConstraints attribute is used to request that the query process any relevant constraints for the match object(s) in order to return only the applicable portion of any referenced codelists. A value of "true" indicates that constraints should be processed. |
| detail (default: Full) | MaintainableReturnDe tailType | The detail attribute indicates the amount of detail that should be returned for reference objects. A value of "Full" indicates that the full details of all reference objects should be returned. A value of "CompleteStub" indicates that the identification information, name, description, and annotations for the reference object should be returned. A value of "Stub" indicates that just the identification information and name should be returned for the reference objects. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| None | com:EmptyType | None indicates that no reference objects should be returned. |
| All | com:EmptyType | All is a convenience to indicate that the sets indicated by the ParentsAndSiblings and Descendants should be returned. |
| Parents | com:EmptyType | Parents is a convenience to indicate that any object that refers to the matched object should be returned. This is typically used when the query the goal is to find object refer to a set of unknown objects. |
| ParentsAndSiblings | com:EmptyType | ParentsAndSiblings is a convenience to indicate that any object that refers to the matched object should be returned, along with any other objects referenced by those referring objects. |
| Children | com:EmptyType | Children is a convenience to indicate that all object referred to by the matched object should be returned. |
| Descendants | com:EmptyType | Descendants is a convenience to indicate that all object referred to by the matched object should be returned, along with any objects referenced by the referred objects, and so on. This is a deep resolution, where all outgoing references starting at the matched object are resolved. |
| SpecificObjects | com: MaintainableObjectTy peListType | SpecificObjects is used to enumerate specific types of object to be returned. Theses objects will either refer to or are referred by the matched object. Only the maintainable objects listed here will be returned. |

***AnnotableWhereType*:**AnnotableWhereType is an abstract base type for querying an annotable artefact.

Content:

Annotation?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |

**AnnotationWhereType:**AnnotationWhereType defines the structure for querying the details of an annotation.

Content:

Type?, Title?, Text?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Type | QueryStringType | Type is a parameter for matching the type field of an annotation. |
| Title | QueryStringType | Title is a parameter for matching the title field of an annotation. |
| Text | QueryTextType | Text is a parameter for matching the text field of an annotation. |

***IdentifiableWhereType*:**IdentifiableWhereType is an abstract base type that serves as the basis for any query for an identifiable object.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType*



Content:

Annotation?, URN?, ID?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |

***NameableWhereType*:**NameableWhereType is an abstract base type that serves as the basis for any query for a nameable object.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType*



Content:

Annotation?, URN?, ID?, Name?, Description?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |

***VersionableWhereType*:**VersionableQueryType is an abstract base type that serves as the basis for any query for a versionable object.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType*



Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |

***MaintainableWhereType*:**MaintainableQueryType is an abstract base type that serves as the basis for any query for a maintainable object.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |

***ItemSchemeWhereType*:**ItemSchemeQueryType is an abstract base type that serves as the basis for any query for an item scheme.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *ItemSchemeWhereType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?,*ItemWhere\**

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| *ItemWhere* | *ItemWhereType* | ItemWhere is an abstract substitution head to query for an item within a parent item where or an item scheme query. An item where query is implicitly an and-query - that is the conditions within the item where must all be met in order to return a match. If this is nested within another item where, the meaning is that the conditions specified must be matched within the hierarchy provided. |

***ItemWhereType*:**ItemQueryType is an abstract base type that serves as the basis for a query for an item within an item scheme query. A nested item where is provided to query for items nested within other items. The conditions within an item query are implied to be in an and-query. If an id and a child item where condition are supplied, then both conditions will have to met in order for the item query to return true. If, for instance, a query based on names in multiple languages is required, then multiple instances of the element utilizing this type should be used within an or-query container.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *ItemWhereType*



Content:

Annotation?, URN?, ID?, Name?, Description?, (Parent |*ItemWhere+*)?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Parent | *com: LocalItemReferenceTy pe* | Parent is used to query for an item where it declares the item referenced here as its parent. This is used for items that are not nested in a hierarchy. If child items are sought for an item that is contained in a nested hierarchy (e.g. a category) on can query directly for the parent category and request that the child items be returned by specifying cascadeMatchedItems in the detail field of the return details. |
| *ItemWhere* | *ItemWhereType* | ItemWhere is an abstract substitution head to query for an item within a parent item where or an item scheme query. An item where query is implicitly an and-query - that is the conditions within the item where must all be met in order to return a match. If this is nested within another item where, the meaning is that the conditions specified must be matched within the hierarchy provided. |

***StructureWhereType*:**StructureWhereType is an abstract base type that serves as the basis for a query for a structure object.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *StructureWhereType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, UsedConcept\*, UsedRepresentation\*, *ComponentListWhere\**,*ComponentWhere\**

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| UsedConcept | com: ConceptReferenceType | UsedConcept is used to query for a structure that uses the referenced concept as the basis of one of its components. |
| UsedRepresentation | *com: ItemSchemeReferenceB aseType* | UsedRepresentation is used to query for a structure that uses the referenced item scheme for the representation of one of its components. |
| *ComponentListWhere* | *ComponentListWhereTy pe* | ComponentListWhere is an abstract substitution head to query for a component list within a structure. A component list where query is implicitly an and-query - that is the conditions within the component list where must all be met in order to return a match. |
| *ComponentWhere* | *ComponentWhereType* | ComponentWhere is an abstract substitution head to query for a component within a component list or a structure. A component where query is implicitly an and-query - that is the conditions within the component where must all be met in order to return a match. |

***ComponentListWhereType*:**ComponentListWhereType is an abstract base type that serves as the basis for a query for a component list within a structure query. A list of component where children are provided to query for the list's child components. The conditions within a component list query are implied to be in an and-query. If an id and a child component where condition are supplied, then both conditions will have to met in order for the component list query to return true. If, for instance, a query based on names in multiple languages is required, then multiple instances of the element utilizing this type should be used within an or-query container.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentListWhereType*



Content:

Annotation?, URN?, ID?, *ComponentWhere\**

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| *ComponentWhere* | *ComponentWhereType* | ComponentWhere is an abstract substitution head to query for a component within a component list or a structure. A component where query is implicitly an and-query - that is the conditions within the component where must all be met in order to return a match. |

***ComponentWhereType*:**ComponentWhereType is an abstract base type that serves as the basis for a query for a component within a component list where or a structure query. A concept identity and a local representation condition are available to seek a component that utilizes a particular concept or representation scheme. The conditions within a component query are implied to be in an and-query. If an id and a concept identity condition are supplied, then both conditions will have to met in order for the component query to return true. If, for instance, a query based on names in multiple languages is required, then multiple instances of the element utilizing this type should be used within an or-query container.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType*



Content:

Annotation?, URN?, ID?, ConceptIdentity?, Enumeration?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| ConceptIdentity | com: ConceptReferenceType | ConceptIdentity is used to query for a structure component based on the concept from which it takes its semantic. |
| Enumeration | *com: ItemSchemeReferenceB aseType* | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |

***StructureUsageWhereType*:**StructureUsageWhereType is an abstract base type that serves as the basis for a query for a structure usage object.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *StructureUsageWhereType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, Structure?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| Structure | *com: StructureReferenceBa seType* | Structure is used to match the structure referenced by a structure usage object. Only structure usages which reference the supplied structure will be returned. |

**ConceptValueType:**ConceptValueType describes the structure of a query for the value of the underlying concept of a component. It provides a reference to a concept in a concept scheme via a URN and/or a complete set of reference fields, as well as a numeric, text, or un-typed value.

Content:

Concept, (NumericValue[1..2] | TextValue+ | TimeValue[1..2] | Value)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Concept | com: ConceptReferenceType |  |
| NumericValue | NumericValueType | NumericValue is used to query for a the value of a concept or component based on a numeric search. This is typically used when the value needs to be searched explicitly as a number, such as when data is sought with an observed value greater than some threshold. If only a simple match is required (i.e. the numeric value equals 'x') then the Value element can be used. |
| TextValue | QueryTextType | TextValue is used to query for the value of a concept or component based on textual parameters. The text value can be language specific (where parallel multi-lingual values are available) and is qualified with an operator indicating how the supplied text should be matched against the sought components. If only a simple equality check is necessary, regardless of language, the Value element can be used. |
| TimeValue | TimePeriodValueType | TimeValue is used to query for the value of a concept or component based on time parameters. This is typically used when the value needs to be treated explicitly as a time, for example when searching for data after a particular point in time. If only a simple equality check is necessary, the Value element can be used. |
| Value | SimpleValueType | Value is used to query for the value of a component. This should be used for concepts or components based on a simple value (e.g. a code or a simple string). This should be used when a simple equal/not equal operator is all that is necessary for matching the sought value. |

**CodeValueType:**CodeValueType is used to query for data or reference metadata where a component which uses the referenced codelist as its representation enumeration has the value provided. Note that this is only applicable when the value is a coded value, which is to say that it does not apply to a codelist which is specified as the representation or an identifiable object target in a metadata target.

Attributes:

value

Content:

Codelist

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| value | xs:string | The value attribute indicates the coded value that is to be queried for. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Codelist | com: CodelistReferenceTyp e | Codelist references the codelist for which the coded value is being sought. |

**SimpleValueType:**SimpleValueType describes the structure of a simple value query. A value is provided as the content in string format.

Derivation:

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



Attributes:

operator?

Content:

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| operator (default: equal) | com: SimpleOperatorType | The operator attribute indicates the operator to apply to the string value query. The options are equal and notEqual. |

**NumericValueType:**NumericValueType describes the structure of a numeric query. A value is provided as the content in decimal format.

Derivation:

xs:anySimpleType (restriction)   
   xs:decimal (extension)   
         NumericValueType



Attributes:

operator?

Content:

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| operator (default: equal) | com: OrderedOperatorType | The operator attribute indicates the operator to apply to the numeric value query, such as equal to or greater than. |

**QueryStringType:**QueryStringType defines the structure of a string query.

Derivation:

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



Attributes:

operator?

Content:

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| operator (default: equal) | com:TextOperatorType | The operator attribute indicates how the supplied value should be applied to the objects being searched in order to constitute a match. For example, a value of "EqualTo" means the value of the field being search should exactly match the value supplied. See the defining type for further details. |

**QueryIDType:**QueryIDType defines the structure of a query for an identifier.

Derivation:

xs:anySimpleType (restriction)   
   xs:string (restriction)   
         com:NestedIDType (restriction)   
               com:IDType (extension)   
                     QueryIDType



Attributes:

operator?

Content:

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| operator (default: equal) | com:TextOperatorType | The operator attribute indicates how the supplied value should be applied to the objects being searched in order to constitute a match. For example, a value of "EqualTo" means the value of the field being search should exactly match the value supplied. See the defining type for further details. |

**QueryNestedIDType:**QueryNestedIDType defines the structure of a query for a nested identifier.

Derivation:

xs:anySimpleType (restriction)   
   xs:string (restriction)   
         com:NestedIDType (extension)   
               QueryNestedIDType



Attributes:

operator?

Content:

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| operator (default: equal) | com:TextOperatorType | The operator attribute indicates how the supplied value should be applied to the objects being searched in order to constitute a match. For example, a value of "EqualTo" means the value of the field being search should exactly match the value supplied. See the defining type for further details. |

**QueryTextType:**QueryTextType describes the structure of a textual query value. A language must be specified if parallel multi-lingual values are available, otherwise it is ignored.

Derivation:

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



Attributes:

xml:lang?, operator?

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. |
| operator (default: equal) | com:TextOperatorType | The operator attribute indicates how the supplied value should be applied to the objects being searched in order to constitute a match. For example, a value of "EqualTo" means the value of the field being search should exactly match the value supplied. See the defining type for further details. |

**TimePeriodValueType:**TimePeriodValueType describes the structure of a time period query. A value is provided as the content in the SDMX time period format.

Derivation:

xs:anySimpleType (restriction)   
   com:ObservationalTimePeriodType (extension)   
         TimePeriodValueType



Attributes:

operator?, reportingYearStartDay?

Content:

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| operator (default: equal) | com: TimeOperatorType | The operator attribute indicates the operator to apply to the value query, such as equal to or greater than. |
| reportingYearStartDay (default: Any) | ReportingYearStartDa yQueryType | The reportingYearStartDay attribute allows a reporting year start day to be specified for the reporting period time value. If this time value provided is not a report period, this value can be ignored. If an explicit value is provided, this will effectively turn the time parameter into a distinct time range. For example if the time parameter value is "2010-Q1" and this attribute has a value of "--04-01", the parameter will be treated as "2010-04-01/2010-06-30". If a value of "Any" is provided, then data will be matched regardless of its reporting year start day. For example, a query of 2011-A1 would return all data that belongs to a reporting year of 2011, regardless of the start day of the reporting year. For the puroses of matching data reporting against a Gregorian period against a time parameter value that is a reporting period, a value of "Any" will be treated as a start day of January 1. Therefore, if the time paramter value was 2011-A1, data reported against 2011 would be matched but data reporting against '2011-06/P1Y' would not be matched. |

**CategorisationQueryType:**CategorisationQueryType defines the structure of a categorisation query. The parameters for the query are contained in the CategorisationWhere element. The References element is used to indicate how objects that are referenced from the matched categorisations should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   CategorisationQueryType



Content:

ReturnDetails, CategorisationWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| CategorisationWhere | CategorisationWhereT ype | CategorisationWhere defines the parameters for a categorisation query. All parameters must be matched for an object to satisfy the query.In addition to querying based on the basic maintainable properties, it is also possible to search based on the source object being categorized and target category the object is categorized against. |

***CategorisationWhereBaseType*:**CategorisationWhereBaseType is an abstract base type which forms the basis for the CategorisationWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *CategorisationWhereBaseType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: Categorisation) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |

**CategorisationWhereType:**CategorisationWhereType contains a set of parameters for a categorisation query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *CategorisationWhereBaseType* (extension)   
                                 CategorisationWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, CategoryScheme?, TargetCategory?, ObjectReference?, CategorisedObjectType\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: Categorisation) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| CategoryScheme | com: CategorySchemeRefere nceType | CategoryScheme references a category scheme for which categorisations or sought for any of the scheme's categories. Any categorisation which has a target of a category defined in the scheme will be returned. |
| TargetCategory | com: CategoryReferenceTyp e | TargetCategory references the category that defines the target of the categorisation (the category which an object is categorized against). Only categorisations which refence the category supplied here will be returned. |
| ObjectReference | com: ObjectReferenceType | ObjectReference references the object that is the source of the categorisation (the object which is categorized). Only categorisations which reference the object supplied here will be returned. |
| CategorisedObjectTyp e | com: ObjectTypeCodelistTy pe | CategorisedObjectType is used to specify the type of objects that are categorised by the categorisations being sought. For example, this could be used to find an caategorisation which classifies a code list. |

