



CTS2 Code System and Code System Version Catalog Services

Version 1.0

OMG Document Number: formal/2012-11-02
Standard document URL: <http://www.omg.org/spec/cts2/1.0/>

Copyright © 2012, Mayo Clinic
Copyright © 2012, Object Management Group

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT

LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE.

IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 109 Highland Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™, Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IIOP™, IMM™, MOF™, OMG Interface Definition Language (OMG IDL)™, and OMG Systems Modeling Language (OMG SysML)™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

Table of Contents

1. Introduction	1
1.1 Notation	1
1.2 Identifiers	1
1.3 Stereotypes	1
1.4 Read Only Attributes	2
2. Code System Information Model	3
2.1 Code System Catalog Entry	3
2.1.1 Class CodeSystemCatalogEntry	3
2.1.2 Code System Directory and List	5
3. Code System Catalog Services	7
3.1 Code System Catalog Read Service	7
3.1.1 Interface CodeSystemCatalogReadService	7
3.2 Code System Catalog Query Service	9
3.2.1 Interface CodeSystemCatalogQueryService	9
3.3 Code System Catalog History Service	12
3.3.1 Interface CodeSystemCatalogHistoryService	12
3.4 Code System Catalog Maintenance Service	14
3.4.1 Interface CodeSystemCatalogMaintenanceService	15
4. Code System Version Information Model	17
4.1 Code System Version Catalog Entry	17
4.1.1 Class CodeSystemCatalogEntry	17
4.2 Code System Version Directory and List	19
4.2.1 Class CodeSystemVersionCatalogEntryDirectory	19
4.2.2 Class CodeSystemVersionCatalogEntryList	20
5. Code System Version Catalog Services	21
5.1 Code System Version Catalog Read Services	21
5.1.1 Interface CodeSystemVersionCatalogReadService	21
5.2 Code System Version Catalog Query Services	25
5.2.1 Interface CodeSystemVersionCatalogQueryService	26
5.3 Code System Version Catalog History Services	29

5.3.1 Interface CodeSystemVersionCatalogHistoryService	29
5.4 Code System Version Catalog Maintenance Services	31
5.4.1 Interface CodeSystemVersionCatalogMaintenanceService	31
5.4.2 Class UpdateCodeSystemVersionRequest	33

1 Introduction

The CTS2 specification is divided into a number of separably implementable profiles. This profile describes Code System Catalog Services and Code System Version Catalog Services, which represent metadata about abstract code systems and/or specific versions (releases) of these code systems.

1.1 Notation

Model elements are referenced using Typewriter font.

1.2 Identifiers

- Class names are capitalized camel case (`CodeSystemCatalog`, `String`).
- Attribute and role names are lower camel case (`version`, `sourceAndRole`).
- Enumeration values are all capital letters with underscores ('_') introduced as needed for clarity. (`EMPTY`, `STOP_ON_ERROR`).
- Method names are lowerCamelCase.

1.3 Stereotypes

The following stereotypes are used throughout the model:

- `dataType` - UML `dataType` semantics.
- `enumeration` - UML enumeration semantics.
- `interface` - UML interface semantics.
- `mixin` - a class that provides a certain functionality to be inherited by a subclass, while not meant for instantiation.
- `opt` - an optional parameter in a method call. This stereotype should always be accompanied by a 0..1 cardinality and is provided to make optionality visible in the modeling tool.
- `optparam` - an optional attribute in a class instance. This indicates that an implementation must be able to distinguish three alternatives: (1) add or change the attribute to whatever is in the class, (2) remove the attribute if its minimum cardinality is zero, and (3) leave the attribute unchanged.
- `delta` - a method call that changes the state of the service. Note that the semantics of this stereotype is the inverse of the `is query model attribute`.
- `exception` - an exception.
- `exceptionSet` - a set of exceptions, the members of which are represented by aggregation.

1.4 Read Only Attributes

The CTS2 PIM is based on a RESTful architectural style. Resources can be created, updated, read, and/or deleted, and every resource has one or more immutable characteristics - characteristics that, if changed, would change the identity of the resource itself. In the information model we distinguish these characteristics from those that can be modified by declaring the identifying characteristics as *ReadOnly*. The computational model utilizes a pattern based on this information where the *create* operation(s) supply the identifying characteristics plus any of the non-optional mutable characteristics. The *update* operations then supply a unique identifier and a set of one or more mutable characteristics to be changed.

2 Code System Information Model

2.1 Code System Catalog Entry



Figure 2.1 - Code System Catalog Entry

A code system is a resource that is maintained by individuals and/or organizations, typically has a specific goal or purpose and is published and/or updated at periodic intervals. Its purpose is to declare a collection of codes or identifiers that represent classes, categories, or individuals that are used for reporting, organizing, and/or reasoning about knowledge in some discipline, specialty, or domain. The `CodeSystemCatalogEntry` model carries metadata about the code system itself, while `CodeSystemVersionCatalogEntry` carries information about the content of a code system.

2.1.1 Class `CodeSystemCatalogEntry`

Metadata and access information about a code system.

Superclasses

- Every instance of `CodeSystemCatalogEntry` is also an instance of `AbstractResourceDescription`.

Attributes

- **codeSystemName** - the local identifier that uniquely identifies the code system within the context of the implementing service. Note that the `about` URI is the globally unique identifier.
- **codeSystemCategory** - the category or type of resource that the code system represents.
- **ontologyDomain** - the subject domain of the ontology.
- **ontologyType** - the nature of the content of the ontology.
- **designedForOntologyTask** - the purpose for which the ontology was originally designed.
- **hasOntologyLanguage** - the formal ontology language.
- **includes** - a reference to an external code system that is included as a component of the referenced code system. The Wine Ontology, as an example, includes the Food Ontology as one of its components. Similarly, the US Edition of SNOMED-CT includes the international edition.
- **versions** - a `DirectoryURI` that references the known versions of this code system.
- **currentVersion** - a reference to the code system version marked as `CURRENT` in the service instance.
- **usedOntologyEngineeringTool** - information about a tool used to create the ontology.

Invariants

1. The `resourceID` of a `CodeSystem` is the `codeSystemName`.
2. The described `ResourceType` of a `CodeSystem` is `CODE_SYSTEM`.
3. One of the resource types must be `owl:Ontology` or `skos:ConceptScheme`.
4. `CodeSystems` cannot have workflow status.

