

# HL7 Common Terminology Services 2 Service Functional Model (SFM)

Service Functional Model Specification

Common Terminology Services Release 2 (CTS 2)

Version 1.0

23-July-2008

Project Leads	Russell Hamm <a href="mailto:rhamm@apelon.com">rhamm@apelon.com</a> (Apelon, Inc.) Craig Stancl <a href="mailto:stancl.craig@mayo.edu">stancl.craig@mayo.edu</a> (Mayo Clinic/Foundation)
Principal Contributors	Davera Gabriel <a href="mailto:davera@ucdavis.edu">davera@ucdavis.edu</a> (University of California, Davis) Russell Hamm <a href="mailto:rhamm@apelon.com">rhamm@apelon.com</a> (Apelon, Inc.) Senthil Nachimuthu <a href="mailto:snachimuthu@mmm.com">snachimuthu@mmm.com</a> (3M Health Information Systems) Craig Stancl <a href="mailto:stancl.craig@mayo.edu">stancl.craig@mayo.edu</a> (Mayo Clinic/Foundation)
Contributors	John Carter <a href="mailto:jcarter@apelon.com">jcarter@apelon.com</a> (Apelon, Inc.) Kristi Eckerson <a href="mailto:keckers@sph.emory.edu">keckers@sph.emory.edu</a> (Emory University) Sundak Ganesan <a href="mailto:dsq3@CDC.GOV">dsq3@CDC.GOV</a> (CDC) Stanley Huff, M.D. <a href="mailto:stan.huff@ihc.com">stan.huff@ihc.com</a> (Intermountain Health Care)
HL7 Vocabulary Co-Chairs	Heather Grain <a href="mailto:heather@lginformatics.com">heather@lginformatics.com</a> () Russell Hamm <a href="mailto:rhamm@apelon.com">rhamm@apelon.com</a> (Apelon, Inc.) Ted Klein <a href="mailto:ted@tklein.com">ted@tklein.com</a> (Klein Consulting) Beverly Knight <a href="mailto:beverly.knight@procurement.ca">beverly.knight@procurement.ca</a> (Canada Health Infoway)

11  
12 **Preface**

13 **Notes to Readers**

14 This document is the Service Functional Model for the Common Terminology Services 2  
15 specification, which is specified under the Service Development Framework process under the  
16 auspices of the Healthcare Services Specification Project (HSSP). Further context is given in the  
17 overview section below, but one key point to note is that the SFM provides a Service **Interface**  
18 specification, NOT the specification of a Service implementation. This is a critical distinction in  
19 terms of Service Oriented Architecture. There could be many different ways of implementing all  
20 or part of the functionality to support the behavior described in this specification.

21  
22 **NOTE: For the purposes of this specification, the terms *vocabulary* and *terminology* are**  
23 **used interchangeably.**

24  
25 **Changes from Previous Release**

26 This is the first public release of this document.

27  
28 **Acknowledgments**

29 This document is the result of the collaboration of many individuals and organizations. The  
30 terminology and standards community - all involved in the numerous meetings and  
31 teleconferences are to be thanked for their contributions and support. In addition to the listed  
32 authors, the following individuals are acknowledged for their contributions and support during  
33 the development of this specification.

- 34 • The HL7 Vocabulary Technical Committee
- 35 • Christopher Chute, M.D., PhD. (Mayo Clinic/Foundation)
- 36 • John Koisch (OCTL Consulting)
- 37 • Ken Rubin (EDS)
- 38 • Tony Weida, PhD. (Apelon, Inc.)

39  
40 **Note:** Sections of this document in **blue** indicate text that is consistent across HSSP  
41 specifications.

42

# Contents

- [1 Overview](#)
  - [1.1 Introduction and Scope](#)
    - [1.1.1 HL7-OMG Healthcare Services Specification Project \(HSSP\)](#)
    - [1.1.2 Service Definition Principles](#)
  - [1.2 Overall disclaimers](#)
  - [1.3 Context of this SFM within HSSP Roadmap](#)
- [2 Service Overview and Business case](#)
  - [2.1 Service Overview](#)
    - [2.1.1 CTS 2 Service Description and Purpose](#)
    - [2.1.2 Scope](#)
  - [2.2 The reason why the service is necessary](#)
  - [2.3 Structure of the CTS 2 Service](#)
  - [2.4 Implementation Considerations](#)
    - [2.4.1 Interface Interoperability Considerations](#)
    - [2.4.2 Terminology Structure Considerations](#)
      - [2.4.2.1 Code System](#)
      - [2.4.2.2 Code System Version](#)
      - [2.4.2.3 Concept](#)
      - [2.4.2.4 Designation](#)
      - [2.4.2.5 Concept Property](#)
      - [2.4.2.6 Concept Association](#)
      - [2.4.2.7 Value Set](#)
- [3 Business Scenarios](#)
  - [3.1 Scenario Actors](#)
  - [3.2 Primary Scenarios](#)
    - [3.2.1 Administrative Scenarios](#)
      - [3.2.1.1 Import Content](#)
      - [3.2.1.2 Export Content](#)
      - [3.2.1.3 Remove Content](#)
      - [3.2.1.4 Change Content Status](#)
      - [3.2.1.5 Update Notification](#)
      - [3.2.1.6 Update Notification Management](#)
      - [3.2.1.7 Content Dependency Notification](#)
    - [3.2.2 Search / Query Scenarios](#)
      - [3.2.2.1 Code System Search / Query](#)
        - [3.2.2.1.1 Resolve Available Code Systems](#)
        - [3.2.2.1.2 Retrieve Coded Concepts from Code System](#)
        - [3.2.2.1.3 Validate Concept in Code System](#)
        - [3.2.2.1.4 Identify Concept Language Translations](#)
        - [3.2.2.1.5 Resolve Concept Representations](#)
        - [3.2.2.1.6 Compare Code System Versions](#)
      - [3.2.2.2 Value Set Search / Query](#)
        - [3.2.2.2.1 Resolve Available Value Sets](#)
        - [3.2.2.2.2 Retrieve Coded Concepts from Value Set](#)

- [3.2.2.2.3 Validate Coded Concept in Value Set](#)
- [3.2.2.2.4 Compare Value Set Versions](#)
- [3.2.2.2.5 Resolve Concept Representations](#)
- [3.2.3 Authoring / Curation Scenarios](#)
  - [3.2.3.1 Code System Authoring / Curation](#)
    - [3.2.3.1.1 Create Code System](#)
    - [3.2.3.1.2 Maintain Code System](#)
    - [3.2.3.1.3 Create Concept](#)
    - [3.2.3.1.4 Maintain Concept](#)
    - [3.2.3.1.5 Deprecate Concept](#)
  - [3.2.3.2 Value Set Authoring / Curation](#)
    - [3.2.3.2.1 Create Value Set by Intension](#)
    - [3.2.3.2.2 Create Value Set by Extension](#)
    - [3.2.3.2.3 Maintain Value Set \(Definition\)](#)
    - [3.2.3.2.4 Maintain Value Set \(Enumeration\)](#)
  - [3.2.3.3 Change Request Processing](#)
    - [3.2.3.3.1 Create Change Request](#)
    - [3.2.3.3.2 Edit Change Request](#)
    - [3.2.3.3.3 Submit Change Request](#)
    - [3.2.3.3.4 Package Change Request](#)
- [3.2.4 Association Scenarios](#)
  - [3.2.4.1 Association Administrative Scenarios](#)
    - [3.2.4.1.1 Enumerate Code System Coded Concept Relationship Types](#)
    - [3.2.4.1.2 Identify / Retrieve Concept Associations for a Single Concept](#)
    - [3.2.4.1.3 Identify / Retrieve Associations Between Two or More Coded Concepts](#)
    - [3.2.4.1.4 Import Coded Concept Associations](#)
    - [3.2.4.1.5 Export Coded Concept Associations](#)
    - [3.2.4.1.6 Remove Coded Concept Associations](#)
    - [3.2.4.1.7 Change Status of Coded Concept Associations](#)
    - [3.2.4.1.8 Register for Association Update Notification](#)
  - [3.2.4.2 Association Search / Query Scenarios](#)
    - [3.2.4.2.1 Resolve Available Associations](#)
    - [3.2.4.2.2 Validate Associations](#)
    - [3.2.4.2.3 Retrieve Association Metadata](#)
    - [3.2.4.2.4 Compare Association Versions](#)
    - [3.2.4.2.5 Request / Retrieve Association Instance](#)
  - [3.2.4.3 Association Author / Curation Scenarios](#)
    - [3.2.4.3.1 Create / Maintain an Association between Coded Concepts](#)
    - [3.2.4.3.2 Create Relationship Type](#)
    - [3.2.4.3.3 Create Lexical Association](#)
    - [3.2.4.3.4 Create Rules Based Association](#)
    - [3.2.4.3.5 Validate Relationship Associations between](#)

- [3.2.4.3.6 Validate Map Associations Between Coded Concepts](#)
- [4 Assumptions and Dependencies](#)
  - [4.1 Dependencies on other Service Frameworks](#)
  - [4.2 CTS Backwards Compatibility](#)
    - [4.2.1 Message API Support \(MAPI\)](#)
    - [4.2.2 General CTS API Support](#)
    - [4.2.3 HL7 Datatypes](#)
- [5 Functional Overloading and Metadata Discovery](#)
  - [5.1 Functional Overloading](#)
  - [5.2 Metadata Discovery](#)
- [6 Detailed Functional Model for each Interface](#)
  - [6.1 Administration Functions](#)
    - [6.1.1 Import Terminology](#)
    - [6.1.2 Import Terminology Revision](#)
    - [6.1.3 Export Terminology](#)
    - [6.1.4 Remove Terminology / Terminology Version](#)
    - [6.1.5 Change Terminology Status](#)
    - [6.1.6 Convert Terminology Format](#)
    - [6.1.7 Register for Update Notification](#)
    - [6.1.8 Revise or Remove Update Notification](#)
    - [6.1.9 Register for Concept Dependency Notification](#)
    - [6.1.10 Revise or Remove Concept Dependency Notification](#)
  - [6.2 Search / Access](#)
    - [6.2.1 Code System Search / Access](#)
    - [6.2.2 Resolve Available Code Systems](#)
    - [6.2.3 Resolve Code System Metadata](#)
    - [6.2.4 Resolve Code System Concepts](#)
    - [6.2.5 Resolve Concept Details](#)
    - [6.2.6 Resolve Coded Concept from Code System](#)
    - [6.2.7 Resolve Relations Between Coded Concepts](#)
    - [6.2.8 Compare Code Systems](#)
    - [6.2.9 Compare Code System Contents](#)
  - [6.3 Value Set Search / Access](#)
    - [6.3.1 Compare Value Sets](#)
    - [6.3.2 Compare Value Set Contents](#)
    - [6.3.3 Resolve Available Value Sets](#)
    - [6.3.4 Resolve Value Set Metadata](#)
    - [6.3.5 Resolve Value Set Entries](#)
    - [6.3.6 Resolve Value Set Entry](#)
  - [6.4 Authoring/Curation](#)
    - [6.4.1 Code System Authoring/Curation](#)
      - [6.4.1.1 Create Code System](#)
      - [6.4.1.2 Maintain Code System](#)
      - [6.4.1.3 Create Concept](#)

- [6.4.1.4 Maintain Concept](#)
- [6.4.1.5 Deprecate Concept](#)
- [6.4.1.6 Create Relationship Type](#)
- [6.4.2 Value Set Authoring/Curation](#)
  - [6.4.2.1 Create Value Set by Intension](#)
  - [6.4.2.2 Create Value Set by Extension](#)
  - [6.4.2.3 Maintain Value Set \(Intension\)](#)
  - [6.4.2.4 Maintain Value Set \(Extension\)](#)
- [6.4.3 Change Request Processing](#)
  - [6.4.3.1 Create Change Request](#)
  - [6.4.3.2 Edit Change Request](#)
  - [6.4.3.3 Submit Change Request](#)
  - [6.4.3.4 Package Change Request](#)
- [6.5 Code System Relationships and Maps](#)
  - [6.5.1 Concept Relationships](#)
    - [6.5.1.1 Resolve Available Concept Relationships](#)
    - [6.5.1.2 Retrieve Concept Relationships for a Single Coded Concept](#)
    - [6.5.1.3 Retrieve Concept Relationships Between Two Coded Concepts](#)
    - [6.5.1.4 Retrieve Concept Relationship Metadata](#)
    - [6.5.1.5 Import Concept Relationship](#)
    - [6.5.1.6 Import Concept Relationship Revision](#)
    - [6.5.1.7 Export Concept Relationship](#)
    - [6.5.1.8 Import Concept Relationship Metadata](#)
    - [6.5.1.9 Remove Concept Relationship Version](#)
    - [6.5.1.10 Change Concept Relationship Status](#)
    - [6.5.1.11 Create Concept Relationship Between Coded Concepts](#)
    - [6.5.1.12 Create Lexical Relationship Between Coded Concepts](#)
    - [6.5.1.13 Create Rules Based Relationship Between Coded Concepts](#)
    - [6.5.1.14 Compare Relationships Between Coded Concepts](#)
    - [6.5.1.15 Compare Metadata Between Relationships](#)
    - [6.5.1.16 Validate Relationships Between Coded Concepts](#)
    - [6.5.1.17 Validate Lexical Based Relationships Between Coded Concepts](#)
    - [6.5.1.18 Validate Rules Based Relationships Between Coded Concepts](#)
    - [6.5.1.19 Register For Concept Relationship Update Notification](#)
    - [6.5.1.20 Revise or Remove Concept Relationship Update Notification](#)
    - [6.5.1.21 Register For Concept Dependency Relationship Notification](#)
    - [6.5.1.22 Revise or Remove Concept Dependency Relationship Notification](#)
  - [6.5.2 Concept Maps](#)

- [6.5.2.1 Resolve Available Concept Maps](#)
- [6.5.2.2 Retrieve Concept Maps for a Specified Coded Concept](#)
- [6.5.2.3 Retrieve Concept Maps Between Multiple Coded Concepts](#)
- [6.5.2.4 Retrieve Map Metadata](#)
- [6.5.2.5 Import Map](#)
- [6.5.2.6 Import Concept Map Metadata](#)
- [6.5.2.7 Convert Mapping Format](#)
- [6.5.2.8 Import Map Revision](#)
- [6.5.2.9 Export Map](#)
- [6.5.2.10 Export Map Metadata](#)
- [6.5.2.11 Remove Map Version](#)
- [6.5.2.12 Change Map Status](#)
- [6.5.2.13 Create Map Between Coded Concepts](#)
- [6.5.2.14 Create Lexical Mapping Between Coded Concepts](#)
- [6.5.2.15 Create Rules Based Mapping Between Coded Concepts](#)
- [6.5.2.16 Compare Maps Between Coded Concepts](#)
- [6.5.2.17 Compare Metadata Between Maps](#)
- [6.5.2.18 Validate Mappings Between Coded Concepts](#)
- [6.5.2.19 Validate Lexical Based Mappings Between Coded Concepts](#)
- [6.5.2.20 Validate Rules Based Mappings Between Coded Concepts](#)
- [6.5.2.21 Register for Concept Dependency Map Notification](#)
- [6.5.2.22 Revise or Remove Map Update Notification](#)
- [6.5.2.23 Register For Concept Dependency Map Notification](#)
- [6.5.2.24 Revise or Remove Concept Dependency Map Notification](#)
- [7 Profiles](#)
  - [7.1 Introduction](#)
  - [7.2 CTS 2 Functional Profiles](#)
    - [7.2.1 Minimal CTS 2 Profile](#)
    - [7.2.2 Vocabulary Facilitator Profile](#)
    - [7.2.3 Terminology Administration Profile](#)
    - [7.2.4 Terminology Authoring Profile](#)
  - [7.3 CTS 2 Semantic Profiles](#)
    - [7.3.1 Mature Terminology Profile](#)
    - [7.3.2 Developing Terminology Profile](#)
  - [7.4 CTS 2 Conformance Profiles](#)
    - [7.4.1 Conformance Interoperability](#)
    - [7.4.2 Conformance Assertion](#)
- [8 The Services Framework Functional Model](#)
- [9 Relationship to Information Content](#)
- [10 Recommendations for Technical RFP Issuance](#)
  - [10.1 Semantic Signifiers: Disparate Terminologies](#)
    - [10.1.1 Semantic Signifiers: HL7 Terminologies](#)
  - [10.2 Conformance Profiles and Service Level Agreements](#)
  - [10.3 Operationalizing CTS 2: Considerations in Implementation](#)
    - [10.3.1 Optimization](#)

- [10.3.2 Internationalization](#)
- [10.4 Service Description and Discovery](#)
- [10.5 Federated Terminologies](#)
- [10.6 Terminology Structure Considerations](#)
- [10.7](#)
- [11 Appendix A - Relevant Standards](#)
  - [11.1 HL7 Common Terminology Services](#)
  - [11.2 The Lexical Grid](#)
- [12 Appendix B – Glossary](#)
  - [12.1 Actor](#)
  - [12.2 Code System](#)
  - [12.3 Concept Map](#)
  - [12.4 Concept Relationship](#)
  - [12.5 Nested Value Sets](#)
  - [12.6 Value Set](#)
  - [12.7 Value Set Specification](#)
  - [12.8 Nested Value Sets](#)
- [13 Appendix C - HL7 EHR Functional Model Traceability](#)

## 43 **Overview**

### 44 **Introduction and Scope**

45 The Service Specification Development Framework Methodology is the methodology followed  
46 to define HSSP specifications. The methodology sets out an overall process, and also defines the  
47 responsibilities of the Service Functional Model (SFM). Section 2 sets out the business context  
48 for this particular specification, but firstly it is important to understand the overall context within  
49 which this specification is written, i.e. its purpose from a methodology standpoint.

50

### 51 **HL7-OMG Healthcare Services Specification Project (HSSP)**

52 The Healthcare Services Specification Project (HSSP) [<http://hssp.wikispaces.com>] is a joint  
53 endeavor between Health Level Seven (HL7) [<http://www.hl7.org>] and the Object Management  
54 Group (OMG) [<http://www.omg.org>]. The HSSP was chartered at the January 2005 HL7 meeting  
55 under the Electronic Health Records Technical Committee, and the project was subsequently  
56 validated by the Board of Directors of both organizations.

57 The HSSP has several objectives. These objectives include the following:

- 58 • To stimulate the adoption and use of standardized “plug-and-play” services by healthcare  
59 software product vendors



- 60 • To facilitate the development of a set of implementable interface standards supporting  
61 agreed-upon services specifications to form the basis for provider purchasing and  
62 procurement decisions.
- 63 • To complement and not conflict with existing HL7 work products and activities,  
64 leveraging content and lessons learned from elsewhere within the organization.

65 Within the process, HL7 has primary responsibility for (1) identifying and prioritizing services as  
66 candidates for standardization; (2) specifying the functional requirements and conformance  
67 criteria for these services in the form of Service Functional Model (SFM) specifications such as  
68 this document; and (3) adopting these SFMs as balloted HL7 standards. These activities are  
69 coordinated by the HL7 Services Oriented Architecture SIG in collaboration with other HL7  
70 committees, which currently include the Vocabulary TC and the Clinical Decision Support TC.

71 Based on the HL7 SFMs, OMG will develop “Requests for Proposals” (RFPs) that are the basis  
72 of the OMG standardization process. This process allows vendors and other submitters to  
73 propose solutions that satisfy the mandatory and optional requirements expressed in the RFP  
74 while leaving design flexibility to the submitters and implementation flexibility to the users of  
75 the standard. The result of this collaboration is an RFP Submission, which will be referred to in  
76 the HSSP process as a Service Technical Model (STM). HL7 members, content, and concerns  
77 are integral to this process, and will explicitly included in the RFP creation and evaluation  
78 process.

79 It is important to note that the HL7 SFMs specify the *functional* requirements of a service, the  
80 OMG RFPs specify the *technical* requirements of a service, and the STM represents the resulting  
81 technical model, except as specified below. In many cases, SFMs describe an overall coherent  
82 set of functional capabilities and / or define a minimum set of behaviors necessary to guarantee a  
83 minimal level of service in a deployment scenario. These capabilities may be specialized or  
84 subdivided from both functional and informational (semantic) perspectives to provide  
85 conformance “profiles” that may be used as the basis for the OMG RFP process and/or  
86 implemented.

87

## 88 **Service Definition Principles**

89 The high level principles regarding service definition that have been adopted by the Services  
90 Specification Project are as follows:

- 91 • Service Specifications shall be well defined and clearly scoped and with well understood  
92 requirements and responsibilities.
- 93 • Services should have a unity of purpose (e.g., fulfilling one domain or area) but services  
94 themselves may be composable.
- 95 • Services will be specified sufficiently to address functional, semantic, and structural  
96 interoperability.
- 97 • It must be possible to replace one conformant service implementation with another  
98 meeting the same service specification while maintaining functionality of the system.

99 A Service at the SFM level is regarded as a system component; the meaning of the term  
100 “(system) component” in this context is consistent with UML usage<sup>[11]</sup>. A component is a modular  
101 unit with well-defined interfaces that is replaceable within its environment. A component can  
102 always be considered an autonomous unit within a system or subsystem. It has one or more  
103 provided and/or required interfaces, and its internals are hidden and inaccessible other than as  
104 provided by its interfaces.

105 Each Service’s Functional Model defines the interfaces that the service exposes to its  
106 environment, and the service’s dependencies on services provided by other components in its  
107 environment. Dependencies in the Functional Model relate to services that have or may in future  
108 have a Functional Model at a similar level; detail dependencies on low-level utility services  
109 should not be included, as that level of design is not in scope for the Functional Model.

110 The manner in which services and interfaces are deployed, discovered, and so forth is outside the  
111 scope of the Functional Model. However, HSSP Functional Models may reference content from  
112 other areas of HSSP work that deals with architecture, deployment, naming and so forth. Except  
113 where explicitly specified, these references are to be considered informative only. All other  
114 interactions within the scope of the scenarios identified above are in the scope of the Functional  
115 Model.

116 Reference may be made to other specifications for interface descriptions, for example where an  
117 interface is governed by an existing standard.

## 118 **Overall disclaimers**

- 119 • Examples are illustrative and not normative unless otherwise specified
- 120 • The scope of information content of HSSP service specifications is not limited to HL7  
121 content models. At a minimum, however, specifications should provide a semantic profile  
122 as part of its conformance profile to provide support for HL7 content models where  
123 applicable.

## 124 **Context of this SFM within HSSP Roadmap**

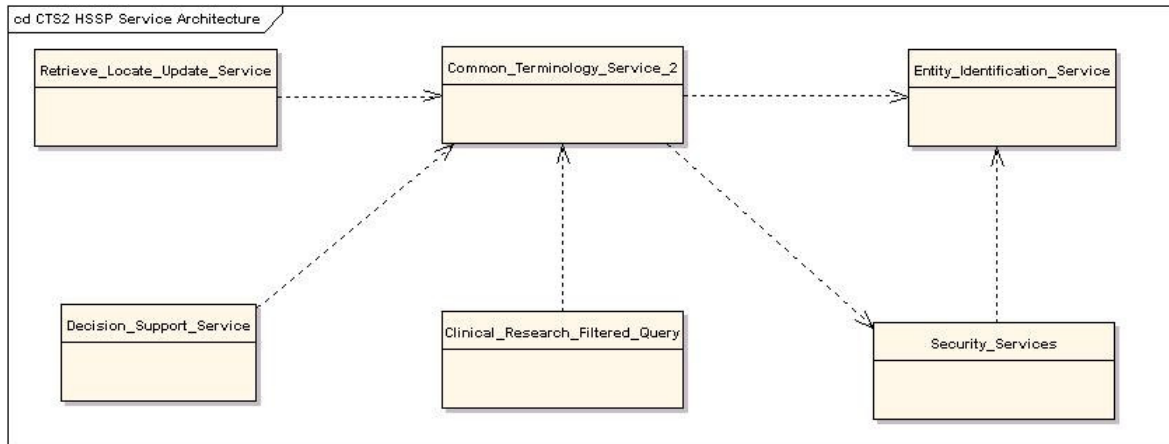
125 As described above, the purpose of an HL7 SFM is to identify and document the functional  
126 requirements of services important to healthcare. Accordingly, the CTS 2 service provides a  
127 critical component within the larger context of service specifications in that it defines both the  
128 expected behaviors of a terminology service and a standardized method of accessing terminology  
129 content. This consistent approach to terminology interaction will benefit other business context  
130 services by providing a level of terminology interoperability that currently only exists in a  
131 limited form.

132 Once adopted as an HL7 standard, it is anticipated that the CTS 2 service will serve as the basis  
133 for one or more OMG technical specifications. It is expected that CTS 2 will effectively leverage  
134 other HSSP specifications to enhance overall functionality in integration environments. In

135 particular, the CTS 2 service is expected to interact with one or more infrastructure services as  
136 outlined below.

137 At a minimum, it is expected that CTS 2 will be made available via an Entity Identification  
138 Service, which in turn references a set of Security Services. CTS 2 itself will make use of the  
139 Security Services to implement its own functional profile restrictions. Additionally, services such  
140 as a Decision Support Service, Clinical Research Functional Query, and Resource Locate and  
141 Update Service may find the use of the CTS 2 service a key resource in improving content  
142 disambiguation.

143



144

145 This specification will provide an important foundation component for many healthcare  
146 interoperability scenarios, both within and across organizations. Although in many business  
147 scenarios CTS 2 may be used in conjunction with other services, it has been specified to provide  
148 stand alone capabilities when referenced solely for terminology access and management  
149 purposes.  
150

## 151 Service Overview and Business case

### 152 Service Overview

#### 153 CTS 2 Service Description and Purpose

154 The goal of the Common Terminology Services 2 (CTS 2) Specification is to expand on the  
155 original functionality outlined in HL7's Common Terminology Service (CTS) Specification.  
156 CTS 2 defines the functional requirements of a set of service interfaces to allow the  
157 representation, access, and maintenance of terminology content either locally, or across a  
158 federation of terminology service nodes.

159 The CTS 2 specification strives to expand on the original functionality outlined in HL7's  
160 Common Terminology Service specification, specifically looking to:

- 161 1. Establish the minimal common structural model for terminology behavior, and how it is  
162 related to meta-data (information about data) and data (the information itself)
- 163 2. Integrate into CTS 2 the functional coverage outlined in the existing CTS specification.
- 164 3. Specify both an information and functional model that addresses the relationships and use  
165 of terminology, e.g. how value sets are built and queried, and how terminological  
166 information is validated.
- 167 4. Specify the interactions between terminology providers and consumers – how  
168 terminology users can submit unambiguous requests for corrections and extensions and  
169 how revisions to content are identified, distributed and integrated into running systems.
- 170 5. Specify how mapping between compatible terminologies and data models is defined,  
171 exchanged and revised.
- 172 6. Specify how logic-based terminologies can be queried about subsumption and inferred  
173 relationships.
- 174 7. Engage broad community participation to describe the dimensions of use and purpose for  
175 vocabularies and value sets. This aim will attempt to harmonize these efforts in terms of  
176 models, use cases, and requirements for creating a functional model for CTS 2.

## 177 **Scope**

178 To address the above stated purpose of CTS2, the scope of functionality addresses several broad  
179 categories.

180 Terminology services represent functions necessary to manage, search, and access terminology  
181 content. Terminology services provide a consistent specification for using terminology content  
182 independent of the terminology content and underlying technology stack. Terminology content  
183 represents various resources including lists, value sets, taxonomies, and formal description logic  
184 based ontologies. The following thematic areas are considered in scope for CTS 2.

185

- 186 • **Administration:** This is a set of functionality that provides the ability to manage content  
187 as part of a terminology service. Administration functions include the ability to load  
188 terminologies, export terminologies, activate terminologies, and retire terminologies.  
189 These functions are generally protected and accessible by service administrators with  
190 appropriate authorization.
- 191 • **Search / Query:** This is a set of functionality that provides the ability to find concepts  
192 based on some search criteria. This includes restrictions to specific associations or other  
193 attributes of the terminology, including navigation of associations for result sets. This  
194 represents the primary utility for using terminology content in a number of application  
195 contexts.
- 196 • **Authoring / Maintenance:** This is a set of functionality that provides the ability to create  
197 and maintain content. From a terminology service perspective, this would include the  
198 appropriate APIs to add, change, or delete concepts and associations. This would also  
199 include the processing of change events from various terminology providers.

- **Associations:** This is a set of functionality that provides the ability to map concepts and the concept's associated attributes from a source terminology to a concept in a target terminology, or create relationships between concepts within a single code system.

CTS 2 is intended to allow the look up and management of a wide variety of terminology components, including, but not limited to, Concepts, Associations, and Value Sets. At the functional level, the service interface will explicitly allow the query, definition, publication, and modification of the different terminology components that are required of terminologies and terminology services.

Conformance profiles will be defined which may limit specific implementations of CTS 2 to a specific class of functionality and pre-define minimum trait sets for each specified functionality class, such as query, authoring and mapping. This will also allow for performance optimizations to be defined for terminology searches and queries (which are implementation considerations which will be considered in the technical specification arising from the OMG RFP process.) The scope of this functional specification covers support for multiple terminology sources and a federated terminology environment.

## **The reason why the service is necessary**

The original HL7 CTS specification deliberately steered clear of developing a generic model of terminology, and avoided issues related to terminology distribution and versioning. The value set, or sub-setting section of CTS focused on static value sets and didn't fully address the definition or resolution of value sets that define post-coordinated expressions – issues that are now in scope due to the maturing Terminfo/SNOMED Concept Model Working Group (CMWG) model.