**CategorySchemeQueryType:**CategorySchemeQueryType defines the structure of a category scheme query. The parameters for the query are contained in the CategorySchemeWhere element. The References element is used to indicate how objects that reference the matched category scheme should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   CategorySchemeQueryType



Content:

ReturnDetails, CategorySchemeWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | StructureReturnDetai lsType |  |
| CategorySchemeWhere | CategorySchemeWhereT ype | CategorySchemeWhere contains the parameters for a category scheme query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a category scheme based on the details of its categories. In any case, the category scheme will be returned according the indicated return detail. |

**CategorySchemeWhereType:**CategorySchemeWhereType defines the parameters of a category scheme query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *ItemSchemeWhereType* (restriction)   
                                 CategorySchemeWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, CategoryWhere\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: CategoryScheme) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| CategoryWhere | CategoryWhereType | CategoryWhere is used to query for categories matching the parameters supplied. It allows for nested category queries so that hierarchical categories can be queried explicitly by their nested level, although a top level category will always result in a search for categories at any level. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match. |

**CategoryWhereType:**CategoryQueryWhereType contains a set of parameters for matching a category. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *ItemWhereType* (restriction)   
                     CategoryWhereType



Content:

Annotation?, ID?, Name?, Description?, CategoryWhere\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| CategoryWhere | CategoryWhereType | CategoryWhere is used to query for categories matching the parameters supplied. It allows for nested category queries so that hierarchical categories can be queried explicitly by their nested level, although a top level category will always result in a search for categories at any level. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match. |

**CodelistQueryType:**CodelistQueryType defines the structure of a codelist query. The parameters for the query are contained in the CodelistWhere element. The References element is used to indicate how objects that reference the matched codelist should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   CodelistQueryType



Content:

ReturnDetails, CodelistWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | StructureReturnDetai lsType |  |
| CodelistWhere | CodelistWhereType | CodelistWhere defines the parameters for a codelist query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a codlist based on the details of its codes. In any case, the codelist will be returned according the indicated return detail. |

**CodelistWhereType:**CodelistWhereType contains the parameters of a codelist query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *ItemSchemeWhereType* (restriction)   
                                 CodelistWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, CodeWhere\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: Codelist) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| CodeWhere | CodeWhereType | CodeWhere is used to query for codes matching the parameters supplied. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match. |

**CodeWhereType:**CodeWhereType defines a set of parameters for matching a code. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *ItemWhereType* (restriction)   
                     CodeWhereType



Content:

Annotation?, ID?, Name?, Description?, Parent?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Parent | com: LocalCodeReferenceTy pe | Parent is used to query for an item where it declares the item referenced here as its parent. This is used for items that are not nested in a hierarchy. If child items are sought for an item that is contained in a nested hierarchy (e.g. a category) on can query directly for the parent category and request that the child items be returned by specifying cascadeMatchedItems in the detail field of the return details. |

**ConceptSchemeQueryType:**ConceptSchemeQueryType defines the structure of a category scheme query. The parameters for the query are contained in the ConceptSchemeWhere element. The References element is used to indicate how objects that reference or are referenced from the matched concept scheme should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   ConceptSchemeQueryType



Content:

ReturnDetails, ConceptSchemeWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | StructureReturnDetai lsType |  |
| ConceptSchemeWhere | ConceptSchemeWhereTy pe | ConceptSchemeWhere defines the parameters for a concept scheme query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a concept scheme based on the details of its concepts. In any case, the concept scheme will be returned according the indicated return detail. |

**ConceptSchemeWhereType:**ConceptSchemeWhereType contains the parameters of a concept scheme query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *ItemSchemeWhereType* (restriction)   
                                 ConceptSchemeWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, ConceptWhere\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: ConceptScheme) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| ConceptWhere | ConceptWhereType | ConceptWhere is used to query for concepts matching the parameters supplied. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match. |

***ConceptWhereBaseType*:**ConceptWhereBaseType is an abstract base type that forms the basis for the ConceptWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *ItemWhereType* (restriction)   
                     *ConceptWhereBaseType*



Content:

Annotation?, ID?, Name?, Description?, Parent?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Parent | com: LocalConceptReferenc eType | Parent is used to query for an item where it declares the item referenced here as its parent. This is used for items that are not nested in a hierarchy. If child items are sought for an item that is contained in a nested hierarchy (e.g. a category) on can query directly for the parent category and request that the child items be returned by specifying cascadeMatchedItems in the detail field of the return details. |

**ConceptWhereType:**ConceptWhereType defines a set of parameters for matching a category. All supplied parameters must be matched in order for an object to satisfy the query. In addition to the base parameters for an item, a concept can be queried based on the item scheme that is used as a core representation.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *ItemWhereType* (restriction)   
                     *ConceptWhereBaseType* (extension)   
                           ConceptWhereType



Content:

Annotation?, ID?, Name?, Description?, Parent?, Enumeration?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Parent | com: LocalConceptReferenc eType | Parent is used to query for an item where it declares the item referenced here as its parent. This is used for items that are not nested in a hierarchy. If child items are sought for an item that is contained in a nested hierarchy (e.g. a category) on can query directly for the parent category and request that the child items be returned by specifying cascadeMatchedItems in the detail field of the return details. |
| Enumeration | com: CodelistReferenceTyp e | Enumeration is used to query for a concept based on the codelist that is uses as the enumeration for its core representation. |

**ConstraintQueryType:**ConstraintQueryType defines the structure of a constraint query. The parameters for the query are contained in the ConstraintWhere element. The References element is used to indicate how objects that are referenced from the matched constraint should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   ConstraintQueryType



Content:

ReturnDetails, ConstraintWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| ConstraintWhere | ConstraintWhereType | AttachmentConstraintWhere contains the parameters for a constraint query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a constraint based on the objects it applies to. |

***ConstraintWhereBaseType*:**ConstraintWhereBaseType is an abstract base type which forms the basis for the ConstraintWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *ConstraintWhereBaseType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (default: Constraint) | ConstraintTypeCodeli stType | The type attribute indicates the type of constraint that is being queried for, with a default of Any, meaning both content and attachment constraints will be searched. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |

**ConstraintWhereType:**ConstraintWhereType contains the parameters of a constraint query. All supplied parameters must be matched in order for an object to stratify the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *ConstraintWhereBaseType* (extension)   
                                 ConstraintWhereType



Attributes:

type?, allowable?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, ConstraintAttachmentWhere?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (default: Constraint) | ConstraintTypeCodeli stType | The type attribute indicates the type of constraint that is being queried for, with a default of Any, meaning both content and attachment constraints will be searched. |
| allowable | xs:boolean | The allowable attribute indicates whether the returned search should be limited to only allowable constraints. This only applies to content constraint. If this attribute has a value of true, this indicates that only matching allowable content constraints should be returned. If this is false, than only actual content constraints returned. If no value is provided, all matching content constraints will be returned, regardless of whether they are stating actual or allowable content. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| ConstraintAttachment Where | ConstraintAttachment WhereType | ConstraintAttachmentWhere identifies a collection of objects to which a constraint may be attached. This container is an implicit and-query, meaning all of the objects referenced in here must be part of the constraint attachment in order for a constraint to match. |

**ConstraintAttachmentWhereType:**ConstraintAttachmentWhereType describes the structure for querying for a constraint based on the objects it is attached to. A constraint attachment query is implicitly an and-query meaning all of the referenced objects must be part of a constraints attachment in order to return a match. It is treated as a condition within its parent query.

Content:

(DataProvider | Dataflow | DataStructure | Metadataflow | MetadataStructure | ProvisionAgreement | DataSet | MetadataSet | DataSourceURL)+

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| DataProvider | com: DataProviderReferenc eType | DataProviderReference references a data provider to which a constraint is attached. It is referenced via a URN and/or a full set of reference fields. If a constraint is attached to the data provider referenced, a match will be returned. |
| Dataflow | com: DataflowReferenceTyp e | DataflowReference references a data flow to which a constraint is attached. It is referenced via a URN and/or a full set of reference fields. If a constraint is attached to the data flow referenced, a match will be returned. |
| DataStructure | com: DataStructureReferen ceType | DataStructureReference references a data structure definition to which a constraint is attached. It is referenced via a URN and/or a full set of reference fields. If a constraint is attached to the data structure definition referenced, a match will be returned. |
| Metadataflow | com: MetadataflowReferenc eType | MetadataflowReference references a metadata flow to which a constraint is attached. It is referenced via a URN and/or a full set of reference fields. If a constraint is attached to the metadata flow referenced, a match will be returned. |
| MetadataStructure | com: MetadataStructureRef erenceType | MetadataStructureReference references a metadata structure definition to which a constraint is attached. It is referenced via a URN and/or a full set of reference fields. If a constraint is attached to the metadata structure definition referenced, a match will be returned. |
| ProvisionAgreement | com: ProvisionAgreementRe ferenceType | ProvisionAgreementReference references a provision agreement to which a constraint is attached. It is referenced via a URN and/or a full set of reference fields. If a constraint is attached to the provision agreement referenced, a match will be returned. |
| DataSet | com:SetReferenceType | DataSetReference references a data set to which a constraint is attached. If a constraint is attached to the data set referenced, a match will be returned. |
| MetadataSet | com:SetReferenceType | MetadataSetReference references a reference metadata set to which a constraint is attached. If a constraint is attached to the metadata set referenced, a match will be returned. |
| DataSourceURL | xs:anyURI | DataSourceURL references a queryable data source to which a constraint it attached. The data source is referenced by its data URL. If a constraint is attached to the data source described (by matching the data URL), a match will be returned. |

**DataQueryType:**DataQueryType defines the structure of a query for data. This is generally appliable for any data request, but can be refined depending on the type of data being queried (generic or structured, time series specific or not) to the requirements of the requested format.

Content:

ReturnDetails, DataWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | DataReturnDetailsTyp e | ReturnDetails contains the details of how the returned data should be structured, what type of data (e.g. active or deleted observations), and the limit of the amount of observations to be returned. |
| DataWhere | DataParametersAndTyp e | DataWhere contains the details of the data query. |

**TimeSeriesDataQueryType:**TimeSeriesDataQueryType defines the structure of a query for data. This specifically applies to requesting time series only structured data.

Derivation:

DataQueryType (restriction)   
   TimeSeriesDataQueryType



Content:

ReturnDetails, DataWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | TimeSeriesDataReturn DetailsType | ReturnDetails contains the details of how the returned data should be structured, what type of data (e.g. active or deleted observations), and the limit of the amount of observations to be returned. |
| DataWhere | DataParametersAndTyp e | DataWhere contains the details of the data query. |

**GenericDataQueryType:**GenericDataQueryType defines the structure of a query for data formatted in the generic format. This structure generally applies to any type of generic data request, but can be refined to request time series only data.

Derivation:

DataQueryType (restriction)   
   GenericDataQueryType



Content:

ReturnDetails, DataWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | GenericDataReturnDet ailsType | ReturnDetails contains the details of how the returned data should be structured, what type of data (e.g. active or deleted observations), and the limit of the amount of observations to be returned. |
| DataWhere | DataParametersAndTyp e | DataWhere contains the details of the data query. |

**GenericTimeSeriesDataQueryType:**GenericTimeSeriesDataQueryType defines the structure of a query for time series only data formatted in the generic format.

Derivation:

DataQueryType (restriction)   
   GenericDataQueryType (restriction)   
         GenericTimeSeriesDataQueryType



Content:

ReturnDetails, DataWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | GenericTimeSeriesDat aReturnDetailsType | ReturnDetails contains the details of how the returned data should be structured, what type of data (e.g. active or deleted observations), and the limit of the amount of observations to be returned. |
| DataWhere | DataParametersAndTyp e | DataWhere contains the details of the data query. |

***DataReturnDetailsBaseType*:**DataReturnDetailsBaseType is an abstract base type which forms the basis of the DataReturnDetailsType.

Derivation:

*ReturnDetailsBaseType* (restriction)   
   *DataReturnDetailsBaseType*



Attributes:

defaultLimit?, detail?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| defaultLimit | xs:integer | The defaultLimit attribute is the suggested maximum response size in kilobytes. |
| detail (default: Full) | DataReturnDetailType | The detail attribute is used to indicate whether the response to the query should return the full details of matched data sets, or just a subset of the information should be returned. A value of "Full" indicates that the complete data set (including data and documentation) will be returned. A value of "DataOnly" indicates that only the observation values and keys should be returned. A value of "SeriesKeyOnly" indicates that only the Series elements and their keys (i.e. Dimension values) should be returned. A value of "NoData" indicates that only documentation should be returned (i.e. the DataSet, Group, and Series level Attributes). |

**DataReturnDetailsType:**DataReturnDetailsType specifies the specifics of the how data should be returned, including how it should be structured and how many and what type (e.g. active or deleted) observations should be returned.

Derivation:

*ReturnDetailsBaseType* (restriction)   
   *DataReturnDetailsBaseType* (extension)   
         DataReturnDetailsType



Attributes:

defaultLimit?, detail?, observationAction?