2.1.2 Code System Directory and List

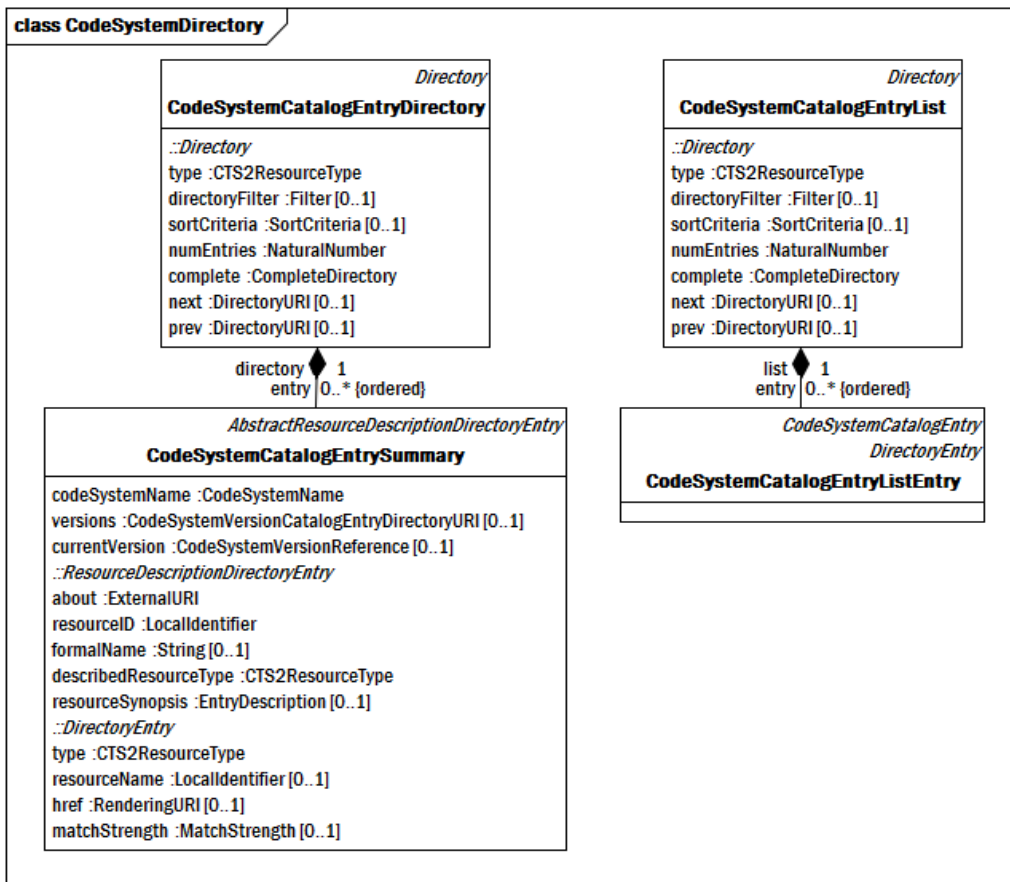


Figure 2.2 - Code System Directory

A directory of all or some of the code systems known to the CTS2 service.

2.1.2.1 Class CodeSystemCatalogEntryDirectory

A directory that summarizes a set of catalog entries.

Superclasses

- Every instance of CodeSystemCatalogEntryDirectory is also an instance of Directory.

Attributes

- **entry** - an entry in the directory.

Invariants

1. The type of a CodeSystemDirectory must always be CODE_SYSTEM.
2. All entries in a code system directory must be type CODE_SYSTEM.

2.1.2.2 Class `CodeSystemCatalogEntryList`

A set of code system catalog entries.

Superclasses

- Every instance of `CodeSystemCatalogEntryList` is also an instance of `Directory`.

Attributes

- **entry**- an entry in the list.

Invariants

1. The type of a `CodeSystemList` must be `CODE_SYSTEM`.
2. All entries in a `CodeSystemList` must be type `CODE_SYSTEM`.

2.1.2.3 Class `CodeSystemCatalogEntryListEntry`

An instance of `CodeSystem` that meets a specified filter criteria.

Superclasses

- Every instance of `CodeSystemCatalogEntryListEntry` is also an instance of `DirectoryEntry`.
- Every instance of `CodeSystemCatalogEntryListEntry` is also an instance of `CodeSystemCatalogEntry`.

Invariants

1. The type of a `CodeSystemListEntry` must be `CODE_SYSTEM`.

2.1.2.4 Class `CodeSystemCatalogEntrySummary`

A synopsis of a `CodeSystem` along with information about how to access the complete resource.

Superclasses

- Every instance of `CodeSystemCatalogEntrySummary` is also an instance of `AbstractResourceDescriptionDirectoryEntry`.

Attributes

- **codeSystemName** - the local identifier that uniquely identifies the code system within the context of the implementing service. Note that the about URI is the globally unique identifier.
- **versions** - a directory URI that resolves to an ordered list of code system versions associated with this code system. Versions may only be present if the service supports the `CODESYSTEMVERSION QUERY` profile.
- **currentVersion** - a reference to the code system version marked as `CURRENT` in the service instance.

Invariants

1. The type of a `CodeSystemDirectoryEntry` must be `CODE_SYSTEM`.

3 Code System Catalog Services

3.1 Code System Catalog Read Service

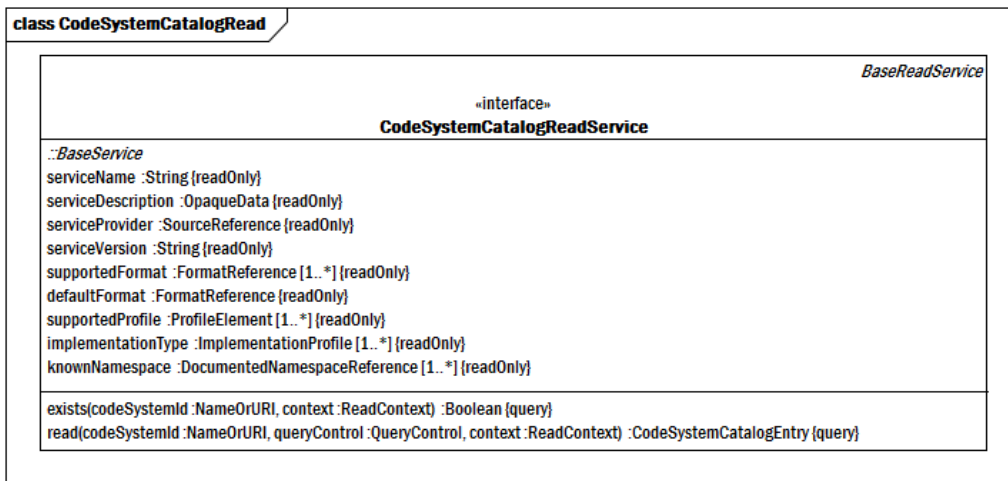


Figure 3.1 - Code System Catalog Read

The `CodeSystemCatalogReadService` provides basic access to the CTS2 code system catalog via a locally unique name or a primary or alternative URI.

3.1.1 Interface `CodeSystemCatalogReadService`

A service that provides direct read access to the contents of the code system catalog.

Superclasses

- Every instance of `CodeSystemCatalogReadService` is also an instance of `BaseReadService`.

Invariants

1. Service must be present in `supportedProfile` list.
2. Every code system name and uri must be unique at a given point in time.
3. The service does not know about change sets unless either the update or authoring profile is supported.
4. The service can never represent a point in time in the future.
5. If the service doesn't support the temporal profile, there can be at most one reference date represented in the service.

3.1.1.1 Operation: `exists`

Determine whether a catalog entry exists that has an `about` or `alternate` Id that matches the supplied URI or that has a `codeSystemName` that matches the supplied name that is (was) applicable in the supplied context.

Input Parameters

- **codeSystemId** - the local name or URI to check for existence (Type: NameOrURI).
- **context_{OPT}** - the context in which the check is performed. (Type: ReadContext).

Return Type: Boolean

Exceptions

- **UnsupportedLanguage** - The referenceLanguage is not supported by the service
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The changeSetContext is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more changeSetContext is not supported by the service.

Preconditions

1. profile and dates are set to SP_CODE_SYSTEM and the known service dates.

Postconditions

1. The return is true if there was a matching code system at the reference time that met the appropriate activity criteria.

3.1.1.2 Operation: read

Return the catalog entry that has an about or alternate Id that matches the supplied URI or that has a codeSystemName that matches the supplied name that is (was) applicable in the supplied context.