Adopting organizations have recognized the existing HL7 CTS standard serves an important role in defining the common functional characteristics that a terminology service (either internal or external) must be able to provide. However, these organizations are also realizing that CTS fails to address many of the issues that are required for a truly interoperable terminology service.

While CTS defined a standard API to access terminology source content, in practice it is often necessary to implement those APIs specific to the target source terminology being accessed. This is necessary because CTS does not – by intention - define a “normal form” model that the terminology content can be represented in. Controlled terminologies are developed with specific purposes and use cases in mind. As such, different terminology sources define different model attributes and structure, specific to the purpose and intent of each source.

CTS 2 as a commonly accepted standard for terminology services, will enhance the capabilities of the initial CTS specification for sub-setting and mapping, and extend the specification into domains such as terminology distribution, authoring, versioning, and classification. Standardizing the functionality at this level will allow applications using terminology services to build on a common infrastructure, and improve interoperability at the terminology layer across applications.

239 CTS 2 will provide the terminology community with a defined set of standards interfaces that  
240 can be used to evaluate terminology source structure, terminology source content, and  
241 terminology tools.

## 242 **Structure of the CTS 2 Service**

243 In order to provide for the maximum implementation flexibility, this functional model defines  
244 several enumerated functional profiles for CTS 2. These profiles each identify a subset of the  
245 CTS 2 available functionality as pertinent to a specific semantic profile. These profiles include:

- 246 • **Minimal CTS 2 Profile** - The minimal functional coverage necessary for a service to  
247 declare itself as being a conformant CTS 2 service. The minimal CTS 2 includes  
248 capabilities for searching and query terminology content, representing terminology  
249 content in the appropriate HL7 Datatypes, and structuring terminology content  
250 appropriately when HL7 Datatypes are not available for representing the necessary  
251 terminology content being queried (i.e. value sets.)
- 252 • **Vocabulary Facilitator Profile** - The ability for Vocabulary Facilitators to create,  
253 modify, package and submit change requests to a Terminology Provider. Change requests  
254 to the terminology do not modify the terminology content directly, but result in a  
255 collaborative community consensus recommendation to the Terminology Provider that  
256 outlines a requested modification to the source terminology. These change requests can  
257 then be reviewed by the Terminology Provider, and when appropriate, included in the  
258 next release of the source terminology.
- 259 • **Terminology Administration Profile** - The functional operations necessary for  
260 terminology administrators to be able to access and make available terminology content  
261 obtained from a Terminology Provider. Terminology Administrators are required to  
262 interface with Terminology Provider systems in order to obtain the terminology content,  
263 then load that terminology content on local Terminology Servers.
- 264 • **Terminology Authoring Profile** - The functional operations necessary for terminology  
265 authors to analyze the existing terminology content, as well as directly edit terminology  
266 content.

267

268 The degree to which an organization's interoperability deployment supports a conformance  
269 profile, then, is directly related to the other agreements implemented with a business partner. A  
270 single CTS 2 service may respond to different real-world business partners depending on the  
271 underlying agreements and needs. For example, an organization may implement a CTS 2  
272 (Authoring) compliant service with a trusted partner (i.e., a Terminology Provider). A separate  
273 partner may only be allowed CTS 2 (Minimal) access to the content from that Terminology  
274 Provider as dictated by other factors.

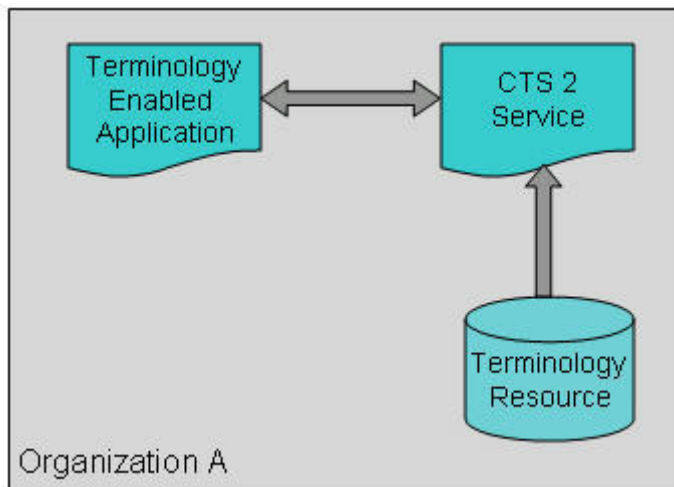
275 Additionally, CTS 2 explicitly makes no distinctions at the functional level regarding semantics  
276 of the underlying systems. Instead, it provides for a semantic profile as part of CTS 2  
277 conformance profiles. This allows definition, publication, and discovery of vital semantic  
278 artifacts between sharing partners through CTS 2 interfaces without requiring strict, tightly  
279 coupled integration. Thus, CTS 2 does not preclude a strategy for semantic interoperability to be  
280 realized, though it would likely depend on other factors (for example, a security service and / or  
281 an entity identification service). This improves CTS 2 as an interoperability mechanism by  
282 relegating the issue of semantic interoperability to the trading partners, allowing semantic  
283 transformations to be performed at the least cost for the most derived value.

## 284 **Implementation Considerations**

### 285 **Interface Interoperability Considerations**

286 CTS 2 is an interface specification, not an implementation specification. As such, it is intended  
287 to be an interoperability mechanism for terminology resources between applications. There is  
288 nothing inherent in the CTS 2 specification that restricts its use to be within a single  
289 organization. To the contrary, CTS 2 is intended to expose a single or multiple terminology  
290 sources for use by various applications that may or may not be within the same organization,  
291 providing a standardized method for terminology access.

292



293

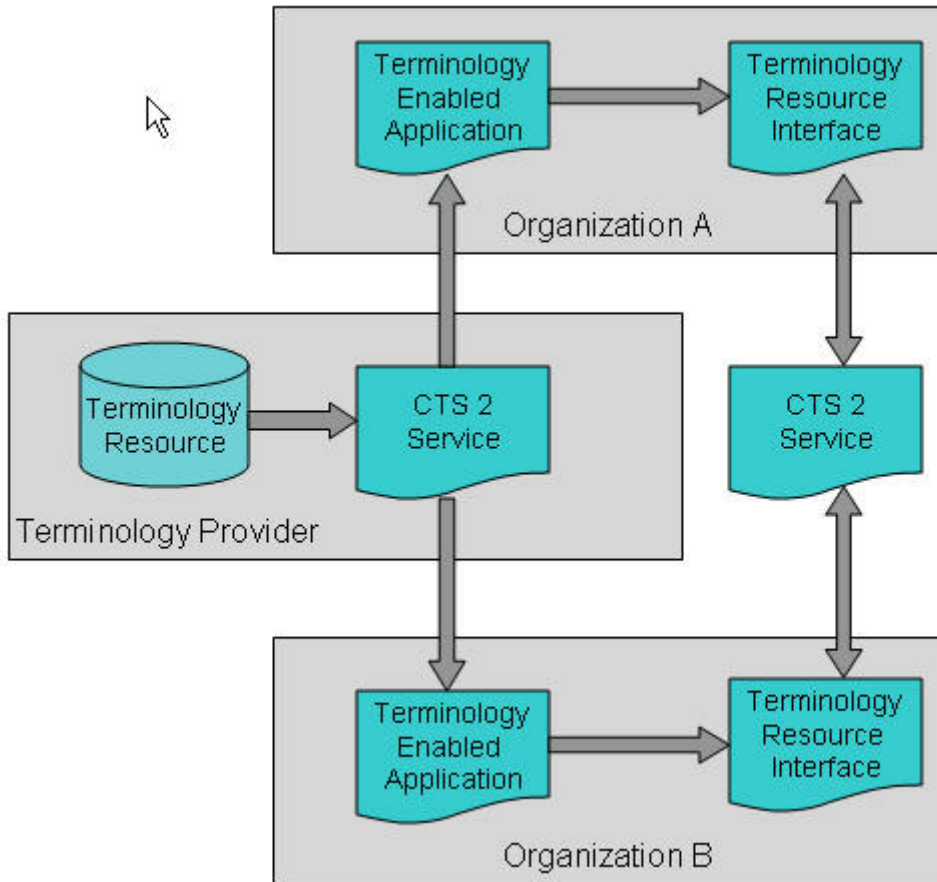
294 **Figure 2.4-1 CTS 2 Service Accessed by a Single Organization**

295

296 CTS 2 will provide for terminology interoperability between organizations. While coded  
297 concepts from structured terminology can unambiguously identify the concept(s) being  
298 communicated, a standard way of structuring and communicating those coded entries is required.

299  
300 CTS 2 can be used in an inter-organizational setting where each organization maintains its own  
301 security and application specific provisions. CTS 2 will enable consistent access to a high  
302 availability or international standard terminology resource, made available to subscribers via a  
303 CTS 2 interface.

304



305

306 **Figure 2.4-2 Multi Organizational access to a CTS 2 Service**

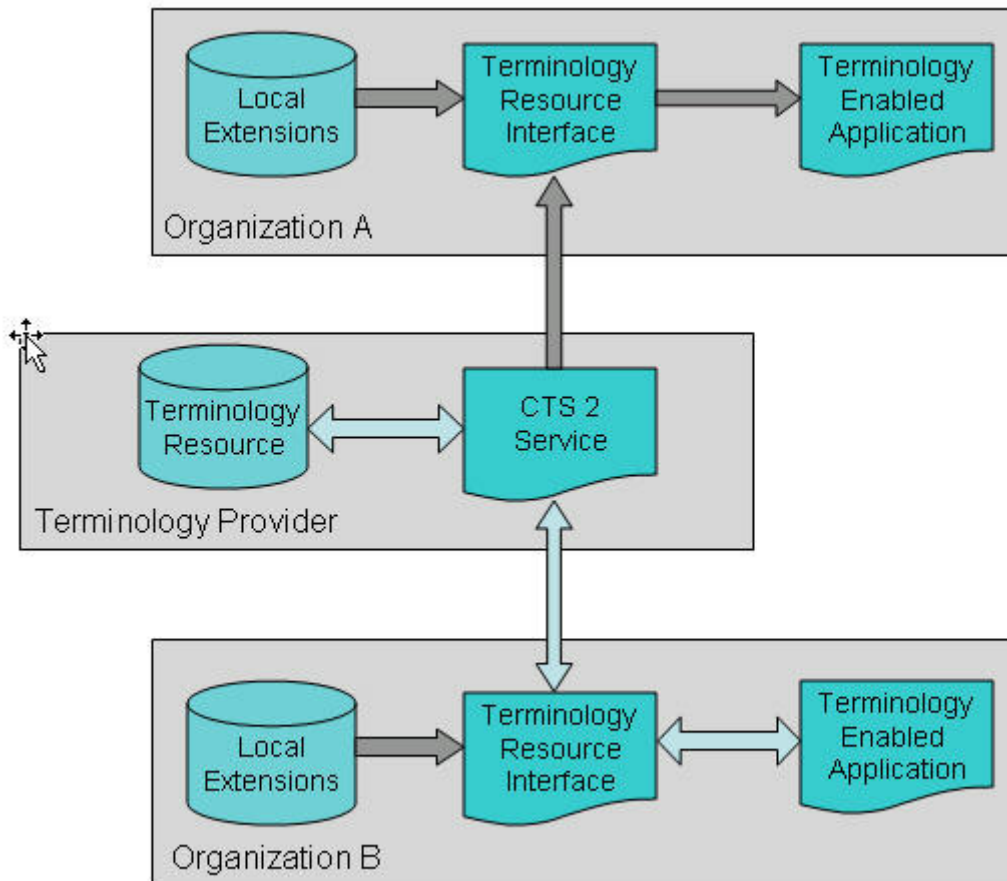
307  
308 Since terminology content is not static, CTS 2 will also provide functionality to maintain and  
309 update terminology content. Updates and update requests to terminology sources need to be  
310 reviewable and traceable over time. Often, a terminology source provider will want to maintain  
311 the “gold standard” or master release of a code system, as to maintain a consistent standard  
312 terminology that can be used across multiple organizations and realms. Notwithstanding, users of  
313 any given source terminology may wish to extend that terminology for their own use, and may  
314 even wish to recommend the addition of those “local” extensions to the terminology provider to  
315 be included as part of the release.

316 CTS 2 will provide a mechanism to allow for terminology users to extend a given terminology,  
317 share those extensions with others, or feed those extensions back to the source provider in a



318 structured format to be reviewed, modified as necessary, and fed into a CTS 2 server as input to  
319 update the source terminology with the content contained in the change request. As depicted in  
320 Figure 2.4-3, Organization A is applying its own local extensions to a terminology resource  
321 being served by a CTS 2 service. In addition to applying its own local extensions, Organization  
322 B is feeding some of those local extensions back to the terminology provider as suggestions to be  
323 included in the next release of the code system.

324



325

326 **Figure 2.4-3 Multi Organization Access with Write Permissions by One Organization**

### 327 **Terminology Structure Considerations**

328 Terminologies are created for many purposes, and as such are often structured very differently,  
329 from a flat list of concepts, to complex poly-hierarchies. The attributes of the entities of code  
330 systems vary as well. Even the formats of the identifiers are different, with some concept  
331 identifiers being meaningless identifiers, to others which have explicit or implied meaning.

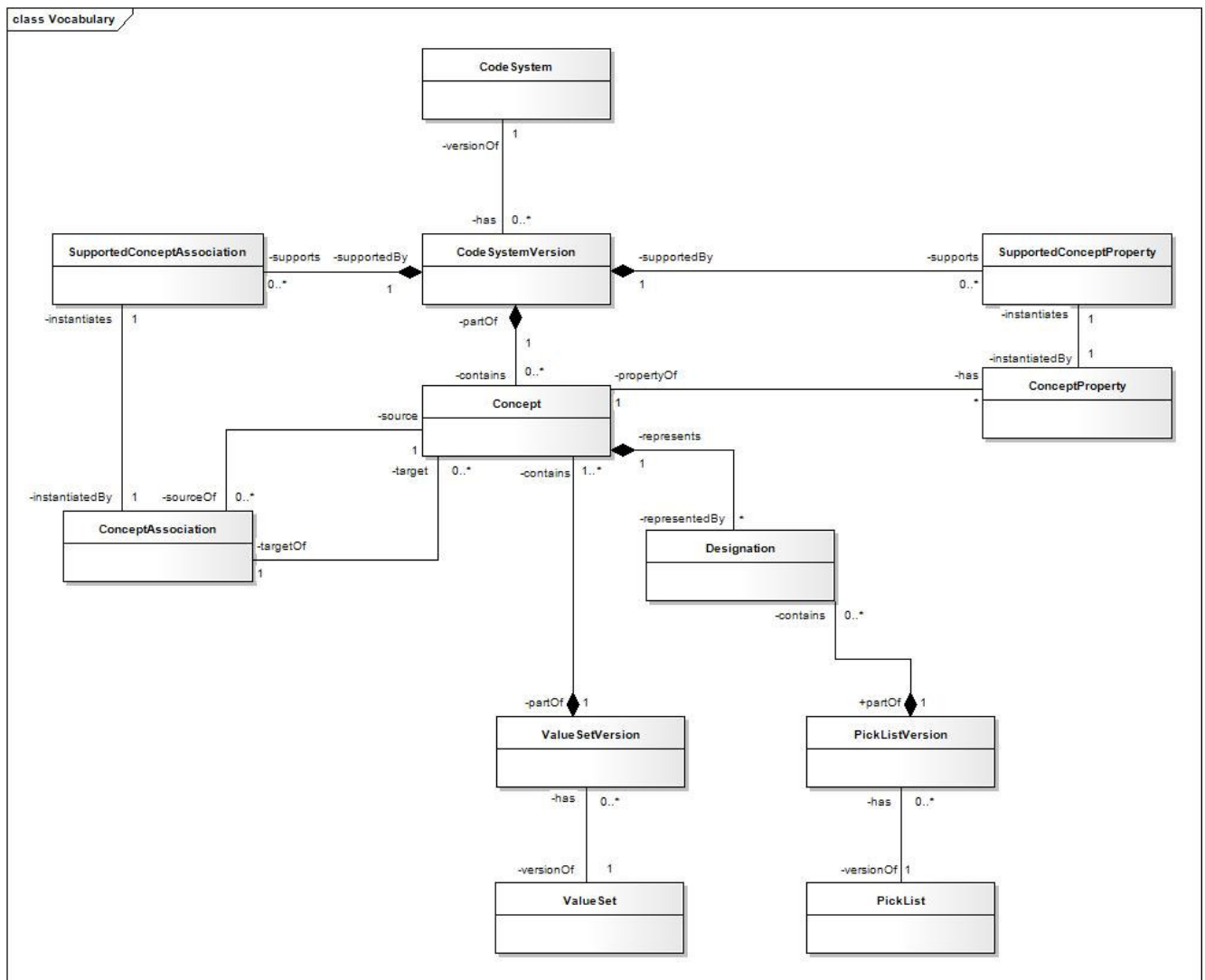
332

333 The functional components of CTS 2 must be able to operate on this broad spectrum of  
334 terminology sources. At a minimum, CTS 2 must specify a concept based terminology model

335 that is capable of representing most varieties of structured terminologies. The basic structure of  
 336 the code system is illustrated in the **CTS 2 Upper Level Class Model** below. This model  
 337 outlines the various components and the cardinality between them but does not dictate particular  
 338 levels of data normalization or other technical details of implementation.

339 **This model is intended to assist with outlining the minimal functional behaviors of a**  
 340 **terminology that is served by a CTS 2 terminology service, and is informative. An**  
 341 **implementation model will be necessary as a response to the RFP for CTS 2.**

342

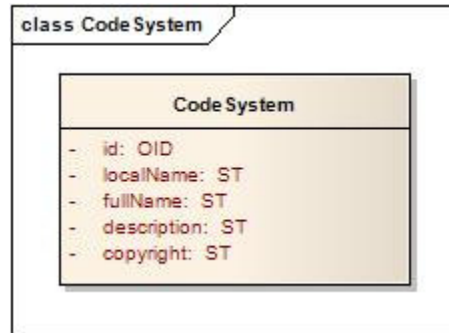


343  
344

**Figure 1 CTS 2 Upper Level Class Model**

346

347 **Code System**



348

349 A code system is defined as a collection of uniquely identifiable concepts with associated  
350 designations, associations and meanings. Examples of code systems include ICD-9 CM,  
351 SNOMED CT, LOINC, and CPT. To meet the requirements of a code system as defined by HL7,  
352 a given code must resolve to one and only one meaning within the code system.

353 In the terminology model, a code system is represented by the **CodeSystem** class. Code systems  
354 themselves can be concepts and each can be represented by a **Concept** class with a unique  
355 identifier.

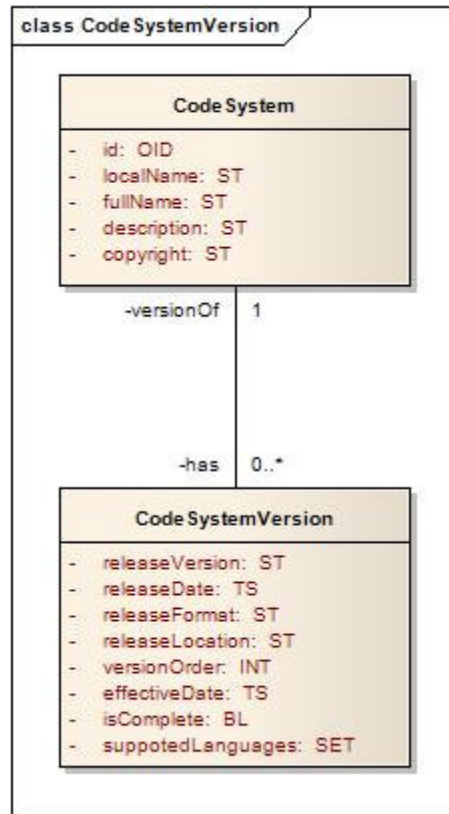
356 At a minimum, Code Systems have the following attributes:

- 357 • An identifier (**id**) that uniquely identified the Code System
- 358 • A name (**localName**) that the Code system is normally referred to
- 359 • A name (**fullName**) that is the official name of the code system as assigned by the  
360 terminology provider.
- 361 • A description (**description**) that describes the Code System. This may include the code  
362 system uses and intent.
- 363 • Copyright information (**copyright**) pertaining to the Code System

364

365

366 **Code System Version**



367

368

369 Code systems are not generally static entities and change over time. A **CodeSystemVersion** is a  
370 static snapshot of a **CodeSystem** at a given point of time, and specifies the version of the code  
371 system in which any given concept can be found.

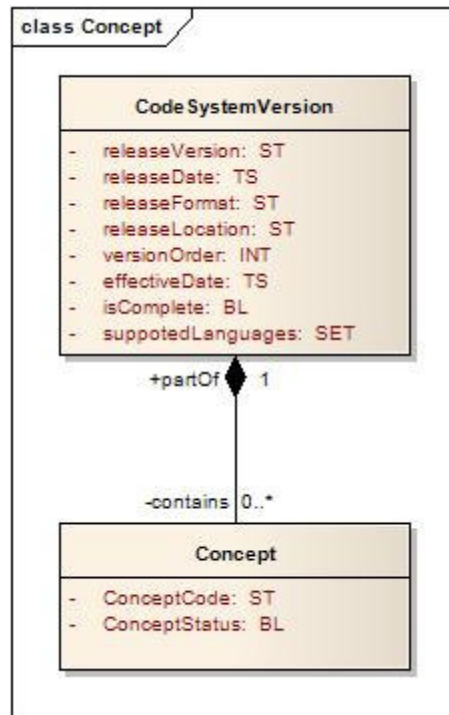
372 A CodeSystemVersion are represented by attributes including:

- 373 • A version identifier (**releaseVersion**) that uniquely identifies each version of a Code
- 374 System
- 375 • A date (**releaseDate**) that represents the date when the version of the Code System
- 376 became available
- 377 • The format (**releaseFormat**) that indicates the format(s) that the version of the Code
- 378 System is available in.
- 379 • The official location (**releaseLocation**) where the version of the code system is available
- 380 from
- 381 • An optional ordering parameter (**versionOrder**) that identifies the order which the
- 382 version should be applied (used for version deltas).
- 383 • The start date (**effectiveDate**) when the version is deemed to be valid for use.
- 384 • An flag (**isComplete**) indicating that the version in question is complete (i.e. standalone)
- 385 or requires other previous or later versions to be complete.
- 386 • The different languages (**supportedLanguages**) supported by the Code System

387

388

389 **Concept**



390

391

392 A concept defines a unitary mental representation of a real or abstract thing; an atomic unit of  
393 thought. Concepts should be unique within a given code system, but may have synonyms in  
394 terms of representation. Concept may be primitive or compositional in nature. For example, the  
395 concept of “hypertension” evokes the same meaning to all clinicians even though it may be  
396 expressed as “high blood pressure,” “hypertensive disorder,” or “HTN.”

397

398 Each **CodeSystem** entity will have a set of **Concepts** associated with it. Each **Concept** is  
associated with a **CodeSystem** through a specific **CodeSystemVersion** in a one-to-one manner.

399

Concepts are represented by attributes including:

400

- A unique concept identifier (conceptCode)

401

- A concept status flag (conceptStatus)

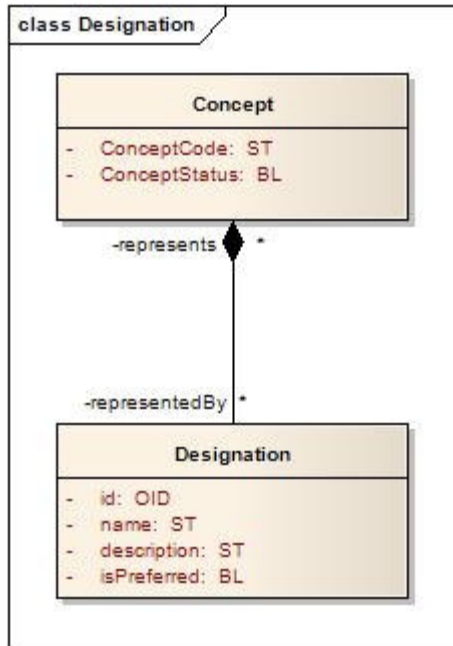
402

Terminology best practices dictate that concepts are not deleted from code systems, but are  
403 instead deprecated or retired from use.

404

405

406 **Designation**



407

408 Concept designations are representations of concepts. The designation identifier must uniquely  
409 map to a given text string, bitmap, etc. within the context of the containing concept.

410 In some terminologies, every unique text string will have exactly one presentation identifier,  
411 which means that the same presentation identifier may occur under more than one concept.

412 In other terminologies, there may be more than one identifier for a given text string, meaning that  
413 the presentation identifier uniquely determines the concept. Service software must not assume  
414 either model.

415 For example, in SNOMED CT, the concept of “fever” has the fully specified name of “fever  
416 (finding),” a preferred name of “fever,” and synonyms of “febrile” and “pyrexia.” These are all  
417 designations for the concept of “fever.”

418 In the terminology model, designations are represented by the **Designation** class. Each  
419 **Designation** is a representation of the **Concept** and is assigned a unique designation identifier.  
420 In most instances, concept designations are human readable forms, but machine readable forms  
421 may also be present.

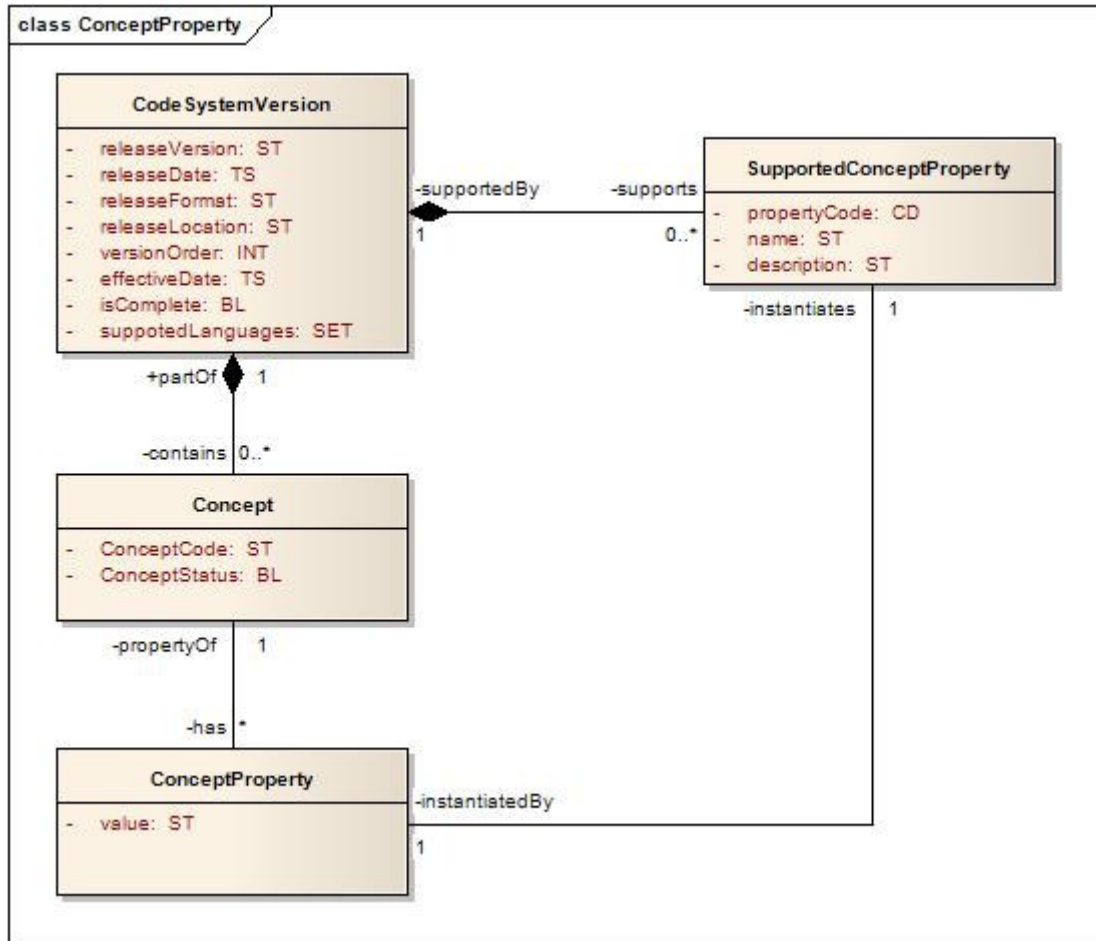
422 The **Designation** class is minimally defined by the following attributes:

- 423 • A unique identifier (**id**) for the designation
- 424 • A name (**name**) for the designation
- 425 • A description (**description**) for the designation
- 426 • A format (**format**) for the designation
- 427 • A flag (**isPreferred**) indicating if the designation is preferred for the concept

428

429

430 **Concept Property**



431

432

433 A concept property is a named characteristic of a concept that can be assigned a value. In the  
 434 terminology model a concept property is represented by the *ConceptProperty*' class. The  
 435 allowable or supported concept properties for any given **CodeSystem** are specified by the  
 436 **SupportedConceptProperty** class, and specific to **CodeSystemVersion**'.

437

438 For example, the result of Hematocrit with a LOINC code of 11271-4 has a specimen property  
 439 with the value of “blood” and a method property with the value of “automated count.”