Content:

FirstNObservations?, LastNObservations?, Structure\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| defaultLimit | xs:integer | The defaultLimit attribute is the suggested maximum response size in kilobytes. |
| detail (default: Full) | DataReturnDetailType | The detail attribute is used to indicate whether the response to the query should return the full details of matched data sets, or just a subset of the information should be returned. A value of "Full" indicates that the complete data set (including data and documentation) will be returned. A value of "DataOnly" indicates that only the observation values and keys should be returned. A value of "SeriesKeyOnly" indicates that only the Series elements and their keys (i.e. Dimension values) should be returned. A value of "NoData" indicates that only documentation should be returned (i.e. the DataSet, Group, and Series level Attributes). |
| observationAction (default: Active) | ObservationActionCod eType | The observationAction attribute specifies the type of observations (added, deleted, udpated, or current) to be returned. In the absence of the Updated parameter, this will return all observations that have ever existed for Added, any observations that have ever been updated for Updated, and any observations that have ever been deleted for Deleted. Note that since observations themselves contain no status in the data messages, it is only possible to query for active observations or deleted observations, but not both. It is possible to subset active observation into recently added and recently updated, however it is only possible to retrieve on or the other. If active observations are returned, there will be no distinction as to whether an individual observation was updated or added during the requested Update parameter. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| FirstNObservations | xs:int | FirstNObservations specifies that number of observations indicated should be returned, starting from the earliest observation. Note that this can be used in conjunction with the LastNObservations. For example, if both FirstNObservations and LastNObservations had a value of 1, then the first and the last observation would be returned. |
| LastNObservations | xs:int | LastNObservations specifies that number of observations indicated should be returned, starting from the latest observation and working back. Note that this can be used in conjunction with the FirstNObservations. For example, if both FirstNObservations and LastNObservations had a value of 1, then the first and the last observation would be returned. |
| Structure | com: DataStructureRequest Type | Structure defines how the is requested to be oriented in the returned message. For each data structure, dataflow, or provision agreement, a dimension at the observation level can be specified, and in the case that the dimension is a measure and the query is for structured data, the measure can be specified as being explicit. For any matched data in which there is not a structure specification, the query service can orient that data in any manner, although it is recommended that time be used as the observation dimension. |

**GenericDataReturnDetailsType:**GenericDataReturnDetailsType specifies the specifics of the how data should be returned as it pertains to a request for generic data.

Derivation:

*ReturnDetailsBaseType* (restriction)   
   *DataReturnDetailsBaseType* (extension)   
         DataReturnDetailsType (restriction)   
               GenericDataReturnDetailsType



Attributes:

defaultLimit?, detail?, observationAction?

Content:

FirstNObservations?, LastNObservations?, Structure\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| defaultLimit | xs:integer | The defaultLimit attribute is the suggested maximum response size in kilobytes. |
| detail (default: Full) | DataReturnDetailType | The detail attribute is used to indicate whether the response to the query should return the full details of matched data sets, or just a subset of the information should be returned. A value of "Full" indicates that the complete data set (including data and documentation) will be returned. A value of "DataOnly" indicates that only the observation values and keys should be returned. A value of "SeriesKeyOnly" indicates that only the Series elements and their keys (i.e. Dimension values) should be returned. A value of "NoData" indicates that only documentation should be returned (i.e. the DataSet, Group, and Series level Attributes). |
| observationAction (default: Active) | ObservationActionCod eType | The observationAction attribute specifies the type of observations (added, deleted, udpated, or current) to be returned. In the absence of the Updated parameter, this will return all observations that have ever existed for Added, any observations that have ever been updated for Updated, and any observations that have ever been deleted for Deleted. Note that since observations themselves contain no status in the data messages, it is only possible to query for active observations or deleted observations, but not both. It is possible to subset active observation into recently added and recently updated, however it is only possible to retrieve on or the other. If active observations are returned, there will be no distinction as to whether an individual observation was updated or added during the requested Update parameter. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| FirstNObservations | xs:int | FirstNObservations specifies that number of observations indicated should be returned, starting from the earliest observation. Note that this can be used in conjunction with the LastNObservations. For example, if both FirstNObservations and LastNObservations had a value of 1, then the first and the last observation would be returned. |
| LastNObservations | xs:int | LastNObservations specifies that number of observations indicated should be returned, starting from the latest observation and working back. Note that this can be used in conjunction with the FirstNObservations. For example, if both FirstNObservations and LastNObservations had a value of 1, then the first and the last observation would be returned. |
| Structure | com: GenericDataStructure RequestType | Structure defines how the is requested to be oriented in the returned message. For each data structure, dataflow, or provision agreement, a dimension at the observation level can be specified, and in the case that the dimension is a measure and the query is for structured data, the measure can be specified as being explicit. For any matched data in which there is not a structure specification, the query service can orient that data in any manner, although it is recommended that time be used as the observation dimension. |

**GenericTimeSeriesDataReturnDetailsType:**GenericTimeSeriesDataReturnDetailsType specifies the specifics of the how data should be returned as it pertains to a request for time series only oriented data in the generic format.

Derivation:

*ReturnDetailsBaseType* (restriction)   
   *DataReturnDetailsBaseType* (extension)   
         DataReturnDetailsType (restriction)   
               GenericDataReturnDetailsType (restriction)   
                     GenericTimeSeriesDataReturnDetailsType



Attributes:

defaultLimit?, detail?, observationAction?

Content:

FirstNObservations?, LastNObservations?, Structure\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| defaultLimit | xs:integer | The defaultLimit attribute is the suggested maximum response size in kilobytes. |
| detail (default: Full) | DataReturnDetailType | The detail attribute is used to indicate whether the response to the query should return the full details of matched data sets, or just a subset of the information should be returned. A value of "Full" indicates that the complete data set (including data and documentation) will be returned. A value of "DataOnly" indicates that only the observation values and keys should be returned. A value of "SeriesKeyOnly" indicates that only the Series elements and their keys (i.e. Dimension values) should be returned. A value of "NoData" indicates that only documentation should be returned (i.e. the DataSet, Group, and Series level Attributes). |
| observationAction (default: Active) | ObservationActionCod eType | The observationAction attribute specifies the type of observations (added, deleted, udpated, or current) to be returned. In the absence of the Updated parameter, this will return all observations that have ever existed for Added, any observations that have ever been updated for Updated, and any observations that have ever been deleted for Deleted. Note that since observations themselves contain no status in the data messages, it is only possible to query for active observations or deleted observations, but not both. It is possible to subset active observation into recently added and recently updated, however it is only possible to retrieve on or the other. If active observations are returned, there will be no distinction as to whether an individual observation was updated or added during the requested Update parameter. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| FirstNObservations | xs:int | FirstNObservations specifies that number of observations indicated should be returned, starting from the earliest observation. Note that this can be used in conjunction with the LastNObservations. For example, if both FirstNObservations and LastNObservations had a value of 1, then the first and the last observation would be returned. |
| LastNObservations | xs:int | LastNObservations specifies that number of observations indicated should be returned, starting from the latest observation and working back. Note that this can be used in conjunction with the FirstNObservations. For example, if both FirstNObservations and LastNObservations had a value of 1, then the first and the last observation would be returned. |
| Structure | com: TimeSeriesGenericDat aStructureRequestTyp e | Structure defines how the is requested to be oriented in the returned message. For each data structure, dataflow, or provision agreement, a dimension at the observation level can be specified, and in the case that the dimension is a measure and the query is for structured data, the measure can be specified as being explicit. For any matched data in which there is not a structure specification, the query service can orient that data in any manner, although it is recommended that time be used as the observation dimension. |

**TimeSeriesDataReturnDetailsType:**TimeSeriesDataReturnDetailsType specifies the specifics of the how data should be returned as it pertains to a request for time series only oriented data in the structured format.

Derivation:

*ReturnDetailsBaseType* (restriction)   
   *DataReturnDetailsBaseType* (extension)   
         DataReturnDetailsType (restriction)   
               TimeSeriesDataReturnDetailsType



Attributes:

defaultLimit?, detail?, observationAction?

Content:

FirstNObservations?, LastNObservations?, Structure\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| defaultLimit | xs:integer | The defaultLimit attribute is the suggested maximum response size in kilobytes. |
| detail (default: Full) | DataReturnDetailType | The detail attribute is used to indicate whether the response to the query should return the full details of matched data sets, or just a subset of the information should be returned. A value of "Full" indicates that the complete data set (including data and documentation) will be returned. A value of "DataOnly" indicates that only the observation values and keys should be returned. A value of "SeriesKeyOnly" indicates that only the Series elements and their keys (i.e. Dimension values) should be returned. A value of "NoData" indicates that only documentation should be returned (i.e. the DataSet, Group, and Series level Attributes). |
| observationAction (default: Active) | ObservationActionCod eType | The observationAction attribute specifies the type of observations (added, deleted, udpated, or current) to be returned. In the absence of the Updated parameter, this will return all observations that have ever existed for Added, any observations that have ever been updated for Updated, and any observations that have ever been deleted for Deleted. Note that since observations themselves contain no status in the data messages, it is only possible to query for active observations or deleted observations, but not both. It is possible to subset active observation into recently added and recently updated, however it is only possible to retrieve on or the other. If active observations are returned, there will be no distinction as to whether an individual observation was updated or added during the requested Update parameter. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| FirstNObservations | xs:int | FirstNObservations specifies that number of observations indicated should be returned, starting from the earliest observation. Note that this can be used in conjunction with the LastNObservations. For example, if both FirstNObservations and LastNObservations had a value of 1, then the first and the last observation would be returned. |
| LastNObservations | xs:int | LastNObservations specifies that number of observations indicated should be returned, starting from the latest observation and working back. Note that this can be used in conjunction with the FirstNObservations. For example, if both FirstNObservations and LastNObservations had a value of 1, then the first and the last observation would be returned. |
| Structure | com: TimeSeriesDataStruct ureRequestType | Structure defines how the is requested to be oriented in the returned message. For each data structure, dataflow, or provision agreement, a dimension at the observation level can be specified, and in the case that the dimension is a measure and the query is for structured data, the measure can be specified as being explicit. For any matched data in which there is not a structure specification, the query service can orient that data in any manner, although it is recommended that time be used as the observation dimension. |

***DataParametersType*:**DataParametersType defines the parameters for querying for data. This structure is refined by separate And/Or constructs which make logical restrictions on which parameters apply in such cases.

Content:

DataSetID\*, DataProvider\*, DataStructure\*, Dataflow\*, ProvisionAgreement\*, Category\*, Updated[0..2], ConceptValue\*, RepresentationValue\*, DimensionValue\*, TimeDimensionValue\*, AttributeValue\*, PrimaryMeasureValue\*, AttachmentConstraint\*, TimeFormat\*, Or\*, And\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| DataSetID | QueryIDType | DataSetID is used to match the id of the data set. Only data from data sets with an identifier satisfying these conditions will be matched. |
| DataProvider | com: DataProviderReferenc eType | DataProvider is used to match the provider of data to the referenced data provider. Only data from data sets provided by the referenced data provider will be matched. |
| DataStructure | com: DataStructureReferen ceType | DataStructure is used to match the underlying structure of the data. Only data from data sets that conform to referenced data structure definition will be matched. |
| Dataflow | com: DataflowReferenceTyp e | Dataflow is used to match the flow which data is reported against. Only data from data sets report against referenced dataflow will be matched. |
| ProvisionAgreement | com: ProvisionAgreementRe ferenceType | ProvisionAgreement is used to match the provision agreement which data is reported against. Only data from data sets report against the referenced provision agreement will be matched. |
| Category | com: CategoryReferenceTyp e | Category is used to match a data based on the categorization of its underlying structure (data structure definition), or the usage of that structure (data flow). Only data whose underlying structure or structure usage are categorized against the referenced category will be matched. |
| Updated | com: TimeRangeValueType | Updated is used to match data based on when it was last updated (including additions and deletions). Only data which satisfies the conditions for the last update parameters supplied here will be matched. |
| ConceptValue | ConceptValueType | ConceptValue is used to match data based on the value of a particular concept. This concept may be used as a dimension, attribute, or measure for the data. So long as the referenced concept has the specified value for a given data point, it will be matched. |
| RepresentationValue | CodeValueType | RepresentationValue is used to match data based on a representation scheme having a particular value. This representation scheme may be used as the representation of a dimension, attribute, or measure. So long as the value of the concept using the referenced codelist has the value specified, any data point for the concept will be matched. |
| DimensionValue | DimensionValueType | DimensionValue is used to match data based on the value of a dimension. Any data with the dimension with the supplied identifier satisfies the conditions supplied will be matched. |
| TimeDimensionValue | TimeDimensionValueTy pe | TimeDimensionValue is used to match data based on the value of the time dimension. Any data with a time value satisfying the conditions supplied will be matched. |
| AttributeValue | AttributeValueType | AttributeValue is used to match data based on the value of an attribute. Any data with an attribute with the supplied identifier satisfies the conditions supplied will be matched. |
| PrimaryMeasureValue | PrimaryMeasureValueT ype | PrimaryMeasureValue is used to match data based on the value of the primary measure. Any data with its value satisfying the conditions supplied will be matched. |
| AttachmentConstraint | com: AttachmentConstraint ReferenceType | AttachmentConstraint references an attachment constraint in order to match data which matches the effective data keys or cube regions defined in the constraint. Data will be returned by first matching data on the keys and cube regions that are marked as included (or all data if none), and then excluding the data that satisfies the conditions of the excluded keys and cube regions. |
| TimeFormat | com:TimeDataType | TimeFormat is used to match data when a frequency dimension is not explicitly defined. Only data reported against the supplied time data type will be returned. |
| Or | DataParametersOrType | Or contains a collection of additional parameters, any one of which can be satisfied to result in a match. |
| And | DataParametersAndTyp e | And contains a collection of additional parameters, all of which must be satisfied to result in a match. |

**DataParametersOrType:**DataParametersOrType refines the base data parameters to define a set of parameters joined by an "or" condition. Only one of the parameters supplied in an instance of this type can be satisfied to result in a match.