Input Parameters

- **codeSystemId** - the local identifier or URI of a code system described in the catalog (Type: NameOrURI).
- **queryControl_{OPT}** - parameters that control the ordering, timeout and query behavior (Type: QueryControl).
- **context_{OPT}** - parameters that control the language, date and other contextual variables (Type: ReadContext).

Return Type: CodeSystemCatalogEntry

Exceptions

- **UnknownCodeSystem** - The referenced CodeSystem is not recognized by the service.
- **UnsupportedLanguage** - The referenceLanguage is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The changeSetContext is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more changeSetContext is not supported by the service.
- **UnsupportedMatchAlgorithm** - The matchAlgorithm is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.

- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.

3.2 Code System Catalog Query Service

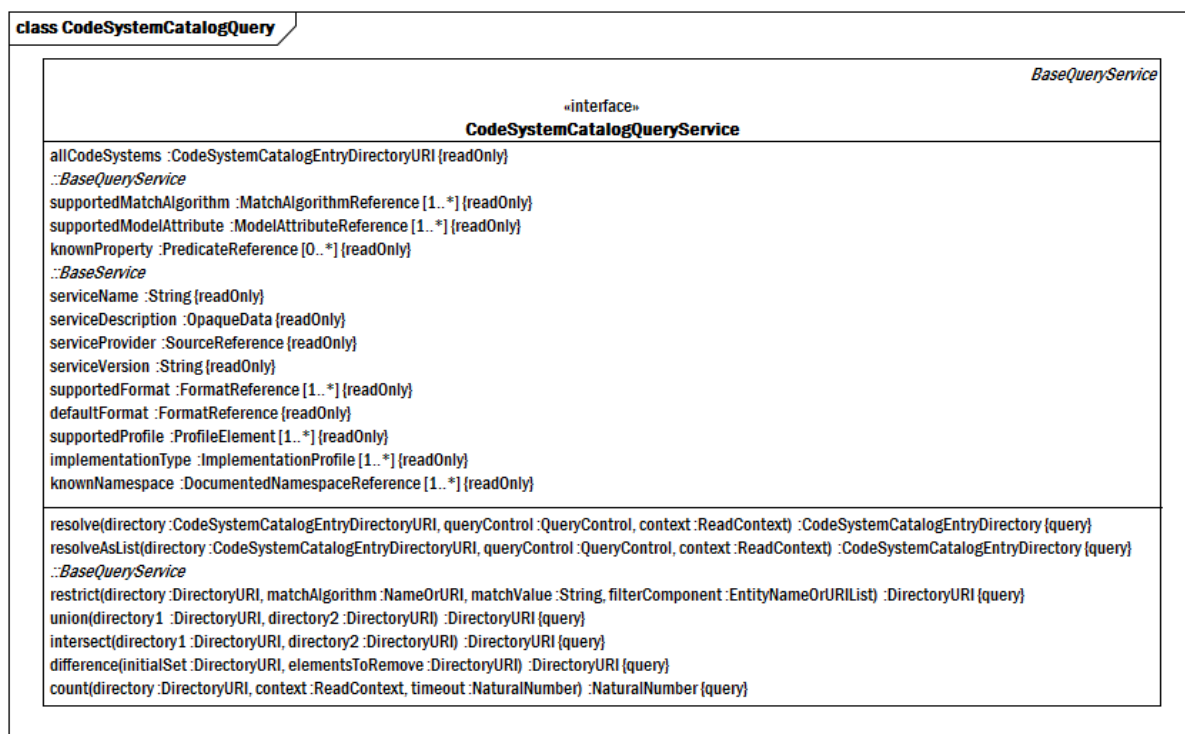


Figure 3.2 - Code System Catalog Query

The `CodeSystemCatalogQueryServices` provides the ability to query code system catalogs based on the contents of the catalog entries.

3.2.1 Interface CodeSystemCatalogQueryService

A service that allows the query and filtering of entries in the code system catalog.

Superclasses

- Every instance of `CodeSystemCatalogQueryService` is also an instance of `BaseQueryService`.

Attributes

- **allCodeSystems** - A `DirectoryURI` that references all of the code system catalog entries known to the service implementation.

3.2.1.1 Operation: resolve

Resolve a `DirectoryURI` that references a set of code system catalog entries and return a summary of the references.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of `CodeSystemCatalogEntries` (Type: `CodeSystemCatalogEntryDirectoryURI`).
- **queryControl**_{OPT} - parameters that control the ordering, timeout and query behavior (Type: `QueryControl`).
- **context**_{OPT} - parameters that control the language, date and other contextual variables (Type: `ReadContext`).

Return Type: `CodeSystemCatalogEntryDirectory`

Exceptions

- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.
- **InvalidDirectoryURI** - The supplied `directory` URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied `directory` URI is not the one required by the service invocation.

3.2.1.2 Operation: resolveAsList

Resolve a `DirectoryURI` that references a set of code system catalog entries and return the set of corresponding catalog entries.

Input Parameters

- **directory** - a DirectoryURI that resolves to a set of CodeSystemCatalogEntries (Type: CodeSystemCatalogEntryDirectoryURI).
- **queryControl**_{OPT} - parameters that control the ordering, timeout and query behavior (Type: QueryControl).
- **context**_{OPT} - parameters that control the language, date and other contextual variables (Type: ReadContext).

Return Type: CodeSystemCatalogEntryDirectory

Exceptions

- **UnsupportedLanguage** - The referenceLanguage is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The changeSetContext is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more changeSetContext is not supported by the service.
- **UnsupportedMatchAlgorithm** - The matchAlgorithm is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized byUnsupportedNamespaceName - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException**-The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The format is not supported by the service implementation.
- **QueryTimeout** - The timeLimit was exceeded by the service.
- **InvalidDirectoryURI** - The supplied directory URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied directory URI is not the one required by the service invocation.