440

441 In the above example, the specimen and method properties are part of how LOINC assigns the  
 442 code and, when these properties change, a different LOINC code will be assigned. This implies  
 443 that concept properties do not change over time. However, exceptions may be possible when the

444 addition of a new property does not change the concept and code. For example, if LOINC  
 445 decides to add “analyte chemical structure” as a new property, there may not be a need to change  
 446 the existing LOINC codes since the new information can apply to all of the LOINC concepts.

447  
 448 Each *Concept* may have zero to many concept properties, and each **CodeSystem** may have its  
 449 own unique set of concept properties associated with its **Concepts** for any specific  
 450 **CodeSystemVersion**.

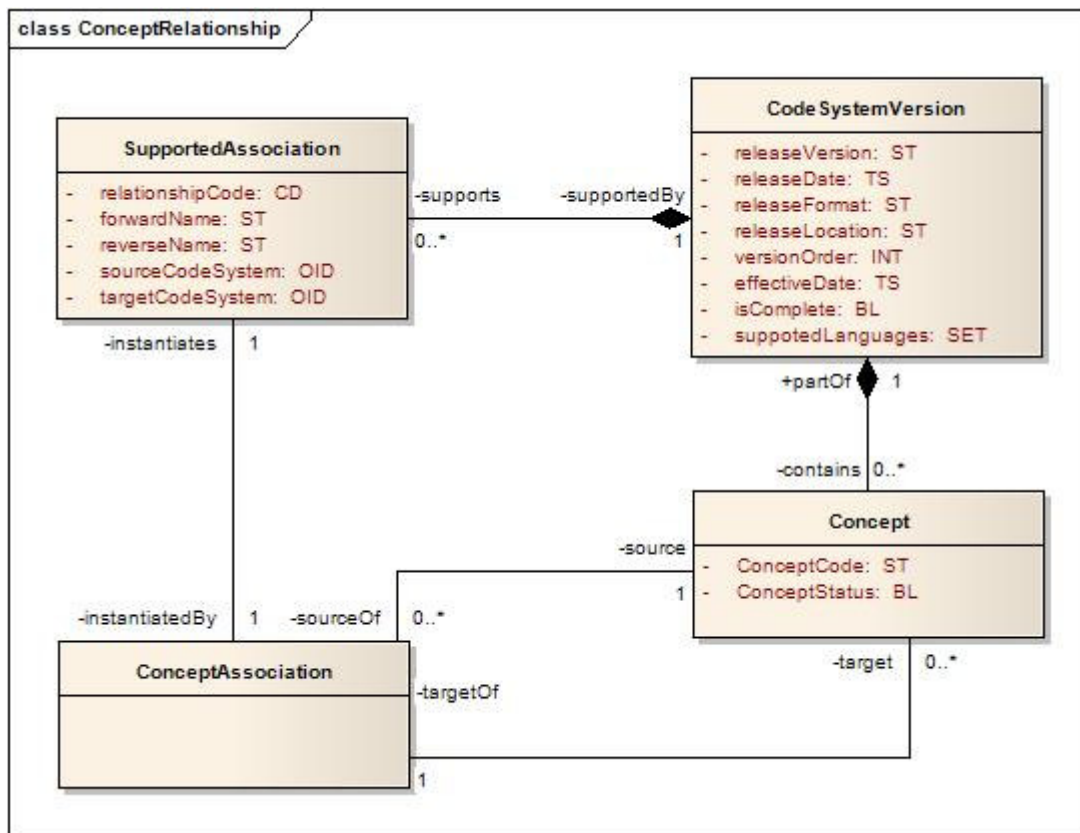
451  
 452 Concept properties are represented with attributes including:

- 453 • A code (**propertyCode**) that uniquely identifies the property
- 454 • A name (**name**) for the property
- 455 • A description (**description**) of the property
- 456 • A value (**value**) of the property.

457

458

459 **Concept Association**



460



461 Associations define the relationships or linkages between concepts. For example, in SNOMED  
462 CT, the concept of “pneumonia” has an “is-a” relationship to the concept of “lung  
463 consolidation,” and “lung consolidation” has an “is-a” relationship to the concept of “disorder of  
464 lung.” This represents the logical conclusion that “pneumonia” is a “disorder of lung.”

465 In the terminology model, relationships are represented by the **ConceptAssociation** entity and  
466 are defined as a directed semantic relationship triples between two concepts. The allowable or  
467 supported concept associations for any given **CodeSystem** are specified by the  
468 **SupportedConceptAssociation** class, and specific to **CodeSystemVersion**.

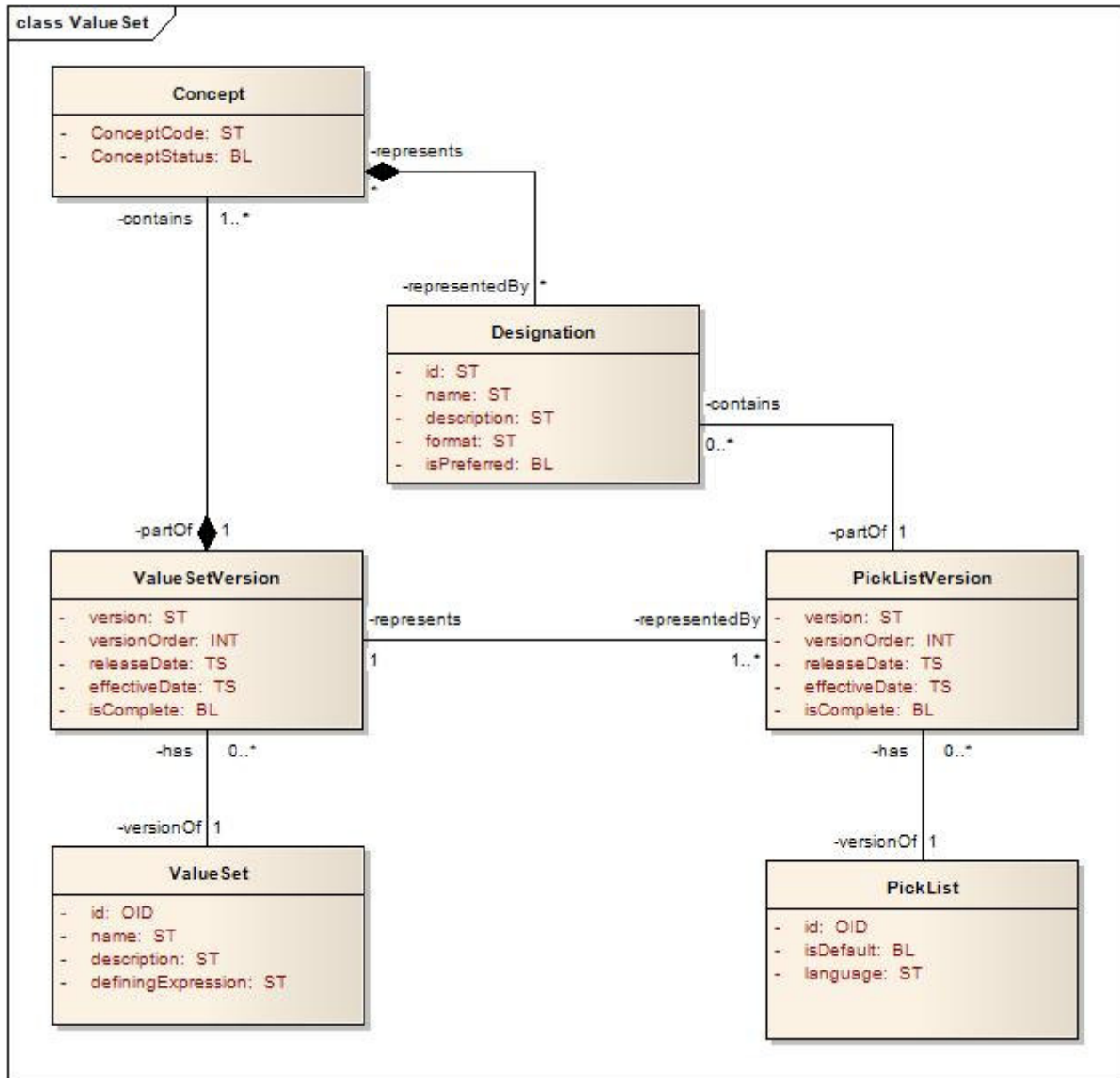
469 It is not necessary for concepts to have associations to other concepts. However, when  
470 associations exist, the cardinality and the explicit declaration of source and target would indicate  
471 the directionality that restricts the designation of the association. For example, from the concept  
472 relationship (an association between concepts within a single code system) in the above example,  
473 we can infer that “pneumonia” is a “disorder of lung,” but the inverse concept relationship of  
474 “disorder of lung” is-a “pneumonia” cannot be inferred. If we want the inverse concept  
475 relationship, it must be explicitly stated, that is, there has to be a specific relation of “disorder of  
476 lung” “is-a” “pneumonia. In the case of Concept Maps (where the source and target concepts are  
477 from different code systems) the direction and designation of the relationship have similar  
478 restrictions, except in the case where the Concept Map indicates semantic equivalence. The equal  
479 association in this case obviates the requirement for interpreting the association direction.

480 A **ConceptAssociation** links a source **Concept** to a target **Concept**. The supported  
481 **ConceptAssociations** define the relationships that can be instantiated between any two concepts  
482 for a given **CodeSystemVersion**.

483 Concept associations are minimally defined by attributes including:

- 484 • A code (**associationCode**) that uniquely identifies either the association instance (for  
485 concept maps) or type (for concept relationships).
- 486 • A name (**forwardName**) that represents how the association should be represented when  
487 reading from source concept to target concept.
- 488 • A name (**reverseName**) that represents how the association should be represented when  
489 reading from target concept to source concept.
- 490 • The unique identifier of the code system (**sourceCodeSystem**) where the source concept  
491 originated.
- 492 • The unique identifier of the code system (**targetCodeSystem**) where the target concept  
493 originated.

494 **Value Set**



495  
496

497

498 A value set represents a uniquely identifiable set of valid concept representations (codes), where  
499 any concept representation can be tested to determine whether or not it is a member of the value  
500 set.

501

502 Value set complexity may range from a simple flat list of concept codes drawn from a single  
503 code system, to an unbounded hierarchical set of possibly post-coordinated expressions drawn  
from multiple code systems.

504

505 In the terminology model, a value set is represented by the **ValueSet** class. Value sets have

identifiers and is a collection of codes for a given **Concept**.

506 A **ValueSet** is represented by a given **ValueSetVersion**. The **ValueSetVersion** concepts that are  
507 available in the value set for any specific version of the value set. As discussed above, **Concepts**  
508 are represented by **Designations**. Designations for the concepts in a value set are housed in a  
509 **PickList**. The designations available for a given pick list is controlled by the **PickListVersion**  
510 class. The **PickListVersion** represents the **Concepts** available to a **ValueSet** for a given  
511 **ValueSetVersion**.

512 Value sets are represented by attributes including:

- 513 • An identifier (**id**) that uniquely identifies the value set.
- 514 • A name (**name**) for the value Set
- 515 • A description (**description**) for the value set.
- 516 • An optional expression (**definingExpression**) that defines the value set
- 517 • A version (**ValueSetVersion.version**) for the value set
- 518 • An optional order (**ValueSetVersion.versionOrder**) that identifies the order of which
- 519 the version should be applied
- 520 • A date (**ValueSetVersion.releaseDate**) when the version of the value set was released
- 521 • An effective date (**ValueSetVersion.effectiveDate**) that identifies when the value set
- 522 version became effective
- 523 • A flag (**ValueSetVersion.isComplete**) that indicates whether the version of the value set
- 524 is complete or not.

525

526 Pick lists are represented by attributes including:

- 527 • An identifier (**id**) that uniquely identifies the pick list.
- 528 • A flag (**isDefault**) that identifies whether the pick list in question is the default pick list
- 529 for the value set.
- 530 • The human language (**language**) that the pick list is using.
- 531 • A version (**PickListVersion.version**) for the pick list
- 532 • An optional order (**PickListVersion.versionOrder**) that identifies the order of which the
- 533 version should be applied
- 534 • A date (**PickList.releaseDate**) when the version of the pick list was released
- 535 • An effective date (**PickListVersion.effectiveDate**) that identifies when the pick list
- 536 version became effective
- 537 • A flag (**PickListVersion.isComplete**) that indicates whether the version of the pick list
- 538 is complete or not.

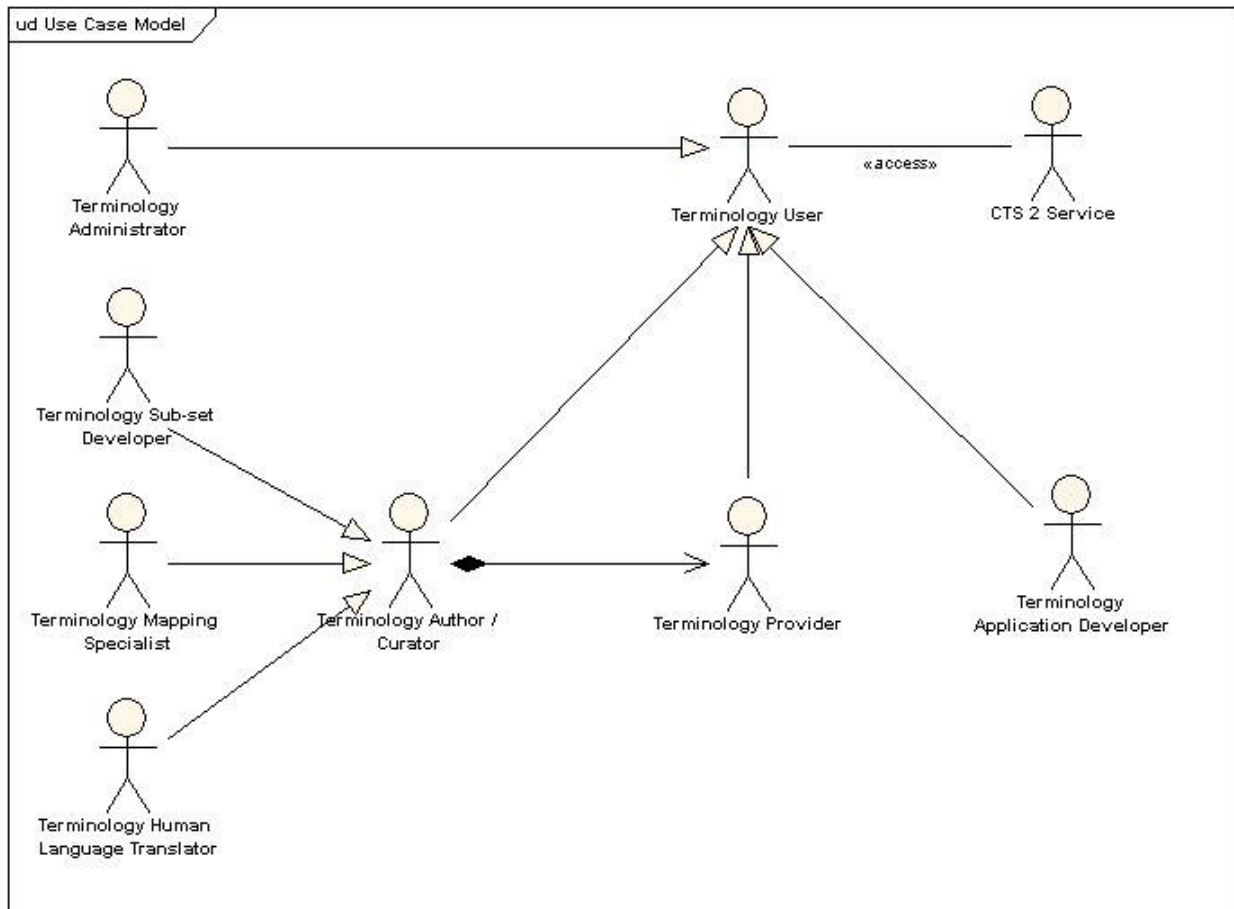
## 539 **Business Scenarios**

### 540 **Scenario Actors**

541 Actors will use the CTS 2 service for different purposes. These different actors can be  
542 generalized into a basic *Terminology User* actor that is simply an individual, organization, or

543 application that requires access to terminology content for some purpose. Specializations of the  
544 *Terminology User* actor participate in additional operational specific scenarios that are defined  
545 by this Service Functional Model to address the Scope that is outlined in section 2.1.2. Actors  
546 described in this section are not necessarily human actors, but also include organizations and  
547 systems Figure3.1-1 outlines the specializations and composition of the different actors used in  
548 this specification. These actors are described below.

549



550

551

**Figure 3.1-2**

552

553 The following actors take a role in the CTS 2 scenarios.

554

- 555 • **CTS 2 Service**

556 The *CTS 2 Service* is a specific implementation of the CTS 2 Terminology Server.

557

558       • **Terminology User**

559       A *Terminology User* is an actor such as a subject matter expert, terminologist or terminology  
560       enabled application. *Terminology User* activities include, but are not limited to, querying for  
561       specific concept codes and browsing or comparing value sets. Specializations of the *Terminology*  
562       *User* actor follow below.

563

564       • **Terminology Administrator**

565       The *Terminology Administrator* is an actor responsible for ensuring the availability and overall  
566       maintenance of the terminology server. This includes, but is not limited to loading content into  
567       the terminology server, and making available the required functionality to address the specific  
568       conformance profiles implemented by the Terminology Server instance.

569

570       • **Terminology Enabled Application Developer**

571       A *Terminology Enabled Application Developer* is an actor who is responsible for the  
572       development of software applications that make explicit use of controlled terminologies.

573

574       • **Terminology Author / Curator**

575       A *Terminology Author / Curator* is an actor who is responsible maintaining terminology content,  
576       including but not limited to, the development of new concepts that may be submitted to the  
577       *Terminology Provider* or the extension of an existing terminology with local concepts. This may  
578       also who can validation and quality control of terminology content. Terminology Authors /  
579       Curators may not necessarily belong to the *Terminology Provider's* organization.

580

581       • **Terminology Human Language Translator**

582       A *Terminology Human Language Translator* is an actor with domain knowledge who is also  
583       familiar with the languages and dialects which they are responsible for translating.

584

585       • **Terminology Mapper**

586 A *Terminology Mapper* is an actor (human or system) that is responsible for creating or  
587 maintaining specialized associations, or "mappings" between concepts from different code  
588 systems.

589

590 • **Terminology Provider**

591 The *Terminology Provider* is the actor the individuals or organization that is responsible for the  
592 development of Terminology Content.

593

594 • **Terminology Value Set Developer**

595 A *Terminology Value Set Developer* is an actor with specific domain knowledge, as well as  
596 expertise in controlled terminologies who develops and maintains domain-or application-specific  
597 terminology value sets.

598 **Primary Scenarios**

599 Primary scenarios are tied to one or more conformance profiles. Note, that as an aid to reading  
600 this specification, Actors that are identified in the text are italicized. In addition, when a scenario  
601 references another scenario, that referenced scenario is in bolded italics.

602

603 **Administrative Scenarios**

604 The administration scenarios are intended to provide the functional operations necessary for  
605 terminology administrators to be able to access and make available terminology content obtained  
606 from a Terminology Provider. Terminology Administrators are required to interface with  
607 Terminology Provider systems in order to obtain the terminology content, then load that  
608 terminology content on local Terminology Servers.

609 **Import Content**

610 A *Terminology Administrator* is required to make available new terminology content from a  
611 *Terminology Provider* available to *Terminology Users* through a *Terminology Server*. This may  
612 or may not include the removal of previously loaded terminology content from the terminology  
613 server. To accomplish this, the *Terminology Administrator* may be required to convert the  
614 content from the format provided by the *Terminology Provider* to a format that the *Terminology*  
615 *Server* is capable of importing. Example of terminology content that may be available for  
616 loading into the *Terminology Server* include but is not limited to:

- 617 • Source terminologies (complete sources and deltas)

- 618 • Value sets

619  
620 These content sources may either be new sources, or updated versions of a previously existing  
621 content sources.

622 Associated Functional Models: [Import Terminology](#), [Import Terminology Revision](#), [Convert](#)  
623 [Terminology Format](#)

## 624 **Export Content**

625 A *Terminology Administrator* wants to be required to export a terminology or terminology subset  
626 from the *Terminology Server*. This may require filtering of the content and converting the format  
627 of the export.

628 Associated Functional Models: [Export Terminology](#)

## 629 **Remove Content**

630 A *Terminology Administrator* is required to remove a terminology or terminology version from  
631 the terminology service, rendering it unavailable for subsequent access by other service  
632 functions.

633 Associated Functional Models: [Remove Terminology / Terminology Version](#)

## 634 **Change Content Status**

635 A *Terminology Administrator* is required to activate or inactivate a given terminology, thus  
636 changing its availability for access by other terminology service functions.

637 Associated Functional Models: [Change Terminology Status](#)

## 638 **Update Notification**

639 A *Terminology User* has a dependency on a specific terminology element that is available to a  
640 Terminology Server. The *Terminology User* is interested in knowing when this terminology  
641 element is modified in any way, and would like to receive an electronic notification in the event  
642 of that change to that terminology element Associated Functional Models: [Register for Update](#)  
643 [Notification](#)

## 644 **Update Notification Management**

645 A *Terminology User* is required to update the notification information pertinent to their  
646 notification account.

647 Associated Functional Models: [Revise or Remove Update Notification](#)

648 **Content Dependency Notification**

649 A *Terminology Administrator* wants is required to run a dependency check to compare updated  
650 content for a given code system, against the version of that code system currently used by the  
651 *Terminology Administrator's* organization. For example, to provide a list of all terminology  
652 elements which are somehow affected by upgrading to a newer version of a terminology.

653 Associated Functional Models: [Register for Concept Dependency Notification](#), [Revise or](#)  
654 [Remove Concept Dependency Notification](#)

655 **Search / Query Scenarios**

656 The scenarios in this section describe the ability to query code system and value set. These  
657 scenarios attempt to outline the information requirements for querying. The detailed function  
658 models in section 5.2 call out the distinct functional requirements.

659 A given CTS 2 implementation will be required to advertise the specific search algorithms that is  
660 supports.

661 In each scenario below, the *Terminology User* may need to specify additional information  
662 pertaining to the query. This information may include:

- 663 • The ability to determine the status of metadata or contents of a code system, value set as  
664 it existed in a specified *version*, where *version* represents a meta-data component used to  
665 filter the result set of the query.

666 **NOTE:** Details of the available meta-data requirements will be identified as part of the Binding  
667 Document and Model harmonization activity.

668

669 **Code System Search / Query**

670 This section outlines Search / Query operations pertaining to Code Systems.

671 **Resolve Available Code Systems**

672 A *Terminology User* wants to determine what code systems are available through a specific  
673 instance of a Terminology Service. The *Terminology User* is interested in seeing a listing of the  
674 available code systems, as well as the details pertaining to each code systems available through a  
675 specific Terminology Service instance.

676 Associated Functional Models: [Resolve Available Code Systems](#), [Resolve Code System](#)  
677 [Metadata](#)

678 **Retrieve Coded Concepts from Code System**



679 A *Terminology User* wants to browse or query the content of a specific code system. The  
680 *Terminology User* is interested in seeing a listing of specific coded concepts, associated  
681 attributes, as well as the metadata pertaining to each coded concept that meets some search  
682 criteria. For example, after a retrieval of concepts has been performed, the result set could be fed  
683 to a terminology browsing GUI

684 Associated Functional Models: [Resolve Code System Concepts](#), [Resolve Coded Concept from](#)  
685 [Code System](#), [Resolve Concept Details](#)

#### 686 **Validate Concept in Code System**

687 A *Terminology User* wants to validate that a given concept exists in a given code system.

688 Associated Functional Models: [Resolve Coded Concept from Code System](#)

689

#### 690 **Identify Concept Language Translations**

691 A *Terminology User* wants to determine what (if any) alternate language representations exist for  
692 a given Concept.

693 Associated Functional Models: [Resolve Concept Details](#)

#### 694 **Resolve Concept Representations**

695 A *Terminology User* wants to determine what (if any) alternate representations exist for a given  
696 Coded Concept. Examples of alternate representations for a concept may include abbreviations,  
697 or synonyms.

698 Associated Functional Models: [Resolve Concept Details](#)

#### 699 **Compare Code System Versions**

700 A *Terminology User* wants to determine what differences exist between different versions or  
701 instances of a code system.

702 Associated Functional Models: [Compare Code Systems](#), [Compare Code System Contents](#)

#### 703 **Value Set Search / Query**

704 This section outlines Search / Query operations pertaining to Value Sets.

705

#### 706 **Resolve Available Value Sets**

707 A *Terminology User* wants to determine what value sets are available through a specific instance  
708 of a Terminology Service. The *Terminology User* is interested in seeing a listing of the available  
709 value sets that match some search criteria, as well as the details pertaining to each value set  
710 available through a specific *CTS 2 Service* instance.

711 Associated Functional Models: [Resolve Available Value Sets](#), [Resolve Value Set Metadata](#)

#### 712 **Retrieve Coded Concepts from Value Set**

713 A *Terminology User* wants to browse or query the content of one or more value sets. The  
714 *Terminology User* is interested in seeing a listing of specific coded concepts, as well as the  
715 details pertaining to each coded concept in any of the given value sets. For example, the  
716 *Terminology User* may want to search for some criteria over a set of value sets.

717 Associated Functional Models: [Resolve Value Set Entries](#), [Resolve Concept Details](#)

#### 718 **Validate Coded Concept in Value Set**

719 A *Terminology User* wants to validate that a given concept exists in a given value set.

720 Associated Functional Models: [Resolve Value Set Entry](#)

#### 721 **Compare Value Set Versions**

722 A *Terminology User* wants to determine what differences exist between different versions of a  
723 value set.

724 Value Sets can be defined as either enumerations of concepts (Enumerated Value Set), or by  
725 expression syntax that defines the content of the Value Set.

726 In the case of an Enumerated Value Set, the specific Value Set version identifier can be used as a  
727 compare point for the two value sets.

728 For Intensionally defined Value Sets, the compare point is either the Code System version when  
729 the Value Set definition is bound to a specific Code System version, or the date when the Value  
730 Set definition is bound to a code system with no specific version specified.

731 Associated Functional Models: [Compare Value Sets](#), [Compare Value Set Contents](#)

#### 732 **Resolve Concept Representations**

733 A *Terminology User* wants to determine what (if any) alternate representations exist for a given  
734 Coded Concept in a value set. Examples of alternate representations for a concept may include  
735 abbreviations, or synonyms.

736 Associated Functional Models: [Resolve Concept Details](#)

## 737 **Authoring / Curation Scenarios**

738 This section outlines the requirements of terminology systems that provide the capability of  
739 making changes to terminology elements such as code system or value sets. This includes both  
740 the direct modification of terminology content for use by individuals responsible for terminology  
741 authoring and curation. Such functionality includes:

- 742 • adding new concepts into a code system
- 743 • adding new relationships into a code system
- 744 • extending a code system with local terms
- 745 • creating or modifying value sets
- 746 • modifying other code system content and attributes

747 In addition to direct modification of terminology content, this section also specifies functionality  
748 with the capability of creating structured change requests for consideration by terminology  
749 maintainers. This functionality is key in allowing Terminology Providers to solicit feedback  
750 pertaining to terminology structure and content from Terminology Users in a controlled and  
751 structured manner.

## 752 **Code System Authoring / Curation**

753 This section outlines the business scenarios specific to terminology systems that provide the  
754 capability of making changes to code system components which include coded concepts,  
755 representations (textual), Associations or Relationships, and value sets.

### 756 **Create Code System**

757 A *Terminology Author* is required to create a new Code System to contain a set of new coded  
758 concepts. The Code System is created by defining the set of meta-data properties that describe it.

759 Associated Functional Models: [Create Code System](#)

### 760 **Maintain Code System**

761 As part of ongoing terminology maintenance, a *Terminology Author* is required to perform  
762 maintenance to the defining characteristics of an existing code system.

763 Associated Functional Models: [Maintain Code System](#)

### 765 **Create Concept**

766 A *Terminology Author* is required to create concept to be included in a Code System.

767 For example, as part of providing *Terminology Service* infrastructure to another department, a  
768 *Terminology Author* is required to add additional concept codes to a code system to represent the  
769 domain concepts that are important to the new department.

770 The new concept is defined by the set of meta-data properties that describe it, which may include  
771 its proper placement via association binding within the hierarchy of the Code System.

772  
773 Associated Functional Models: [Create Concept](#)

#### 774 **Maintain Concept**

775 A *Terminology Author* is required to maintain a concept. This includes but is not limited to  
776 functionality such as:

- 777 • making updates to the associated concept attributes,
- 778 • changing the presentation,
- 779 • changing preferred name,
- 780 • changing synonymy,
- 781 • technical corrections to the concept
- 782 • modifying the associations bound to concepts

783 These types of changes result in a new version of the of the code system being modified.

784  
785 Associated Functional Models: [Maintain Concept](#)

#### 786 **Deprecate Concept**

787 A *Terminology Author* is required to deprecate a coded concept. Concepts may be required to be  
788 deprecated if they become obsolete or are ambiguous. In many cases, the deprecated concept is  
789 replaced with other new concepts.

790 NOTE: In keeping with good vocabulary practice, codes or identifiers for concepts cannot be  
791 reused. Additionally, in hierarchical Code Systems, it may be necessary to re-associate any  
792 concepts related to the concept being deprecated to prevent a part of the code system hierarchy  
793 from being orphaned.

794  
795 Associated Functional Models: [Deprecate Concept](#)

#### 796 **Value Set Authoring / Curation**

797 This section outlines the business scenarios specific to terminology systems that provide the  
798 capability of creating and maintaining sub-sets of a code system, otherwise known as value sets.