Derivation:

*DataParametersType* (restriction)   
   DataParametersOrType



Content:

DataSetID\*, DataProvider\*, DataStructure\*, Dataflow\*, ProvisionAgreement\*, Category\*, Updated[0..2], ConceptValue\*, RepresentationValue\*, DimensionValue\*, TimeDimensionValue\*, AttributeValue\*, PrimaryMeasureValue\*, AttachmentConstraint\*, TimeFormat\*, And\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| DataSetID | QueryIDType | DataSetID is used to match the id of the data set. Only data from data sets with an identifier satisfying these conditions will be matched. |
| DataProvider | com: DataProviderReferenc eType | DataProvider is used to match the provider of data to the referenced data provider. Only data from data sets provided by the referenced data provider will be matched. |
| DataStructure | com: DataStructureReferen ceType | DataStructure is used to match the underlying structure of the data. Only data from data sets that conform to referenced data structure definition will be matched. |
| Dataflow | com: DataflowReferenceTyp e | Dataflow is used to match the flow which data is reported against. Only data from data sets report against referenced dataflow will be matched. |
| ProvisionAgreement | com: ProvisionAgreementRe ferenceType | ProvisionAgreement is used to match the provision agreement which data is reported against. Only data from data sets report against the referenced provision agreement will be matched. |
| Category | com: CategoryReferenceTyp e | Category is used to match a data based on the categorization of its underlying structure (data structure definition), or the usage of that structure (data flow). Only data whose underlying structure or structure usage are categorized against the referenced category will be matched. |
| Updated | com: TimeRangeValueType | Updated is used to match data based on when it was last updated (including additions and deletions). Only data which satisfies the conditions for the last update parameters supplied here will be matched. |
| ConceptValue | ConceptValueType | ConceptValue is used to match data based on the value of a particular concept. This concept may be used as a dimension, attribute, or measure for the data. So long as the referenced concept has the specified value for a given data point, it will be matched. |
| RepresentationValue | CodeValueType | RepresentationValue is used to match data based on a representation scheme having a particular value. This representation scheme may be used as the representation of a dimension, attribute, or measure. So long as the value of the concept using the referenced codelist has the value specified, any data point for the concept will be matched. |
| DimensionValue | DimensionValueType | DimensionValue is used to match data based on the value of a dimension. Any data with the dimension with the supplied identifier satisfies the conditions supplied will be matched. |
| TimeDimensionValue | TimeDimensionValueTy pe | TimeDimensionValue is used to match data based on the value of the time dimension. Any data with a time value satisfying the conditions supplied will be matched. |
| AttributeValue | AttributeValueType | AttributeValue is used to match data based on the value of an attribute. Any data with an attribute with the supplied identifier satisfies the conditions supplied will be matched. |
| PrimaryMeasureValue | PrimaryMeasureValueT ype | PrimaryMeasureValue is used to match data based on the value of the primary measure. Any data with its value satisfying the conditions supplied will be matched. |
| AttachmentConstraint | com: AttachmentConstraint ReferenceType | AttachmentConstraint references an attachment constraint in order to match data which matches the effective data keys or cube regions defined in the constraint. Data will be returned by first matching data on the keys and cube regions that are marked as included (or all data if none), and then excluding the data that satisfies the conditions of the excluded keys and cube regions. |
| TimeFormat | com:TimeDataType | TimeFormat is used to match data when a frequency dimension is not explicitly defined. Only data reported against the supplied time data type will be returned. |
| And | DataParametersAndTyp e | And contains a collection of additional parameters, all of which must be satisfied to result in a match. |

**DataParametersAndType:**DataParametersAndType refines the base data parameters to define a set of parameters joined by an "and" conditions. All of the parameters supplied in an instance of this type must be satisfied to result in a match. As a result of this condition, the maximum occurrence of some parameters has been reduced so as to not allow for impossible conditions to be specified (for example data cannot be matched is it is specified that the data set identifier should be "xyz" and the data identifier should be "abc".

Derivation:

*DataParametersType* (restriction)   
   DataParametersAndType



Content:

DataSetID?, DataProvider?, DataStructure?, Dataflow?, ProvisionAgreement?, Category\*, Updated[0..2], ConceptValue\*, RepresentationValue\*, DimensionValue\*, TimeDimensionValue?, AttributeValue\*, PrimaryMeasureValue?, AttachmentConstraint\*, TimeFormat?, Or\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| DataSetID | QueryIDType | DataSetID is used to match the id of the data set. Only data from data sets with an identifier satisfying these conditions will be matched. |
| DataProvider | com: DataProviderReferenc eType | DataProvider is used to match the provider of data to the referenced data provider. Only data from data sets provided by the referenced data provider will be matched. |
| DataStructure | com: DataStructureReferen ceType | DataStructure is used to match the underlying structure of the data. Only data from data sets that conform to referenced data structure definition will be matched. |
| Dataflow | com: DataflowReferenceTyp e | Dataflow is used to match the flow which data is reported against. Only data from data sets report against referenced dataflow will be matched. |
| ProvisionAgreement | com: ProvisionAgreementRe ferenceType | ProvisionAgreement is used to match the provision agreement which data is reported against. Only data from data sets report against the referenced provision agreement will be matched. |
| Category | com: CategoryReferenceTyp e | Category is used to match a data based on the categorization of its underlying structure (data structure definition), or the usage of that structure (data flow). Only data whose underlying structure or structure usage are categorized against the referenced category will be matched. |
| Updated | com: TimeRangeValueType | Updated is used to match data based on when it was last updated (including additions and deletions). Only data which satisfies the conditions for the last update parameters supplied here will be matched. |
| ConceptValue | ConceptValueType | ConceptValue is used to match data based on the value of a particular concept. This concept may be used as a dimension, attribute, or measure for the data. So long as the referenced concept has the specified value for a given data point, it will be matched. |
| RepresentationValue | CodeValueType | RepresentationValue is used to match data based on a representation scheme having a particular value. This representation scheme may be used as the representation of a dimension, attribute, or measure. So long as the value of the concept using the referenced codelist has the value specified, any data point for the concept will be matched. |
| DimensionValue | DimensionValueType | DimensionValue is used to match data based on the value of a dimension. Any data with the dimension with the supplied identifier satisfies the conditions supplied will be matched. |
| TimeDimensionValue | TimeDimensionValueTy pe | TimeDimensionValue is used to match data based on the value of the time dimension. Any data with a time value satisfying the conditions supplied will be matched. |
| AttributeValue | AttributeValueType | AttributeValue is used to match data based on the value of an attribute. Any data with an attribute with the supplied identifier satisfies the conditions supplied will be matched. |
| PrimaryMeasureValue | PrimaryMeasureValueT ype | PrimaryMeasureValue is used to match data based on the value of the primary measure. Any data with its value satisfying the conditions supplied will be matched. |
| AttachmentConstraint | com: AttachmentConstraint ReferenceType | AttachmentConstraint references an attachment constraint in order to match data which matches the effective data keys or cube regions defined in the constraint. Data will be returned by first matching data on the keys and cube regions that are marked as included (or all data if none), and then excluding the data that satisfies the conditions of the excluded keys and cube regions. |
| TimeFormat | com:TimeDataType | TimeFormat is used to match data when a frequency dimension is not explicitly defined. Only data reported against the supplied time data type will be returned. |
| Or | DataParametersOrType | Or contains a collection of additional parameters, any one of which can be satisfied to result in a match. |

***DataStructureComponentValueQueryType*:**DataStructureComponentValueQueryType is an abstract base type that is the basis for query a data structure definition component for a particular value. Multiple values may be supplied, but the component value must match all of the value conditions supplied in order to return a match.

Content:

ID?, (NumericValue[1..2] | TextValue+ | TimeValue[1..2] | Value)?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ID | com:NCNameIDType | ID provides the identifier for component for which the value is sought. |
| NumericValue | NumericValueType | NumericValue is used to query for a the value of a concept or component based on a numeric search. This is typically used when the value needs to be searched explicitly as a number, such as when data is sought with an observed value greater than some threshold. If only a simple match is required (i.e. the numeric value equals 'x') then the Value element can be used. |
| TextValue | QueryTextType | TextValue is used to query for the value of a concept or component based on textual parameters. The text value can be language specific (where parallel multi-lingual values are available) and is qualified with an operator indicating how the supplied text should be matched against the sought components. If only a simple equality check is necessary, regardless of language, the Value element can be used. |
| TimeValue | TimePeriodValueType | TimeValue is used to query for the value of a concept or component based on time parameters. This is typically used when the value needs to be treated explicitly as a time, for example when searching for data after a particular point in time. If only a simple equality check is necessary, the Value element can be used. |
| Value | SimpleValueType | Value is used to query for the value of a component. This should be used for concepts or components based on a simple value (e.g. a code or a simple string). This should be used when a simple equal/not equal operator is all that is necessary for matching the sought value. |

**DimensionValueType:**DimensionValueType is used to query for data where a given dimension has a particular value.

Derivation:

*DataStructureComponentValueQueryType* (restriction)   
   DimensionValueType



Content:

ID, (NumericValue[1..2] | TimeValue[1..2] | Value)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ID | com:NCNameIDType | ID provides the identifier for component for which the value is sought. |
| NumericValue | NumericValueType | NumericValue is used to query for a the value of a concept or component based on a numeric search. This is typically used when the value needs to be searched explicitly as a number, such as when data is sought with an observed value greater than some threshold. If only a simple match is required (i.e. the numeric value equals 'x') then the Value element can be used. |
| TimeValue | TimePeriodValueType | TimeValue is used to query for the value of a concept or component based on time parameters. This is typically used when the value needs to be treated explicitly as a time, for example when searching for data after a particular point in time. If only a simple equality check is necessary, the Value element can be used. |
| Value | SimpleValueType | Value is used to query for the value of a component. This should be used for concepts or components based on a simple value (e.g. a code or a simple string). This should be used when a simple equal/not equal operator is all that is necessary for matching the sought value. |

**AttributeValueType:**AttributeValueType is used to query for data where a given attribute has a particular value.

Derivation:

*DataStructureComponentValueQueryType* (restriction)   
   AttributeValueType



Content:

ID, (NumericValue[1..2] | TextValue | TimeValue[1..2] | Value)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ID | com:NCNameIDType | ID provides the identifier for component for which the value is sought. |
| NumericValue | NumericValueType | NumericValue is used to query for a the value of a concept or component based on a numeric search. This is typically used when the value needs to be searched explicitly as a number, such as when data is sought with an observed value greater than some threshold. If only a simple match is required (i.e. the numeric value equals 'x') then the Value element can be used. |
| TextValue | QueryTextType | TextValue is used to query for the value of a concept or component based on textual parameters. The text value can be language specific (where parallel multi-lingual values are available) and is qualified with an operator indicating how the supplied text should be matched against the sought components. If only a simple equality check is necessary, regardless of language, the Value element can be used. |
| TimeValue | TimePeriodValueType | TimeValue is used to query for the value of a concept or component based on time parameters. This is typically used when the value needs to be treated explicitly as a time, for example when searching for data after a particular point in time. If only a simple equality check is necessary, the Value element can be used. |
| Value | SimpleValueType | Value is used to query for the value of a component. This should be used for concepts or components based on a simple value (e.g. a code or a simple string). This should be used when a simple equal/not equal operator is all that is necessary for matching the sought value. |

**PrimaryMeasureValueType:**PrimaryMeasureValueType is used to query for data where the primary measure (i.e. the observed value) has a particular value.

Derivation:

*DataStructureComponentValueQueryType* (restriction)   
   PrimaryMeasureValueType



Content:

ID?, (NumericValue[1..2] | TextValue | TimeValue[1..2] | Value)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ID | com:NCNameIDType | ID provides the identifier for component for which the value is sought. |
| NumericValue | NumericValueType | NumericValue is used to query for a the value of a concept or component based on a numeric search. This is typically used when the value needs to be searched explicitly as a number, such as when data is sought with an observed value greater than some threshold. If only a simple match is required (i.e. the numeric value equals 'x') then the Value element can be used. |
| TextValue | QueryTextType | TextValue is used to query for the value of a concept or component based on textual parameters. The text value can be language specific (where parallel multi-lingual values are available) and is qualified with an operator indicating how the supplied text should be matched against the sought components. If only a simple equality check is necessary, regardless of language, the Value element can be used. |
| TimeValue | TimePeriodValueType | TimeValue is used to query for the value of a concept or component based on time parameters. This is typically used when the value needs to be treated explicitly as a time, for example when searching for data after a particular point in time. If only a simple equality check is necessary, the Value element can be used. |
| Value | SimpleValueType | Value is used to query for the value of a component. This should be used for concepts or components based on a simple value (e.g. a code or a simple string). This should be used when a simple equal/not equal operator is all that is necessary for matching the sought value. |

**TimeDimensionValueType:**TimeDimensionValueType is used to query for data where the time dimension has a particular value.

Derivation:

*DataStructureComponentValueQueryType* (restriction)   
   TimeDimensionValueType



Content:

ID?, TimeValue[1..2]

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ID | com:NCNameIDType | ID provides the identifier for component for which the value is sought. |
| TimeValue | TimePeriodValueType | TimeValue is used to query for the value of a concept or component based on time parameters. This is typically used when the value needs to be treated explicitly as a time, for example when searching for data after a particular point in time. If only a simple equality check is necessary, the Value element can be used. |

**DataflowQueryType:**DataflowQueryType defines the structure of a dataflow query. The parameters for the query are contained in the DataflowWhere element. The References element is used to indicate how objects that reference or are referenced from the matched dataflow should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   DataflowQueryType



Content:

ReturnDetails, DataflowWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| DataflowWhere | DataflowWhereType | DataflowWhere defines the parameters for a dataflow query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a dataflow based on the key family it defines the usage of. |

**DataflowWhereType:**DataflowWhereType contains the parameters of a dataflow query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *StructureUsageWhereType* (restriction)   
                                 DataflowWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, Structure?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: Dataflow) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| Structure | com: DataStructureReferen ceType | Structure is used to indicate which key family the dataflow must define a usage for in order to constitute a match. |

**DataStructureQueryType:**DataStructureQueryType defines the structure of a data structure definition query. The parameters for the query are contained in the DataStructureWhere element. The References element is used to indicate how objects that reference or are referenced from the matched data structure definition should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   DataStructureQueryType



Content:

ReturnDetails, DataStructureWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| DataStructureWhere | DataStructureWhereTy pe | DataStructureWhere contains the parameters for a data structure definition query. All parameters must be matched for an object to satisfy the query. The query is simply a refinement of the base structure query to make the parameters specific to the data structure definition. |

***DataStructureWhereBaseType*:**DataStructureWhereBaseType is an abstract base type that forms the basis of the DataStructureWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *StructureWhereType* (restriction)   
                                 *DataStructureWhereBaseType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, UsedConcept\*, UsedRepresentation\*, GroupWhere\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: DataStructure) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| UsedConcept | com: ConceptReferenceType | UsedConcept is used to query for a structure that uses the referenced concept as the basis of one of its components. |
| UsedRepresentation | com: DataStructureEnumera tionSchemeReferenceT ype | UsedRepresentation is used to query for a structure that uses the referenced item scheme for the representation of one of its components. |
| GroupWhere | GroupWhereType | GroupWhere is used to query for a data structure definition that contains a group meeting the conditions detailed in this container. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |

**DataStructureWhereType:**DataStructureWhereType defines the parameters of a data structure definition query. In addition to querying based on the identification, it is also possible to search for data structure definitions based on information about its components.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *StructureWhereType* (restriction)   
                                 *DataStructureWhereBaseType* (extension)   
                                       DataStructureWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, UsedConcept\*, UsedRepresentation\*, GroupWhere\*, AttributeWhere\*, DimensionWhere\*, MeasureDimensionWhere?, TimeDimensionWhere?, PrimaryMeasureWhere?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: DataStructure) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| UsedConcept | com: ConceptReferenceType | UsedConcept is used to query for a structure that uses the referenced concept as the basis of one of its components. |
| UsedRepresentation | com: DataStructureEnumera tionSchemeReferenceT ype | UsedRepresentation is used to query for a structure that uses the referenced item scheme for the representation of one of its components. |
| GroupWhere | GroupWhereType | GroupWhere is used to query for a data structure definition that contains a group meeting the conditions detailed in this container. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |
| AttributeWhere | AttributeWhereType | AttributeWhere is used to query for a data structure definition that contains an attribute meeting the conditions contained in this structure. The attribute can be queried based on its identification, the concept from which it takes its semantic, its attachment level, the role it plays, and the code list it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |
| DimensionWhere | DimensionWhereType | DimensionWhere is used to query for a data structure definition that contains a dimension meeting the conditions contained in this structure. The dimension can be queried based on its identification, the concept from which it takes its semantic, the role it plays, and the code list it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |
| MeasureDimensionWher e | MeasureDimensionWher eType | MeasureDimensionWhere is used to query for a data structure definition that contains a measure dimension meeting the conditions contained in this structure. The cross-sectional measure can be queried based on its identification, the concept from which it takes its semantic, and the concept scheme it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |
| TimeDimensionWhere | TimeDimensionWhereTy pe | TimeDimensionWhere is used to query for a data structure definition that contains a time dimension meeting the conditions contained in this structure. The time dimension can be queried based on its identification and the concept from which it takes its semantic. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |
| PrimaryMeasureWhere | PrimaryMeasureWhereT ype | PrimaryMeasureWhere is used to query for a data structure definition that contains a primary measure meeting the conditions contained in this structure. The primary measure can be queried based on its identification, the concept from which it takes its semantic, and the code list it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |

***GroupWhereBaseType*:**GroupWhereBaseType is an abstract base type that forms the basis for the GroupWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentListWhereType* (restriction)   
               *GroupWhereBaseType*



Content:

Annotation?, ID?, GroupDimensionWhere\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| GroupDimensionWhere | DimensionWhereType | GroupDimensionWhere is used to query a group based on the details of the dimensions it groups. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |

**GroupWhereType:**GroupWhereType defines the parameters querying for a data structure definition based a group meeting the conditions detailed. Parameters include identification, dimensions used in the group, and the group's referenced attachment constraint. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentListWhereType* (restriction)   
               *GroupWhereBaseType* (extension)   
                     GroupWhereType



Content:

Annotation?, ID?, GroupDimensionWhere\*, AttachmentConstraint?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| GroupDimensionWhere | DimensionWhereType | GroupDimensionWhere is used to query a group based on the details of the dimensions it groups. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |
| AttachmentConstraint | com: AttachmentConstraint ReferenceType | AttachmentConstraint queries for a group where the referenced attachment constraint defines the contents of the group. |

**DataStructureComponentWhereType:**DataStructureComponentWhereType defines the basic information for querying for a data structure definition component. The component can be queried based on its identification, the concept from which it takes its semantic, and the code list it uses as its representation,. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               DataStructureComponentWhereType



Content:

Annotation?, ID?, ConceptIdentity?, Enumeration?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| ConceptIdentity | com: ConceptReferenceType | ConceptIdentity is used to query for a structure component based on the concept from which it takes its semantic. |
| Enumeration | com: CodelistReferenceTyp e | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |

**AttributeWhereType:**AttributeWhereType describes the structure of an attribute query. An attribute can be queried based on its identification, the concept from which it takes its semantic, the role it plays, and the code list it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               DataStructureComponentWhereType (extension)   
                     AttributeWhereType



Content:

Annotation?, ID?, ConceptIdentity?, Enumeration?, Role\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| ConceptIdentity | com: ConceptReferenceType | ConceptIdentity is used to query for a structure component based on the concept from which it takes its semantic. |
| Enumeration | com: CodelistReferenceTyp e | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |
| Role | com: ConceptReferenceType | Role is used to specify the role of the attribute. |

**DimensionWhereType:**DimensionWhereType describes the structure of a dimension query. A dimension can be queried based on its identification, the concept from which it takes its semantic, the role it plays, and the code list it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               DataStructureComponentWhereType (extension)   
                     DimensionWhereType



Content:

Annotation?, ID?, ConceptIdentity?, Enumeration?, Role\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| ConceptIdentity | com: ConceptReferenceType | ConceptIdentity is used to query for a structure component based on the concept from which it takes its semantic. |
| Enumeration | com: CodelistReferenceTyp e | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |
| Role | com: ConceptReferenceType | Role is used to specify the role of the dimension. |

**TimeDimensionWhereType:**TimeDimensionWhereType describes the structure of a time dimension query. The time dimension can be queried based on the concept from which it takes its semantic. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               DataStructureComponentWhereType (restriction)   
                     TimeDimensionWhereType



Content:

Annotation?, ConceptIdentity?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ConceptIdentity | com: ConceptReferenceType | ConceptIdentity is used to query for a structure component based on the concept from which it takes its semantic. |

***MeasureDimensionWhereBaseType*:**MeasureDimensionWhereBaseType is an abstract base type which forms the basis for a measure dimension query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               *MeasureDimensionWhereBaseType*



Content:

Annotation?, ID?, ConceptIdentity?, Enumeration?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| ConceptIdentity | com: ConceptReferenceType | ConceptIdentity is used to query for a structure component based on the concept from which it takes its semantic. |
| Enumeration | com: ConceptSchemeReferen ceType | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |

**MeasureDimensionWhereType:**MeasureDimensionWhereType describes the structure of a measure dimension query. A measure dimension can be queried based on its identification, the concept from which it takes its semantic, the role it plays, and the concept scheme which defines its measure concepts. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               *MeasureDimensionWhereBaseType* (extension)   
                     MeasureDimensionWhereType



Content:

Annotation?, ID?, ConceptIdentity?, Enumeration?, Role\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| ConceptIdentity | com: ConceptReferenceType | ConceptIdentity is used to query for a structure component based on the concept from which it takes its semantic. |
| Enumeration | com: ConceptSchemeReferen ceType | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |
| Role | com: ConceptReferenceType | Role is used to specify the role of the dimension. |

**PrimaryMeasureWhereType:**MeasureWhereType describes the structure of a measure query. The primary measure can be queried based on the concept from which it takes its semantic, and the code list it uses as the enumeration of its representation. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               DataStructureComponentWhereType (restriction)   
                     PrimaryMeasureWhereType



Content:

Annotation?, ConceptIdentity?, Enumeration?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ConceptIdentity | com: ConceptReferenceType | ConceptIdentity is used to query for a structure component based on the concept from which it takes its semantic. |
| Enumeration | com: CodelistReferenceTyp e | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |

**HierarchicalCodelistQueryType:**HierarchicalCodelistQueryType defines the structure of a hierarchical codelist query. The parameters for the query are contained in the HierarchicalCodelistWhere element. The References element is used to indicate how objects that are referenced from the matched hierarchical codelist should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   HierarchicalCodelistQueryType



Content:

ReturnDetails, HierarchicalCodelistWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| HierarchicalCodelist Where | HierarchicalCodelist WhereType | HierarchicalCodelistWhere defines the parameters for a hierarchical codelist query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a hierarchical codelist based on the codelists it arranges into hierarchies. |

***HierarchicalCodelistWhereBaseType*:**HierarchicalCodelistWhereBaseType is an abstract base type which forms the basis for the HierarchicalCodelistWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *HierarchicalCodelistWhereBaseType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: HierarchicalCodelist) | com: MaintainableTypeCode listType | The type attribute indicates the type of constraint that is being queried for, with a default of Any, meaning both content and attachment constraints will be searched. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |

**HierarchicalCodelistWhereType:**HierarchicalCodelistWhereType contains the parameters of a hierarchical codelist query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *HierarchicalCodelistWhereBaseType* (extension)   
                                 HierarchicalCodelistWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, IncludedCodelist\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: HierarchicalCodelist) | com: MaintainableTypeCode listType | The type attribute indicates the type of constraint that is being queried for, with a default of Any, meaning both content and attachment constraints will be searched. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| IncludedCodelist | com: CodelistReferenceTyp e | IncludedCodelist is used to reference a codelist which the hierarchical codelist to be matched references. |

**MetadataQueryType:**MetadataQueryType defines the structure of a reference metadata query. Reference metadata is queried as individual reports. The result of this query will be a collection of metadata sets, with only the relevant metadata reports contained within them. If no report level parameters are specified, then the query will result in entire metadata sets being returned.

Content:

ReturnDetails, MetadataParameters

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MetadataReturnDetail sType | ReturnDetails specifies the details of how the reference metadata should be returned. |
| MetadataParameters | MetadataParametersAn dType | MetadataParameters contains the parameters that are to be matched in order to determine which metadata reports are to be returned as the result of the query. |

**MetadataReturnDetailsType:**MetadataReturnDetailsType is a structure for detailing how reference metadata should be returned. Only a default size limit can be specified.

Derivation:

*ReturnDetailsBaseType* (extension)   
   MetadataReturnDetailsType



Attributes:

defaultLimit?, detail?

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| defaultLimit | xs:integer | The defaultLimit attribute is the suggested maximum response size in kilobytes. |
| detail | xs:string | >The detail attribute is used to indicate how much of the matched object should be returned. |

***MetadataParametersType*:**MetadataParametersType defines the parameters for querying for reference metadata. This structure is refined by separate And/Or constructs which make logical restrictions on which parameters apply in such cases.

Content:

MetadataSetID\*, DataProvider\*, MetadataStructure\*, Metadataflow\*, ProvisionAgreement\*, Category\*, Updated\*, ConceptValue\*, RepresentationValue\*, MetadataTargetValue\*, ReportStructureValue\*, AttachmentConstraint\*, AttachedObject\*, AttachedDataKey\*, AttachedDataSet\*, AttachedReportingPeriod\*, Or\*, And\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| MetadataSetID | QueryIDType | MetadataSetID is used to match the id of the metadata set. Only metadata reports from data sets with an identifier satisfying these conditions will be matched. |
| DataProvider | com: DataProviderReferenc eType | DataProviderReference is used to match the provider of reference metadata to the referenced data provider. Only metadata reports from metadata sets provided by the referenced data provider will be matched. |
| MetadataStructure | com: MetadataStructureRef erenceType | MetadataStructureReference is used to match the underlying structure of the data. Only metadata reports from metadata sets that conform to referenced metadata structure definition will be matched. |
| Metadataflow | com: MetadataflowReferenc eType | MetadataflowReference is used to match the flow which reference metadata is reported against. Only metadata reports from metadata sets report against referenced metadata flow will be matched. |
| ProvisionAgreement | com: ProvisionAgreementRe ferenceType | ProvisionAgreement is used to match the provision agreement which metadata is reported against. Only metadata from metadata sets report against the referenced provision agreement will be matched. |
| Category | com: CategoryReferenceTyp e | CategoryReference is used to match a reference metadata based on the categorization of its underlying structure (metadata structure definition), or the usage of that structure (metadata flow). Only metadata reports whose underlying structure or structure usage are categorized against the referenced category will be matched. |
| Updated | com: TimeRangeValueType | Updated is used to match reference metadata based on when it was last updated. Only metadata reports which satisfy the conditions for the last update parameters supplied here will be matched. |
| ConceptValue | ConceptValueType | ConceptValue is used to match reference metadata based on the value of a particular concept which defines a metadata attribute. So long as the referenced concept has the specified value for a metadata attribute anywhere within the report structure, the metadata report which contains that attribute will be matched. |
| RepresentationValue | CodeValueType | RepresentationValue is used to match reference metadata based on a codelist having a particular value. This codelist only applies in the context of the representation of a metadata attribute. So long as the value of any metadata attribute within the report structure which uses the referenced codelist has the value specified, any metadata report containing the attribute will be matched. |
| MetadataTargetValue | MetadataTargetValueT ype | MetadataTargetValue is used to match reference metadata based on the target object values of a particular metadata target. If not report structure is specified in the query, this will result in a search across all metadata reports which may use this metadata target. If a value is not given for a target object which is part of this metadata target, it is assumed that all values are allowed for that target object. Thus, if no target object values are given in the entire metadata target, the query will simply match ant report where the reference metadata target is used. All target object value conditions must be met to constitute a match. |
| ReportStructureValue | ReportStructureValue Type | ReportStructureValue is used to match particular reference metadata reports. Only metadata reports based on the referenced report structure will be matched. It is also possible to detail the values of the metadata attributes contained within the reference report. In this case, only metadata reports based on the referenced report structure which have the metadata attribute values specified will be matched. |
| AttachmentConstraint | com: AttachmentConstraint ReferenceType | AttachmentConstraint references an attachment constraint in order to match reference metadata which matches the effective metadata keys or metadata target regions defined in the constraint. Metadata will be returned by first matching metadata on the keys and metadata target regions that are marked as included (or all metadata if none), and then excluding the metadata that satisfies the conditions of the excluded keys and metadata target regions. |
| AttachedObject | com: ObjectReferenceType | AttachedObject is used to match reference metadata based on an object which it is attached to, regardless of the report structure or metadata target in which the object is referenced. Any metadata reports attached to the referenced objects will be matched. |
| AttachedDataKey | com:DataKeyType | AttachedDataKey is used to match reference metadata based on a data key it is attached to, regardless of the report structure or metadata target in which the data key is referenced. Any metadata reports attached to the supplied data key will be matched. |
| AttachedDataSet | com:SetReferenceType | AttachedDataSet is used to query for reference metadata based on a data set it is attached to, regardless of the report structure or metadata target in which the data set is referenced. Any metadata reports attached to the supplied metadata key will be matched. |
| AttachedReportingPer iod | com: TimeRangeValueType | AttachedReportingPeriod is used to query for reference metadata based on the reporting period to which it is attached, regardless of the report structure or metadata target in which the reference period is referenced. Any metadata reports attached to a reporting period which falls within the range specified will be matched. |
| Or | MetadataParametersOr Type | Or contains a collection of additional parameters, any one of which can be satisfied to result in a match. |
| And | MetadataParametersAn dType | And contains a collection of additional parameters, all of which must be satisfied to result in a match. |

**MetadataParametersOrType:**MetadataParametersOrType refines the base data parameters to define a set of parameters joined by an "or" condition. Only one of the parameters supplied in an instance of this type can be satisfied to result in a match.