3.3 Code System Catalog History Service

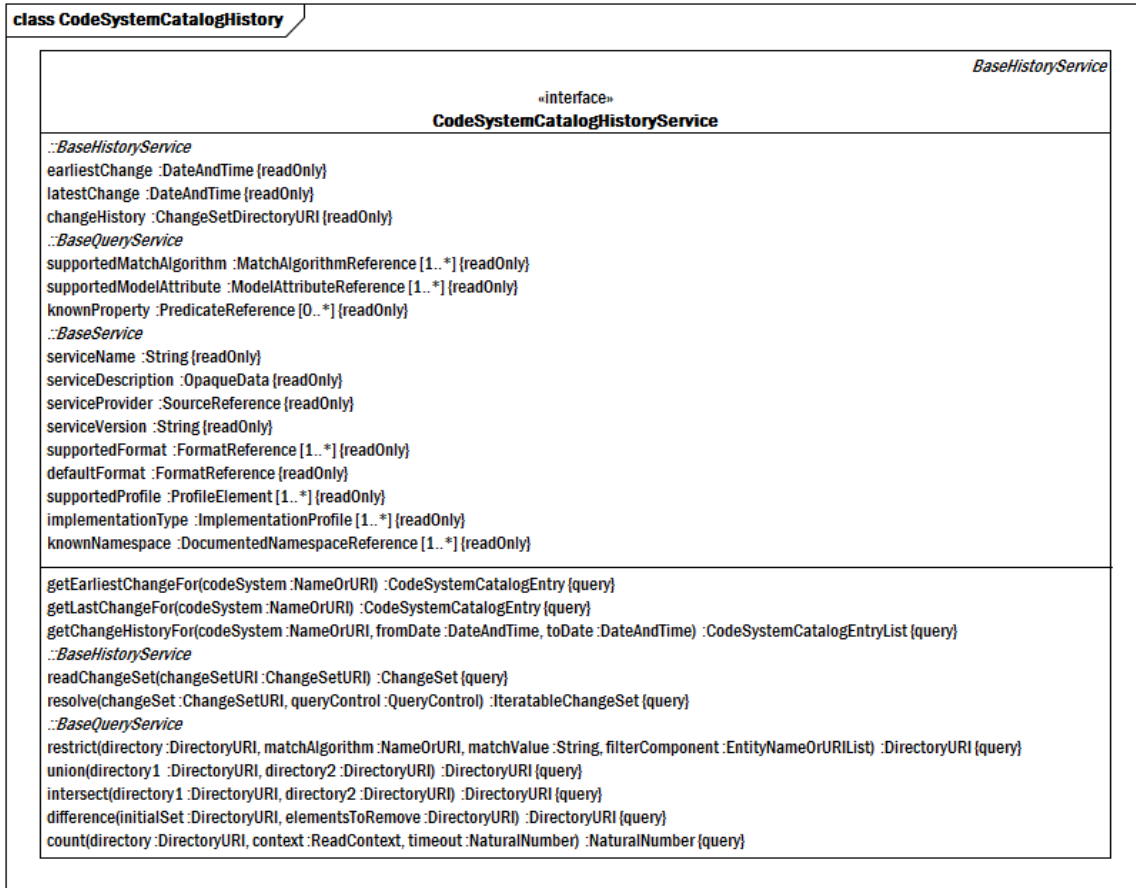


Figure 3.3 - Code System Catalog History

The CodeSystemCatalogHistoryService provides the ability to determine how a catalog entry has evolved over time.

3.3.1 Interface CodeSystemCatalogHistoryService

A service that allows access to the history of changes on a code system catalog.

Superclasses

- Every instance of CodeSystemCatalogHistoryService is also an instance of BaseHistoryService.

3.3.1.1 Operation: getEarliestChangeFor

Return a catalog entry that reflects the state of the entry when it first appeared in the service.

Input Parameters

- **codeSystem** - the name or URI of the code system described by the catalog entry (Type: NameOrURI).

Return Type: `CodeSystemCatalogEntry`

Exceptions

- **UnknownCodeSystem** - The referenced `CodeSystem` is not recognized by the service.

3.3.1.2 Operation: `getLastChangeFor`

Return an entry that reflects the result of the last change that was made to the catalog.

Input Parameters

- **codeSystem** - the name or URI of the code system that the entry describes (Type: `NameOrURI`).

Return Type: `CodeSystemCatalogEntry`

Exceptions

- **UnknownCodeSystem** - The referenced `CodeSystem` is not recognized by the service.

3.3.1.3 Operation: `getChangeHistoryFor`

Return a list of records that reflects the change in the catalog entry over the specified time span. Records will be sorted from earliest to latest if (a) `fromDate` is omitted, or (b) `toDate` is omitted, or (c) `fromDate` is earlier than `toDate`. In all other cases, records will be sorted from latest to earliest.

Input Parameters

- **codeSystem** - a `codeSystemName` or URI that uniquely identifies the code system in question (Type: `NameOrURI`).
- **fromDate_{OPT}** - the earliest to return changes for. If omitted, the history starts at the first known entry. (Type: `DateAndTime`).
- **toDate_{OPT}** - the last date to return changes for. If omitted, all changes up to the last one made are returned. (Type: `DateAndTime`).

Return Type: `CodeSystemCatalogEntryList`

Exceptions

- **UnknownCodeSystem** - The referenced `CodeSystem` is not recognized by the service.

3.4 Code System Catalog Maintenance Service

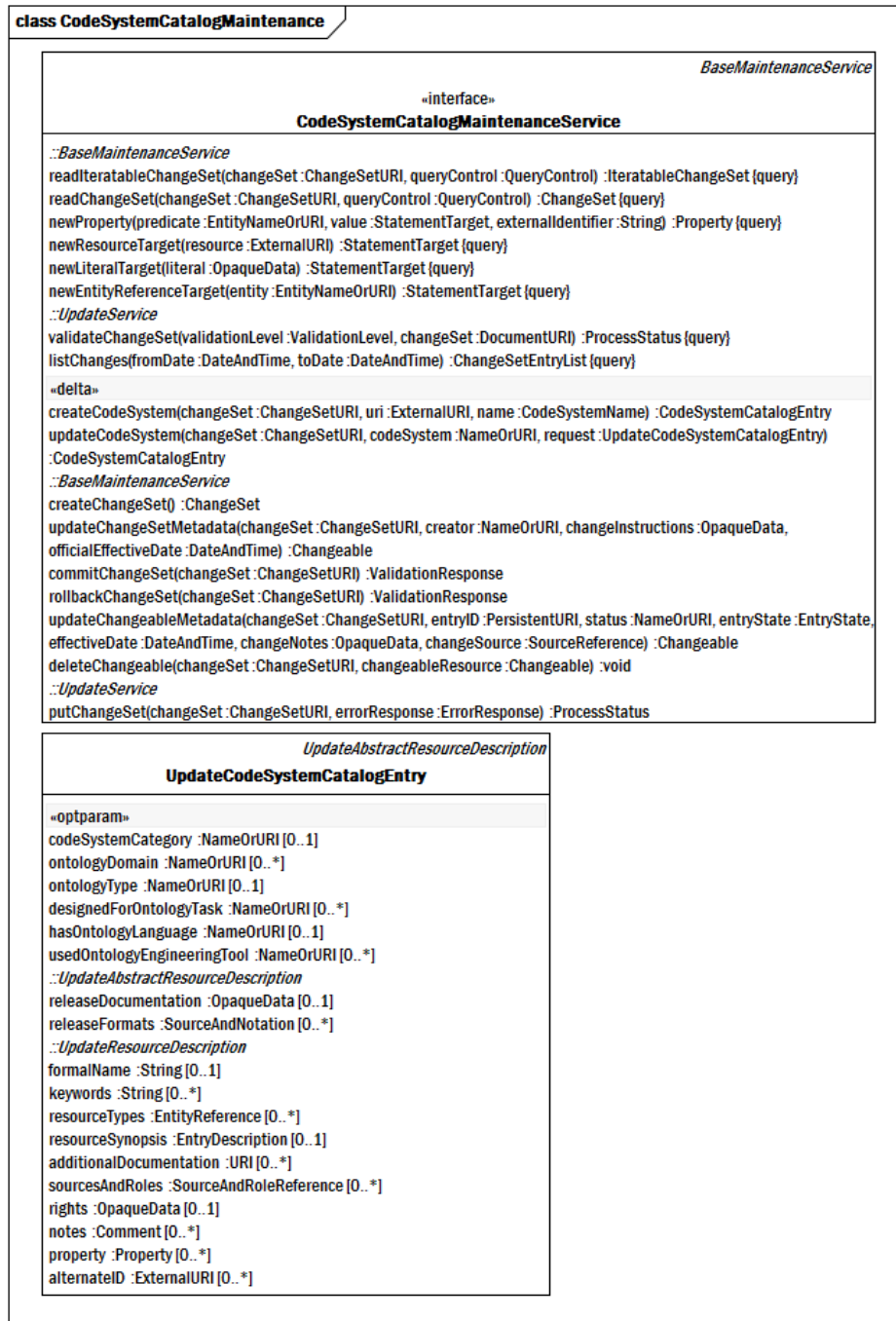


Figure 3.4 - Code System Catalog Maintenance Service

The `CodeSystemCatalogMaintenanceService` allows the creation, modification, and removal of entries in the service catalog.