##### 799 **Create Value Set by Intension**

800 A *Terminology User* is required to create a dynamic value set that is defined by a computable  
801 expression that can be resolved to an exact list of coded concepts at any given point in time.

802 For example, an intensional value set might be expressed as, “SNOMED CT concepts that are  
803 children of the SNOMED CT concept “Diabetes Mellitus.”

804 Note: When creating an intensionally defined value set, the *Terminology User* may or may not  
805 bind the value set definition to a specific version of the Code System(s) from which the concepts  
806 are being drawn.

807 If the value set expression is bound to a specific version of the Code System(s), the value set will  
808 always resolve the same set of concept codes for any given version of the value set.

809 If the value set expression is **not** bound to a specific version of the Code System(s), the value set  
810 will resolve a different set of concept codes as the version of the Code System changes.

811 Associated Functional Models: [Create Value Set by Intension](#)

#### 812 **Create Value Set by Extension**

813 A *Terminology User* is required to create an enumerated (static) value set that is comprised of an  
814 explicitly enumerated set of codes.

815 For example, A *Terminology Author* is interested in creating a value set based on the SNOMED-  
816 CT code system. The *Terminology Author* builds the value set by selecting the individual  
817 concepts that best represent the concepts that are required for the value set.

818  
819 Associated Functional Models: [Create Value Set by Extension](#)

#### 820 **Maintain Value Set (Definition)**

821 A *Terminology User* is required to maintain (i.e. remove or update) a value set (by definition).

822 For example, a *Terminology User* identifies an error in how a value set (by definition) is defined.  
823 A *Terminology Author* re-defines the value set to be accurate to the understanding of the  
824 *Terminology User*.

825 Associated Functional Models: [Maintain Value Set \(Intension\)](#)

#### 826 **Maintain Value Set (Enumeration)**

827 A *Terminology User* is required to maintain (i.e. add, remove or update) an enumerated coded  
828 concept in value set.

829 For example, a *Terminology User* identifies a concept code that is not included in an enumerated  
830 value set. The *Terminology Author* browses the code system to select the concept code identified  
831 by the *Terminology User*. The *Terminology Author* selects the concept code to be included in the  
832 existing value set.

833 **Note:** In order to create a coded concept in a value set it must first exist in the code system  
834 (include concept section).

835  
836 Associated Functional Models: [Maintain Value Set \(Extension\)](#)

### 837 **Change Request Processing**

838 This section outlines the business scenarios specific to terminology systems that provide the  
839 capability of creating and processing change requests used to author / curate a terminology  
840 service.

#### 841 **Create Change Request**

842 A terminology user identifies a potential improvement in the vocabulary, and would like to  
843 create a change request that can be reviewed by other terminology users and ultimately submitted  
844 to the Terminology Provider for consideration as a change to the terminology.

845 Associated Functional Models: [Create Change Request](#)

#### 846 **Edit Change Request**

847 A Terminology User is reviewing an existing change request that outlines a potential change to  
848 the terminology content from a Terminology Provider. The Terminology User would like to edit  
849 the content of the change request prior to it being submitted to the Terminology Provider.

850 Associated Functional Models: [Edit Change Request](#)

#### 851 **Submit Change Request**

852 A Terminology User can submit a change request or a package of several change requests to the  
853 Terminology Provider for review. This action has the effect of changing the status of the affected  
854 proposal(s) so that they can no longer be modified by other Terminology Users.

855 Associated Functional Models: [Submit Change Request](#)

#### 856 **Package Change Request**

857 A Terminology User identifies a set of related change requests. The Package Change Request  
858 operation will group a set of change requests together to be submitted to the Terminology  
859 Provider to be considered as a set of changes to the terminology. Any individual change request  
860 can only be a part of one package.

861 Associated Functional Models: [Package Change Request](#)

### 862 **Association Scenarios**

863 The scenarios in this section describe the ability to create, query and maintain associations  
864 between coded concepts. These coded concepts may or may not come from the same Code  
865 System. As such, these scenarios can describe intra-code system associations, or concept  
866 relationships, as well as inter-code system associations across different systems, or concept  
867 maps. The premise for this is that information requirements and functions for concept  
868 relationships and concept mapping are the similar, although the context of use and elements for  
869 each are different. These scenarios attempt to outline the information requirements for  
870 associations. The detailed functional models in section six call out the distinct functional  
871 requirements that specifically differentiate internal code system concept relationships from  
872 concept maps. In each scenario below, the Terminology Mapper may need to specify additional  
873 information pertaining to the source / target association of interest. This information may  
874 include:

- 875 • The version of the source and target Code Systems being used to create the association,  
876 or,
- 877 • The code system of interest, whether that pertains to a single code system or more than  
878 one code system
- 879 • The version of the source and target code systems being used to create the association, or,
- 880 • The cardinality of the association, i.e.: if the concept association is one-to-one, one-to-  
881 many, many-to-one, or many-to-many.

882 Additionally, the type of associations may include, but are not limited to:

- 883 • if the source concept is an *exact match* to the target concept,
- 884 • if the source concept is *equivalent* to the target concept,
- 885 • if the source concept is *broader than* the target concept,
- 886 • if the source concept is *narrower than* the target concept
- 887 • Other examples are *generic-to-brand name*, *ingredient-variant-of*, etc.

## 888 **Association Administrative Scenarios**

### 889 **Enumerate Code System Coded Concept Relationship Types**

890 A *Terminology User* wants to determine the set of concept relationship types that are available  
891 within a given code system.

892 Associated Functional Models: [Resolve Available Concept Relationships](#)

### 893 **Identify / Retrieve Concept Associations for a Single Concept**

894 A *Terminology User* wants to identify all the associations that exist for a given concept. This  
895 includes both direct and indirect relationships, and may be depth limited where appropriate. This  
896 includes concept relationships (associations for the concept that are within its native code  
897 system) or concept maps (associations between the specified concept code system and another  
898 code system) or both. Returns a set of triples: the source, the target and the association

899 Associated Functional Models: [Retrieve Concept Relationships for a Single Coded Concept](#)



900 **Identify / Retrieve Associations between Two or More Coded Concepts**

901 A *Terminology User* is required to provide a listing of the concept associations that exist  
902 between coded concepts. For example, these associations may be required as part of a  
903 government regulatory compliance review or audit.

904 This includes concept relationships (associations for the concept that are within its native code  
905 system) or concept maps (associations between the specified concept code system and another  
906 code system) or both. Returns a set of triples: the source, the target and the association.

907 Associated Functional Models: [Retrieve Concept Relationships Between Two Coded Concepts](#),  
908 [Retrieve Concept Maps Between Multiple Coded Concepts](#), [Retrieve Concept Relationship](#)  
909 [Metadata](#), [Retrieve Map Metadata](#)

910 **Import Coded Concept Associations**

911 A *Terminology User* is required to make new coded concept associations available through a  
912 Terminology Server. This may or may not include the removal of previously loaded coded  
913 concept associations from the terminology server. To accomplish this, the Terminology User  
914 may be required to convert the content from the format to a format that the Terminology Server  
915 is capable of importing

916 Associated Functional Models: [Import Concept Relationship](#), [Import Concept Relationship](#)  
917 [Metadata](#), [Import Map](#), [Import Concept Map Metadata](#)

918 **Export Coded Concept Associations**

919 A *Terminology User* wants to export coded concept associations from the Terminology Server.  
920 This may require filtering of the content and converting the format of the export.

921 Associated Functional Models: [Export Concept Relationship](#), [Export Map](#), [Export Map Metadata](#)

922 **Remove Coded Concept Associations**

923 A *Terminology User* is required to remove coded concept associations or coded concept  
924 association versions from the terminology service, rendering them unavailable for subsequent  
925 access by other service functions.

926 Associated Functional Models: [Remove Concept Relationship Version](#), [Remove Map Version](#)

927 **Change Status of Coded Concept Associations**

928 A *Terminology User* is required to activate or inactivate coded concept associations, thus  
929 changing their availability for access by other terminology service functions.

930 Associated Functional Models: [Change Concept Relationship Status](#), [Change Map Status](#)



931 **Register for Association Update Notification**

932 A *Terminology User* wants to receive notification that an element of an association has changed  
933 and thus may require review.

934 Associated Functional Models: [Register For Concept Relationship Update Notification](#), [Revise](#)  
935 [or Remove Concept Relationship Update Notification](#), [Register For Concept Dependency](#)  
936 [Relationship Notification](#), [Revise or Remove Concept Dependency Relationship Notification](#),  
937 [Register for Concept Dependency Map Notification](#), [Revise or Remove Map Update](#)  
938 [Notification](#), [Register For Concept Dependency Map Notification](#), [Revise or Remove Concept](#)  
939 [Dependency Map Notification](#)

940 **Association Search / Query Scenarios**

941 This section outlines Search / Query operations specific to associations and association content.

942 **Resolve Available Associations**

943 A *Terminology User* wants to determine what associations are available on the terminology  
944 service by browsing a list of available associations on the CTS 2 instance. The service  
945 differentiates between coded concept relationships and coded concept maps available for any  
946 specified concept.

947 Associated Functional Models: [Resolve Available Concept Relationships](#), [Resolve Available](#)  
948 [Concept Maps](#)

949 **Validate Associations**

950 A *Terminology User* wants to validate that a given association or set of associations are available  
951 on the CTS 2 service instance based upon specific search criteria.

952 Associated Functional Models: [Validate Relationships Between Coded Concepts](#), [Validate](#)  
953 [Lexical Based Relationships Between Coded Concepts](#), [Validate Rules Based Relationships](#)  
954 [Between Coded Concepts](#), [Validate Mappings Between Coded Concepts](#), [Validate Lexical Based](#)  
955 [Mappings Between Coded Concepts](#), [Validate Rules Based Mappings Between Coded Concepts](#)

956 **Retrieve Association Metadata**

957 A *Terminology User* wants to retrieve metadata on available associations in the CTS 2 service  
958 instance. This may include metadata regarding the code system(s) employed, versions, authoring  
959 / curation content or additional data hosted on the CTS server designated to be used by external  
960 systems (i.e.: XML encoded or OWL formatted mapping rule content).

961 Associated Functional Models: [Retrieve Concept Relationship Metadata](#), [Retrieve Map Metadata](#)

962 **Compare Association Versions**

963 A *Terminology User* wants to compare two or more versions of an association on a CTS 2  
964 service instance by viewing each association version's identifying information or metadata.

965 Associated Functional Models: [Compare Relationships Between Coded Concepts](#), [Compare](#)  
966 [Metadata Between Relationships](#), [Compare Maps Between Coded Concepts](#), [Compare Metadata](#)  
967 [Between Maps](#)

968 **Request / Retrieve Association Instance**

969 A *Terminology User* would like to request or retrieve an association when the metadata for such  
970 is retrieved and viewed from a CTS 2 instance.

971 Associated Functional Models: [Validate Relationships Between Coded Concepts](#), [Validate](#)  
972 [Lexical Based Relationships Between Coded Concepts](#), [Validate Rules Based Relationships](#)  
973 [Between Coded Concepts](#), [Validate Mappings Between Coded Concepts](#), [Validate Lexical Based](#)  
974 [Mappings Between Coded Concepts](#), [Validate Rules Based Mappings Between Coded Concepts](#)

975 **Association Author / Curation Scenarios**

976 **Create / Maintain an Association between Coded Concepts**

977 A *Terminology User* wants to create or maintain (i.e. remove or update) an association between  
978 coded concepts. For example, these associations may be required to map a local Code System to  
979 standard Code Systems in order to be compliant with regulatory reporting policies.

980 Search criteria may be accompanied by a "match algorithm code" that determines how the search  
981 text will be applied. The table below provides an example set of match algorithms. **NOTE:** This  
982 match algorithm list is not exhaustive. It is permissible for service implementations to extend the  
983 list below with additional, custom match algorithms as appropriate, although implementers are  
984 strongly encouraged to register the algorithm code to ensure interoperability.

Match Algorithm Code	Description
IdenticalIgnoreCase	The lower case representation of the target text must match the lower case representation matchText exactly.
Identical	The target text must match the matchText exactly.
StartsWithIgnoreCase	The lower case representation of target text must begin with the lower case representation of matchText.
StartsWith	The target text must begin with the matchText.
EndsWithIgnoreCase	The lower case representation of the target text must end with the lower case representation of matchText.
EndsWith	The target text must end with the matchText.
ContainsPhraseIgnoreCase	The lower case representation of the target text must contain the lower case representation of the matchText.
ContainsPhrase	The target text must contain the matchText.

WordsAnyOrderIgnoreCase	The target text must contain all of the words in the match text, but in any order.
WildCardsIgnoreCase	The match text may contain zero or more 'wild cards', designated by an asterisk (*). Wild cards match 0 or more characters in the target string. The escape character is a backslash(\) meaning that the matchText "a\\*b*" would match any string that begins with the string "a*b".
RegularExpression	The match text may contain regular expressions, as defined in XML Schema Part 2: Datatypes.
NYSIIS	New York State Identification and Intelligence System phonetic encoding

985 Associated Functional Models: [Create Relationship Map Between Coded Concepts](#), [Create](#)  
986 [Lexical Relationship Between Coded Concepts](#), [Create Rules Based Relationship Between](#)  
987 [Coded Concepts](#) [Import Concept Relationship](#), [Import Concept Relationship Revision](#), [Import](#)  
988 [Concept Relationship Metadata](#), [Remove Concept Relationship Version](#), [Import Map](#), [Import](#)  
989 [Concept Map Metadata](#), [Convert Mapping Format](#), [Import Map Revision](#), [Remove Map Version](#)

990 **Create Relationship Type**

991 A *Terminology Author* is required to create a new relationship type that may be used to link two  
992 concepts.

993 Associated Functional Models: [Create Relationship Type](#)

994

995 **Create Lexical Association**

996 A *Terminology User* wants to instantiate an association between two sets of coded concepts  
997 using a set of lexical rules (matching algorithms) to generate the associations

998 Associated Functional Models: [Create Lexical Relationship Between Coded Concepts](#), [Create](#)  
999 [Lexical Mapping Between Coded Concepts](#)

1000 **Create Rules Based Association**

1001 A *Terminology User* wants to instantiate an association between two sets of coded concepts  
1002 using a set of description logic or inference rules that either assert or infer mappings between two  
1003 Code Systems. **NOTE:** These associations may be subject to human review to verify validity

1004 Associated Functional Models: [Create Rules Based Relationship Between Coded Concepts](#),  
1005 [Create Rules Based Mapping Between Coded Concepts](#)

1006 **Validate Relationship Associations between Concepts**

1007 *A Terminology User* wants to determine if a specified relationship type exists between two  
1008 concepts in a code system. This includes both direct and indirect relationships

1009 Associated Functional Models: [Validate Relationships Between Coded Concepts](#), [Validate](#)  
1010 [Lexical Based Relationships Between Coded Concepts](#)

1011 **Validate Map Associations between Coded Concepts**

1012 *A Terminology User* wants to validate that a given Coded Concept has a mapping to a specified  
1013 Coded Concept

1014 Associated Functional Models: [Validate Mappings Between Coded Concepts](#), [Validate Lexical](#)  
1015 [Based Mappings Between Coded Concepts](#)

## 1016 **Assumptions and Dependencies**

### 1017 **Dependencies on other Service Frameworks**

1018 As a service specification, the original CTS specified service discovery APIs. We assume that for  
1019 CTS 2 a service discovery framework is available to aid with discovery and query of the CTS 2  
1020 service, and that the service is queryable by the common service metadata attributes outlined in  
1021 the Service Discovery framework, in addition to terminology service specific metadata outlined  
1022 in section 7.

1023 CTS 2 offers a robust set of API requirements, many of which should be restricted to specific  
1024 user classes. This specification outlines a set of functional profiles that specify the types of  
1025 operations users may perform as part of a profile. CTS 2 assumes that security, identity  
1026 management, and auditing services are available that can implement the necessary user role  
1027 based access requirements outlined in section 6.

### 1028 **CTS Backwards Compatibility**

#### 1029 **Message API Support (MAPI)**

1030 In the original CTS, the Message API component is specific to HL7. Its primary purpose is to  
1031 allow a wide variety of message processing applications to create, validate and translate CD-  
1032 derived data types in a consistent and reproducible fashion. It is assumed that this level of  
1033 functionality will remain specific to HL7, and as such will be managed by developing a profile  
1034 specific to HL7.

#### 1035 **General CTS API Support**

1036 Unless otherwise indicated, it is assumed that CTS 2 provides the functional coverage required  
1037 for backwards compatibility to CTS. It is assumed that areas where CTS 2 compatibility with  
1038 CTS will vary include areas such as:

- 1039 • HL7 Datatypes - Where the version of the datatypes has been updates since CTS was  
1040 developed.
- 1041 • Service Discovery - The CTS service discovery APIs are no longer needed assuming the  
1042 existence of Service Discovery infrastructure.
- 1043 • Separating MAPI APIs into an HL7 specific terminology profile.

## 1044 **HL7 Datatypes**

1045 As an HL7 specification, CTS 2 will make use of the HL7 Datatypes where possible.  
1046 Recognizing that they HL7 Datatypes are an evolving standard, CTS 2 technical  
1047 implementations of CTS 2 will be required to indicate what version of the HL7 Datatypes that  
1048 are currently implemented. This restricts complete backward compatibility to the original CTS,  
1049 as CTS 2 implementations will be implementing a more current version of the HL7 datatypes.

1050 In the event that the development of the CTS 2 Technical Specification identifies gaps in the  
1051 HL7 datatypes, CTS 2 will specify its requirements and feed those requirements to HL7 for  
1052 inclusion in the datatypes specification.

# 1053 **Functional Overloading and Metadata** 1054 **Discovery**

## 1055 **Functional Overloading**

1056 CTS 2 services need to support terminologies with very different designs. Some terminologies  
1057 are well designed, whereas some others are not. The terminologies that are well designed have  
1058 concept uniqueness, concept permanence, unique identifiers, formal definitions, and track history  
1059 well supporting a 'graceful evolution'.

1060 Terminologies that are not well designed lack one or more of these good design practices, and  
1061 need additional modifications in the functional definitions of CTS 2 functions. For example,  
1062 terminologies that reuse concept identifiers among different domains (e.g. 'M' may mean male,  
1063 million or meter) need the domain identifier in addition to the concept identifier to uniquely  
1064 identify a concept.

1065 The objective of CTS 2 is to support various terminologies within a single terminology service,  
1066 and not to standardize terminology design. However, to support terminologies with varying  
1067 designs, several CTS2 functions have to be modified to accept additional input parameters (for  
1068 the terminologies with non-standard designs) to return the output. Thus, the input parameters of a  
1069 given CTS 2 function vary based on the design of the terminology that is being queried. The CTS  
1070 2 function will have required and optional input parameters - the required parameters apply to all

1071 terminologies, but the optional parameters apply only to some terminologies with non-standard  
1072 designs.

1073 However, specifying input parameters as just required or optional leads to confusion in  
1074 implementation. This is insufficient for the user to understand which combinations of optional  
1075 parameters are to be used for a specific terminology. As the number of optional parameters for a  
1076 function increases linearly, the number of combinations of input parameters increases  
1077 exponentially. For a function with  $n$  optional input parameters,  $2^n$  combinations of input  
1078 parameters are possible. It becomes hard for the user to decipher which combinations are valid  
1079 and which combinations should be used for different terminologies. In reality, most of these  
1080 combinations are invalid and cannot be used. In addition, such a function is hard to automate,  
1081 and requires human intervention to specify the parameters that are applicable to each  
1082 terminology. Thus, specifying required and optional parameters alone leads to unnecessary and  
1083 confusing combinatorial explosion and lends poorly to automation.

1084 The problem with optional and required parameters is overcome by 'function overloading'. This  
1085 is similar to overloading a method in Object Oriented Programming (OOP). A given method in  
1086 OOP may have many overloaded variants - each with different inputs, but all returning the same  
1087 output. A variant is called based on the input parameters available to the calling function.  
1088 Similarly, a CTS 2 function will have several functional variants which take different  
1089 combinations of input parameters, but all returning the same output. Only those variants with  
1090 valid combinations of input parameters will be created. The variant to be called depends on the  
1091 terminology that is queried. This technique is known as 'functional overloading' in CTS 2.

1092 Functional overloading requires more effort to create the functional variants, but this reduces the  
1093 combinatorial explosion of optional input parameters, avoids creation of invalid combinations of  
1094 input parameters, and provides a solution to tie a specific terminology to a specific variant for  
1095 each function. This also lends better to automation than specifying the input parameters as just  
1096 optional or required.

1097 Functional overloading still cannot be automated at runtime, because we need a way to tie a  
1098 specific variant of a function to a specific terminology. This is achieved through a metadata  
1099 discovery service.

## 1100 **Metadata Discovery**

1101 The metadata discovery service helps the calling program to discover the available variants of a  
1102 CTS 2 function, the input parameters required for each function, and the terminologies that each  
1103 variant applies to. For example, a function that returns the descriptions of a given concept will  
1104 require the concept identifier as a coded datatype (ConceptCode, CodeSystemId,  
1105 CodeSystemVersion) as the input. However, if a given terminology reuses the concept identifier  
1106 in different domains, we need the domain identifier also to uniquely identify the concept and to  
1107 answer the CTS 2 function. The metadata discovery service will provide the definitions of these  
1108 two variants (the input parameter and datatype), and the terminologies that these two variants  
1109 apply to, in this case.

1110 However, creating a relationship between a given terminology and a functional variant can be  
 1111 daunting given the number of CTS 2 functions and the number of terminologies supported. This  
 1112 can be overcome by grouping the terminologies together based on common design and structural  
 1113 characteristics. This is achieved by using semantic profiles. A given semantic profile groups  
 1114 together terminologies with similar designs. Many such semantic profiles are thus possible. The  
 1115 metadata discovery function will list the variants applicable to different semantic profiles, rather  
 1116 than different terminologies.

1117 The CTS 2 authors define the semantic profiles as a set of design characteristics, and assign the  
 1118 better known terminologies into these profiles. The CTS 2 functions will then have overloaded  
 1119 variants based on common design characteristics, rather than those based on individual  
 1120 terminologies. The metadata discovery service will provide the available variants of each CTS 2  
 1121 function and the different semantic profiles they apply to (rather than the different  
 1122 terminologies).

1123 Functions that do not have overloaded variants will just have the single (default) variant returned  
 1124 by the metadata discovery service. Terminologies that are not yet classified need to be classified  
 1125 into a semantic profile. This is discussed in detail under the Semantic Profiles section.

1126 By using functional overloading, metadata discovery and semantic profiles together, the  
 1127 combinatorial explosion and invalid combinations are avoided, the ambiguity and the amount of  
 1128 effort required are reduced, and automation is made easier.

## 1129 Detailed Functional Model for each Interface

### 1130 Administration Functions

#### 1131 Import Terminology

<b>Description</b>	Installs a terminology into the terminology service for subsequent access by other service functions.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology source</li> <li>2. Terminology Version</li> <li>3. Source URI</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgement indicating whether the terminology has been successfully loaded or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Terminology source is available in a format directly consumable by CTS 2 import tools.</li> </ol>

<b>Post Conditions</b>	1. The terminology is available for access via the CTS 2 service functions.
<b>Exception Conditions</b>	1. Terminology source is not consumable by CTS 2 import tools. 2. Information pertaining to the failure is logged and reported for analysis and serviceability.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Administration
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Import Content</a>

1132 **Import Terminology Revision**

<b>Description</b>	Installs a new version of an already loaded terminology into the terminology server repository.
<b>Inputs</b>	1. Terminology source. 2. Terminology version. 3. Terminology revision source. 4. Terminology revision version. 5. Source URI
<b>Outputs</b>	1. An acknowledgment indicating weather the terminology revision has been successfully loaded or not.
<b>Invariants</b>	
<b>Precondition</b>	1. CTS 2 Service installed and running. 2. Existing Terminology is loaded into the terminology service. 3. Existing Terminology must be active. 4. Terminology revision source is available in a format directly consumable by CTS 2 import tools
<b>Post Conditions</b>	1. The terminology revision is available for access via the CTS 2 service functions.
<b>Exception Conditions</b>	1. Terminology does not exist. 2. Terminology is not active.



	<ol style="list-style-type: none"> <li>3. Terminology source is not consumable by CTS 2 import tools.</li> <li>4. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Administration
Miscellaneous notes	<p>Terminology revisions may be available as either:</p> <ol style="list-style-type: none"> <li>1. complete code systems</li> <li>2. set of deltas to be applied sequentially to the previous version.</li> </ol> <p>In either case, all previous versions/iterations should be available until specifically removed.</p>
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Import Content</a>

1133 **Export Terminology**

<b>Description</b>	Exports a terminology, terminology subset or map from the Terminology Server by filtering the content and converting to the requested format for export.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology source.</li> <li>2. Terminology version.</li> <li>3. Terminology subset criteria.</li> <li>4. Terminology map.</li> <li>5. Export Format.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating whether the terminology, terminology subset or map has been successfully exported or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Existing Terminology is loaded into the terminology service.</li> <li>3. Existing Terminology must be active.</li> <li>4. Terminology source is available for export.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. The terminology is available for access via the CTS 2 service</li> </ol>

	functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology does not exist.</li> <li>2. Terminology is not active.</li> <li>3. Terminology source is not exportable by CTS 2 export tools.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Administration
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Export Content</a>

1134 **Remove Terminology / Terminology Version**

<b>Description</b>	Removes a terminology or terminology version from the terminology service, rendering it unavailable for subsequent access by other service functions.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology identifier to be removed.</li> <li>2. Terminology version (optional).</li> </ol>
<b>Outputs</b>	An acknowledgement indicating whether the terminology / terminology version has been successfully removed or not.
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Terminology to be removed is available in CTS 2 service.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. The terminology / terminology revision is no longer available for access via the CTS 2 service functions.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology does not exist.</li> <li>2. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Administration

Miscellaneous notes	If no version is specified, all versions of a specified terminology will be removed.
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Remove Content</a>

1135 **Change Terminology Status**

<b>Description</b>	Make a code system either active or inactive. This allows a <i>Terminology Administrator</i> to activate or inactivate a given terminology, thus changing its availability for access by other terminology service functions.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Code system version.</li> <li>3. Flag to indicate whether to activate or inactivate a code system or code system version.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgement indicating weather the source terminology has been successfully activated/inactivated or not.</li> </ol>
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> </ol>
Post Conditions	<ol style="list-style-type: none"> <li>1. The terminology source is active/inactive making it either available or unavailable by other terminology service operations.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology does not exist.</li> <li>2. Terminology already active.</li> <li>3. Terminology already not active.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Administration
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Change Content Status</a>

1136 **Convert Terminology Format**

<b>Description</b>	Converts a terminology from its source format into or terminology format that can directly imported (consumed) by the CTS 2 importer.
<b>Inputs</b>	1. Terminology source
<b>Outputs</b>	1. An acknowledgement indicating weather the source terminology has been successfully converted or not.
<b>Invariants</b>	
<b>Precondition</b>	1. CTS 2 Service installed and running. 2. Terminology source is available in its original source format.
<b>Post Conditions</b>	1. The terminology source is available in a format that is readily ingestible by the CTS 2 importers.
<b>Exception Conditions</b>	1. Terminology source is not consumable by CTS 2 convert tools.
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Administration
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Import Content</a>

1137 **Register for Update Notification**

<b>Description</b>	Register to be notified whenever a vocabulary element (code system or value set) is modified in any way.
<b>Inputs</b>	1. URL or other electronic address which to send the terminology element modification notification to. 2. Code System Identifier. 3. Code System Version. 4. Concept Identifier. 5. Value Set Identifier. 6. Value Set Version
<b>Outputs</b>	1. An acknowledgement indicating weather the terminology element notification request was successfully created.

Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> <li>4. Value Set must be loaded into the terminology service.</li> <li>5. Value Set must be active.</li> <li>6. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
Post Conditions	<ol style="list-style-type: none"> <li>1. Notification records are updated appropriately</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Code System does not exist.</li> <li>2. Code System version does not exist.</li> <li>3. Value Set does not exist.</li> <li>4. Value Set version does not exist.</li> <li>5. Coded Concept does not exist.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Administration
Miscellaneous notes	Subsequent notifications do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Update Notification</a>

1138 **Revise or Remove Update Notification**

<b>Description</b>	Revise or remove a notification entry for a particular vocabulary element.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Notification Entry Identifier</li> <li>2. URL or other electronic address which to send the terminology element modification notification to.</li> <li>3. Code System Identifier.</li> <li>4. Code System Version.</li> <li>5. Concept Identifier.</li> </ol>

	<ol style="list-style-type: none"> <li>6. Value Set Identifier.</li> <li>7. Value Set Version.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating whether the terminology element notification revision request was successfully received.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> <li>4. Value Set must be loaded into the terminology service.</li> <li>5. Value Set must be active.</li> <li>6. Notification Entry exists.</li> <li>7. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. Notification records are updated appropriately.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Notification Entry Identifier does not exist.</li> <li>2. Code System does not exist.</li> <li>3. Code System version does not exist.</li> <li>4. Value Set does not exist.</li> <li>5. Value Set version does not exist.</li> <li>6. Coded Concept does not exist.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Administration
<b>Miscellaneous notes</b>	Subsequent notifications do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Update Notification Management</a>

1139 **Register for Concept Dependency Notification**

<b>Description</b>	Register to be notified whenever a concept dependency is updated.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. URL or other electronic address which to send the terminology element modification notification to.</li> <li>2. Code System Identifier.</li> <li>3. Code System Version.</li> <li>4. Concept Identifier.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating weather the concept dependency notification request was received or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> <li>4. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. Notification records are updated appropriately.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Code System does not exist.</li> <li>2. Code System version does not exist.</li> <li>3. Coded Concept does not exist.</li> <li>4. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Administration
<b>Miscellaneous notes</b>	Subsequent notifications do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Content Dependency Notification</a>

1140 **Revise or Remove Concept Dependency Notification**

<b>Description</b>	Revise or remove a notification entry for a particular vocabulary element
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Notification Entry Identifier</li> <li>2. URL or other electronic address which to send the terminology element modification notification to.</li> <li>3. Code System Identifier.</li> <li>4. Code System Version.</li> <li>5. Concept Identifier.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating weather the terminology element notification revision request was received or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> <li>4. Notification Entry exists.</li> <li>5. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. Notification records are updated appropriately.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Notification Entry Identifier does not exist.</li> <li>2. Code System does not exist.</li> <li>3. Code System version does not exist.</li> <li>4. Coded Concept does not exist.</li> <li>5. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Administration
<b>Miscellaneous notes</b>	Subsequent notifications do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Other relevant content</b>	
<b>Associated</b>	<a href="#">Content Dependency Notification</a>



<b>Scenario</b>	
-----------------	--

1141 **Search / Access**

1142 **Code System Search / Access**

1143 **Resolve Available Code Systems**

<b>Description</b>	Resolve the code systems available by this instance of the CTS 2 Service
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Value Set Identifier (Optional)</li> <li>3. Value Set Name (Optional)</li> <li>4. Value Set version (Optional)</li> <li>5. Metadata attributes/properties of the code system (Optional)</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. A listing of the code systems and code system metadata properties available on the specified instance of the terminology service.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System Identifier not found.</li> <li>3. Code System Name not found.</li> <li>4. Code System version not found.</li> <li>5. Metadata attributes/properties of the value set not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Query/Search
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Resolve Available Code Systems</a>

1144 **Resolve Code System Metadata**

<b>Description</b>	Resolve the metadata attributes for a given code system available on the terminology service
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code System Name</li> <li>3. Code System Identifier</li> <li>4. Code System Version</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Detailed code system description (resolved meta data or attributes for the code system.)</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System Identifier not found.</li> <li>3. Code System Name not found.</li> <li>4. Code System version not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Query/Search
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Resolve Available Code Systems</a>

1145 **Resolve Code System Concepts**

<b>Description</b>	Returns the set of all (or all active) concepts in the specified code system.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code System Name</li> <li>3. Code System Identifier</li> <li>4. Code System Version</li> <li>5. Boolean for active concepts only.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. The set of all (or all active) concepts in the specified code</li> </ol>

	system.
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System Identifier not found.</li> <li>3. Code System Name not found.</li> <li>4. Code System version not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Query/Search
<b>Miscellaneous notes</b>	Returning all concepts in a code system is generally impractical for large code sets. Indexing, query optimization is necessary.
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Retrieve Coded Concepts from Code System</a>

1146 **Resolve Concept Details**

<b>Description</b>	Resolve the details for the known attributes (metadata) of a coded concept
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code System Name</li> <li>3. Code System Identifier</li> <li>4. Code System Version</li> <li>5. Coded Concept Identifier</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>
	<p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code System Name</li> <li>3. Code System Identifier</li> </ol>

	<ol style="list-style-type: none"> <li>4. Code System Version</li> <li>5. Coded Concept Identifier</li> <li>6. Domain Identifier</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. The details of the attributes (metadata) of the coded concept</li> </ol>
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> <li>4. Coded concept must exist.</li> </ol>
Post Conditions	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System Name not found.</li> <li>3. Code System Identifier not found.</li> <li>4. Code System Version not found.</li> <li>5. Coded Concept Identifier not found.</li> <li>6. Domain Identifier not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Query/Search
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Retrieve Coded Concepts from Value Set</a> , <a href="#">Resolve Concept Representations</a>

1147 **Resolve Coded Concept from Code System**

<b>Description</b>	Given a set of attributes for a coded concept, allow for the search of entries that match the criteria specified on the query attributes
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code System Name</li> <li>3. Code System Identifier</li> <li>4. Code System Version</li> <li>5. List of attribute(s)</li> </ol>

	6. Matching algorithm (optional)
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgement that a list of coded concepts for the search predicate has been found or no matches.</li> <li>2. List of coded concepts from the code system that match the search criteria.</li> <li>3. List of coded concepts from the code system that match the search attributes.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> <li>4. Coded Concept Attribute must exist.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System Name does not exist.</li> <li>3. Code System Identifier does not exist.</li> <li>4. Code System Version does not exist.</li> <li>5. Coded Concept Identifier does not exist.</li> <li>6. Domain Identifier does not exist.</li> <li>7. List of attribute(s) does not exist.</li> <li>8. Matching algorithm does not exist.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Query/Search
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Retrieve Coded Concepts from Code System</a>

1148 **Resolve Relations between Coded Concepts**

<b>Description</b>	Determine whether there is a directed relation (or transitive closure relation) from the source code to the target code.
<b>Inputs</b>	<p>Variant 1</p> <ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code System Name</li> <li>3. Code System Identifier</li> </ol>

	<ol style="list-style-type: none"> <li>4. Code System Version</li> <li>5. Source Coded Concept Identifier</li> <li>6. Target Coded Concept Identifier</li> <li>7. Association Identifier</li> <li>8. Boolean Indicator to indicate if direct associations are considered or whether the transitive closure of the relation are used.</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2</p> <ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code System Name</li> <li>3. Code System Identifier</li> <li>4. Code System Version</li> <li>5. Source Coded Concept Identifier</li> <li>6. Domain Identifier of Source Coded Concept</li> <li>7. Target Coded Concept Identifier</li> <li>8. Domain Identifier of Target Coded Concept</li> <li>9. Association Identifier</li> <li>10. Boolean Indicator to indicate if direct associations are considered or whether the transitive closure of the relation are used.</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Return True if a directed relation exists</li> <li>2. Return False if a directed relation does not exist</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> <li>4. Source Coded Concept must exist.</li> <li>5. Target Coded Concept must exist.</li> <li>6. Association must exist.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System Name not found.</li> <li>3. Code System Identifier not found.</li> <li>4. Code System Version not found.</li> <li>5. Source Coded Concept Identifier not found.</li> <li>6. Domain Identifier of Source Coded Concept not found.</li> <li>7. Target Coded Concept Identifier not found.</li> </ol>

	<ul style="list-style-type: none"> <li>8. Domain Identifier of Target Coded Concept not found.</li> <li>9. Association Identifier not found.</li> </ul>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Query/Search
Miscellaneous notes	Boolean input parameter determines whether only direct associations are considered or whether the transitive closure of the relation is used
<b>Other relevant content</b>	

1149 **Compare Code Systems**

<b>Description</b>	Compare two or more code systems via the metadata properties of the code systems.
<b>Inputs</b>	<ul style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code System id (1)</li> <li>3. Code System version (1)</li> <li>4. Code System id (2)</li> <li>5. Code System version (2)</li> <li>6. Code System id (n)</li> <li>7. Code System version (n)</li> </ul>
<b>Outputs</b>	<ul style="list-style-type: none"> <li>1. The comparison result from the two code systems is returned.</li> </ul>
<b>Invariants</b>	
<b>Precondition</b>	<ul style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> </ul>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ul style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System not found.</li> <li>3. Code System version not found.</li> </ul>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Query /Search
Miscellaneous notes	Comparison of Code Systems may include set comparisons such as

	intersection, difference and union. This comparison is ONLY at the Code System container level, and does NOT include comparison of the contents of the Code System.
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Compare Code System Versions</a>

1150 **Compare Code System Contents**

<b>Description</b>	Compare the contents of two or more Code Systems.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code System id (1)</li> <li>3. Code System version (1)</li> <li>4. Code System id (2)</li> <li>5. Code System version (2)</li> <li>6. Code System id (n)</li> <li>7. Code System version (n)</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. The result of the compare of the contents of the Code Systems.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System not found</li> <li>3. Code System version not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Query/Search
<b>Miscellaneous notes</b>	Comparison of Code Systems may include set comparisons such as intersection and difference and union. This comparison is on the contents of the Code Systems (Concepts) and is not a comparison of the Code System metadata attributes.
<b>Other relevant</b>	



<b>content</b>	
<b>Associated Scenario</b>	<a href="#">Compare Code System Versions</a>

1151 **Value Set Search / Access**

1152 **Compare Value Sets**

<b>Description</b>	Compare two or more value sets via the metadata properties of the value sets.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Value Set id (1)</li> <li>3. Value Set version (1)</li> <li>4. Value Set id (2)</li> <li>5. Value Set version (2)</li> <li>6. Value Set id (n)</li> <li>7. Value Set version (n)</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. The comparison result from the two value sets is returned.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> <li>4. Value Sets for comparison must exist.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Value Set not found.</li> <li>3. Value Set version not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Query /Search
<b>Miscellaneous notes</b>	Comparison of value set may include set comparisons such as intersection, difference and union. This comparison is ONLY at the value set container level, and does NOT include comparison of the contents of the value sets
<b>Other relevant content</b>	

<b>Associated Scenario</b>	<a href="#">Compare Value Set Versions</a>
----------------------------	--

1153 **Compare Value Set Contents**

<b>Description</b>	Compare the contents of two or more value sets.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Value Set id (1)</li> <li>3. Value Set version (1)</li> <li>4. Value Set id (2)</li> <li>5. Value Set version (2)</li> <li>6. Value Set id (n)</li> <li>7. Value Set version (n)</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. The result of the compare of the contents of the value sets.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Value Set must be loaded into the terminology service.</li> <li>3. Value Set must be active.</li> <li>4. Value Sets for comparison must exist.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Value Set not found</li> <li>3. Value Set version not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Query/Search
<b>Miscellaneous notes</b>	Comparison of value sets may include set comparisons such as intersection and difference and union. This comparison is on the contents of the value sets (Value Set Entries and PickList Entries) and is not a comparison of the Value Set metadata attributes (Value Set container object.)
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Compare Value Set Versions</a>

1154 **Resolve Available Value Sets**

<b>Description</b>	Resolve the value sets that are available to the CTS 2 service.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Value Set Identifier (Optional)</li> <li>3. Value Set Name (Optional)</li> <li>4. Value Set version (Optional)</li> <li>5. Code Systems that comprise the values of the value set (Optional)</li> <li>6. Metadata attributes/properties of the value set (Optional)</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Listing of the available value sets on this instance of the terminology server.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Value Set must be loaded into the terminology service.</li> <li>3. Value Set must be active.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Value Set Identifier not found.</li> <li>3. Value Set Name not found.</li> <li>4. Value Set version not found.</li> <li>5. Code Systems that comprise the values of the value set not found.</li> <li>6. Metadata attributes/properties of the value set not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Query/Search
<b>Miscellaneous notes</b>	<p>When search attributes are applied, the result set is restricted to the value sets that match the search attribute criteria. Examples include:</p> <ol style="list-style-type: none"> <li>1. restricting to matching properties such as: <ol style="list-style-type: none"> <li>1. Value Set Identifier</li> <li>2. Value Set Name</li> <li>3. value Set version</li> <li>4. Code Systems that comprise the values of the value set</li> <li>5. Metadata attributes/properties of the value set</li> </ol> </li> </ol>

<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Resolve Available Value Sets</a>

1155 **Resolve Value Set Metadata**

<b>Description</b>	Look up detailed information (metadata) for a given value set.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Value Set Identifier</li> <li>3. Value Set Name</li> <li>4. Value Set version</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Detailed value set description (resolved meta data or attributes for the value set.)</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Value Set must be loaded into the terminology service.</li> <li>3. Value Set must be active.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Value Set Identifier not found.</li> <li>3. Value Set Name not found.</li> <li>4. Value Set version not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Query/Search
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Resolve Available Value Sets</a>

1156 **Resolve Value Set Entries**

<b>Description</b>	Resolve the contents (entries) of a given value set
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Value Set Identifier</li> </ol>

	<ol style="list-style-type: none"> <li>3. Value Set Name</li> <li>4. Value Set version (Optional)</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. A set representing all entries for the given value set</li> </ol>
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Value Set must be loaded into the terminology service.</li> <li>3. Value Set must be active.</li> </ol>
Post Conditions	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Value Set Identifier not found.</li> <li>3. Value Set Name not found.</li> <li>4. Value Set version not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Query/Search
Miscellaneous notes	Value sets may not be finite (e.g. the set of all reals between 1 and 10) Obviously we don't want to list them all. We need to limit the result set to a reasonable amount.
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Retrieve Coded Concepts from Value Set</a>

1157 **Resolve Value Set Entry**

<b>Description</b>	Determine whether the supplied coded concept exists in the supplied value set
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code System Name</li> <li>3. Code System Identifier</li> <li>4. Code System Version</li> <li>5. Value Set Name</li> <li>6. Value Set Identifier</li> <li>7. Value Set Version</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Return True if coded concept exists in value set</li> </ol>

	2. Return False if coded concept does not exist in value set
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> <li>4. Value Set must be loaded into the terminology service.</li> <li>5. Value Set must be active.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System Identifier not found.</li> <li>3. Code System Name not found.</li> <li>4. Code System version not found.</li> <li>5. Value Set Identifier not found.</li> <li>6. Value Set Name not found.</li> <li>7. Value Set version not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Query/Search
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Validate Coded Concept in Value Set</a>

1158 **Authoring/Curation**

1159 **Code System Authoring/Curation**

1160 **Create Code System**

<b>Description</b>	Create a new Code System to contain a set of new coded concepts. The Code System is created by defining the set of meta-data properties that describe it.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Code System Name</li> <li>3. Code System Version</li> <li>4. Code System properties</li> </ol>

<b>Outputs</b>	1. An acknowledgment indicating weather the code system was created or not.
Invariants	
<b>Precondition</b>	1. CTS 2 Service installed and running.
<b>Post Conditions</b>	1. The code system is available for access via the CTS 2 service functions.
<b>Exception Conditions</b>	1. Terminology service not available. 2. Code System already exists.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Authoring
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Create Code System</a>

1161 Maintain Code System

<b>Description</b>	Update Code System meta-data properties.
<b>Inputs</b>	1. Terminology service instance identifier 2. Code System Name 3. Code System Version 4. Code System properties
<b>Outputs</b>	1. An acknowledgment indicating weather the code system was updated or not.
Invariants	
<b>Precondition</b>	1. CTS 2 Service installed and running. 2. Code System must be loaded into the terminology service. 3. Code System must be active.
<b>Post Conditions</b>	1. The code system is available for access via the CTS 2

	service functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System does not exist.</li> <li>3. Code System version does not exist.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Authoring
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Maintain Code System</a>

1162 **Create Concept**

<b>Description</b>	Create concept to be included in a Code System. The new concept is defined by the set of meta-data properties that describe it, which may include its proper placement via association binding within the hierarchy of the Code System.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Code System Name</li> <li>3. Code System Version</li> <li>4. Concept Name</li> <li>5. Concept Properties</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating weather the concept was created or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. The concept is available in the code system and is available for access via the CTS 2 service functions.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System does not exist.</li> <li>3. Code System version does not exist.</li> </ol>



	4. Concept already exists.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Authoring
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Create Concept</a>

1163 **Maintain Concept**

<b>Description</b>	Update Concept meta-data properties.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Code System Name</li> <li>3. Code System Version</li> <li>4. Concept Name</li> <li>5. Concept properties</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating weather the concept was updated or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. The concept is updated appropriately.</li> <li>2. A new version of the code system is created.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System does not exist.</li> <li>3. Code System version does not exist.</li> <li>4. Concept does not exist.</li> </ol>
<b>Aspects left to RFP Submitters</b>	

Relationship to levels of conformance	Authoring
Miscellaneous notes	<p>Updates include but is not limited to functionality such as:</p> <ol style="list-style-type: none"> <li>1. making updates to the associated concept attributes,</li> <li>2. changing the presentation</li> <li>3. changing preferred name</li> <li>4. changing synonymy</li> <li>5. technical corrections to the concept</li> <li>6. modifying the associations bound to concepts</li> </ol>
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Maintain Concept</a>

1164 **Deprecate Concept**

<b>Description</b>	Deprecated an obsolete or ambiguous concept. In many cases, the deprecated concept is replaced with other new concepts.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Code System Name</li> <li>3. Code System Version</li> <li>4. Concept Identifier</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating weather the concept was deprecated or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. The concept is updated appropriately.</li> <li>2. A new version of the code system is created.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System does not exist.</li> <li>3. Code System version does not exist.</li> <li>4. Concept does not exist.</li> </ol>
<b>Aspects left to RFP Submitters</b>	

Relationship to levels of conformance	Authoring
Miscellaneous notes	In keeping with good vocabulary practice, codes or identifiers for concepts cannot be reused. Additionally, in hierarchical Code Systems, it may be necessary to re-associate any concepts related to the concept being deprecated to prevent a part of the code system hierarchy from being orphaned
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Deprecate Concept</a>

1165 Create Relationship Type

<b>Description</b>	Create a new relationship type that may be used to link two concepts.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Code System Name</li> <li>3. Code System Version</li> <li>4. Relationship Type Name</li> <li>5. Relationship Type Properties</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating whether the relationship type was created or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. The relationship type is available in the code system and is available for access via the CTS 2 service functions.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Code System does not exist.</li> <li>3. Code System version does not exist.</li> <li>4. Relationship Type already exists.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of	Authoring

conformance	
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Create Relationship Type</a>

1166 **Value Set Authoring/Curation**

1167 **Create Value Set by Intension**

<b>Description</b>	Create a dynamic Value Set that is defined by a computable expression that can be resolved to an exact list of coded concepts at any given point in time.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Value Set Name</li> <li>3. Value Set Version</li> <li>4. Value Set Properties</li> <li>5. Value Set Expression</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating weather the value set was created or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. The value set is available for access via the CTS 2 service functions.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Value Set already exists.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Authoring
Miscellaneous notes	<ol style="list-style-type: none"> <li>1. Example expression: an intensional value set might be expressed as, “SNOMED CT concepts that are children of the SNOMED CT concept “Diabetes Mellitus.”</li> <li>2. When creating an intensionally defined value set, the Terminology User may or may not bind the value set definition to a specific version of the Code System(s) from which the concepts are being drawn.</li> </ol>

	<ol style="list-style-type: none"> <li>3. If the value set expression is bound to a specific version of the Code System(s), the value set will always resolve the same set of concept codes for any given version of the value set.</li> <li>4. If the value set expression is not bound to a specific version of the Code System(s), the value set will resolve a different set of concept codes as the version of the Code System changes.</li> </ol>
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Create Value Set by Intension</a>

1168 **Create Value Set by Extension**

<b>Description</b>	Create an enumerated (static) value set that is comprised of an explicitly enumerated set of codes.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Value Set Name</li> <li>3. Value Set Version</li> <li>4. Value Set Properties</li> <li>5. Enumerated set of concepts</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating weather the value set was created or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. The value set is available for access via the CTS 2 service functions.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Value Set already exists.</li> <li>3. Concept not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Authoring
Miscellaneous notes	
<b>Other relevant content</b>	

<b>Associated Scenario</b>	<a href="#">Create Value Set by Extension</a>
----------------------------	---

1169 **Maintain Value Set (Intension)**

<b>Description</b>	Update properties or expression of a value set by definition.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Value Set Name</li> <li>3. Value Set Version</li> <li>4. Value Set Properties</li> <li>5. Value Set Expression</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating weather the value set was updated or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. The value set is available for access via the CTS 2 service functions.</li> <li>2. A new value set version is created.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Value Set not available.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Authoring
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Maintain Value Set (Definition)</a>

1170 **Maintain Value Set (Extension)**

<b>Description</b>	Update properties or concepts of an enumerated (static) value set.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Value Set Name</li> <li>3. Value Set Version</li> <li>4. Value Set Properties</li> </ol>

	5. Enumerated set of concepts
<b>Outputs</b>	1. An acknowledgment indicating whether the value set was updated or not.
Invariants	
<b>Precondition</b>	1. CTS 2 Service installed and running.
<b>Post Conditions</b>	1. The value set is available for access via the CTS 2 service functions. 2. A new value set version is created.
<b>Exception Conditions</b>	1. Terminology service not available. 2. Value Set not available. 3. Concept not found.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Authoring
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Maintain Value Set (Enumeration)</a>

1171 **Change Request Processing**

1172 **Create Change Request**

<b>Description</b>	Create a change request that can be reviewed by other terminology users and ultimately submitted to the Terminology Provider for consideration as a change to the terminology.
<b>Inputs</b>	1. Terminology service instance identifier 2. Change requirements.
<b>Outputs</b>	1. A CTS 2 formatted change request.
Invariants	
<b>Precondition</b>	1. CTS 2 Service installed and running.
<b>Post Conditions</b>	1. A new change request is created.

<b>Exception Conditions</b>	1. Terminology service not available.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Authoring
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Create Change Request</a>

1173 Edit Change Request

<b>Description</b>	Edit the content of the change request prior to it being submitted to the Terminology Provider.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Change Request identifier</li> <li>3. Change Request version</li> <li>4. Change requirements.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. A CTS 2 formatted change request.</li> </ol>
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. A new change request is created.</li> <li>2. A new change request version is created.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Change request is not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Authoring
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Edit Change Request</a>



1174 **Submit Change Request**

<b>Description</b>	Submit a change request or a package of several change requests to the Terminology Provider for review.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Change Request identifier</li> <li>3. Change Request version</li> <li>4. Change Request Package</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgment indicating whether the change request or package was submitted or not.</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. Change Request or Package is submitted.</li> <li>2. Change Request or Package status is changed for the affected proposal(s) so that they can no longer be modified by other Terminology Users.</li> </ol>
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Change request is not found.</li> <li>3. Package is not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Authoring
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Submit Change Request</a>

1175 **Package Change Request**

<b>Description</b>	Group a set of change requests together to be submitted to the Terminology Provider to be considered as a set of changes to the terminology.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier</li> <li>2. Change Request identifiers and corresponding versions</li> </ol>

<b>Outputs</b>	1. An acknowledgment indicating whether the package was created or not.
Invariants	
<b>Precondition</b>	1. CTS 2 Service installed and running.
<b>Post Conditions</b>	1. Package is created.
<b>Exception Conditions</b>	1. Terminology service not available. 2. Change request is not found.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Authoring
Miscellaneous notes	
<b>Other relevant content</b>	1. Any individual change request can only be a part of one package.
<b>Associated Scenario</b>	<a href="#">Package Change Request</a>

1176 **Code System Relationships and Maps**

1177 **Concept Relationships**

1178 **Resolve Available Concept Relationships**

<b>Description</b>	Resolve the concept relationships available by this instance of the CTS 2 Service
<b>Inputs</b>	1. Terminology service instance identifier. 2. Code System identifier
<b>Outputs</b>	A listing of the concept relationships for a specified code system available on the specified instance of the terminology service
Invariants	
<b>Precondition</b>	1. CTS 2 Service installed and running. 2. Code System must be loaded into the terminology service. 3. Code System must have coded concept relationships.

Post Conditions	none
Exception Conditions	<ol style="list-style-type: none"> <li>1. Terminology service not available</li> <li>2. Code system specified not available</li> <li>3. Code system specified does not have concept relationships on terminology service</li> <li>4. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
Aspects left to RFP Submitters	
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	
Associated Scenario	<a href="#">Enumerate Code System Coded Concept Relationship Types</a> , <a href="#">Resolve Available Associations</a>

1179 **Retrieve Concept Relationships for a Single Coded Concept**

<b>Description</b>	Returns all concept relationships for a given coded concept
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Terminology Service identifier</li> <li>2. Code System identifier</li> <li>3. Code System version</li> <li>4. Coded Concept identifier</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>
	<p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Terminology Service identifier</li> <li>2. Code System identifier</li> <li>3. Code System version</li> <li>4. Coded Concept identifier</li> <li>5. Domain Identifier of Coded Concept</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Code system description</li> <li>2. Code system identifier</li> <li>3. Code system version</li> <li>4. A list of concept relationships for specified concept</li> </ol>

	5. Target concepts for each concept relationship
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 service is installed and running</li> <li>2. Code systems loaded and available on one or more instances of a terminology service.</li> <li>3. Specified concept is on terminology service</li> <li>4. Specified concept has concept one or more relationships on terminology service</li> </ol>
<b>Post Conditions</b>	System displays a list of relationships
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> <li>2. Source or target coded concept not found.</li> <li>3. Concept relationships for specified coded concepts not found</li> <li>4. Information pertaining to the failure is logged and reported for analysis and serviceability</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Identify / Retrieve Concept Associations for a Single Concept</a>

1180 **Retrieve Concept Relationships between Two Coded Concepts**

<b>Description</b>	Given two or more coded concepts, returns the set of all concept relationships between the concepts within their native code system.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Terminology service identifier</li> <li>2. Code system identifier*</li> <li>3. Code system version*</li> <li>4. Source coded concept*</li> <li>5. Target Coded concept*</li> </ol> <ul style="list-style-type: none"> <li>• indicates required fields</li> </ul> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>

	<p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Terminology service identifier</li> <li>2. Code system identifier*</li> <li>3. Code system version*</li> <li>4. Source coded concept*</li> <li>5. Domain identifier of source coded concept*</li> <li>6. Target Coded concept*</li> <li>7. Domain identifier of target coded concept*</li> </ol> <ul style="list-style-type: none"> <li>• indicates required fields</li> </ul> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Code system description</li> <li>2. Code system identifier</li> <li>3. Code system version</li> <li>4. A list of concept relationships with concept relationship version for each</li> </ol>
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 service is installed and running</li> <li>2. Code systems loaded and available on one or more instances of a terminology service.</li> <li>3. Specified source and target concepts are on terminology service</li> <li>4. Specified source and target concepts have concept relationships on terminology service</li> </ol>
Post Conditions	System displays a list of relationships.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> <li>2. Source or target coded concept not found.</li> <li>3. Concept relationships for specified coded concepts not found</li> <li>4. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Relationship
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Identify / Retrieve Associations Between Two or More Coded Concepts</a>

1181 Retrieve Concept Relationship Metadata

<b>Description</b>	Look up detailed information (metadata) for a given concept relationship
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Concept relationship identifier</li> <li>3. Concept relationship version</li> </ol>
<b>Outputs</b>	<p>All available concept relationship information (resolved meta data or attributes for the concept relationship.) Including:</p> <ol style="list-style-type: none"> <li>1. Code system description</li> <li>2. Code system identifier</li> <li>3. Code system version</li> <li>4. Coded concept relationship description</li> <li>5. Coded concept relationship identifier</li> <li>6. Coded concept relationship version</li> <li>7. Authoring / curation information</li> <li>8. External systems coded concept relationship data hosted on the CTS server (i.e.: XML encoded or OWL formatted relationship rule content).</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Terminology native to concept and concept relationship must be loaded into the terminology service</li> <li>3. Concept relationship must be loaded into the terminology service.</li> </ol>
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Concept relationship does not exist.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Identify / Retrieve Associations Between Two or More Coded Concepts</a> , <a href="#">Retrieve Association Metadata</a>

1182 **Import Concept Relationship**

<b>Description</b>	Installs a concept relationship into the terminology service for subsequent access by other service functions.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Relationship Identifier*</li> <li>2. Relationship Descriptor*</li> <li>3. Relationship Source*</li> <li>4. Relationship Target*</li> <li>5. Relationship Version *</li> <li>6. Relationship Type</li> <li>7. Relationship Restrictions</li> <li>8. Relationship Cardinality</li> <li>9. Relationship Group</li> <li>10. Relationship Order</li> <li>11. Relationship is Reciprocal</li> <li>12. Relationship is Refinable</li> <li>13. Relationship is Transitive</li> <li>14. Relationship is Cyclic</li> <li>15. Relationship is Inheritable</li> <li>16. Relationship Curation / Authoring Information</li> <li>17. External systems coded concept relationship data hosted on the CTS server (i.e.: XML encoded or OWL formatted relationship rule content).</li> </ol> <ul style="list-style-type: none"> <li>• indicates required inputs</li> </ul> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Relationship Identifier*</li> <li>2. Relationship Descriptor*</li> <li>3. Relationship Source*</li> <li>4. Domain identifier of Relationship Source</li> <li>5. Relationship Target*</li> <li>6. Domain identifier of Relationship Target</li> <li>7. Relationship Version *</li> <li>8. Relationship Type</li> <li>9. Relationship Restrictions</li> <li>10. Relationship Cardinality</li> <li>11. Relationship Group</li> <li>12. Relationship Order</li> <li>13. Relationship is Reciprocal</li> <li>14. Relationship is Refinable</li> </ol>

	<p>15. Relationship is Transitive 16. Relationship is Cyclic 17. Relationship is Inheritable 18. Relationship Curation / Authoring Information 19. External systems coded concept relationship data hosted on the CTS server (i.e.: XML encoded or OWL formatted mapping rule content).</p> <ul style="list-style-type: none"> <li>• indicates required inputs</li> </ul> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	An acknowledgement indicating whether the concept relationship has been successfully loaded or not.
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Terminology source and targets are available in a format directly consumable by CTS 2 import tools.</li> </ol>
Post Conditions	The concept relationship is available for access via the CTS 2 service functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> <li>2. A listing of the input field(s) which did not load correctly is made available and / or displayed</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Import Coded Concept Associations</a>

1183 **Import Concept Relationship Revision**

<b>Description</b>	Installs a new version of an already loaded concept relationship into the terminology server repository.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Relationship Identifier*</li> <li>2. Relationship Descriptor*</li> </ol>



3. Relationship Source\*
  4. Relationship Target\*
  5. Relationship Version \*
  6. Relationship Type
  7. Relationship Restrictions
  8. Relationship Cardinality
  9. Relationship Group
  10. Relationship Order
  11. Relationship is Reciprocal
  12. Relationship is Refinable
  13. Relationship is Transitive
  14. Relationship is Cyclic
  15. Relationship is Inheritable
  16. Relationship Curation / Authoring Information
  17. External systems coded concept relationship data hosted on the CTS server (i.e.: XML encoded or OWL formatted relationship rule content).
- indicates required inputs

Applies to: [Mature Terminology Profile](#)

Variant 2:

1. Relationship Identifier\*
2. Relationship Descriptor\*
3. Relationship Source\*
4. Domain identifier of relationship source
5. Relationship Target\*
6. Domain identifier of relationship target
7. Relationship Version \*
8. Relationship Type
9. Relationship Restrictions
10. Relationship Cardinality
11. Relationship Group
12. Relationship Order
13. Relationship is Reciprocal
14. Relationship is Refinable
15. Relationship is Transitive
16. Relationship is Cyclic
17. Relationship is Inheritable
18. Relationship Curation / Authoring Information

External systems coded concept relationship data hosted on the CTS server (i.e.: XML encoded or OWL formatted mapping rule content).