Derivation:

*MetadataParametersType* (restriction)   
   MetadataParametersOrType



Content:

MetadataSetID\*, DataProvider\*, MetadataStructure\*, Metadataflow\*, ProvisionAgreement\*, Category\*, Updated\*, ConceptValue\*, RepresentationValue\*, MetadataTargetValue\*, ReportStructureValue\*, AttachmentConstraint\*, AttachedObject\*, AttachedDataKey\*, AttachedDataSet\*, AttachedReportingPeriod\*, And\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| MetadataSetID | QueryIDType | MetadataSetID is used to match the id of the metadata set. Only metadata reports from data sets with an identifier satisfying these conditions will be matched. |
| DataProvider | com: DataProviderReferenc eType | DataProviderReference is used to match the provider of reference metadata to the referenced data provider. Only metadata reports from metadata sets provided by the referenced data provider will be matched. |
| MetadataStructure | com: MetadataStructureRef erenceType | MetadataStructureReference is used to match the underlying structure of the data. Only metadata reports from metadata sets that conform to referenced metadata structure definition will be matched. |
| Metadataflow | com: MetadataflowReferenc eType | MetadataflowReference is used to match the flow which reference metadata is reported against. Only metadata reports from metadata sets report against referenced metadata flow will be matched. |
| ProvisionAgreement | com: ProvisionAgreementRe ferenceType | ProvisionAgreement is used to match the provision agreement which metadata is reported against. Only metadata from metadata sets report against the referenced provision agreement will be matched. |
| Category | com: CategoryReferenceTyp e | CategoryReference is used to match a reference metadata based on the categorization of its underlying structure (metadata structure definition), or the usage of that structure (metadata flow). Only metadata reports whose underlying structure or structure usage are categorized against the referenced category will be matched. |
| Updated | com: TimeRangeValueType | Updated is used to match reference metadata based on when it was last updated. Only metadata reports which satisfy the conditions for the last update parameters supplied here will be matched. |
| ConceptValue | ConceptValueType | ConceptValue is used to match reference metadata based on the value of a particular concept which defines a metadata attribute. So long as the referenced concept has the specified value for a metadata attribute anywhere within the report structure, the metadata report which contains that attribute will be matched. |
| RepresentationValue | CodeValueType | RepresentationValue is used to match reference metadata based on a codelist having a particular value. This codelist only applies in the context of the representation of a metadata attribute. So long as the value of any metadata attribute within the report structure which uses the referenced codelist has the value specified, any metadata report containing the attribute will be matched. |
| MetadataTargetValue | MetadataTargetValueT ype | MetadataTargetValue is used to match reference metadata based on the target object values of a particular metadata target. If not report structure is specified in the query, this will result in a search across all metadata reports which may use this metadata target. If a value is not given for a target object which is part of this metadata target, it is assumed that all values are allowed for that target object. Thus, if no target object values are given in the entire metadata target, the query will simply match ant report where the reference metadata target is used. All target object value conditions must be met to constitute a match. |
| ReportStructureValue | ReportStructureValue Type | ReportStructureValue is used to match particular reference metadata reports. Only metadata reports based on the referenced report structure will be matched. It is also possible to detail the values of the metadata attributes contained within the reference report. In this case, only metadata reports based on the referenced report structure which have the metadata attribute values specified will be matched. |
| AttachmentConstraint | com: AttachmentConstraint ReferenceType | AttachmentConstraint references an attachment constraint in order to match reference metadata which matches the effective metadata keys or metadata target regions defined in the constraint. Metadata will be returned by first matching metadata on the keys and metadata target regions that are marked as included (or all metadata if none), and then excluding the metadata that satisfies the conditions of the excluded keys and metadata target regions. |
| AttachedObject | com: ObjectReferenceType | AttachedObject is used to match reference metadata based on an object which it is attached to, regardless of the report structure or metadata target in which the object is referenced. Any metadata reports attached to the referenced objects will be matched. |
| AttachedDataKey | com:DataKeyType | AttachedDataKey is used to match reference metadata based on a data key it is attached to, regardless of the report structure or metadata target in which the data key is referenced. Any metadata reports attached to the supplied data key will be matched. |
| AttachedDataSet | com:SetReferenceType | AttachedDataSet is used to query for reference metadata based on a data set it is attached to, regardless of the report structure or metadata target in which the data set is referenced. Any metadata reports attached to the supplied metadata key will be matched. |
| AttachedReportingPer iod | com: TimeRangeValueType | AttachedReportingPeriod is used to query for reference metadata based on the reporting period to which it is attached, regardless of the report structure or metadata target in which the reference period is referenced. Any metadata reports attached to a reporting period which falls within the range specified will be matched. |
| And | MetadataParametersAn dType | And contains a collection of additional parameters, all of which must be satisfied to result in a match. |

**MetadataParametersAndType:**MetadataParametersAndType refines the base data parameters to define a set of parameters joined by an "and" conditions. All of the parameters supplied in an instance of this type must be satisfied to result in a match. As a result of this condition, the maximum occurrence of some parameters has been reduced so as to not allow for impossible conditions to be specified (for example data cannot be matched is it is specified that the data set identifier should be "xyz" and the data identifier should be "abc".

Derivation:

*MetadataParametersType* (restriction)   
   MetadataParametersAndType



Content:

MetadataSetID?, DataProvider?, MetadataStructure?, Metadataflow?, ProvisionAgreement\*, Category\*, Updated?, ConceptValue\*, RepresentationValue\*, MetadataTargetValue?, ReportStructureValue?, AttachmentConstraint\*, AttachedObject\*, AttachedDataKey?, AttachedDataSet?, AttachedReportingPeriod?, Or\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| MetadataSetID | QueryIDType | MetadataSetID is used to match the id of the metadata set. Only metadata reports from data sets with an identifier satisfying these conditions will be matched. |
| DataProvider | com: DataProviderReferenc eType | DataProviderReference is used to match the provider of reference metadata to the referenced data provider. Only metadata reports from metadata sets provided by the referenced data provider will be matched. |
| MetadataStructure | com: MetadataStructureRef erenceType | MetadataStructureReference is used to match the underlying structure of the data. Only metadata reports from metadata sets that conform to referenced metadata structure definition will be matched. |
| Metadataflow | com: MetadataflowReferenc eType | MetadataflowReference is used to match the flow which reference metadata is reported against. Only metadata reports from metadata sets report against referenced metadata flow will be matched. |
| ProvisionAgreement | com: ProvisionAgreementRe ferenceType | ProvisionAgreement is used to match the provision agreement which metadata is reported against. Only metadata from metadata sets report against the referenced provision agreement will be matched. |
| Category | com: CategoryReferenceTyp e | CategoryReference is used to match a reference metadata based on the categorization of its underlying structure (metadata structure definition), or the usage of that structure (metadata flow). Only metadata reports whose underlying structure or structure usage are categorized against the referenced category will be matched. |
| Updated | com: TimeRangeValueType | Updated is used to match reference metadata based on when it was last updated. Only metadata reports which satisfy the conditions for the last update parameters supplied here will be matched. |
| ConceptValue | ConceptValueType | ConceptValue is used to match reference metadata based on the value of a particular concept which defines a metadata attribute. So long as the referenced concept has the specified value for a metadata attribute anywhere within the report structure, the metadata report which contains that attribute will be matched. |
| RepresentationValue | CodeValueType | RepresentationValue is used to match reference metadata based on a codelist having a particular value. This codelist only applies in the context of the representation of a metadata attribute. So long as the value of any metadata attribute within the report structure which uses the referenced codelist has the value specified, any metadata report containing the attribute will be matched. |
| MetadataTargetValue | MetadataTargetValueT ype | MetadataTargetValue is used to match reference metadata based on the target object values of a particular metadata target. If not report structure is specified in the query, this will result in a search across all metadata reports which may use this metadata target. If a value is not given for a target object which is part of this metadata target, it is assumed that all values are allowed for that target object. Thus, if no target object values are given in the entire metadata target, the query will simply match ant report where the reference metadata target is used. All target object value conditions must be met to constitute a match. |
| ReportStructureValue | ReportStructureValue Type | ReportStructureValue is used to match particular reference metadata reports. Only metadata reports based on the referenced report structure will be matched. It is also possible to detail the values of the metadata attributes contained within the reference report. In this case, only metadata reports based on the referenced report structure which have the metadata attribute values specified will be matched. |
| AttachmentConstraint | com: AttachmentConstraint ReferenceType | AttachmentConstraint references an attachment constraint in order to match reference metadata which matches the effective metadata keys or metadata target regions defined in the constraint. Metadata will be returned by first matching metadata on the keys and metadata target regions that are marked as included (or all metadata if none), and then excluding the metadata that satisfies the conditions of the excluded keys and metadata target regions. |
| AttachedObject | com: ObjectReferenceType | AttachedObject is used to match reference metadata based on an object which it is attached to, regardless of the report structure or metadata target in which the object is referenced. Any metadata reports attached to the referenced objects will be matched. |
| AttachedDataKey | com:DataKeyType | AttachedDataKey is used to match reference metadata based on a data key it is attached to, regardless of the report structure or metadata target in which the data key is referenced. Any metadata reports attached to the supplied data key will be matched. |
| AttachedDataSet | com:SetReferenceType | AttachedDataSet is used to query for reference metadata based on a data set it is attached to, regardless of the report structure or metadata target in which the data set is referenced. Any metadata reports attached to the supplied metadata key will be matched. |
| AttachedReportingPer iod | com: TimeRangeValueType | AttachedReportingPeriod is used to query for reference metadata based on the reporting period to which it is attached, regardless of the report structure or metadata target in which the reference period is referenced. Any metadata reports attached to a reporting period which falls within the range specified will be matched. |
| Or | MetadataParametersOr Type | Or contains a collection of additional parameters, any one of which can be satisfied to result in a match. |

**MetadataTargetValueType:**MetadataTargetValueType describes the structure that is used to match reference metadata where a given metadata target's target object have particular values. If a value is not given for a target object which is part of the metadata target, it is assumed that all values are allowed for that target object. Thus, if no target object values are given in the entire metadata target, the query will simply match ant report where the reference metadata target is used. All target object value conditions must be met to constitute a match.

Content:

ID, TargetObjectValue\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ID | com:NCNameIDType |  |
| TargetObjectValue | TargetObjectValueTyp e | TargetObjectValue is used to match reference metadata where a given target object in a metadata target references a particular object or time period. |

**TargetObjectValueType:**IdentifierComponentValueType describes the structure that is used to match reference metadata where a given identifier component has a particular value.

Content:

ID, (DataSet | DataKey | Object | TimeValue[1..2])

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ID | com:NCNameIDType | ID identifies the metadata target object. |
| DataSet | com:SetReferenceType | DataSet provides a reference to a data set which the target object should reference to result in a match. |
| DataKey | com:DataKeyType | DataKey provides a data key (set of dimension values) which the target object should reference to result in a match. |
| Object | com: ObjectReferenceType | Object provides a reference to any SDMX identifiable object which the target object should reference to result in a match. |
| TimeValue | TimePeriodValueType | TimeValue is used to provide a time value or range for matching a reporting period which the target object should reference to result in a match. |

**ReportStructureValueType:**ReportStructureValueType describes the structure that is used to match reference metadata where the metadata attributes of a report structure have particular values. All metadata attribute value conditions must be met to constitute a match.

Content:

ID, MetadataAttributeValue\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ID | com:NCNameIDType |  |
| MetadataAttributeVal ue | MetadataAttributeVal ueType | MetadataAttributeValue is used to match reference metadata where a metadata attribute has a particular value within the referenced report structure. |

**MetadataAttributeValueType:**MetadataAttributeValueType describes the structure that is used to match reference metadata where a metadata attribute has a particular value. Metadata attribute value queries can be nested for querying nested metadata attributes. If no value is provided, then simply the presence of the metadata attribute within the given context will result in a match. All nested metadata attribute value conditions must be met to constitute a match.

Content:

ID, (Value | TextValue+ | NumericValue[1..2] | TimeValue[1..2])?, MetadataAttributeValue\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ID | com:NCNameIDType |  |
| Value | SimpleValueType | Value is used to query for the value of a component. This should be used for concepts or components based on a simple value (e.g. a code or a simple string). This should be used when a simple equal/not equal operator is all that is necessary for matching the sought value. |
| TextValue | QueryTextType | TextValue is used to query for the value of a concept or component based on textual parameters. The text value can be language specific (where parallel multi-lingual values are available) and is qualified with an operator indicating how the supplied text should be matched against the sought components. If only a simple equality check is necessary, regardless of language, the Value element can be used. |
| NumericValue | NumericValueType | NumericValue is used to query for a the value of a concept or component based on a numeric search. This is typically used when the value needs to be searched explicitly as a number, such as when data is sought with an observed value greater than some threshold. If only a simple match is required (i.e. the numeric value equals 'x') then the Value element can be used. |
| TimeValue | TimePeriodValueType | TimeValue is used to query for the value of a concept or component based on time parameters. This is typically used when the value needs to be treated explicitly as a time, for example when searching for data after a particular point in time. If only a simple equality check is necessary, the Value element can be used. |
| MetadataAttributeVal ue | MetadataAttributeVal ueType |  |

**MetadataflowQueryType:**MetadataflowQueryType defines the structure of a metadataflow query. The parameters for the query are contained in the MetadataflowWhere element. The References element is used to indicate how objects that reference or are referenced from the matched metadataflow should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   MetadataflowQueryType



Content:

ReturnDetails, MetadataflowWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| MetadataflowWhere | MetadataflowWhereTyp e | MetadataflowWhere contains the parameters for a metadataflow query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a metadataflow based on the metadata structure definition it defines the usage of. |

**MetadataflowWhereType:**MetadataflowWhereType contains the parameters of a metadataflow query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *StructureUsageWhereType* (restriction)   
                                 MetadataflowWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, Structure?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: Metadataflow) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| Structure | com: MetadataStructureRef erenceType | MetadataStructureReference is used to indicate which metadata structure definition the metadataflow must define a usage for in order to constitute a match. |

**MetadataStructureQueryType:**MetadataStructureQueryType defines the structure of a metadata structure definition query. The parameters for the query are contained in the MetadataStructureDefinitionWhere element. The References element is used to indicate how objects that reference or are referenced from the matched metadata structure definition should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   MetadataStructureQueryType



Content:

ReturnDetails, MetadataStructureWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| MetadataStructureWhe re | MetadataStructureWhe reType | MetadataStructureWhere contains the parameters for a metadata structure definition query. All parameters must be matched for an object to satisfy the query. The query is simply a refinement of the base structure query to make the parameters specific to the metadata structure definition. |