3.4.1 Interface `CodeSystemCatalogMaintenanceService`

A service that allows the creation and maintenance of entries in a code system catalog.

Superclasses

- Every instance of `CodeSystemCatalogMaintenanceService` is also an instance of `BaseMaintenanceService`.

3.4.1.1 Operation: `createCodeSystem`

Create a new entry in the code system catalog.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set to record the changes (Type: `ChangeSetURI`).
- **uri** - the about URI of the catalog entry to be created (Type: `ExternalURI`).
- **name** - the local identifier of the catalog entry in the context of the service (Type: `CodeSystemName`).

Return Type: `CodeSystemCatalogEntry`

Exceptions

- **AlternateURINotUnique** - One of the supplied `alternateIDs` is not unique - it appears as an `entryID` or `alternateID` for a different resource of the same type.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ResourceNameAlreadyExists** - The local identifier for the resource already exists in the service.
- **ResourceURIAAlreadyExists** - The URI (`entityID`) of the supplied resource already exists in the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.

3.4.1.2 Operation: `updateCodeSystem`

Update the contents of an existing catalog entry.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set to record the changes (Type: `ChangeSetURI`).
- **codeSystem** - the name or URI of the code system to update (Type: `NameOrURI`).
- **request** - the parameters to update (Type: `UpdateCodeSystemCatalogEntry`).

Return Type: `CodeSystemCatalogEntry`

3.4.1.3 Class `UpdateCodeSystemCatalogEntry`

The set of attributes that can be updated in an existing code system catalog entry.

Superclasses

- Every instance of `UpdateCodeSystemCatalogEntry` is also an instance of `UpdateAbstractResourceDescription`.

Attributes

- **`codeSystemCategory`** - the category or type of resource that the code system represents.
- **`ontologyDomain`** - the subject domain of the code system or ontology.
- **`ontologyType`** - the nature of the content of the code system or ontology.
- **`designedForOntologyTask`** - the purpose for which the code system or ontology was originally designed.
- **`hasOntologyLanguage`** - the formal language used to express the code system or ontology.
- **`usedOntologyEngineeringTool`** - The name or URI of an ontology engineering tool.

4 Code System Version Information Model

4.1 Code System Version Catalog Entry



Figure 4.1 - Code System Version Catalog Entry

4.1.1 Class CodeSystemCatalogEntry

`CodeSystemVersion` is a collection of assertions about one or more entities. If the state of a `CodeSystemVersion` is `Final`, the set of assertions made the resource is fixed, meaning that if one knows the `documentURI` assigned to the version and the specific `sourceAndNotation` from which the version instance was derived, the set of assertions is known as well.

Superclasses

- Every instance of `CodeSystemVersionCatalogEntry` is also an instance of `ResourceVersionDescription`.

Attributes

- **codeSystemVersionName** - an identifier that uniquely names this particular code system version within the context of the CTS2 service instance.
- **versionOf** - the name, URI and, if supported by the service, link to the code system that this version represents.
- **imports** - a reference to a code system version whose assertions are included as part of this code system version.
- **defaultLanguage** - the default spoken or written language used in this version.
- **supportedLanguage** - all languages recognized by this particular code system version
- **associations** - a `DirectoryURI` that resolves to a list of all associations asserted by this particular code system version.
- **entityDescriptions** - a `DirectoryURI` that resolves to the set of all entities and their descriptions asserted by this code system version. This attribute will only be present if and only if the service implementation supports the ENTITYQUERY service profile.
- **classes** - a `DirectoryURI` that resolves to the set of `EntityDescriptions` that are of type `owl:Class` within this code system. Will only be present in code systems that represent ontologies.
- **roles** - a `DirectoryURI` that resolves to the set of `EntityDescriptions` that are of type `owl:ObjectProperty` or `owl:DataProperty` within this code system. Will only be present in code systems that represent ontologies.
- **individuals** - a `DirectoryURI` that resolves to the set of `EntityDescriptions` that are of type `owl:Individual` within this code system. Will only be present in code systems that represent ontologies.

Invariants

1. The `codeSystemVersionName` is the unique local identifier of the resource.
2. The `describedResourceType` of `CodeSystemVersion` is `CODE_SYSTEM_VERSION`.
3. A code system version cannot import itself - either directly or indirectly, although this constraint only reflects the direct option.
4. A code system version cannot import another version of the same code system.
5. Every `CodeSystemVersion` must be an instance of `owl:Ontology` and/or `skos:ConceptSystem`.

4.2 Code System Version Directory and List

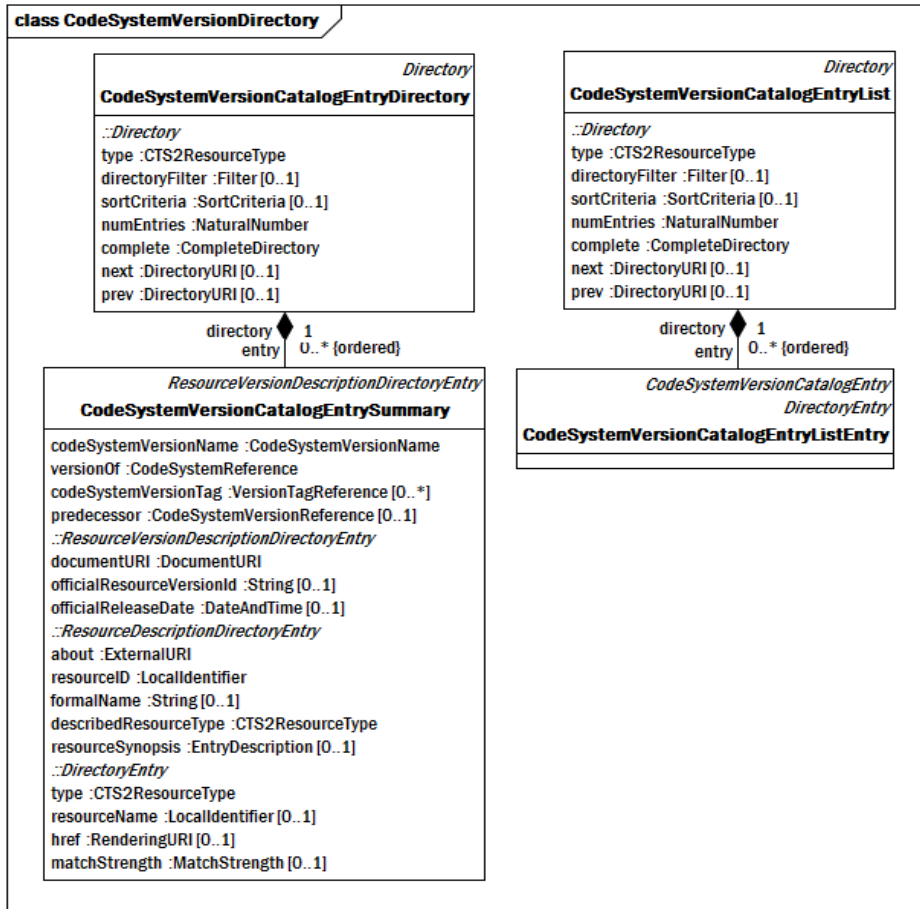


Figure 4.2 - Code System Version Directory

4.2.1 Class CodeSystemVersionCatalogEntryDirectory

A directory of CodeSystemVersion resources that meet a specified criteria.