	<ul style="list-style-type: none"> <li>indicates required inputs</li> </ul> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>An acknowledgement indicating whether the concept relationship has been successfully loaded or not.</li> <li>Relationship identifier, descriptor, version, date and time of successful concept relationship update is made available and / or displayed</li> </ol>
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>CTS 2 Service installed and running.</li> <li>Concept Relationships are available in CTS 2 repository</li> </ol>
Post Conditions	The revision is available for access via the CTS 2 service functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>Information pertaining to the failure is logged and reported for analysis and serviceability.</li> <li>A listing of the input field(s) which did not load correctly is made available and / or displayed</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	

1184 **Export Concept Relationship**

<b>Description</b>	Exports a specified version of an already loaded concept relationship from the terminology server repository
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>Relationship Identifier</li> <li>Relationship Descriptor</li> <li>Relationship Source</li> <li>Relationship Target</li> <li>Relationship Version</li> <li>Requestor</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p>

	<ol style="list-style-type: none"> <li>1. Relationship Identifier</li> <li>2. Relationship Descriptor</li> <li>3. Relationship Source</li> <li>4. Domain identifier of relationship source</li> <li>5. Relationship Target</li> <li>6. Relationship Version</li> <li>7. Requestor</li> <li>8. Domain identifier of relationship target</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. All available data in terminology server repository for concept relationship version specified.</li> <li>2. Requestor name, date and time of export request</li> </ol>
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Concept relationships are available in CTS 2 repository</li> <li>3. Requested concept relationship and version is available in CTS2 repository</li> </ol>
Post Conditions	The concept relationship is available for access by CTS 2 external functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Information pertaining to the export failure is logged and reported for analysis and serviceability.</li> <li>2. A listing of the cause(s) of export failure is made available and / or displayed</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Export Coded Concept Associations</a>

1185 **Import Concept Relationship Metadata**

<b>Description</b>	Installs an updated version of metadata to an already loaded concept relationship into the terminology server repository.
<b>Inputs</b>	Variant 1:

	<ol style="list-style-type: none"> <li>1. Relationship Identifier</li> <li>2. Relationship Source</li> <li>3. Relationship Target</li> <li>4. Relationship Curation / Metadata elements to be updated</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>
	<p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Relationship Identifier</li> <li>2. Relationship Source</li> <li>3. Domain identifier of relationship source</li> <li>4. Relationship Target</li> <li>5. Domain identifier of relationship target</li> <li>6. Relationship Curation / Metadata elements to be updated</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgement indicating whether the concept relationship metadata has been successfully loaded or not.</li> <li>2. Relationship identifier, descriptor, version, date and time of successful concept relationship update is made available and / or displayed</li> </ol>
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Specified concept Relationship is available in CTS 2 repository</li> </ol>
Post Conditions	The revision is available for access via the CTS 2 service functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> <li>2. A listing of the input field(s) which did not load correctly is made available and / or displayed</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Import Coded Concept Associations</a>

<b>Description</b>	Removes a concept relationship version from the terminology service, rendering it unavailable for subsequent access by other service functions
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept relationship identifier.</li> <li>2. Concept relationship version.</li> </ol>
<b>Outputs</b>	An acknowledgement indicating whether the concept relationship version has been successfully removed or not.
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Concept relationship to be removed is available in CTS 2 service.</li> </ol>
<b>Post Conditions</b>	The concept relationship is no longer available for access via the CTS 2 service functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept relationship does not exist.</li> <li>2. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Remove Coded Concept Associations</a>

1187 **Change Concept Relationship Status**

<b>Description</b>	Make a concept relationship either active or inactive. This allows a Terminology User to activate or inactivate a given concept relationship, thus changing its availability for access by other terminology service functions
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept relationship identifier.</li> <li>2. Concept relationship version.</li> <li>3. Flag to indicate whether to activate or inactivate a concept relationship within specified code system(s) or code system version(s).</li> </ol>
<b>Outputs</b>	An acknowledgement indicating whether the concept relationship has been successfully activated/inactivated or not.

Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Concept relationship must be loaded into the terminology service.</li> </ol>
Post Conditions	The concept relationship is active/inactive making it either available or unavailable by other terminology service operations.
<b>Exception Conditions</b>	
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Change Status of Coded Concept Associations</a>

1188 **Create Concept Relationship between Coded Concepts**

<b>Description</b>	Relates a coded concept within a specified code system (source) to a corresponding coded concept (target) within that system.
<b>Inputs</b>	<p>Variant 1</p> <ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Source code system terminology service identifier.</li> <li>3. Target Code system terminology service identifier.</li> <li>4. Source coded concept.</li> <li>5. Target Coded concept.</li> <li>6. Optional source code system version.</li> <li>7. Optional target code system version.</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>
	<p>Variant 2</p> <ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Source code system terminology service identifier.</li> <li>3. Target Code system terminology service identifier.</li> <li>4. Source coded concept.</li> <li>5. Domain Identifier of Source Coded Concept</li> <li>6. Target Coded concept.</li> </ol>

	<ol style="list-style-type: none"> <li>7. Domain Identifier of Target Coded Concept</li> <li>8. Optional source code system version.</li> <li>9. Optional target code system version.</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	A concept relationship is created between a coded concept and a coded concept in the specified code system
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must be active.</li> <li>4. Source Coded Concept must exist.</li> <li>5. Target Coded Concept must exist.</li> </ol>
Post Conditions	A concept relationship is created between a coded concepts in the specified code system and are present for use on the terminology service..
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source Code System does not exist.</li> <li>2. Target Code System does not exist.</li> <li>3. Source Coded concept not found.</li> <li>4. Target Coded concept not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	RelationshipMapping
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Maintain Concept Relationship/Association</a>

1189 **Create Lexical Relationship between Coded Concepts**

<b>Description</b>	Relates a coded concept within a specified code system (source)to a corresponding coded concept (target) within that system using a set of lexical rules (matching algorithms) to generate the relationships.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Code system identifier</li> <li>2. Source code system terminology service identifier (if source and target are on different systems)</li> </ol>

	<ol style="list-style-type: none"> <li>3. Target Code system terminology service identifier (if source and target are on different systems)</li> <li>4. Source coded concept</li> <li>5. Target Coded concept</li> <li>6. Search Criteria</li> <li>7. Match Algorithm Code</li> <li>8. Optional source code system version (if source and target are on different systems)</li> <li>9. Optional target code system version (if source and target are on different systems)</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Code system identifier</li> <li>2. Source code system terminology service identifier (if source and target are on different systems)</li> <li>3. Target Code system terminology service identifier (if source and target are on different systems)</li> <li>4. Source coded concept</li> <li>5. Domain identifier of source coded concept</li> <li>6. Target Coded concept</li> <li>7. Domain identifier of target coded concept</li> <li>8. Search Criteria</li> <li>9. Match Algorithm Code</li> <li>10. Optional source code system version (if source and target are on different systems)</li> <li>11. Optional target code system version (if source and target are on different systems)</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	Relationships are created between created between a coded concepts in the specified code system and are present for use on the terminology service.
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Source Code System must be loaded into the terminology service.</li> <li>3. Source Code System must be active.</li> <li>4. Target Code System must be loaded into the terminology service.</li> <li>5. Target Code System must be active.</li> <li>6. Source Coded Concept must exist.</li> <li>7. Target Coded Concept must exist.</li> </ol>
<b>Post Conditions</b>	Relationships are created between one or more coded concepts from a source code system and one or more coded concepts in the target code system.



<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source Code System does not exist.</li> <li>2. Target Code System does not exist.</li> <li>3. Source Coded concept not found.</li> <li>4. Target Coded concept not found.</li> <li>5. No coded concepts match the search criteria for the specified match algorithm.</li> </ol>	
<b>Aspects left to RFP Submitters</b>		
Relationship to levels of conformance	Relationship	
Miscellaneous notes		
<b>Other relevant content</b>	<b>Match Algorithm Code</b>	<b>Description</b>
	IdenticalIgnoreCase	The lower case representation of the target text must match the lower case representation matchText exactly.
	Identical	The target text must match the matchText exactly.
	StartsWithIgnoreCase	The lower case representation of target text must begin with the lower case representation of matchText.
	StartsWith	The target text must begin with the matchText.
	EndsWithIgnoreCase	The lower case representation of the target text must end with the lower case representation of matchText.
	EndsWith	The target text must end with the matchText.
	ContainsPhraseIgnoreCase	The lower case representation of the target text must contain the lower case representation of the matchText.
	ContainsPhrase	The target text must contain the matchText.
	WordsAnyOrderIgnoreCase	The target text must contain all of the words in the match text, but in any order.
	WildCardsIgnoreCase	The match text may contain zero or more 'wild cards', designated by an asterisk (*). Wild cards match 0 or more characters in the target string. The escape character is a backslash('\) meaning that the matchText "a*b*" would match any string that begins with the string "a*b".
	RegularExpression	The match text may contain regular

		expressions, as defined in XML Schema Part 2: Datatypes.
	NYSIIS	New York State Identification and Intelligence System phonetic encoding
<b>Associated Scenario</b>	<a href="#">Maintain Concept Relationship/Association</a>	

1190 **Create Rules Based Relationship Between Coded Concepts**

<b>Description</b>	Relates a coded concept within a specified code system (source) to a corresponding coded concept (target) within that system using a set of description logic or inference rules that either assert or infer relationships
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Source code system terminology service identifier (when source and target reside on different services).</li> <li>3. Target Code system terminology service identifier (when source and target reside on different services).</li> <li>4. Source coded concept.</li> <li>5. Target Coded concept.</li> <li>6. Description Logic (text string)</li> <li>7. Inference Rules (text string)</li> <li>8. Optional source code system version (when source and target reside on different services).</li> <li>9. Optional target code system version (when source and target reside on different services).</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Source code system terminology service identifier (when source and target reside on different services).</li> <li>3. Target Code system terminology service identifier (when source and target reside on different services).</li> <li>4. Source coded concept.</li> <li>5. Domain identifier of Source coded concept.</li> <li>6. Target Coded concept.</li> <li>7. Domain identifier of Target coded concept.</li> <li>8. Description Logic (text string)</li> <li>9. Inference Rules (text string)</li> <li>10. Optional source code system version (when source and target reside on different services).</li> </ol>

	<p>11. Optional target code system version (when source and target reside on different services).</p> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	Relationships are created between created between a coded concepts in the specified code system.
Invariants	
<b>Precondition</b>	Specified code system is loaded and available on one or more instances of a terminology service.
Post Conditions	Relationships are created between one or more coded concepts from a source code system and one or more coded concepts in the target code system.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> <li>2. Source or target coded concept not found.</li> <li>3. No coded concepts satisfy the description logic or inference rules.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Relationship
Miscellaneous notes	
<b>Other relevant content</b>	These relationships are subject to human review to verify validity.
<b>Associated Scenario</b>	<a href="#">Maintain Concept Relationship/Association</a>

1191 **Compare Relationships Between Coded Concepts**

<b>Description</b>	Compare two or more concept relationships
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept relationships id (1)</li> <li>2. Concept relationships version (1)</li> <li>3. Concept relationships id (2)</li> <li>4. Concept relationships (2)</li> <li>5. Concept relationships id (n)</li> <li>6. Concept relationships version (n)</li> </ol>
<b>Outputs</b>	Identifying information about the two or more concept relationships specified are returned.
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Concept relationships for comparison must exist.</li> </ol>
Post Conditions	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept relationship(s) not found.</li> <li>2. Concept relationship(s) version not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Compare Association Versions</a>

1192 **Compare Metadata Between Relationships**

<b>Description</b>	Compare metadata between two or more concept relationships
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept relationships id (1)</li> <li>2. Concept relationships version (1)</li> <li>3. Concept relationships id (2)</li> <li>4. Concept relationships (2)</li> <li>5. Concept relationships id (n)</li> <li>6. Concept relationships version (n)</li> </ol>
<b>Outputs</b>	All data from the two or more specified concept relationships are returned.
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Concept relationships for comparison must exist</li> </ol>
Post Conditions	All metadata for specified concept relationships is returned / displayed.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept relationship(s) not found.</li> <li>2. Concept relationship(s) version not found.</li> <li>3. Concept relationship(s) do not have metadata on system</li> </ol>
<b>Aspects left to RFP Submitters</b>	

Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Compare Association Versions</a>

1193 **Validate Relationships Between Coded Concepts**

<b>Description</b>	Given two or more coded concepts and relationship type, determine if any of the specified relationships exist between the concepts
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Source code system terminology service identifier (when source and target reside on different services).</li> <li>3. Target Code system terminology service identifier (when source and target reside on different services).</li> <li>4. Source coded concept.</li> <li>5. Target Coded concept.</li> <li>6. Relationship type.</li> <li>7. Optional code system version.</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Source code system terminology service identifier (when source and target reside on different services).</li> <li>3. Target Code system terminology service identifier (when source and target reside on different services).</li> <li>4. Source coded concept.</li> <li>5. Domain identifier of Source coded concept.</li> <li>6. Target Coded concept.</li> <li>7. Domain identifier of Target coded concept.</li> <li>8. Relationship type.</li> <li>9. Optional source code system version (when source and target reside on different services).</li> <li>10. Optional target code system version (when source and target reside on different services).</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	Boolean
<b>Invariants</b>	

<b>Precondition</b>	Source and target code systems loaded and available on one or more instances of a terminology service.
Post Conditions	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Code systems not found.</li> <li>2. Coded concept not found.</li> <li>3. Relationship type not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Relationship
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Validate Associations</a>

1194 **Validate Lexical Based Relationships Between Coded Concepts**

<b>Description</b>	Given two or more coded concepts and lexical search criteria and match algorithm, determine if any of the specified lexical based relationships exist between the concepts.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Source code system terminology service identifier (when source and target reside on different services).</li> <li>3. Target Code system terminology service identifier (when source and target reside on different services).</li> <li>4. Source coded concept.</li> <li>5. Target Coded concept.</li> <li>6. Lexical Search Criteria and Match Algorithm.</li> <li>7. Optional code system version.</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>
	<p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Source code system terminology service identifier (when source and target reside on different services).</li> <li>3. Target Code system terminology service identifier (when source and target reside on different services).</li> <li>4. Source coded concept.</li> <li>5. Domain identifier of Source coded concept.</li> <li>6. Target Coded concept.</li> </ol>

	<ol style="list-style-type: none"> <li>7. Domain identifier of Target coded concept.</li> <li>8. Lexical Search Criteria and Match Algorithm.</li> <li>9. Optional code system version.</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	Boolean
Invariants	
<b>Precondition</b>	Source and target code systems loaded and available on one or more instances of a terminology service.
Post Conditions	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> <li>2. Source or target coded concepts not found.</li> <li>3. Lexical Search Criteria not found.</li> <li>4. Match Algorithm not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Relationship
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Validate Associations</a>

1195 **Validate Rules Based Relationships Between Coded Concepts**

<b>Description</b>	Given two or more coded concepts and description logic or inference rules (and optional code system identifiers,) determine if any of the specified rules based relationships exist between the concepts
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Code system terminology service identifier.</li> <li>3. Source coded concept.</li> <li>4. Target Coded concept.</li> <li>5. Description Logic.</li> <li>6. Inference Rules.</li> <li>7. Optional code system version.</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p>

	<ol style="list-style-type: none"> <li>1. Code system identifier.</li> <li>2. Source Code system terminology service identifier (when source and target reside on different services).</li> <li>3. Target Code system terminology service identifier (when source and target reside on different services).</li> <li>4. Source coded concept.</li> <li>5. Domain identifier of Source coded concept.</li> <li>6. Target Coded concept.</li> <li>7. Domain identifier of Target coded concept.</li> <li>8. Description Logic.</li> <li>9. Inference Rules.</li> <li>10. Optional source code system version.</li> <li>11. Optional target code system version.</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	Boolean
Invariants	
<b>Precondition</b>	Source and target code system(s) loaded and available on one or more instances of a terminology service.
Post Conditions	
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> <li>2. Source or target coded concept not found.</li> <li>3. Description Logic not found.</li> <li>4. Inference Rules not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Relationship
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Validate Associations</a>

1196 **Register for Concept Relationship Update Notification**

<b>Description</b>	Register to be notified whenever a concept relationship is updated in the registry.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. URL or other electronic address which to send the Concept Relationship modification notification to.</li> <li>2. Concept Relationship Identifier.</li> </ol>



	3. Concept Relationship Version.
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Display of Concept Relationship Update Notification Identifier</li> <li>2. A record of the transmission of an update notification</li> <li>3. An acknowledgement indicating whether the concept relationship notification request was received or not</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. CTS 2 Service has sufficient network access to allow notification message to be sent to specified URL or electronic address.</li> <li>3. Concept relationship must be loaded into the terminology service.</li> <li>4. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept relationship notification identifier is created if none previously existed.</li> <li>2. Concept relationship update notification identifier records are updated appropriately</li> <li>3. Transmission of notification is recorded</li> <li>4. Acknowledgement of notification message transmission is recorded</li> </ol>
<b>Exception Conditions</b>	Information pertaining to any failures are logged and reported for analysis and serviceability.
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	Subsequent notifications for updates to the same concept relationship do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Associated Scenario</b>	<a href="#">Register for Association Update Notification</a>

1197 **Revise or Remove Concept Relationship Update Notification**

<b>Description</b>	Revise or remove a notification entry for a particular concept relationship
--------------------	---

<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept relationship notification entry identifier</li> <li>2. URL or other electronic address which to send the concept relationship notification modification notification to.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Display of Concept Relationship Update Notification Identifier</li> <li>2. A record of the transmission of an update notification revision or removal</li> <li>3. An acknowledgement indicating whether the concept relationship notification revision or removal was received or not</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. CTS 2 Service has sufficient network access to allow notification message to be sent to specified URL or electronic address.</li> <li>3. Code System must be loaded into the terminology service.</li> <li>4. Notification Entry exists.</li> <li>5. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept relationship update notification identifier records are updated appropriately</li> <li>2. Transmission of notification revision or removal is recorded</li> <li>3. Acknowledgement of notification revision or removal message transmission is recorded</li> </ol>
<b>Exception Conditions</b>	Information pertaining to the failure is logged and reported for analysis and serviceability.
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	Subsequent notifications for revisions to the same concept relationship do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Associated Scenario</b>	<a href="#">Register for Association Update Notification</a>

<b>Description</b>	Register to be notified whenever the state of a code system or concept specified as a target or source for a concept relationship is updated in the registry.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. URL or other electronic address which to send the Concept Relationship modification notification to.</li> <li>2. Concept relationship identifier.</li> <li>3. Concept relationship source concept.</li> <li>4. Concept relationship source version.</li> <li>5. Concept relationship target concept.</li> <li>6. Concept relationship target version.</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. URL or other electronic address which to send the Concept Relationship modification notification to.</li> <li>2. Concept relationship identifier.</li> <li>3. Concept relationship source concept.</li> <li>4. Domain identifier of Concept relationship source.</li> <li>5. Concept relationship source version.</li> <li>6. Concept relationship target concept.</li> <li>7. Domain identifier of Concept relationship target.</li> <li>8. Concept relationship target version.</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Display of Concept Relationship Update Notification Identifier</li> <li>2. A record of the transmission of an update notification</li> <li>3. An acknowledgement indicating whether the concept relationship notification request was received or not</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Concept relationship must be loaded into the terminology service.</li> <li>3. Source / Target Code System must be loaded into the terminology service.</li> <li>4. CTS 2 Service has sufficient network access to allow notification message to be sent to specified URL or electronic address.</li> <li>5. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept relationship notification identifier is created if none previously existed.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Concept relationship update notification identifier records are updated appropriately</li> <li>3. Transmission of notification is recorded</li> <li>4. Acknowledgement of notification message transmission is recorded</li> </ol>
<b>Exception Conditions</b>	Information pertaining to any failures are logged and reported for analysis and serviceability.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	Subsequent notifications for updates to the same concept relationship do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Associated Scenario</b>	<a href="#">Register for Association Update Notification</a>

1199 **Revise or Remove Concept Dependency Relationship Notification**

<b>Description</b>	Revise or remove a notification entry for a particular concept relationship created as a result of the change in a source or target concept.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept relationship notification entry identifier</li> <li>2. URL or other electronic address which to send the concept relationship notification modification notification to.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Display of Concept Relationship Update Notification Identifier</li> <li>2. A record of the transmission of an update notification revision or removal</li> <li>3. An acknowledgement indicating whether the concept relationship notification revision or removal was received or not</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. CTS 2 Service has sufficient network access to allow notification message to be sent to specified URL or electronic address.</li> <li>3. Source / Target Code System must be loaded into the terminology service.</li> </ol>

	<ol style="list-style-type: none"> <li>4. Notification Entry exists.</li> <li>5. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
Post Conditions	<ol style="list-style-type: none"> <li>1. Concept relationship update notification identifier records are updated appropriately</li> <li>2. Acknowledgement of notification revision or removal message transmission is recorded</li> </ol>
<b>Exception Conditions</b>	Information pertaining to the failure is logged and reported for analysis and serviceability.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	Subsequent notifications for revisions to the same concept relationship do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Associated Scenario</b>	<a href="#">Register for Association Update Notification</a>

1200 **Concept Maps**

1201 **Resolve Available Concept Maps**

<b>Description</b>	Resolve the concept maps available by this instance of the CTS 2 Service
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Code system identifier</li> </ol>
<b>Outputs</b>	A listing of the concept maps for a specified code system available on the specified instance of the terminology service
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Code System must have concept maps.</li> </ol>

Post Conditions	None.
Exception Conditions	<ol style="list-style-type: none"> <li>1. Terminology service not available</li> <li>2. Code system specified not available</li> <li>3. Code system specified does not have concept maps on terminology service</li> <li>4. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
Aspects left to RFP Submitters	
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	
Associated Scenario	<a href="#">Resolve Available Associations</a>

1202 Retrieve Concept Maps for a Specified Coded Concept

<b>Description</b>	Returns all concept maps for a given coded concept
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Terminology service identifier</li> <li>2. Code system identifier</li> <li>3. Code system version</li> <li>4. Coded concept</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>
	<p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Terminology service identifier</li> <li>2. Code system identifier</li> <li>3. Code system version</li> <li>4. Coded concept</li> <li>5. Domain identifier of Coded concept</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Code system description</li> <li>2. Code system identifier</li> <li>3. Code system version</li> <li>4. A list of concept maps for specified concept</li> <li>5. Target concepts for each concept map</li> </ol>

Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 service is installed and running</li> <li>2. Code systems loaded and available on one or more instances of a terminology service.</li> <li>3. Specified concept is on terminology service</li> <li>4. Specified concept has concept one or more maps on terminology service</li> </ol>
Post Conditions	System displays a list of maps
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> <li>2. Source or target coded concept not found.</li> <li>3. Concept maps for specified coded concepts not found</li> <li>4. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	

1203 **Retrieve Concept Maps between Multiple Coded Concepts**

<b>Description</b>	
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Terminology service identifier</li> <li>2. Code system identifier*</li> <li>3. Code system version*</li> <li>4. Source coded concept*</li> <li>5. Source coded concept version</li> <li>6. Target Coded concept *</li> <li>7. Target coded concept version</li> </ol> <ul style="list-style-type: none"> <li>• indicates required fields</li> </ul> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Terminology service identifier</li> <li>2. Code system identifier*</li> <li>3. Code system version*</li> </ol>

	<ol style="list-style-type: none"> <li>4. Source coded concept*</li> <li>5. Domain identifier of Source coded concept*</li> <li>6. Source coded concept version</li> <li>7. Target Coded concept *</li> <li>8. Domain identifier of Target coded concept*</li> <li>9. Target coded concept version</li> </ol> <ul style="list-style-type: none"> <li>• indicates required fields</li> </ul> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. a list of concept maps</li> <li>2. a concept map version for each map listed</li> <li>3. Code system identifier, version and description for source</li> <li>4. Code system identifier version and description for target(s)</li> <li>5. Terminology service identifier(s)</li> <li>6. Domain identifiers, as applicable</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 service is installed and running</li> <li>2. Code systems loaded and available on one or more instances of a terminology service.</li> <li>3. Specified source and target concepts are on terminology service</li> <li>4. Specified source and target concepts have concept maps on terminology service</li> </ol>
<b>Post Conditions</b>	System displays a list of maps
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> <li>2. Source or target coded concept not found.</li> <li>3. Concept maps for specified coded concepts not found</li> <li>4. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Mapping
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Identify / Retrieve Associations Between Two or More Coded Concepts</a>



<b>Description</b>	Look up detailed information (metadata) for a given concept map
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Terminology service instance identifier.</li> <li>2. Concept map identifier</li> <li>3. Concept map version</li> </ol>
<b>Outputs</b>	<p>All available concept map information (resolved meta data or attributes for the concept map.) Including:</p> <ol style="list-style-type: none"> <li>1. Code system description</li> <li>2. Code system identifier</li> <li>3. Code system version</li> <li>4. Coded concept map description</li> <li>5. Coded concept map identifier</li> <li>6. Coded concept map version</li> <li>7. Authoring / curation information</li> <li>8. External systems coded concept relationship data hosted on the CTS server (i.e.: XML encoded or OWL formatted map rule content).</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Terminologies for source and target of concept map must be loaded into the terminology service</li> <li>3. Concept map must be loaded into the terminology service.</li> </ol>
<b>Post Conditions</b>	None
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Terminology service not available.</li> <li>2. Concept map does not exist.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	
<b>Miscellaneous notes</b>	
<b>Other relevant content'</b>	Look up detailed information (metadata) for a given concept map'
<b>Associated Scenario</b>	<a href="#">Identify / Retrieve Associations Between Two or More Coded Concepts, Retrieve Association Metadata</a>

1205 **Import Map**

<b>Description</b>	Installs a concept map into the terminology service for subsequent
--------------------	--

	access by other service functions.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Map Identifier</li> <li>2. Map Descriptor*</li> <li>3. Map Source*</li> <li>4. Map Target*</li> <li>5. Map Version *</li> <li>6. Map Type</li> <li>7. Map Restrictions</li> <li>8. Map Cardinality</li> <li>9. Map Curation / Authoring information</li> <li>10. External systems coded concept map data hosted on the CTS server (i.e.: XML encoded or OWL formatted mapping rule content).</li> </ol> <ul style="list-style-type: none"> <li>• indicates required inputs</li> </ul> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>
	<p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Map Identifier</li> <li>2. Map Descriptor*</li> <li>3. Map Source*</li> <li>4. Domain identifier of Map Source*</li> <li>5. Map Target*</li> <li>6. Domain identifier of Map Target*</li> <li>7. Map Version *</li> <li>8. Map Type</li> <li>9. Map Restrictions</li> <li>10. Map Cardinality</li> <li>11. Map Curation / Authoring information</li> <li>12. External systems coded concept map data hosted on the CTS server (i.e.: XML encoded or OWL formatted mapping rule content).</li> </ol> <ul style="list-style-type: none"> <li>• indicates required inputs</li> </ul> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	An acknowledgement indicating whether the concept map has been successfully loaded or not.
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Terminology source and targets are available in a format</li> </ol>

	directly consumable by CTS 2 import tools.
Post Conditions	The concept map is available for access via the CTS 2 service functions.
Exception Conditions	<ol style="list-style-type: none"> <li>1. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> <li>2. A listing of the input field(s) that cause the map to not load correctly is made available and / or displayed</li> </ol>
Aspects left to RFP Submitters	
Relationship to levels of conformance	
Miscellaneous notes	
Other relevant content	
Associated Scenario	<a href="#">Import Coded Concept Associations</a>

1206 Import Concept Map Metadata

<b>Description</b>	Installs an updated version of metadata to an already loaded concept map into the terminology server repository.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Map Identifier</li> <li>2. Map Source</li> <li>3. Map Target</li> <li>4. Map Curation / Metadata elements to be updated</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>
	<p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Map Identifier</li> <li>2. Map Source</li> <li>3. Domain identifier of Map source</li> <li>4. Map Target</li> <li>5. Domain identifier of Map target</li> <li>6. Map Curation / Metadata elements to be updated</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgement indicating whether the concept map metadata has been successfully loaded or not.</li> <li>2. Relationship identifier, descriptor, version, date and time of</li> </ol>

	successful concept map update is made available and / or displayed
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Specified concept Relationship is available in CTS 2 repository</li> </ol>
<b>Post Conditions</b>	The revision is available for access via the CTS 2 service functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> <li>2. A listing of the input field(s) which did not load correctly is made available and / or displayed</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Import Coded Concept Associations</a>