***MetadataStructureWhereBaseType*:**MetadataStructureWhereBaseType is an abstract base type the forms the basis for the MetadataStructureWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *StructureWhereType* (restriction)   
                                 *MetadataStructureWhereBaseType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, UsedConcept\*, UsedRepresentation\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: MetadataStructure) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| UsedConcept | com: ConceptReferenceType | UsedConcept is used to query for a structure that uses the referenced concept as the basis of one of its components. |
| UsedRepresentation | com: ItemSchemeReferenceT ype | UsedRepresentation is used to query for a structure that uses the referenced item scheme for the representation of one of its components. |

**MetadataStructureWhereType:**MetadataStructureWhereType defines the parameters of a metadata structure definition query. In addition to querying based on the identification, it is also possible to search for metadata structure definitions based on information about its components.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *StructureWhereType* (restriction)   
                                 *MetadataStructureWhereBaseType* (extension)   
                                       MetadataStructureWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, UsedConcept\*, UsedRepresentation\*, MetadataTargetWhere\*, TargetObjectWhere\*, ReportStructureWhere\*, MetadataAttributeWhere\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: MetadataStructure) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| UsedConcept | com: ConceptReferenceType | UsedConcept is used to query for a structure that uses the referenced concept as the basis of one of its components. |
| UsedRepresentation | com: ItemSchemeReferenceT ype | UsedRepresentation is used to query for a structure that uses the referenced item scheme for the representation of one of its components. |
| MetadataTargetWhere | MetadataTargetWhereT ype | MetadataTargetWhere is used to query for a metadata structure definition that contains a metadata target meeting the conditions contained in this structure. The metadata target can be queried based on its identification and/or the details of its target objects. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |
| TargetObjectWhere | TargetObjectWhereTyp e | IdentifierComponentWhere is used to query for specific target identifiers or metadata structure definitions where a contained identifier component meets the conditions detailed. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |
| ReportStructureWhere | ReportStructureWhere Type | ReportStructureWhere is used to query for metadata structure definitions where a given report structure meets the conditions specified. A report structure can be queried based on identification or details about its metadata attributes. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |
| MetadataAttributeWhe re | MetadataAttributeWhe reType | MetadataAttributeWhere is a parameter which is used in a report structure parameter or to query metadata structure definitions where a contained metadata attribute meets the conditions specified. A metadata attribute can be queried based on its identification, the concept from which it takes its semantic, and an item scheme it uses as its representation. Nested metadata attributes allow for the querying of metadata attributes explicitly at nested level, although a top level metadata attribute query will be processed by querying metadata attributes at any level. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |

**MetadataTargetWhereType:**MetadataTargetWhereType describes the structure that is used to query for metadata structure definitions containing a metadata target meeting the conditions detailed. Conditions include the identification and the details of the target objects which make up the metadata target.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentListWhereType* (restriction)   
               MetadataTargetWhereType



Content:

Annotation?, ID?, TargetObjectWhere\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| TargetObjectWhere | TargetObjectWhereTyp e | IdentifierComponentWhere is used to query for specific target identifiers or metadata structure definitions where a contained identifier component meets the conditions detailed. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |

***TargetObjectWhereBaseType*:**TargetObjectWhereBaseType is an abstract base type that forms the basis for the TargetObjectWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               *TargetObjectWhereBaseType*



Content:

Annotation?, ID?, Enumeration?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Enumeration | com: ItemSchemeReferenceT ype | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |

**TargetObjectWhereType:**TargetObjectWhereType describes the structure of a target object query. A target object can be queried based on its identification, its type (i.e. data set target, key descriptor values target, report period target, or identifiable object target), and in the case of an identifiable object target, an item scheme which enumerates the possible values and/or the class of the target object reference.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               *TargetObjectWhereBaseType* (extension)   
                     TargetObjectWhereType



Attributes:

type?, targetClass?

Content:

Annotation?, ID?, Enumeration?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type | com: TargetObjectTypeCode listType | The type attribute is used to query for a target object of a given type (i.e. data set target, key descriptor values target, report period target, or identifiable object target) |
| targetClass | com: ObjectTypeCodelistTy pe | The targetClass attribute is used to query for an identifiable object target based on the class its target object. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Enumeration | com: ItemSchemeReferenceT ype | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |

**ReportStructureWhereType:**ReportStructureWhereType defines the parameters for matching based on the details of a report structure. This is used to query for metadata structure definitions where a given report structure meets the conditions specified. A report structure can be queried based on identification and details about its metadata attributes. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentListWhereType* (restriction)   
               ReportStructureWhereType



Content:

Annotation?, ID?, MetadataAttributeWhere\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| MetadataAttributeWhe re | MetadataAttributeWhe reType | MetadataAttributeWhere is a parameter which is used in a report structure parameter or to query metadata structure definitions where a contained metadata attribute meets the conditions specified. A metadata attribute can be queried based on its identification, the concept from which it takes its semantic, and an item scheme it uses as its representation. Nested metadata attributes allow for the querying of metadata attributes explicitly at nested level, although a top level metadata attribute query will be processed by querying metadata attributes at any level. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |

***MetadataAttributeWhereBaseType*:**MetadataAttributeWhereBaseType is an abstract base type that forms the basis for the MetadataAttributeWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               *MetadataAttributeWhereBaseType*



Content:

Annotation?, ID?, Enumeration?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Enumeration | com: CodelistReferenceTyp e | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |

**MetadataAttributeWhereType:**MetadataAttributeWhereType describes the parameters for a metadata attribute. A metadata attribute can be queried based on its identification, the concept from which it takes its semantic, and an item scheme it uses as its representation. Nested metadata attributes allow for the querying of metadata attributes explicitly at nested level, although a top level metadata attribute query will be processed by querying metadata attributes at any level. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *ComponentWhereType* (restriction)   
               *MetadataAttributeWhereBaseType* (extension)   
                     MetadataAttributeWhereType



Content:

Annotation?, ID?, Enumeration?, MetadataAttributeWhere\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Enumeration | com: CodelistReferenceTyp e | Enumeration is used to query for a structure component based on the item scheme that is used as the enumeration for its representation. This enumeration may be explicit defined by the component (i.e. its local representation), or inherited from the concept from which the component takes its semantic (i.e. the concept core representation). |
| MetadataAttributeWhe re | MetadataAttributeWhe reType | MetadataAttributeWhere is a parameter which is used in a report structure parameter or to query metadata structure definitions where a contained metadata attribute meets the conditions specified. A metadata attribute can be queried based on its identification, the concept from which it takes its semantic, and an item scheme it uses as its representation. Nested metadata attributes allow for the querying of metadata attributes explicitly at nested level, although a top level metadata attribute query will be processed by querying metadata attributes at any level. This is an implicit set of "and" parameters, that is the conditions within this must all be met in order to return a match. |

**OrganisationSchemeQueryType:**OrganisationSchemeQueryType defines the structure of an organisation scheme query. The parameters for the query are contained in the OrganisationSchemeWhere element. The References element is used to indicate how objects that reference the matched organisation scheme should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   OrganisationSchemeQueryType



Content:

ReturnDetails, OrganisationSchemeWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | StructureReturnDetai lsType |  |
| OrganisationSchemeWh ere | OrganisationSchemeWh ereType | OrganisationSchemeWhere defines the parameters for an organisation scheme query, regardless of the specific type of organisation scheme being sought. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for an organisation scheme based on the details of its organisations. In any case, the organisation scheme will be returned according the indicated return detail. |

**OrganisationSchemeWhereType:**OrganisationSchemeWhereType contains the parameters of an organisation scheme query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *ItemSchemeWhereType* (restriction)   
                                 OrganisationSchemeWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, OrganisationWhere\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (default: OrganisationScheme) | OrganisationSchemeTy peCodeType | The type attribute indicates the type of organisation scheme which is being queried for, with the default being any organisation scheme. Note that agency, data consumer, and data provider scheme all have fixed identifiers and versions, so specifying these types with parameters for the identifier and/or version which do not match these fixed values will never return a result. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| OrganisationWhere | OrganisationWhereTyp e | OrganisationWhere is used to query for organisations matching the parameters supplied. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match. |

**OrganisationWhereType:**OrganisationWhereType defines a set of parameters for matching an organisation. In addition to the base parameters for any item, there is an additional parameter for matching an organisation based on the roles it serves. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *ItemWhereType* (restriction)   
                     OrganisationWhereType



Content:

Annotation?, ID?, Name?, Description?, Parent?

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Parent | com: LocalOrganisationUni tReferenceType | Parent is only applicable when searching for organisation units, and is used to match organisations which have a parent organisation unit which is referenced here. |

**ProcessQueryType:**ProcessQueryType defines the structure of a process query. The parameters for the query are contained in the ProcessWhere element. The References element is used to indicate how objects that are referenced from the matched process scheme should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   ProcessQueryType



Content:

ReturnDetails, ProcessWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| ProcessWhere | ProcessWhereType | ProcessWhere contains the parameters for a process query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to query based on the details of the process steps defined within the process. In any case, the entire process will be returned according the indicated return detail. |

***ProcessWhereBaseType*:**ProcessWhereBaseType is an abstract base type which forms the basis for the ProcessWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *ProcessWhereBaseType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: Process) | com: MaintainableTypeCode listType | The type attribute indicates the type of constraint that is being queried for, with a default of Any, meaning both content and attachment constraints will be searched. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |

**ProcessWhereType:**ProcessWhereType defines the parameters of a process query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *ProcessWhereBaseType* (extension)   
                                 ProcessWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, ProcessStepWhere\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: Process) | com: MaintainableTypeCode listType | The type attribute indicates the type of constraint that is being queried for, with a default of Any, meaning both content and attachment constraints will be searched. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| ProcessStepWhere | ProcessStepWhereType | ProcessStepWhere is used to query for process steps matching the parameters supplied. It allows for nested process step queries so that hierarchical steps can be queried explicitly by their nested level, although a top level step will always result in a search for process steps at any level. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match. |

**ProcessStepWhereType:**ProcessStepWhereType defines a set of parameters for matching a category. All supplied parameters must be matched in order for an object to satisfy the query. In addition to the base item parameters, there are also parameters for matching based on the objects which serve as the input or output to the process step.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         ProcessStepWhereType



Content:

Annotation?, URN?, ID?, InputOrOutputObject\*, ProcessStepWhere\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| InputOrOutputObject | InputOrOutputObjectT ype | InputOrOutputObject is a parameter for matching a process step based on the referenced object, and whether it is an input or an output to the step. |
| ProcessStepWhere | ProcessStepWhereType | ProcessStepWhere is used to query for process steps within a the particular process step. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match. |

**InputOrOutputObjectType:**InputOrOutputObjectType describes the structure of input or output condition for a process step query. It contains reference to an object, as will as an attribute indicates whether the object should be an input, output, or either of the two to the step.

Attributes:

type?

Content:

ObjectReference

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (default: Any) | InputOutputTypeCodeT ype | The type attribute is used to indicate whether the referenced object should be an input, output, or either of the two to the process step. The default for this is Any. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ObjectReference | com: ObjectReferenceType | ObjectReference is used to query for a process containing process steps where the referenced object is input or output to the process step. |

**ProvisionAgreementQueryType:**ProvisionAgreementQueryType defines the structure of a provision agreement query. The parameters for the query are contained in the ProvisionAgreementWhere element. The References element is used to indicate how objects that reference or are referenced from the matched provision agreement should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   ProvisionAgreementQueryType



Content:

ReturnDetails, ProvisionAgreementWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| ProvisionAgreementWh ere | ProvisionAgreementWh ereType | ProvisionAgreementWhere contains the parameters for a provision agreement query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a provision agreement based on the data provider and the structure usage it pairs. |

***ProvisionAgreementWhereBaseType*:**ProvisionAgreementWhereBaseType is an abstract base type which forms the basis for the ProvisionAgreementWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *ProvisionAgreementWhereBaseType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: ProvisionAgreement) | com: MaintainableTypeCode listType | The type attribute indicates the type of constraint that is being queried for, with a default of Any, meaning both content and attachment constraints will be searched. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |

**ProvisionAgreementWhereType:**ProvisionAgreementWhereType defines the parameters of a provision agreement query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *ProvisionAgreementWhereBaseType* (extension)   
                                 ProvisionAgreementWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, StructureUsage?, DataProvider?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: ProvisionAgreement) | com: MaintainableTypeCode listType | The type attribute indicates the type of constraint that is being queried for, with a default of Any, meaning both content and attachment constraints will be searched. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| StructureUsage | com: StructureUsageRefere nceType | StructureUsage is used to indicate which structure usage the provision agreement must reference in order to constitute a match. |
| DataProvider | com: DataProviderReferenc eType | DataProvider is used to indicate which data provider the provision agreement must reference in order to constitute a match. |

**ReportingTaxonomyQueryType:**ReportingTaxonomyQueryType defines the structure of a reporting taxonomy query. The parameters for the query are contained in the ReportingTaxonomyWhere element. The References element is used to indicate how objects that are referenced from the reporting taxonomy should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   ReportingTaxonomyQueryType



Content:

ReturnDetails, ReportingTaxonomyWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | StructureReturnDetai lsType |  |
| ReportingTaxonomyWhe re | ReportingTaxonomyWhe reType | ReportingTaxonomyWhere contains the parameters for a reporting taxonomy query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search for a reporting taxonomy based on the details of its reporting categories. In any case, the reporting taxonomy will be returned according the indicated return detail. |

**ReportingTaxonomyWhereType:**ReportingTaxonomyWhereType defines the parameters of a reporting taxonomy query. All supplied parameters must be matched in order for an object to satisfy the query. In addition to querying based on the base maintainable parameters, it is also possible to search for taxonomies that contain particular reporting categories, and on the root level structure definitions of the taxonomy.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (extension)   
                           *ItemSchemeWhereType* (restriction)   
                                 ReportingTaxonomyWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, ReportingCategoryWhere\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: ReportingTaxonomy) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| ReportingCategoryWhe re | ReportingCategoryWhe reType | ReportingCategoryWhere is used to query for reporting categories matching the parameters supplied. It allows for nested reporting category queries so that hierarchical reporting categories can be queried explicitly by their nested level, although a top level reporting category will always result in a search for reporting categories at any level. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match. |

***ReportingCategoryWhereBaseType*:**ReportingCategoryWhereBaseType is an abstract base type that forms the basis for the ReportingCategoryQueryType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *ItemWhereType* (restriction)   
                     *ReportingCategoryWhereBaseType*



Content:

Annotation?, ID?, Name?, Description?, ReportingCategoryWhere\*

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| ReportingCategoryWhe re | ReportingCategoryWhe reType | ReportingCategoryWhere is used to query for reporting categories matching the parameters supplied. It allows for nested reporting category queries so that hierarchical reporting categories can be queried explicitly by their nested level, although a top level reporting category will always result in a search for reporting categories at any level. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match. |

**ReportingCategoryWhereType:**ReportingCategoryWhereType contains a set of parameters for matching a reporting category. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *ItemWhereType* (restriction)   
                     *ReportingCategoryWhereBaseType* (extension)   
                           ReportingCategoryWhereType



Content:

Annotation?, ID?, Name?, Description?, ReportingCategoryWhere\*, (ProvisioningMetadata\* | StructuralMetadata\*)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| ReportingCategoryWhe re | ReportingCategoryWhe reType | ReportingCategoryWhere is used to query for reporting categories matching the parameters supplied. It allows for nested reporting category queries so that hierarchical reporting categories can be queried explicitly by their nested level, although a top level reporting category will always result in a search for reporting categories at any level. This is an implicit set of "and" parameters, meaning all of the conditions must be met in order to return a match. |
| ProvisioningMetadata | com: StructureUsageRefere nceType | ProvisioningMetadata is used to query for a reporting category where the structure usages referenced are referenced by the reporting category. |
| StructuralMetadata | com: StructureReferenceTy pe | StructuralMetadata is used to query for a reporting category where the structures referenced are referenced by the reporting category. |

**DataSchemaQueryType:**DataSchemaQueryType defines the structure of a query for a structured data schema. This query consists of a single data structure which provides the full details of what type of structured data schema should be returned.