Superclasses

- Every instance of CodeSystemVersionCatalogEntryDirectory is also an instance of Directory.

Attributes

- entry** - an entry in a catalog

Invariants

- The directory type must be CODE_SYSTEM_VERSION.
- All of the entries in the directory must be of type CODE_SYSTEM_VERSION.

4.2.2 Class `CodeSystemVersionCatalogEntryList`

A collection of complete `CodeSystemVersion` resources that meet a specified criteria.

Superclasses

- Every instance of `CodeSystemVersionCatalogEntryList` is also an instance of `Directory`.

Attributes

- **entry** - an entry in a catalog list.

Invariants

1. The resource type must be `CODE_SYSTEM_VERSION`.
2. All entries in a code system version list must be type `CODE_SYSTEM_VERSION`.

4.2.2.1 Class `CodeSystemVersionCatalogEntryListEntry`

An instance of `CodeSystemVersion` that meets a specified filter criteria.

Superclasses

- Every instance of `CodeSystemVersionCatalogEntryListEntry` is also an instance of `DirectoryEntry`.
- Every instance of `CodeSystemVersionCatalogEntryListEntry` is also an instance of `CodeSystemVersionCatalogEntry`.

Invariants

1. The type of a `CodeSystemVersionListEntry` must be `CODE_SYSTEM_VERSION`.

4.2.2.2 Class `CodeSystemVersionCatalogEntrySummary`

A synopsis of a `CodeSystemVersion` along with information about how to access the complete resource.

Superclasses

- Every instance of `CodeSystemVersionCatalogEntrySummary` is also an instance of `ResourceVersionDescriptionDirectoryEntry`.

Attributes

- **codeSystemVersionName** - an identifier that uniquely names this particular code system version within the context of the CTS2 service instance.
- **versionOf** - the name, URI, and, if supported by the service, link to the code system that this version represents.
- **codeSystemVersionTag** - the tag(s) assigned to this version by the service provider. `codeSystemVersionTag` identifies any roles that this version plays within the workflow of the current provider. As an example, a provider might assign the tags of “production” and “current” to a given code system version. With the exception of `CURRENT` the semantics of `VersionTag` are determined by the service provider.
- **predecessor** - a reference to the code system version that immediately preceded the one described here.

Invariants

1. The type of a `CodeSystemVersionDirectoryEntry` must be `CODE_SYSTEM_VERSION`.

5 Code System Version Catalog Services

5.1 Code System Version Catalog Read Services

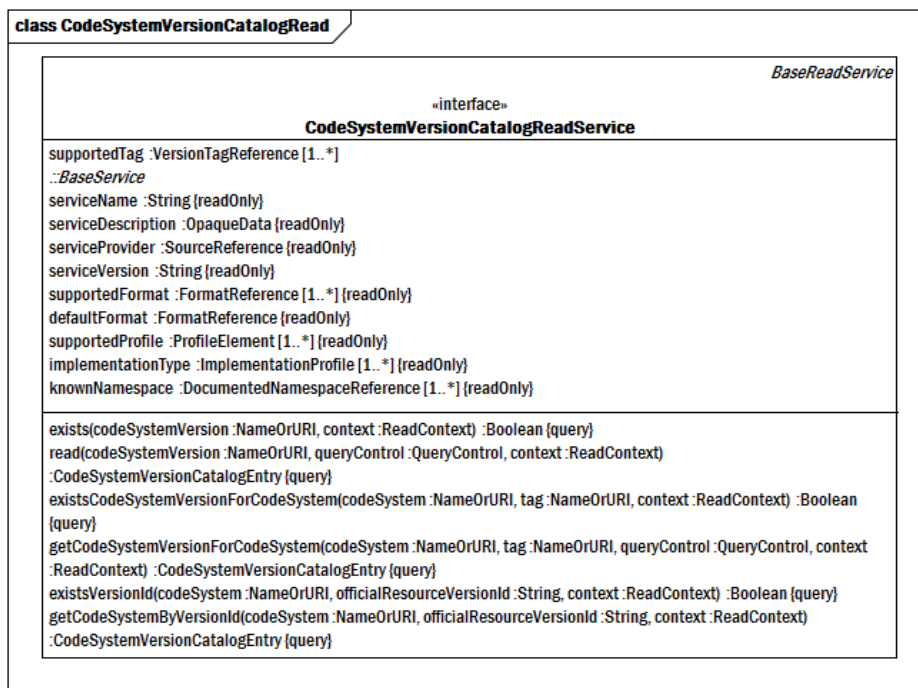


Figure 5.1 - Code System Version Catalog Read

5.1.1 Interface CodeSystemVersionCatalogReadService

A service that provides direct read access to metadata about specific versions of code systems.

Superclasses

- Every instance of **CodeSystemVersionCatalogReadService** is also an instance of **BaseReadService**.

Attributes

- supportedTag** - The set of version tags known to the service. All services must recognize the **CURRENT** tag, although interpretation is service specific.

Invariants

- All services must support the CTS2 **CURRENT** tag, which is the default that will be used when a tag is not supplied in a function call.
- Every code system version represented in the service must have a unique **codeSystemName** and a unique **documentURI**.
- Every supported tag name must be unique.

4. Every tag in `taggedVersion` must have an entry in `supportedTag`.
5. There can be at most one version of a given code system associated with a given tag.

5.1.1.1 Operation: exists

Return TRUE if the `CodeSystemVersion` is known to the service with a matching name or URI.

Input Parameters

- **codeSystemVersion** - the local code system version name or URI (Type: `NameOrURI`).
- **context** - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: `Boolean`

Exceptions

- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.

5.1.1.2 Operation: read

Retrieve the specified code system version details from the service.

Input Parameters

- **codeSystemVersion** - the local code system version name or URI (Type: `NameOrURI`).
- **queryControl_{OPT}** - parameters that control the ordering, timeout, and query behavior (Type: `QueryControl`).
- **context** - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: `CodeSystemVersionCatalogEntry`

Exceptions

- **UnknownCodeSystem** - The referenced `CodeSystem` is not recognized by the service.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.

- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The format is not supported by the service implementation.
- **QueryTimeout** - The timeLimit was exceeded by the service.

Postconditions

- Returns the unique value in knownSystemVersion whose codeSystemVersionName matches the supplied name or whose uri matches the supplied URI.

5.1.1.3 Operation: existsCodeSystemVersionForCodeSystem

Determine if the specified version of the code system exists for the specified code system on the service.

Input Parameters

- **codeSystem** - the name or URI of the target code system (Type: NameOrURI).
- **tag_{OPT}** - the name or URI of a version tag that connects a code system to a code system version. Defaults to CURRENT (Type: NameOrURI).
- **context_{OPT}** - parameters that control the language, date, and other contextual variables (Type: ReadContext).

Return Type: Boolean

Exceptions

- **UnsupportedVersionTag** - The versionTag is not recognized by the service.
- **UnsupportedLanguage** - The referenceLanguage is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The changeSetContext is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more changeSetContext is not supported by the service.

5.1.1.4 Operation: getCodeSystemVersionForCodeSystem

Retrieve the specified code system version details for the specified code system from the service.

Input Parameters

- **codeSystem** - Either the local code system version name or the URI of the resource to be accessed (Type: NameOrURI).
- **tag_{OPT}** - The version tag of the resource. Defaults to the URI for CURRENT if not supplied. (Type: NameOrURI)
- **queryControl** - Operation timeout setting (Type: QueryControl).
- **context** - Context of the access (Type: ReadContext).