1207 **Convert Mapping Format**

<b>Description</b>	Converts a concept map from its source format into a format that can directly imported (consumed) by the CTS 2 importer.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Code system identifier*</li> <li>2. Code system version*</li> <li>3. Source coded concept*</li> <li>4. Source coded concept version</li> <li>5. Target Coded concept *</li> <li>6. Target coded concept version*</li> <li>7. Map Type</li> <li>8. Map Restrictions</li> <li>9. Map Cardinality</li> <li>10. Relationship Curation / Authoring Information</li> </ol> <ul style="list-style-type: none"> <li>• indicates required inputs</li> </ul>
<b>Outputs</b>	An acknowledgement indicating weather the source terminology has been successfully converted or not.
<b>Invariants</b>	

<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Terminology source is available in its original source format.</li> </ol>
Post Conditions	The terminology source is available in a format that is readily ingestible by the CTS 2 importers
<b>Exception Conditions</b>	Terminology source is not consumable by CTS 2 convert tools.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<i>'Other relevant content</i>	

1208 **Import Map Revision**

<b>Description</b>	Installs a new version of an already loaded concept map into the terminology server repository.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Map Identifier*</li> <li>2. Map Descriptor*</li> <li>3. Map Source*</li> <li>4. Map Target*</li> <li>5. Map Version *</li> <li>6. Map Type</li> <li>7. Map Restrictions</li> <li>8. Map Cardinality</li> <li>9. Map Curation / Authoring information</li> <li>10. External systems coded concept map data hosted on the CTS server (i.e.: XML encoded or OWL formatted mapping rule content).</li> </ol> <ul style="list-style-type: none"> <li>• indicates required inputs</li> </ul> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Map Identifier*</li> <li>2. Map Descriptor*</li> <li>3. Map Source*</li> <li>4. Domain identifier of Map Source*</li> <li>5. Map Target*</li> <li>6. Domain identifier of Map Target*</li> </ol>

	<p>7. Map Version *</p> <p>8. Map Type</p> <p>9. Map Restrictions</p> <p>10. Map Cardinality</p> <p>11. Map Curation / Authoring information</p> <p>12. External systems coded concept map data hosted on the CTS server (i.e.: XML encoded or OWL formatted mapping rule content).</p> <ul style="list-style-type: none"> <li>• indicates required inputs</li> </ul> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. An acknowledgement indicating whether the concept map has been successfully loaded or not.</li> <li>2. Map identifier, descriptor, version, date and time of successful concept map update is made available and / or displayed</li> </ol>
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Concept Maps are available in CTS 2 repository</li> </ol>
Post Conditions	The revision is available for access via the CTS 2 service functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> <li>2. A listing of the input field(s) that cause the map to not load correctly is made available and / or displayed</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	

1209 **Export Map**

<b>Description</b>	Exports a specified version of an already loaded concept map from the terminology server repository
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Map Identifier</li> <li>2. Map Descriptor</li> </ol>

	<ol style="list-style-type: none"> <li>3. Map Source</li> <li>4. Map Target</li> <li>5. Map Version</li> <li>6. Requestor</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Map Identifier</li> <li>2. Map Descriptor</li> <li>3. Map Source</li> <li>4. Domain identifier of Map Source</li> <li>5. Map Target</li> <li>6. Domain identifier of Map Target</li> <li>7. Map Version</li> <li>8. Requestor</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. All available data in terminology server repository for concept map version specified.</li> <li>2. Requestor name, date and time of export request</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Concept maps are available in CTS 2 repository</li> <li>3. Requested concept map and version is available in CTS2 repository</li> </ol>
<b>Post Conditions</b>	The concept map is available for access by CTS 2 external functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Information pertaining to the export failure is logged and reported for analysis and serviceability.</li> <li>2. A listing of the cause(s) of export failure is made available and / or displayed</li> </ol>
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Export Coded Concept Associations</a>

--	--

1210 **Export Map Metadata**

<b>Description</b>	Exports the metadata of an already loaded concept map from the terminology server repository.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Map Identifier</li> <li>2. Map Descriptor</li> <li>3. Map Source</li> <li>4. Map Target</li> <li>5. Map Version</li> <li>6. Requestor</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>
	<p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Map Identifier</li> <li>2. Map Descriptor</li> <li>3. Map Source</li> <li>4. Domain identifier of Map Source</li> <li>5. Map Target</li> <li>6. Domain identifier of Map Target</li> <li>7. Map Version</li> <li>8. Requestor</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Metadata in terminology server repository for concept map version specified.</li> <li>2. Requestor name, date and time of export request</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Concept maps are available in CTS 2 repository</li> <li>3. Requested concept map and version is available in CTS2 repository</li> </ol>
<b>Post Conditions</b>	The concept map is available for access by CTS 2 external functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Information pertaining to the export failure is logged and reported for analysis and serviceability.</li> </ol>



	2. A listing of the cause(s) of export failure is made available and / or displayed
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Export Coded Concept Associations</a>

1211 **Remove Map Version**

<b>Description</b>	Removes a concept map version from the terminology service, rendering it unavailable for subsequent access by other service functions
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept map identifier.</li> <li>2. Concept map version.</li> </ol>
<b>Outputs</b>	An acknowledgement indicating whether the concept map version has been successfully removed or not.
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Concept map to be removed is available in CTS 2 service.</li> </ol>
Post Conditions	The concept map is no longer available for access via the CTS 2 service functions.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept map does not exist.</li> <li>2. Information pertaining to the failure is logged and reported for analysis and serviceability.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Remove Coded Concept Associations</a>

1212 **Change Map Status**

<b>Description</b>	Make a concept map either active or inactive. This allows a Terminology User to activate or inactivate a given concept map, thus changing its availability for access by other terminology service functions
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept map identifier.</li> <li>2. Concept map version.</li> <li>3. Flag to indicate whether to activate or inactivate a code system or code system version.</li> </ol>
<b>Outputs</b>	An acknowledgement indicating whether the concept map has been successfully activated/inactivated or not.
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Concept map must be loaded into the terminology service.</li> </ol>
<b>Post Conditions</b>	The concept map is active/inactive making it either available or unavailable by other terminology service operations.
<b>Exception Conditions</b>	
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Change Status of Coded Concept Associations</a>

1213 **Create Map between Coded Concepts**

<b>Description</b>	Maps a coded concept from a specified code system (source) to a coded concept (target) within another system.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Target Coded concept.</li> </ol>

	<p>7. Source code system version. 8. Target code system version.</p> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <p>1. Source code system identifier. 2. Target Code system identifier. 3. Source code system terminology service identifier. 4. Target Code system terminology service identifier. 5. Source coded concept. 6. Domain identifier of Source coded concept. 7. Target Coded concept. 8. Domain identifier of Target Coded concept. 9. Source code system version. 10. Target code system version.</p> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	A mapping is created between a coded concept from a source code system and a coded concept in the target code system
<b>Invariants</b>	
<b>Precondition</b>	<p>1. CTS 2 Service installed and running. 2. Source Code System must be loaded into the terminology service. 3. Source Code System must be active. 4. Target Code System must be loaded into the terminology service. 5. Target Code System must be active. 6. Source Coded Concept must exist. 7. Target Coded Concept must exist.</p>
<b>Post Conditions</b>	A mapping is created between a coded concept from a source code system and a coded concept in the target code system.
<b>Exception Conditions</b>	Source or target code systems not found. Source or target coded concept not found.
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to levels of conformance</b>	Mapping
<b>Miscellaneous notes</b>	
<b>Other relevant content</b>	

1214 **Create Lexical Mapping between Coded Concepts**

<b>Description</b>	Maps the supplied code system id and coded concept to a corresponding coded concept (if any) in the target system using a set of lexical rules (matching algorithms) to generate the relationships.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Target Coded concept.</li> <li>7. Search Criteria (text string),</li> <li>8. Match Algorithm Code,</li> <li>9. Source code system version.</li> <li>10. Target code system version</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <hr/> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Domain identifier of Source coded concept.</li> <li>7. Target Coded concept.</li> <li>8. Domain identifier of Target Coded concept.</li> <li>9. Search Criteria (text string),</li> <li>10. Match Algorithm Code,</li> <li>11. Source code system version.</li> <li>12. Target code system version</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	Mappings are created between one or more coded concepts from a source code system and one or more coded concepts in the target code system.
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Source Code System must be loaded into the terminology service.</li> <li>3. Source Code System must be active.</li> <li>4. Target Code System must be loaded into the terminology service.</li> <li>5. Target Code System must be active.</li> <li>6. Source Coded Concept must exist.</li> </ol>

	7. Target Coded Concept must exist.																						
Post Conditions	Maps are created between one or more coded concepts from a source code system and one or more coded concepts in the target code system.																						
Exception Conditions	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> <li>2. Source or target coded concept not found.</li> <li>3. No coded concepts match the search criteria for the specified match algorithm.</li> </ol>																						
Aspects left to RFP Submitters																							
Relationship to levels of conformance	Mapping																						
Miscellaneous notes																							
Other relevant content	<table border="1"> <thead> <tr> <th>Match Algorithm Code</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>IdenticalIgnoreCase</td> <td>The lower case representation of the target text must match the lower case representation matchText exactly.</td> </tr> <tr> <td>Identical</td> <td>The target text must match the matchText exactly.</td> </tr> <tr> <td>StartsWithIgnoreCase</td> <td>The lower case representation of target text must begin with the lower case representation of matchText.</td> </tr> <tr> <td>StartsWith</td> <td>The target text must begin with the matchText.</td> </tr> <tr> <td>EndsWithIgnoreCase</td> <td>The lower case representation of the target text must end with the lower case representation of matchText.</td> </tr> <tr> <td>EndsWith</td> <td>The target text must end with the matchText.</td> </tr> <tr> <td>ContainsPhraseIgnoreCase</td> <td>The lower case representation of the target text must contain the lower case representation of the matchText.</td> </tr> <tr> <td>ContainsPhrase</td> <td>The target text must contain the matchText.</td> </tr> <tr> <td>WordsAnyOrderIgnoreCase</td> <td>The target text must contain all of the words in the match text, but in any order.</td> </tr> <tr> <td>WildCardsIgnoreCase</td> <td>The match text may contain zero or more 'wild cards', designated by an asterisk (*). Wild cards match 0 or more characters in the target string. The escape character is a backslash('\') meaning</td> </tr> </tbody> </table>	Match Algorithm Code	Description	IdenticalIgnoreCase	The lower case representation of the target text must match the lower case representation matchText exactly.	Identical	The target text must match the matchText exactly.	StartsWithIgnoreCase	The lower case representation of target text must begin with the lower case representation of matchText.	StartsWith	The target text must begin with the matchText.	EndsWithIgnoreCase	The lower case representation of the target text must end with the lower case representation of matchText.	EndsWith	The target text must end with the matchText.	ContainsPhraseIgnoreCase	The lower case representation of the target text must contain the lower case representation of the matchText.	ContainsPhrase	The target text must contain the matchText.	WordsAnyOrderIgnoreCase	The target text must contain all of the words in the match text, but in any order.	WildCardsIgnoreCase	The match text may contain zero or more 'wild cards', designated by an asterisk (*). Wild cards match 0 or more characters in the target string. The escape character is a backslash('\') meaning
	Match Algorithm Code	Description																					
	IdenticalIgnoreCase	The lower case representation of the target text must match the lower case representation matchText exactly.																					
	Identical	The target text must match the matchText exactly.																					
	StartsWithIgnoreCase	The lower case representation of target text must begin with the lower case representation of matchText.																					
	StartsWith	The target text must begin with the matchText.																					
	EndsWithIgnoreCase	The lower case representation of the target text must end with the lower case representation of matchText.																					
	EndsWith	The target text must end with the matchText.																					
	ContainsPhraseIgnoreCase	The lower case representation of the target text must contain the lower case representation of the matchText.																					
	ContainsPhrase	The target text must contain the matchText.																					
	WordsAnyOrderIgnoreCase	The target text must contain all of the words in the match text, but in any order.																					
WildCardsIgnoreCase	The match text may contain zero or more 'wild cards', designated by an asterisk (*). Wild cards match 0 or more characters in the target string. The escape character is a backslash('\') meaning																						

		that the matchText "a*b*" would match any string that begins with the string "a*b".
	RegularExpression	The match text may contain regular expressions, as defined in XML Schema Part 2: Datatypes.
	NYSIIS	New York State Identification and Intelligence System phonetic encoding

1215 **Create Rules Based Mapping between Coded Concepts**

<b>Description</b>	Maps the supplied code system id and coded concept to a corresponding coded concept (if any) in the target system using a set of description logic or inference rules that either assert or infer mappings
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Target Coded concept.</li> <li>7. Description Logic (text string),</li> <li>8. Inference Rules (text string),</li> <li>9. Optional source code system version.</li> <li>10. Optional target code system version.</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Domain identifier of Source coded concept.</li> <li>7. Target Coded concept.</li> <li>8. Domain identifier of Target Coded concept.</li> <li>9. Description Logic (text string),</li> <li>10. Inference Rules (text string),</li> <li>11. Optional source code system version.</li> <li>12. Optional target code system version.</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>

<b>Outputs</b>	A map is created between specified coded concepts.
Invariants	
<b>Precondition</b>	Source and target code systems loaded and available on one or more instances of a terminology service.
Post Conditions	A map is created between one or more coded concepts from a source code system and one or more coded concept in the target code system and is available for use on one or more instances of the terminology server.
<b>Exception Conditions</b>	Source or target code systems not found. Source or target coded concept not found. No coded concepts satisfy the description logic or inference rules.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Mapping
Miscellaneous notes	
<b>Other relevant content</b>	These mappings are subject to human review to verify validity.

1216 Compare Maps between Coded Concepts

<b>Description</b>	Compare two or more concept maps
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept map id (1)</li> <li>2. Concept map version (1)</li> <li>3. Concept map id (2)</li> <li>4. Concept map (2)</li> <li>5. Concept map id (n)</li> <li>6. Concept map version (n)</li> </ol>
<b>Outputs</b>	Identifying information about the two or more concept maps specified are returned.
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Concept maps for comparison must exist.</li> </ol>
Post Conditions	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept map(s) not found.</li> <li>2. Concept map(s) version not found.</li> </ol>

<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Mappings
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Compare Association Versions</a>

1217 Compare Metadata between Maps

<b>Description</b>	Compare metadata between two or more concept maps
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept map id (1)</li> <li>2. Concept map version (1)</li> <li>3. Concept map id (2)</li> <li>4. Concept map (2)</li> <li>5. Concept map id (n)</li> <li>6. Concept map version (n)</li> </ol>
<b>Outputs</b>	All data from the two or more specified concept maps are returned.
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Code System must be loaded into the terminology service.</li> <li>3. Concept maps for comparison must exist.</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept map(s) not found.</li> <li>2. Concept map(s) version not found.</li> <li>3. Concept maps do not contain metadata.</li> </ol>
<b>Exception Conditions</b>	Source or target code systems not found. Source or target coded concept not found. Mapping not found
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Mappings
Miscellaneous notes	
<b>Other relevant content</b>	

1218 Validate Mappings between Coded Concepts

<b>Description</b>	Given two or more coded concepts and mapping (and optional code system identifiers,) determine if any of the specified mappings exist
--------------------	---



	between the concepts.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Target Coded concept.</li> <li>7. Mapping identifier.</li> <li>8. Source code system version.</li> <li>9. Target code system version</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p>
	<p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Domain identifier of Source coded concept.</li> <li>7. Target Coded concept.</li> <li>8. Domain identifier of Target Coded concept.</li> <li>9. Mapping identifier.</li> <li>10. Source code system version.</li> <li>11. Target code system version</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	List of mappings.
Invariants	
<b>Precondition</b>	Source and target code systems loaded and available on one or more instances of a terminology service.
Post Conditions	
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> <li>2. Source or target coded concept not found.</li> <li>3. Mapping not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Mappings
Miscellaneous notes	

<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Validate Associations</a>

1219 **Validate Lexical Based Mappings between Coded Concepts**

<b>Description</b>	Given two or more coded concepts and lexical search criteria and match algorithm, determine if any of the specified lexical based mappings exist between the concepts.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Target Coded concept.</li> <li>7. Lexical Search Criteria. Match Algorithm.</li> <li>8. Source code system version.</li> <li>9. Source target code system version</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Domain identifier of Source coded concept.</li> <li>7. Target Coded concept.</li> <li>8. Domain identifier of Target Coded concept.</li> <li>9. Lexical Search Criteria. Match Algorithm.</li> <li>10. Source code system version.</li> <li>11. Source target code system version</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	Boolean
<b>Invariants</b>	
<b>Precondition</b>	Source and target code systems loaded and available on one or more instances of a terminology service.
<b>Post Conditions</b>	None.
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Source or target coded concept not found.</li> <li>3. Lexical Search Criteria not found.</li> <li>4. Match Algorithm not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Mapping
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Validate Associations</a>

1220 **Validate Rules Based Mappings between Coded Concepts**

<b>Description</b>	Given two or more coded concepts and description logic or inference rules (and optional code system identifiers,) determine if any of the specified rules based mappings exist between the concepts
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Target Coded concept.</li> <li>7. Description Logic.</li> <li>8. Inference Rules.</li> <li>9. Optional source code system version.</li> <li>10. Optional target code system version</li> </ol> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. Source code system identifier.</li> <li>2. Target Code system identifier.</li> <li>3. Source code system terminology service identifier.</li> <li>4. Target Code system terminology service identifier.</li> <li>5. Source coded concept.</li> <li>6. Domain identifier of Source coded concept.</li> <li>7. Target Coded concept.</li> <li>8. Domain identifier of Target Coded concept.</li> <li>9. Description Logic.</li> </ol>

	<p>10. Inference Rules. 11. Optional source code system version. 12. Optional target code system version</p> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	Boolean.
Invariants	
<b>Precondition</b>	Source and target code systems loaded and available on one or more instances of a terminology service.
Post Conditions	
<b>Exception Conditions</b>	<ol style="list-style-type: none"> <li>1. Source or target code systems not found.</li> <li>2. Source or target coded concept not found.</li> <li>3. Description Logic not found.</li> <li>4. Inference Rules not found.</li> </ol>
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	Mapping
Miscellaneous notes	
<b>Other relevant content</b>	
<b>Associated Scenario</b>	<a href="#">Validate Associations</a>

1221 **Register for Concept Dependency Map Notification**

<b>Description</b>	Register to be notified whenever a concept map is updated in the registry.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. URL or other electronic address which to send the Concept Map modification notification to.</li> <li>2. Concept Map Identifier.</li> <li>3. Concept Map Version.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Display of Concept Map Update Notification Identifier</li> <li>2. A record of the transmission of an update notification</li> <li>3. An acknowledgement indicating whether the concept map notification request was received or not</li> </ol>
Invariants	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. CTS 2 Service has sufficient network access to allow notification message to be sent to specified URL or electronic address.</li> </ol>

	<ol style="list-style-type: none"> <li>3. Concept map must be loaded into the terminology service.</li> <li>4. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
Post Conditions	<ol style="list-style-type: none"> <li>1. Concept map notification identifier is created if none previously existed.</li> <li>2. Concept map update notification identifier records are updated appropriately</li> <li>3. Transmission of notification is recorded</li> <li>4. Acknowledgement of notification message transmission is recorded</li> </ol>
<b>Exception Conditions</b>	Information pertaining to any failures are logged and reported for analysis and serviceability.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	Subsequent notifications for updates to the same concept map do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Associated Scenario</b>	<a href="#">Register for Association Update Notification</a>

1222 **Revise or Remove Map Update Notification**

<b>Description</b>	Revise or remove a notification entry for a particular concept map
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept map notification entry identifier</li> <li>2. URL or other electronic address which to send the concept map notification modification notification to.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Display of Concept Map Update Notification Identifier</li> <li>2. A record of the transmission of an update notification revision or removal</li> <li>3. An acknowledgement indicating whether the concept map notification revision or removal was received or not</li> </ol>
<b>Invariants</b>	

<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. CTS 2 Service has sufficient network access to allow notification message to be sent to specified URL or electronic address.</li> <li>3. Code System must be loaded into the terminology service.</li> <li>4. Notification Entry exists.</li> <li>5. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
Post Conditions	<ol style="list-style-type: none"> <li>1. Concept map update notification identifier records are updated appropriately</li> <li>2. Transmission of notification revision or removal is recorded</li> <li>3. Acknowledgement of notification revision or removal message transmission is recorded</li> </ol>
<b>Exception Conditions</b>	Information pertaining to the failure is logged and reported for analysis and serviceability.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content. Revise or remove a notification entry for a particular concept map</b>	Subsequent notifications for revisions to the same concept map do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Associated Scenario</b>	<a href="#">Register for Association Update Notification</a>

1223 **Register for Concept Dependency Map Notification**

<b>Description</b>	Register to be notified whenever the state of a code system or concept specified as a target or source for a concept map is updated in the registry.
<b>Inputs</b>	<p>Variant 1:</p> <ol style="list-style-type: none"> <li>1. URL or other electronic address which to send the Concept Map modification notification to.</li> <li>2. Concept map identifier.</li> <li>3. Concept map source.</li> <li>4. Concept map source version.</li> <li>5. Concept map target.</li> </ol>

	<p>6. Concept map target version.</p> <p>Applies to: <a href="#">Mature Terminology Profile</a></p> <p>Variant 2:</p> <ol style="list-style-type: none"> <li>1. URL or other electronic address which to send the Concept Map modification notification to.</li> <li>2. Concept map identifier.</li> <li>3. Concept map source.</li> <li>4. Domain identifier of Concept map source.</li> <li>5. Concept map source version.</li> <li>6. Concept map target.</li> <li>7. Domain identifier of Concept map target.</li> <li>8. Concept map target version.</li> </ol> <p>Applies to: <a href="#">Developing Terminology Profile</a></p>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Display of Concept Map Update Notification Identifier</li> <li>2. A record of the transmission of an update notification</li> <li>3. An acknowledgement indicating whether the concept map notification request was received or not</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. Concept map must be loaded into the terminology service.</li> <li>3. Source / Target Code System must be loaded into the terminology service.</li> <li>4. CTS 2 Service has sufficient network access to allow notification message to be sent to specified URL or electronic address.</li> <li>5. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept map notification identifier is created if none previously existed.</li> <li>2. Concept map update notification identifier records are updated appropriately</li> <li>3. Transmission of notification is recorded</li> <li>4. Acknowledgement of notification message transmission is recorded</li> </ol>
<b>Exception Conditions</b>	Information pertaining to any failures are logged and reported for analysis and serviceability.
<b>Aspects left to RFP Submitters</b>	
<b>Relationship to</b>	

levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	Subsequent notifications for updates to the same concept map do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Associated Scenario</b>	<a href="#">Register for Association Update Notification</a>

1224 **Revise or Remove Concept Dependency Map Notification**

<b>Description</b>	Revise or remove a notification entry for a particular concept map created as a result of the change in a source or target concept.
<b>Inputs</b>	<ol style="list-style-type: none"> <li>1. Concept map notification entry identifier</li> <li>2. URL or other electronic address which to send the concept map notification modification notification to.</li> </ol>
<b>Outputs</b>	<ol style="list-style-type: none"> <li>1. Display of Concept Map Update Notification Identifier</li> <li>2. A record of the transmission of an update notification revision or removal</li> <li>3. An acknowledgement indicating whether the concept map notification revision or removal was received or not</li> </ol>
<b>Invariants</b>	
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. CTS 2 Service installed and running.</li> <li>2. CTS 2 Service has sufficient network access to allow notification message to be sent to specified URL or electronic address.</li> <li>3. Source / Target Code System must be loaded into the terminology service.</li> <li>4. Notification Entry exists.</li> <li>5. User or appropriate proxy (system administrator, etc) are authorized to access registry</li> </ol>
<b>Post Conditions</b>	<ol style="list-style-type: none"> <li>1. Concept map update notification identifier records are updated appropriately</li> <li>2. Transmission of notification revision or removal is recorded</li> <li>3. Acknowledgement of notification revision or removal message transmission is recorded</li> </ol>
<b>Exception</b>	Information pertaining to the failure is logged and reported for analysis and



<b>Conditions</b>	serviceability.
<b>Aspects left to RFP Submitters</b>	
Relationship to levels of conformance	
Miscellaneous notes	
<b>Other relevant content</b>	Subsequent notifications for revisions to the same concept map do not require a confirmation. Where appropriate, however, negative feedback on the channel (unable to deliver message, unable to connect), should result in attempts to retransmit and/or the placement of a temporary hold on notifications until connection problem is corrected.
<b>Associated Scenario</b>	<a href="#">Register for Association Update Notification</a>

## 1225 Profiles

### 1226 Introduction

1227 A profile is a named set of cohesive capabilities. A profile enables a service to be used at  
 1228 different levels and allows implementers to provide different levels of capabilities in differing  
 1229 contexts. Service-to-service interoperability will be judged at the profile level and not the service  
 1230 level. Note that through the use of profiles, there are no “optional” interfaces. Conditions that  
 1231 might otherwise merit this optionally should be addressed via a dedicated profile.

1232 A set of profiles may be defined that cover specific functions, semantic information and overall  
 1233 conformance. The SSDF explains in detail the meaning of each of these types of profile. In brief,  
 1234 they are as follows:

1235

1236 • **Functional Profile:** a named list of a subset of the operations defined within this  
 1237 specification which must be supported in order to claim conformance to the profile.

1238 • **Semantic Profile:** identification of a named set of information descriptions (e.g.  
 1239 semantic signifiers) that are supported by one or more operations.

1240 • **Conformance Profile:** this is a combination of a set of functional and semantic profiles  
 1241 taken together to give a complete coherent set of capabilities against which conformance  
 1242 can be claimed. This may optionally include additional constraints where relevant.

1243 **CTS 2 Functional Profiles**

1244 **Minimal CTS 2 Profile**

1245 The Minimal CTS 2 Profile specifies the minimal functional coverage necessary for a service to  
 1246 declare itself as being a conformant CTS 2 service. The minimal CTS 2 includes capabilities for  
 1247 searching and query terminology content, representing terminology content in the appropriate  
 1248 HL7 Datatypes, and structuring terminology content appropriately when HL7 Datatypes are not  
 1249 available for representing the necessary terminology content being queried.

1250

Profile	Member Operations	Operation Profile	Notes
<b>Minimal CTS 2 Profile</b>	<a href="#">Resolve Available Code Systems</a>	The ability to provide a listing of the available code systems, as well as the details pertaining to each code systems available on the terminology service.	The Minimal CTS 2 Profile specifies the minimal functional coverage necessary for a service to declare itself as being a conformant CTS 2 service. The Minimal CTS 2 Profile includes the ability to search the contents of code systems and value sets, as well as the ability to administer terminology content with the functions in the <a href="#">Terminology Administration Profile</a>
	<a href="#">Resolve Concepts from Code System</a>	The ability to browse or query the content of a specific code system, including specific concepts, associated attributes (synonyms, associations), as well as the metadata pertaining to each coded concept that meets the desired search criteria	
	<a href="#">Validate Concept in Code System</a>	The ability to validate that a given concept exists in a given version of a code system.	
	<a href="#">Identify Concept Language Translations</a>	The ability to determine the alternate language representations exist for a given Concept.	
	<a href="#">Resolve Concept Representations</a>	The ability to determine what (if any) alternate representations exist for a given concept. Examples of	

		alternate representations for a concept may include abbreviations, or synonyms.
	<a href="#">Compare Code System Versions</a>	The ability to determine what differences exist between different versions or instances of a code system.
	<a href="#">Resolve Available Value Sets</a>	The ability to determine what value sets are available to a Terminology Service. This includes seeing a listing of the available value sets that match some search criteria, as well as the details pertaining to each value set available to the terminology service.
	<a href="#">Retrieve Coded Concepts from Value Set</a>	The ability to see a listing of specific concepts, as well as the details pertaining to each concept in any of the given value sets available to a terminology service.
	<a href="#">Validate Coded Concept in Value Set</a>	The ability to validate that a given concept exists in a given value set.
	<a href="#">Compare Value Set Versions</a>	The ability to determine what differences exist between different versions of a value set.
	<a href="#">Resolve Concept Representations</a>	The ability to determine what (if any) alternate representations exist for a given concept in a value set.
	<a href="#">Terminology Administration Profile</a>	The functional operations necessary for terminology administrators to be able to access and make available terminology content obtained from a

	Terminology Provider.
<a href="#">Resolve Available Associations</a>	The ability to determine what associations are available on the terminology service by browsing a list of available associations on the CTS 2 instance.
<a href="#">Validate Associations</a>	The ability to validate that a given association or set of associations are available on the CTS 2 service instance based upon specific search criteria.
<a href="#">Retrieve Association Metadata</a>	The ability to retrieve metadata on available associations in the CTS 2 service instance.
<a href="#">Compare Association Versions</a>	The ability to compare two or more versions of an association on a CTS 2 service instance by viewing each association version's identifying information or metadata.
<a href="#">Request/Retrieve Association Instance</a>	The ability to request or retrieve an association when the metadata for such is retrieved and viewed from a CTS 2 instance.
<a href="#">Enumerate Concept Relationship Types</a>	The ability to determine the set of concept relationship types that are available within a given code system.
<a href="#">Retrieve Associations for a Given Concept</a>	The ability to identify all the associations that exist for a given concept.
<a href="#">Retrieve Associations Between Multiple Concepts</a>	The ability to provide a listing of the concept associations that exist between a set of coded

		concepts.	
	<a href="#">Validate Relationship Associations between Concepts</a>	The ability to determine if a specified relationship type exists between two concepts in a code system.	
	<a href="#">Validate Map Associations Between Coded Concepts</a>	The ability to validate that a given concept has a mapping to another specified concept.	