Content:

DataStructure

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| DataStructure | DataStructureRequest Type | DataStructure references a data structure either explicitly or through a usage (dataflow) or agreement based on the flow (provision agreement). An observation dimension must be specifies as well as indicators for using explicit measures, deriving the type from the time series specific data set, and processing constraints. The explicit measure option is only applicable if the observation dimension is the measure dimension; otherwise the value provided will be ignored. The time series option is only applicable if the observation dimension is the time dimension; otherwise the value provided will be ignored. Constraints will only be applied in the returned schemas if specifically requested. This means that even if the request specifies a provision agreement, the returned schema will not take into account the constraints on that agreement unless the request explicitly requests this. |

**MetadataSchemaQueryType:**MetadataSchemaQueryType defines the structure of a query for a structured metadata schema. This query consists of a single metadata structure which simply provides a reference to a metadata structure.

Content:

MetadataStructure

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| MetadataStructure | com: GenericMetadataStruc tureType | MetadataStructure references a metadata structure either explicitly or through it usage (metadataflow) or an application of that usage (provision agreement). |

**DataStructureRequestType:**DataStructureRequestType extends the base DataStructureRequestType to add additional parameters that are necessary when querying for a schema.

Derivation:

*com:PayloadStructureType* (restriction)   
   *com:DataStructureType* (restriction)   
         com:DataStructureRequestType (extension)   
               DataStructureRequestType



Attributes:

structureID, dimensionAtObservation, explicitMeasures?, serviceURL?, structureURL?, timeSeries?, processConstraints?

Content:

(com:ProvisionAgrement | com:StructureUsage | com: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 | com: 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. |
| timeSeries (default: false) | xs:boolean | The timeSeries attribute indicates that the requested schema should derived from the time series specific data set. If the observation dimension is anything but the time dimension, this field will be ignored. |
| processConstraints (default: false) | xs:boolean | The processConstraints attribute indicates that constraints should be processed when returning the schema. If this value is false, then the schema will be based on the data structure deflation, regardless of whether the reference was specified as a provision agreement, dataflow, or data structure. If this is true, then the constraints at the requested level will be processed and the returned schema will take these constraints into account (i.e. the appropriate code lists will be sub-setted |

Element Documentation:

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

**StructureSetQueryType:**StructureSetQueryType defines the structure of a structure set query. The parameters for the query are contained in the StructureSetWhere element. The References element is used to indicate how objects that are referenced from the structure set should be returned.

Derivation:

*StructuralMetadataQueryType* (restriction)   
   StructureSetQueryType



Content:

ReturnDetails, StructureSetWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| StructureSetWhere | StructureSetWhereTyp e | StructureSetWhere contains the parameters for a structure query. All parameters must be matched for an object to satisfy the query. In addition to querying based on the basic maintainable properties, it is also possible to search based on the structures that are related by the set or the objects which are mapped by the set's maps. In any case, the structure set will be returned according the indicated return detail. |

***StructureSetWhereBaseType*:**StructureSetWhereBaseType is an abstract base type which forms the basis for the StructureSetWhereType.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *StructureSetWhereBaseType*



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: StructureSet) | com: MaintainableTypeCode listType | The type attribute indicates the type of constraint that is being queried for, with a default of Any, meaning both content and attachment constraints will be searched. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |

**StructureSetWhereType:**StructureSetWhereType defines the parameters of a structure set query. All supplied parameters must be matched in order for an object to satisfy the query. In addition to querying based on the base maintainable parameters, it is also possible to search based on the structures that are related by the set or the objects which are mapped by the set's maps.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           *StructureSetWhereBaseType* (extension)   
                                 StructureSetWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?, RelatedStructures\*, MappedObject\*

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: StructureSet) | com: MaintainableTypeCode listType | The type attribute indicates the type of constraint that is being queried for, with a default of Any, meaning both content and attachment constraints will be searched. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |
| RelatedStructures | com: StructureOrUsageRefe renceType | RelatedStructure is used to query for structure sets where the referenced key families, metadata structure definitions, dataflows, and metadataflows are related to another by the structure set. |
| MappedObject | MappedObjectType | MappedObject is used to query for structure sets where the reference object is mapped in one of the maps defined by the structure set. The referenced object can be specified as being either a source, a target, or either in the queried map. |

**MappedObjectType:**MappedObjectType defines a structure for referencing an object and indicating whether it is the source, target, or either for the purposes of query for structure set containing the referenced object in one of the maps it defines.

Derivation:

*com:ReferenceType* (restriction)   
   *com:MaintainableReferenceBaseType* (restriction)   
         com:MaintainableReferenceType (restriction)   
               MappedObjectReferenceType (extension)   
                     MappedObjectType



Attributes:

type?

Content:

( (Ref, URN?) | URN)

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (default: Any) | SourceTargetType | The type attribute indicates whether the referenced object should be queried as the source, target, or both of a map. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | MappedObjectRefType | 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. |

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

Derivation:

*com:ReferenceType* (restriction)   
   *com:MaintainableReferenceBaseType* (restriction)   
         com:MaintainableReferenceType (restriction)   
               MappedObjectReferenceType



Content:

( (Ref, URN?) | URN)

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Ref | MappedObjectRefType | 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. |

**MappedObjectRefType:**MappedObjectRefType defines a set of reference fields for any type of mappable object.

Derivation:

*com:RefBaseType* (restriction)   
   *com:MaintainableRefBaseType* (restriction)   
         com:MaintainableRefType (restriction)   
               MappedObjectRefType



Attributes:

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

Content:

{Empty}

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| agencyID | com: 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 | com:IDType | The id attribute identifies the object being referenced, and is therefore always required. |
| version (default: 1.0) | com: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 | MappedObjectTypeCode 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 | com: 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. |

**StructuresQueryType:**StructuresQueryType defines the structure of a structures query. The parameters for the query are contained in the StructuresWhere element. The References element, typically used to indicate which objects should be returned, is fixed so that all object matching the parameters are returned..

Derivation:

*StructuralMetadataQueryType* (restriction)   
   StructuresQueryType



Content:

ReturnDetails, StructuresWhere

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| ReturnDetails | MaintainableReturnDe tailsType |  |
| StructuresWhere | StructuresWhereType | StructuresWhere defines the parameters for a structures query. All parameters must be matched for an object to satisfy the query. Only the basic maintainable parameters are available. |

**StructuresWhereType:**StructuresWhereType contains a set of parameters for a structures query. All supplied parameters must be matched in order for an object to satisfy the query.

Derivation:

*AnnotableWhereType* (extension)   
   *IdentifiableWhereType* (extension)   
         *NameableWhereType* (extension)   
               *VersionableWhereType* (extension)   
                     *MaintainableWhereType* (restriction)   
                           StructuresWhereType



Attributes:

type?

Content:

Annotation?, URN?, ID?, Name?, Description?, Version?, VersionTo?, VersionFrom?, VersionActive?, AgencyID?

Attribute Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| type (fixed: Any) | com: MaintainableTypeCode listType | The type attribute optionally defines the type of object being queried. For queries for distinct types of objects, a fixed value should be specified in the derived queries. For queries that serve to query for like types of objects, this should be required. |

Element Documentation:

| **Name** | **Type** | **Documentation** |
| --- | --- | --- |
| Annotation | AnnotationWhereType | Annotation is a parameter for matching the details of an annotatable object's annotations. It allows for querying based on the details of an annotation. |
| URN | xs:anyURI | URN is used to match the urn of any SDMX object. |
| ID | QueryIDType | ID is used to match the id of the identified object. |
| Name | QueryTextType | Name is used to match the name of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each name search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Description | QueryTextType | Description is used to match the description of the identified object. It may occur multiple times for its usage within an or-query or for multi-lingual searches, however if multiple values are supplied in an and-query (explicit or implicit), each description search will have to be found in order to constitute a match. The value here can either be an explicit value (exact match) or a regular expression pattern on which to match. |
| Version | com:VersionQueryType | Version is used to match the version of the versioned object. The version can be specified as either an explicit version number, or a late bound query where the latest version of an object will be returned. |
| VersionTo | com: TimeRangeValueType | VersionTo is used to specify a range which the start date of the validity period of version should fall within to create a successful match. |
| VersionFrom | com: TimeRangeValueType | VersionFrom is used to specify a range which the end date of the validity period of version should fall within to create a successful match. |
| VersionActive | xs:boolean | VersionActive is used to request object with active or inactive versions, base on the version validity dates. A value of true indicates that only objects where the current date is within the validity period of the version will be matched. |
| AgencyID | QueryNestedIDType | AgencyID is used to match the agency id of the maintained object. |

### Simple Types

**StructureReturnDetailType:**StructureReturnDetailType contains a set of enumerations that indicate how much detail should be returned for an object.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Stub | Only the identification information and name should be returned. |
| CompleteStub | Identification information, name, description, and annotations should be returned. |
| Full | The entire detail of the object should be returned. |
| MatchedItems | For an item scheme, only the items matching the item where parameters will be returned. In the case that items are hierarchical, the entire hierarchy leading to the matched item will have to be returned. |
| CascadedMatchedItems | For an item scheme, only the items matching the item where parameters, and their hierarchical child items will be returned. In the case that items are hierarchical, the entire hierarchy leading to the matched item will have to be returned. |

**MaintainableReturnDetailType:**MaintainableReturnDetailType contains a sub set of the enumerations defined in the ReturnDetailType. Enumerations relating specifically to item schemes are not included

Derived by restriction of StructureReturnDetailType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Stub | Only the identification information and name should be returned. |
| CompleteStub | Identification information, name, description, and annotations should be returned. |
| Full | The entire detail of the object should be returned. |

**ReportingYearStartDayQueryType:**ReportingYearStartDayQueryType is a simple type for specifying the reporting year start day in a time query parameter. An explicit value or "Any" can be provided.

Union of:

AnyQueryType, xs:gMonthDay.

**AnyQueryType:**AnyQueryType is a single enumeration of the value "Any" which is meant to be used in union with other simple types when a query allows for any of the possible values.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Any |  |

**ConstraintTypeCodelistType:**ConstraintTypeCodelistType defines a list of types for a constraint for the purpose of querying.

Derived by restriction of com:MaintainableTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Constraint |  |
| AttachmentConstraint |  |
| ContentConstraint |  |

**DataReturnDetailType:**DataReturnDetailType contains a set of enumerations that indicate how much detail should be returned for a data set.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Full | The entire data set (including all data, documentation, and annotations) will be returned. |
| DataOnly | Only the observed values and their keys will be returned. Annotations and documentation (i.e. Attributes) and therefore Groups, will be excluded. |
| SeriesKeyOnly | Only the series elements and the values for the dimensions will be returned. Annotations, documentation, and observations will be excluded. |
| NoData | Returns all documentation at the DataSet, Group, and Series level without any Observations (therefore, Observation level documentation is not returned). Annotations are not returned. |

**ObservationActionCodeType:**ObservationActionCodeType enumerates the type of observations to be returned.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Active | Active observations, regardless of when they were added or updated will be returned. |
| Added | Only newly added observations will be returned. |
| Updated | Only updated observations will be returned. |
| Deleted | Only deleted observations will be returned. |

**OrganisationSchemeTypeCodeType:**OrganisationSchemeTypeCodeType enumerates the possible types of organisation schemes that can be queried for.

Derived by restriction of com:MaintainableTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| OrganisationScheme |  |
| AgencyScheme |  |
| DataConsumerScheme |  |
| DataProviderScheme |  |
| OrganisationUnitScheme |  |

**InputOutputTypeCodeType:**InputOutputTypeCodeType enumerates the role an object plays in a process step.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Input | Input - referenced object is an input to the process step. |
| Output | Output - referenced object is an output to the process step. |
| Any | Any - referenced object is either an input or an output to the process step. |

**MappedObjectTypeCodelistType:**MappedObjectTypeCodelistType is a restriction of the MaintainableTypeCodelistType which contains only the object types which can be mapped in a structure set.

Derived by restriction of com:ConcreteMaintainableTypeCodelistType .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| AgencyScheme |  |
| CategoryScheme |  |
| Codelist |  |
| ConceptScheme |  |
| Dataflow |  |
| DataConsumerScheme |  |
| DataProviderScheme |  |
| DataStructure |  |
| HierarchicalCodelist |  |
| Metadataflow |  |
| MetadataStructure |  |
| OrganisationUnitScheme |  |
| ReportingTaxonomy |  |

**SourceTargetType:**SourceTargetType is an enumeration to indicate whether an object is the source, target, or either of the two options.

Derived by restriction of xs:string .

Enumerations:

|  |  |
| --- | --- |
| **Value** | **Documentation** |
| Any |  |
| Source |  |
| Target |  |