Return Type: `CodeSystemVersionCatalogEntry`

Exceptions

- **UnsupportedVersionTag** - The `versionTag` is not recognized by the service.
- **UnknownCodeSystem** - The referenced `CodeSystem` is not recognized by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.

5.1.1.5 Operation: `existsVersionId`

Determine if the specified version of the code system exists for the specified external identifier.

Input Parameters

- **codeSystem** - the local code system version name or the URI of a code system (Type: `NameOrURI`).
- **officialResourceVersionId** - the official external version identifier assigned to the target version (Type: `String`).
- **context**_{OPT} - parameters that control the language, date and other contextual variables (Type: `ReadContext`).

Return Type: `Boolean`

Exceptions

- **UnknownCodeSystem** - The referenced `CodeSystem` is not recognized by the service.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.

5.1.1.6 Operation: `getCodeSystemByVersionId`

Retrieve the specified version of the code system exists for the specified external identifier.

Input Parameters

- **codeSystem** - the local code system version name or the URI of a code system (Type: `NameOrURI`).

- **officialResourceVersionId** - the official external version identifier assigned to the target version (Type: String).
- **context_{OPT}** - parameters that control the language, date, and other contextual variables (Type: ReadContext).

Return Type: `CodeSystemVersionCatalogEntry`

Exceptions

- **UnknownCodeSystemVersion** - The referenced `CodeSystemVersion` is not recognized by the service.
- **UnknownCodeSystem** - The referenced `CodeSystem` is not recognized by the service.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.

5.2 Code System Version Catalog Query Services

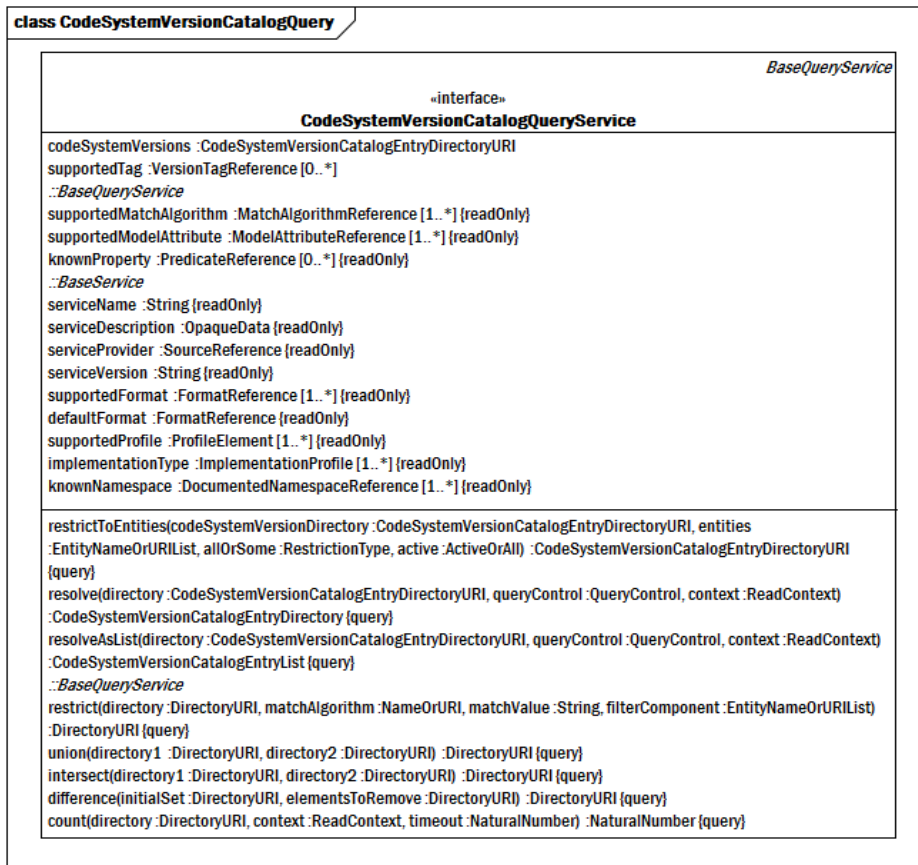


Figure 5.2 - Code System Version Catalog Query

5.2.1 Interface **CodeSystemVersionCatalogQueryService**

Superclasses

- Every instance of **CodeSystemVersionCatalogQueryService** is also an instance of **BaseQueryService**.

Attributes

- **codeSystemVersions** - a **DirectoryURI** that resolves to the set of versions known to the service.
- **supportedTag** - a version tag that is supported by this service instance.

5.2.1.1 Operation: **restrictToEntities**

Retrieve a code system version directory containing the specified entities.

Input Parameters

- **codeSystemVersionDirectory** - a **DirectoryURI** that resolves to a set of code system version catalog entries (Type: **CodeSystemVersionCatalogEntryDirectoryURI**).
- **entities** - a set of namespace/names or URIs for the filter entities (Type: **EntityNameOrURIList**).
- **allOrSome** - an indicator that determines whether any or all of the entities must be included in the code system for it to qualify (Type: **RestrictionType**).
- **active** - an indicator that determines whether just active entities count as a filter or any entity qualifies (Type: **ActiveOrAll**).

Return Type: CodeSystemVersionCatalogEntryDirectoryURI

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnknownEntity** - The **EntityNameOrURI** is not known to the service.
- **InvalidDirectoryURI** - The supplied directory URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied directory URI is not the one required by the service invocation.

5.2.1.2 Operation: **resolve**

Retrieve the specified code system version directory.

Input Parameters

- **directory** - a **DirectoryURI** that resolves to a set of code system versions (Type: **CodeSystemVersionCatalogEntryDirectoryURI**).
- **queryControl**_{OPT} - parameters that control the ordering, timeout, and query behavior (Type: **QueryControl**).
- **context**_{OPT} - parameters that control the language, date, and other contextual variables (Type: **ReadContext**).

Return Type: CodeSystemVersionCatalogEntryDirectory

Exceptions

- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.
- **InvalidDirectoryURI** - The supplied `directory` URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied `directory` URI is not the one required by the service invocation.

5.2.1.3 Operation: resolveAsList

Resolve the supplied `DirectoryURI` and return the set of catalog entries it represents.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of code system versions (Type: `CodeSystemVersionCatalogEntryDirectoryURI`).
- **queryControl**_{OPT} - parameters that control the ordering, timeout, and query behavior (Type: `QueryControl`).
- **context**_{OPT} - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: CodeSystemVersionCatalogEntryList

Exceptions

- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.

- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The format is not supported by the service implementation.
- **QueryTimeout** - The timeLimit was exceeded by the service.
- **InvalidDirectoryURI** - The supplied directory URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied directory URI is not the one required by the service invocation.

5.3 Code System Version Catalog History Services



Figure 5.3 - Code System Version Catalog History

5.3.1 Interface **CodeSystemVersionCatalogHistoryService**

A service that allows the viewing of changes that have occurred to a **CodeSystemVersionCatalogEntry** over time.

Superclasses

- Every instance of **CodeSystemVersionCatalogHistoryService** is also an instance of **BaseHistoryService**.

5.3.1.1 Operation: **getEarliestChangeFor**

Return the catalog entry that reflects the state of the entry when it first became available to the service.

Input Parameters

- **codeSystemVersion** - the local name or URI of a code system version known to the system (Type: **NameOrURI**).

Return Type: CodeSystemVersionCatalogEntry

Exceptions

- **UnknownCodeSystemVersion** - The referenced **CodeSystemVersion** is not recognized by the service.

5.3.1.2 Operation: **getLastChangeFor**

Return the catalog entry that reflects the result of the last change that was made to the referenced entry.

Input Parameters

- **codeSystemVersion** - the local name or URI of a code system version (Type: **NameOrURI**).

Return Type: CodeSystemVersionCatalogEntry

Exceptions

- **UnknownCodeSystemVersion** - The referenced **CodeSystemVersion** is not recognized by the service.

5.3.1.3 Operation: **getChangeHistoryFor**

Return a list of records that reflects changes in the catalog entry over the specified time span. Records will be sorted from earliest to latest if (a) **fromDate** is omitted or (b) **toDate** is omitted or (c) **fromDate** is earlier than **toDate**. In all other cases, records will be sorted from the latest to the earliest.

Input Parameters

- **codeSystemVersion** - then name or URI of a code system version. (Type: **NameOrURI**)
- **fromDate_{OPT}** - the date and time that the historical record begins. If omitted, the history starts at the earliest change. (Type: **DateAndTime**)
- **toDate_{OPT}** - the date and time that the historical record ends. If omitted, the history continues to the latest change. (Type: **DateAndTime**)

Return Type: `CodeSystemVersionCatalogEntryList`

Exceptions

- **UnknownCodeSystemVersion** - The referenced `CodeSystemVersion` is not recognized by the service.

5.4 Code System Version Catalog Maintenance Services

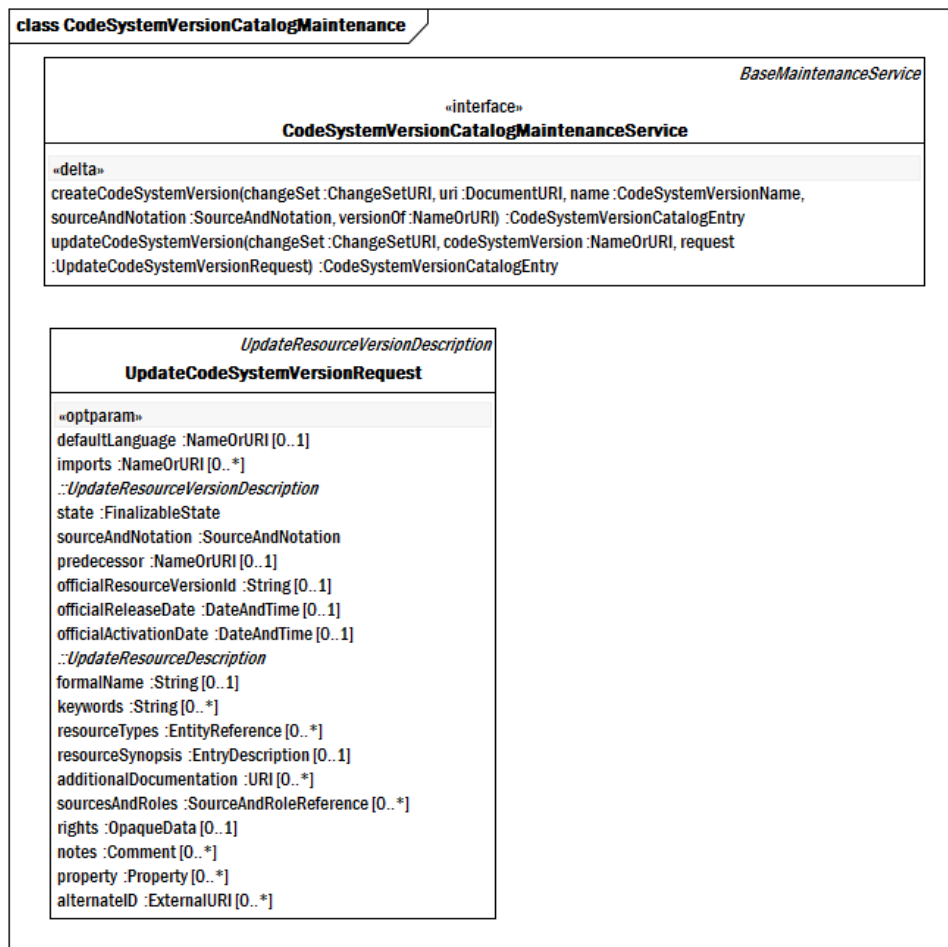


Figure 5.4 - Code System Version Catalog Maintenance

This diagram describes the services that are used to create, update and remove code system catalog entries. Note: the inherited methods and attributes have been excluded from the diagram.

5.4.1 Interface `CodeSystemVersionCatalogMaintenanceService`

Superclasses

- Every instance of `CodeSystemVersionCatalogMaintenanceService` is also an instance of `BaseMaintenanceService`.

5.4.1.1 Operation: createCodeSystemVersion

Create a new code system version catalog entry. Note that the `versionOf` parameter does not have to be validated as there is no service level requirement that code system catalogs be complete or even present.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set that will record this change. (Type: `ChangeSetURI`)
- **uri** - the URI that uniquely identifies the new code system version (Type: `DocumentURI`).
- **name** - a name that uniquely identifies this version in the context of the implementing service. (Type: `CodeSystemVersionName`)
- **sourceAndNotation** - the source and notation used in this code system version. (Type: `SourceAndNotation`)
- **versionOf** - the name or URI of the code system that this is a version of. (Type: `NameOrURI`)

Return Type: `CodeSystemVersionCatalogEntry`

Exceptions

- **UnsupportedSource** - The supplied source is not recognized by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **UnsupportedOntologyLanguage** - The supplied ontology language is not supported by the service.
- **DuplicateCodeSystemVersionName** - The supplied code system version name is already used for another entry.
- **DuplicateCodeSystemVersionURI** - The supplied code system version URI is already a `documentURI` or `alternateID` for an existing code system version.

5.4.1.2 Operation: updateCodeSystemVersion

Update an existing code system version catalog entry.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set where the update will be recorded. (Type: `ChangeSetURI`)
- **codeSystemVersion** - the name or URI of the code system version to be updated. (Type: `NameOrURI`)
- **request** - the parameters to be updated. (Type: `UpdateCodeSystemVersionRequest`)

Return Type: `CodeSystemVersionCatalogEntry`

Exceptions

- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.

- **UnknownCodeSystemVersion** - The referenced `CodeSystemVersion` is not recognized by the service.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **ResourceIsNotOpen** - The target resource version description has been finalized and cannot be updated.
- **UnsupportedOntologyLanguage** - The supplied ontology language is not supported by the service.
- **UnsupportedStatus** - The name or URI of the `ChangeableStatus` property is not recognized by the service.
- **UnsupportedSource** - The supplied source is not recognized by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedRole** - The role name or URI is not recognized by the service.
- **UnsupportedOntologySyntax** - The supplied ontology syntax is not supported by the service.
- **CycleInPredecessor** - The predecessor resource directly or indirectly precedes the resource being updated.
- **WrongPredecessorType** - The predecessor URI is not of the same type as its successor.

5.4.2 Class `UpdateCodeSystemVersionRequest`

Superclasses

- Every instance of `UpdateCodeSystemVersionRequest` is also an instance of `UpdateResourceVersionDescription`.

Attributes

- **defaultLanguage** - the name or URI of the default language used in this version.
- **imports** - the name or URI of a code system version whose content is imported and “asserted by” this code system version.