1251 **Vocabulary Facilitator Profile**

1252 The Vocabulary Facilitator Profile is intended to support the ability for Vocabulary Facilitators  
 1253 to create, modify, package and submit change requests to a Terminology Provider. Change  
 1254 requests to the terminology do not modify the terminology content directly, but result in a  
 1255 collaborative community consensus recommendation to the Terminology Provider that outlines a  
 1256 requested modification to the source terminology. These change requests can then be reviewed  
 1257 by the Terminology Provider, and when appropriate, included in the next release of the source  
 1258 terminology.

1259

Profile	Member Operations	Operation Profile	Notes
Vocabulary Facilitator Profile	<a href="#">Create Change Request</a>	The ability to create a change request against terminology content that can be reviewed by other terminology users and ultimately submitted to the Terminology Provider for consideration as a change to the terminology.	The Vocabulary Facilitator Profile is intended to support the ability for Vocabulary Facilitators to create, modify, package and submit change requests to a Terminology Provider. Change requests to the terminology do not modify the terminology content directly, but result in a collaborative community consensus recommendation to the Terminology Provider that outlines a requested modification to the source terminology. These change requests can then be reviewed by the Terminology Provider, and when appropriate, included in the next release of the source terminology. This profile includes the functionality outlined in the <a href="#">Minimal</a>
	<a href="#">Edit Change Request</a>	The ability to edit and refine the content of a change request prior to it being submitted to the Terminology Provider for consideration.	
	<a href="#">Submit Change Request</a>	The ability to submit a change request or a set of change requests to the	

		Terminology Provider.	<a href="#">CTS 2 Profile</a>
	<a href="#">Package Change Request</a>	The ability to group a set of change requests together to be submitted to the Terminology Provider as a collection of related changes to the terminology.	
	<a href="#">Minimal CTS 2 Profile</a>	The Minimal CTS 2 Profile specifies the minimal functional coverage necessary for a service to declare itself as being a conformant CTS 2 service. The Minimal CTS 2 Profile includes the ability to search the contents of code systems and value sets.	

1260

1261 **Terminology Administration Profile**

1262 The Terminology Administration profile is intended to provide the functional operations  
 1263 necessary for terminology administrators to be able to access and make available terminology  
 1264 content obtained from a Terminology Provider. Terminology Administrators are required to  
 1265 interface with Terminology Provider systems in order to obtain the terminology content, then  
 1266 load that terminology content on local Terminology Servers.

1267

Profile	Member Operations	Operation Profile	Notes
<b>Terminology Administration Profile</b>	<a href="#">Import Content</a>	Terminology content would be loaded into the terminology server either as an entire terminology load, or the loading of a delta or set of changes from the previous version of the terminology.	The Terminology Administration profile utilizes the all of the operations defined in the Administrative Scenario section, as well as the functionality outlined in the <a href="#">Minimal CTS 2 Profile</a> .
	<a href="#">Export Content</a>	Terminology content would be exported either in whole or in part based on filtering	

		against terminology properties. The export format may also be specified.
	<a href="#">Remove Content</a>	A specified terminology component (code system, mapping, value set, etc.) would be removed from the terminology server.
	<a href="#">Change Content Status</a>	Terminology content status would be changed, thus changing its availability for access by other terminology service functions.
	<a href="#">Update Notification</a>	An electronic notification would be sent to subscribe users in the event of a change to the specified terminology element.
	<a href="#">Update Notification Management</a>	Subscription notification information can be updated for a subscriber's notification account.
	<a href="#">Content Dependency Notification</a>	A dependency check would be run to determine if there are any changes between a currently used code system element, and a proposed change to that code system element.
	<a href="#">Minimal CTS 2 Profile</a>	The Minimal CTS 2 Profile specifies the minimal functional coverage necessary for a service to declare itself as being a conformant CTS 2 service. The Minimal CTS 2 Profile includes the ability to search the contents of code systems and value sets.
	<a href="#">Import Concept Associations</a>	Additional concept associations would be imported into the terminology

		server and made available for use in creating associations and mapping between concepts.	
	<a href="#">Export Concept Associations</a>	Concept associations would be exported either in whole or in part based on filtering against terminology properties. The export format may also be specified.	
	<a href="#">Change Status of Concept Association</a>	The status of a concept association would be modified changing their availability for access by other terminology service functions.	
	<a href="#">Register for Association Update Notification</a>	Users could register to receive notification that an element of an association has changed and thus may require review.	

1268 **Terminology Authoring Profile**

1269 Terminology authors require the capability to robustly query and access terminology content, as  
 1270 well as directly modify the terminology content. The Terminology Authoring profile is intended  
 1271 to provide the functional operations necessary for terminology authors to analyze the existing  
 1272 terminology content, as well as directly edit terminology content.

1273

Profile	Member Operations	Operation Profile	Notes
<b>Terminology Authoring Profile</b>	<a href="#">Create Code System</a>	The ability to create a new Code System to contain a set of new coded concepts. The Code System is created by defining the set of meta-data properties that describe it.	The Terminology Authoring Profile is intended to provide the capability to robustly query and access terminology content, as well as directly modify the terminology content. This includes the ability to modify code system content, value set content, as well as the metadata pertaining to each. This profile
	<a href="#">Maintain Code System</a>	The ability to maintain the content and metadata of a	



	code system.	includes the functions necessary to administer and search terminology content as outlined in the <a href="#">Minimal CTS 2 Profile</a> as well as the <a href="#">Terminology Administration Profile</a>
<a href="#">Create Concept</a>	The ability to define and add a new concept to a code system.	
<a href="#">Maintain Concept</a>	The ability to maintain a concept that exists in a code system.	
<a href="#">Deprecate Concept</a>	The ability to deprecate or retire a terminology element from a code system.	
<a href="#">Create Value Set By Intension</a>	The ability to create a dynamic value set that is defined by a computable expression that can be resolved to an exact list of coded concepts at any given point in time.	
<a href="#">Create Value Set by Extension</a>	Operation Profile 3	
<a href="#">Maintain Value Set Definition</a>	The ability to redefine a value set by changing the definition of the value set.	
<a href="#">Maintain Value Set Enumeration</a>	The ability to maintain the content of a value set by changing the enumeration of the concepts that make up the value set.	
<a href="#">Minimal CTS 2 Profile</a>	The Minimal CTS 2 Profile specifies the minimal functional coverage necessary for a service to declare itself as being a conformant CTS 2 service. The Minimal CTS 2 Profile includes the ability to search the contents of code systems and value sets.	
<a href="#">Terminology</a>	The Terminology	

<a href="#"><u>Administration Profile</u></a>	Administration profile is intended to provide the functional operations necessary for terminology administrators to be able to access and make available terminology content obtained from a Terminology Provider.	
<a href="#"><u>Create Maintain Association between Concepts</u></a>	The ability to create or maintain (i.e. remove or update) an association between concepts.	
<a href="#"><u>Create Relationship Type</u></a>	The ability to create a new relationship type that may be used to link two concepts.	
<a href="#"><u>Create Lexical Association</u></a>	The ability to instantiate an association between two sets of coded concepts using a set of lexical rules (matching algorithms) to generate the associations .	
<a href="#"><u>Create Rules Based Association</u></a>	The ability to instantiate an association between two sets of coded concepts using a set of description logic or inference rules that either assert or infer mappings between two Code Systems.	

1274

## 1275 CTS 2 Semantic Profiles

1276 Semantic profiles are created to group together vocabularies with similar designs. Vocabularies  
1277 grouped under a single semantic profile can be queried using the same functional variants of  
1278 CTS2 functions. This approach provides the following advantages:

- 1279 • It allows the CTS2 author to focus on a set of design attributes of terminologies and  
1280 support those using functional variants, rather than having to focus on individual  
1281 terminologies while authoring the standard.

- 1282 • It allows the implementer to implement functional variants of CTS2 functions based on  
 1283 the semantic profiles they want to support rather than to create or implement functional  
 1284 variants for the terminologies that are to be supported by their implementation.
- 1285 • It allows terminology authoring organizations to classify their terminology under a  
 1286 semantic profile and insulates them from the complexities of functional variants of CTS2  
 1287 functions.

1288 These intrinsic qualities of terminologies allow the functional profiles to be implemented in  
 1289 accordance with the properties of the classes of these terminologies. The following Semantic  
 1290 Profiles for terminologies are defined currently:

1291 **Mature Terminology Profile**

Profile	Sample Terminology Criteria	Sample Terminologies Classified Under Profile	Notes
<b>Mature Terminology Profile</b>	<ul style="list-style-type: none"> <li>• Unique identifiers for all concepts</li> <li>• Unique identifiers for all designations</li> <li>• Unique identifiers for all relationships</li> <li>• Identifiers are never reused.</li> </ul>	<ul style="list-style-type: none"> <li>• SNOMED CT, all versions</li> <li>• ICD 9 CM</li> <li>• ICD 10 CM</li> <li>• LOINC</li> <li>• RxNorm</li> <li>• MEDCIN</li> <li>• NDF / NDF-RT</li> <li>• CPT</li> </ul>	Terminologies in the Mature Terminology Profile make an attempt to conform to many of terminology best practices that are, for example outlined in <i>Desiderata for Controlled Medical Vocabularies in the Twenty-First Century</i> , James J. Cimino.

1292 **Developing Terminology Profile**

Profile	Sample Terminology Criteria	Sample Terminologies Classified Under Profile	Notes
<b>Developing Terminology</b>	<ul style="list-style-type: none"> <li>• Identifiers that are not globally unique</li> </ul>	<ul style="list-style-type: none"> <li>• Some HL7 Vocabulary</li> </ul>	Terminologies in the Developing Terminology

<p><b>Profile</b></p>	<p>within the terminology, but are unique within a given domain.</p> <ul style="list-style-type: none"> <li>The concepts can be uniquely identified by combining the concept identifier and the domain identifier.</li> </ul>	<p>tables</p> <ul style="list-style-type: none"> <li>Locally developed terminology sources or code sets</li> </ul>	<p>Profile are either developed using adhoc techniques, or have degraded over time.</p>
-----------------------	---	--	---

1293 **CTS 2 Conformance Profiles**

1294 **Conformance Interoperability**

1295 The capabilities defined within the CTS 2 service functional model have been attributed to  
 1296 different functional profiles. The purpose of functional profiles is to group together functions to  
 1297 form cohesive levels of operational capability against which implementations can be tested for  
 1298 conformance. Thus, interoperability between CTS 2 implementations is assured within a  
 1299 specified conformance profile. In other words, two CTS 2 implementations that conform to the  
 1300 Terminology Authoring profile will be able to interoperate using the functions described in that  
 1301 profile.

1302 These profiles serve to educate the purchasing and implementation communities, allowing for  
 1303 implementation variation while still promoting interoperability. Service Level Agreements made  
 1304 between organizations are then testable because they are informed by these profiles. Governance  
 1305 of these agreements is less ambiguous and more enforceable due to precise functional levels of  
 1306 interoperability that may be expected.

1307 Implementation of this functional specification should explicitly deal with the different  
 1308 interoperability roles that CTS 2 may fill using these conformance profiles. The business rules  
 1309 enforced by an organization’s purchasing, implementation, and governance arms should be  
 1310 discussed, and the ways in which CTS 2 facilitates that enforcement should be made clear.

1311 **Conformance Assertion**

1312 Implementations of CTS 2 conform to a specified conformance profile, which is a combination  
 1313 of a functional and semantic profile. That is, conformance to a specific profile is asserted to  
 1314 against the quality metric of a specified semantic profile in association with the specified  
 1315 functional profile.

1316 There are currently four different functional profiles defined. Each profile can be implemented  
 1317 according to either the **Mature Terminology** or **Developing Terminology** semantic profiles,  
 1318 providing up to eight possible levels of conformance to CTS 2.

1319

	<b>Mature Terminology Semantic Profile</b>	<b>Developing Terminology Semantic Profile</b>
<b>Minimal CTS 2 Functional Profile</b>	Minimal CTS 2 - Mature Terminology Conformance Profile	Minimal CTS 2 - Developing Terminology Conformance Profile
<b>Vocabulary Facilitator Functional Profile</b>	Vocabulary Facilitator - Mature Terminology Conformance Profile	Vocabulary Facilitator - Developing Terminology Conformance Profile
<b>Terminology Administration Functional Profile</b>	Terminology Administration - Mature Terminology Conformance Profile	Terminology Administration - Developing Terminology Conformance Profile
<b>Terminology Authoring Functional Profile</b>	Terminology Authoring - Mature Terminology Conformance Profile	Terminology Authoring - Developing Terminology Conformance Profile

1320

## The Services Framework Functional Model

1321

The Services Framework Functional Model identifies common underlying enterprise infrastructure such as naming, directory, security, etc. that may be assumed and referenced by this Functional Model.

1322

1323

1324

Note that the Services Framework Functional Model is being developed in parallel with other service Functional Models; candidate functionality for the Framework should be submitted to the Infrastructure subgroup for evaluation.

1325

1326

1327

CTS 2 compliant service instances are intended to be healthcare middleware services and to work within the context of supporting infrastructure services that may exist within an enterprise. As a result, a number of underpinning capabilities have been intentionally omitted from the scope of this specification. These include (but are not limited to) capabilities such as identity management, security and record location services.

1328

1329

1330

1331

1332

The CTS 2 specification, by design, can be used as a means to integrate a new capability into a service-oriented architecture, or can be used to provide a service interface to access content in legacy applications. It is not intended as a replacement of any single system, but instead to act as a companion component that facilitates interoperability with data sharing partners through a standardized set of APIs.

1333

1334

1335

1336

1337

CTS 2 serves as a simplifying resource for the organization, as it provide a single point of access for all terminology resources.

1338

## 1339 **Relationship to Information Content**

1340 The following principles shall be followed for specifying the information model to be used by  
1341 the services being specified in this Service Functional Model:

- 1342 1. SFMs shall provide a conformance profile supporting HL7 content where relevant
- 1343 2. We shall not preclude the use of non-HL7 content
- 1344 3. SFMs will reuse to the maximum extent possible the content models as defined in other  
1345 standards (for example, HL7 RMIMs)
- 1346 4. Information content representations shall be represented in platform-agnostic formalisms  
1347 (e.g., UML)
- 1348 5. SFMs may identify content at varying levels of granularity, depending upon the functions  
1349 being specified. (For example, the Common Terminology Service will deal with different  
1350 granularity of information than the Resource Location and Update Service).
- 1351 6. Conformance Profiles may be balloted or adopted after the release of the initial SFM to  
1352 address specialized business needs. (realm-specific profiles, domain-specific profiles,  
1353 etc.)
- 1354 7. Details about semantics specific to this SFM appear in other sections of this document

## 1355 **Recommendations for Technical RFP** 1356 **Issuance**

1357 This section includes Identification of topics requiring elaboration in candidate solutions  
1358 provided through the OMG RFP process. These may be service-specific, deployment related, or  
1359 non-functional.

1360

## 1361 **Semantic Signifiers: Disparate Terminologies**

1362 While defining the semantics of payloads sent through CTS 2 is beyond the scope of this  
1363 publication, the ability of CTS 2 to notify a service partner about the nature of the capabilities of  
1364 that implementation of CTS 2 is essential to fulfilling terminology service interoperability.

1365 CTS 2 could conceivably be used to access and maintain a great variety of terminology sources,  
1366 including SNOMED, ICD, and RxNorm (to name a few). To create true terminological  
1367 interoperability between organizations it is essential to provide a scalable and extensible  
1368 terminology model that can be included in the description of and access to the terminology  
1369 resources available on any given terminology service.

1370 Though a limited number of semantic signifiers have been included in this document as a  
1371 mechanism of defining the necessary behaviors of a terminology, it is expected that HL7, HL7

1372 member organizations, terminology providers, and terminology users will be producing  
1373 representations that will be supported within a given CTS 2 implementation.

1374

### 1375 **Semantic Signifiers: HL7 Terminologies**

1376 Where terminology content exposed through CTS 2 is from an HL7 domain it is necessary to  
1377 include support for Concept Domains, Binding Realms and domain bindings.

1378 RFP submitters should take the requirement for Domain and Binding Realm descriptions as a  
1379 starting point to discuss the additional physical information descriptions. The usage of the two  
1380 should be described and modeled so as to paint a complete picture of the issue of semantic  
1381 description and discovery through the CTS 2 interface.

1382 Additionally, Semantic Signifiers should allow for the use of some sort of logical operators in  
1383 describing their hierarchy or aggregation. For example, Boolean Operators (AND, OR, NOT)  
1384 should be available in creating query parameters.

1385 This should be discussed in detail by RFP Submitters.

1386

### 1387 **Conformance Profiles and Service Level Agreements**

1388 The capabilities defined within the CTS 2 SFM have been attributed to specific conformance  
1389 profiles. The purpose of a conformance profile is to group together functions to form cohesive  
1390 levels of operational capability against which implementations can be tested for conformance.  
1391 Thus, interoperability between CTS 2 implementations is assured within a conformance profile.  
1392 In other words, two CTS 2 implementations that conform to the Authoring profile will be able to  
1393 interoperate using the functions outlined in that profile.

1394 These profiles serve to educate the purchasing and implementation communities, allowing for  
1395 implementation variation while still promoting interoperability. Service Level Agreements made  
1396 between organizations are then testable because they are informed by these profiles. Governance  
1397 of these agreements is less ambiguous and more enforceable due to precise functional levels of  
1398 interoperability that may be expected.

1399 Implementation of this functional specification should explicitly deal with the different  
1400 interoperability roles that CTS 2 may fill using these conformance profiles. The business rules  
1401 enforced by an organization's purchasing, implementation, and governance arms should be  
1402 discussed, and the ways in which CTS 2 facilitates that enforcement should be made clear.

1403

## 1404 **Operationalizing CTS 2: Considerations in Implementation**

### 1405 **Optimization**

1406 Structured terminologies can be quite large in nature in both the number of concepts,  
1407 designations, associations, and other attributes that further describe terminology content. As  
1408 such, efficiently accessing and querying terminology content is critical.

1409 Responders to the RFP should discuss optimization strategies for accessing and updating specific  
1410 terminologies.

1411

### 1412 **Internationalization**

1413 Responders to the RFP will discuss what effect, if any, localization and internationalization of  
1414 terminologies will have on technical implementations of CTS 2?

1415

## 1416 **Service Description and Discovery**

1417 Because CTS 2 exists as a service between organizations, CTS 2 should be considered a perfect  
1418 candidate to benefit from service description and discovery, such as what terminologies are  
1419 available on any given CTS 2 implementation, and the specific profiles implemented by that  
1420 service implementation.

1421 Responders to the RFP should explicitly discuss this deployment case, how to better describe  
1422 CTS 2 to improve service discovery.

1423

## 1424 **Federated Terminologies**

1425 As implementers strive to organize CTS 2 within and between institutions, it is likely that a  
1426 federation of terminology sources and terminology servers will develop. These service interfaces  
1427 will occupy various information and domain levels within and between organizations. Common  
1428 federation patterns are likely to emerge, such as a mesh or a hierarchical structure. However,  
1429 other deployment scenarios are desirable as well. Special attention should be paid to  
1430 implementation in a non-homogeneous environments.

1431 Responders to the RFP discuss how the implementation would support federated terminologies,  
1432 and how it would allow for a hierarchical service topology to satisfy most deployment  
1433 requirements.



## 1434 **Terminology Structure Considerations**

1435 [Section 2.4.2](#) of this document outlines a minimal terminology model and attributes for  
1436 terminologies entities. This model represents the minimal classes, attributes and associations  
1437 necessary to represent conceptual terminologies.

1438 Responders to the RFP should provide a detailed implementation model that can represent  
1439 terminology sources that adhere to terminology best practices, and discuss a strategy for  
1440 representing less mature terminologies in a format that allows them to be consistently accessed  
1441 by the appropriate CTS 2 functions in accordance with the required semantic profiles.

1442

## 1443 **Appendix A - Relevant Standards**

### 1444 **[HL7 Common Terminology Services](#)**

1445 The Common Terminology Services (CTS) specification was developed as an alternative to a  
1446 common data structure. The HL7 Common Terminology Services (HL7 CTS) is an Application  
1447 Programming Interface (API) specification that is intended to describe the basic functionality  
1448 that will be needed by HL7 Version 3 software implementations to query and access  
1449 terminological content. It is specified as an API rather than a set of data structures to enable a  
1450 wide variety of terminological content to be integrated within the HL7 Version 3 messaging  
1451 framework without the need for significant migration or rewrite. Instead of specifying what an  
1452 external terminology must look like, HL7 has chosen to identify the common functional  
1453 characteristics that an external terminology must be able to provide. As an example, an HL7  
1454 compliant terminology service will need to be able to determine whether a given concept code is  
1455 valid within the particular resource. Instead of describing a table keyed by the resource identifier  
1456 and concept code, the CTS specification describes an Application Programming Interface (API)  
1457 call that takes a resource identifier and concept code as input and returns a true/false value. Each  
1458 terminology developer is free to implement this API call in whatever way is most appropriate for  
1459 them. There are two layers between HL7 Version 3 message processing applications and the  
1460 target vocabularies. The upper layer, the Message API communicates with in terms of  
1461 vocabulary domains, realms, coded attributes and other artifacts of the RIM and HL7 messaging  
1462 model. The lower layer, the Vocabulary API communicates in terms of coding system, concept  
1463 codes, designations, and other vocabulary related entities. **NOTE:** THE CTS II specification is  
1464 an extension of the original HL7 Common Terminology Services approved standard

### 1465 **[The Lexical Grid](#)**

1466 The Lexical Grid is a proposal for standard storage of terminologies and ontologies. The  
1467 LexGrid Model defines how terminologies should be formatted and represented  
1468 programmatically. It also defines several different server storage mechanisms and a XML  
1469 format.

## 1470 **Appendix B – Glossary**

### 1471 **Actor**

1472 In the Unified Modeling Language (UML), an actor is something or someone who supplies a  
1473 stimulus to the system. An actor cannot be controlled by the system and is defined as being  
1474 outside the system. An actor is often thought of as a role, rather than an actual person. A single  
1475 person in the real world can be represented by several actors if they have several different roles  
1476 and goals in regards to a system. Source: [http://en.wikipedia.org/wiki/Actor \(UML\)](http://en.wikipedia.org/wiki/Actor_(UML))

1477

### 1478 **Code System**

1479 A *Code System* is defined as a collection of codes with associated designations, meanings and  
1480 associations. The persistent representation of a Code Systems include meta-data about the code  
1481 system itself, as well as the contents of the Code System.

1482 Examples of *Code Systems* include ICD-9 CM, SNOMED CT, LOINC, and CPT. To meet the  
1483 requirements of a *Code System* as defined by HL7, a given *Concept Code* must resolve to one  
1484 and only one meaning within the *Code System*. Given this definition, each table in the HL7  
1485 Version 2 standard represents a different *Code System* since *Concept Codes* are sometimes used  
1486 in different tables to have different meanings. For example, the *Concept Code* “M” in the gender  
1487 *Code System* means “Male”, while “M” in the marital status *Code System* means “Married”

1488

### 1489 **Concept Map**

1490 A concept map is an association between concepts in different code systems, value sets, or a  
1491 combination of these. The endpoints of a concept map are source and targets, implying a  
1492 direction of the relationships from a source to a target, which can have bearing on the meaning  
1493 and appropriate uses of a map. A concept map supports the use of data from disparate systems by  
1494 providing data linkage and information about how the meaning of the concepts from the  
1495 respective systems relate to one another.

### 1496 **Concept Relationship**

1497 A concept relationship is an association between two or more concepts within a single code  
1498 system. The endpoints of a concept relationship are source and target concepts, implying a  
1499 direction of the relationship from a source to a target, which has bearing on the meaning of the  
1500 relationship and the concepts it connects. A concept relationship is definitional, in that the  
1501 relationship gives meaning to the concepts associated. For example, a relationship between a

1502 parent and child concept indicates that the child concept is a refinement or an example of the  
1503 parent concept in a concept hierarchy. A concept relationship can also define other  
1504 characteristics of a concept, as in relationships between concepts in different parent-child  
1505 hierarchies where the child may have a different set of relationship than that one or more of its  
1506 hierarchical parents.

1507

1508

1509

## 1510 **Nested Value Sets**

1511 When a Value Set Entry references another Value Set, the child value set is referred to as a  
1512 *Nested Value Set*. There is no preset limit to the level of nesting allowed within value sets. Value  
1513 sets cannot contain themselves, or any of their ancestors (i.e. they cannot be defined recursively).

1514 Intensional Value Sets can be defined by either fixing the Value Set definition to a specific  
1515 version of the Code System (when the Code System supports versioning), or by decoupling the  
1516 Value Set definition from the version of the code system. This seemingly subtle variation can  
1517 have very significant impact on the final list of concepts which the Value Set ultimately resolves  
1518 to.

1519 When the Value Set definition is tied to the version of the Code System, the value set content  
1520 will remain fixed when instantiated. When the Value Set definition is independent of Code  
1521 System version, the content of the Value Set can vary as the Value Set is resolved against  
1522 different versions of the Code System.

1523

## 1524 **Value Set**

1525 A *Value Set* represents a uniquely identifiable set of valid concept representations, where any  
1526 concept representation can be tested to determine whether or not it is a member of the value set.

1527 Value set complexity may range from a simple flat list of concept codes drawn from a single  
1528 code system, to an unbounded hierarchical set of possibly post-coordinated expressions drawn  
1529 from multiple code systems.

1530 Value sets exist to constrain the content for a coded element in an HL7 static model or data type  
1531 property. Value sets cannot have null content, and must contain at least one concept  
1532 representation where any given concept is generally (but not required to be) represented by only  
1533 a single code within the Value Set.

1534 Sub-value Sets

1535 A *sub-value set* is a sub-set of a parent *Value Set*.

1536 **Value Set Specification**

1537 Value sets can be specified in two ways, either by enumeration (*extension*), or definition  
1538 (*intention*). **Extensional Value Set Representation (Enumeration)**

1539 From ISO (<http://www.tc215wg3.nhs.uk/pages/pdf/vote0204.pdf>), an extensional definition is a  
1540 description of a concept by enumerating all of its subordinate concepts under one criterion of  
1541 subdivision.

1542 Value sets defined by extension are comprised of an explicitly enumerated set of codes. The  
1543 simplest case is when the value set consists of only one code.

1544

Code Value	Description
M	Male
F	Female
U	Unspecified

1545 More complex variations might relate to hierarchical coding systems such as the following  
1546 fictitious example, where “Level” represents the nesting level for a particular Code Value:

1547

Code Value	Level	Description
1123123	1	Education
1343434	2	Diabetic Education
1445455	2	Stroke Education
2135534	1	Counseling
2344566	2	Emotional
3456663	2	Daily Living

1548 **Intensional Value Set Definition (Definition)**

1549 From ISO (<http://www.tc215wg3.nhs.uk/pages/pdf/vote0204.pdf>), an intensional definition  
1550 describes the intension of a concept by stating the superordinate concept and the delimiting  
1551 characteristics.

1552 Value sets defined by intension are value sets that are defined by a computable expression that  
1553 can be resolved to an exact list of codes.

1554 For example, an intensional value set definition might be defined as, “SNOMED CT concepts  
1555 that are children of the SNOMED CT concept “Diabetes Mellitus.”

1556 Some common strategies used to define intensional values sets include:

- 1557 • Reference a head concept and its subordinate concepts in a hierarchy.
- 1558 • Reference only the concepts subordinate to a head code (and not the head code itself).
- 1559 • Create arbitrarily complex unions, intersections, and exclusions of the two previously  
1560 described types of value sets.
- 1561 • Other mechanisms, including statements created using a rich expression language.

## 1562 **Nested Value Sets**

1563 When a Value Set Entry references another Value Set, the child value set is referred to as a  
1564 *Nested Value Set*. There is no preset limit to the level of nesting allowed within value sets. Value  
1565 sets cannot contain themselves, or any of their ancestors (i.e. they cannot be defined recursively).

1566 Intensional Value Sets can be defined by either fixing the Value Set definition to a specific  
1567 version of the Code System (when the Code System supports versioning), or by decoupling the  
1568 Value Set definition from the version of the code system. This seemingly subtle variation can  
1569 have very significant impact on the final list of concepts which the Value Set ultimately resolves  
1570 to.

1571 When the Value Set definition is tied to the version of the Code System, the value set content  
1572 will remain fixed when instantiated. When the Value Set definition is independent of Code  
1573 System version, the content of the Value Set can vary as the Value Set is resolved against  
1574 different versions of the Code System.

1575

1576

## 1577 **Appendix C - HL7 EHR Functional Model** 1578 **Traceability**

1579 *This section lists the EHR Functions that are related to this service.*

1580 *Note that in general there will not be a direct correspondence between EHR Functions and*  
1581 *HSSP Services, since Services are specified from a different system viewpoint. The mapping*  
1582 *provided here enables the HSSP Services to be understood in the context of the **EHR-S***  
1583 ***Functional Model DSTU**. The table below references Version \_\_\_\_\_ of the EHR Functional*  
1584 *Model.*

1585

			Notes
EHR Function ID	EHR Function Name	EHR Function Statement	<i>For every row, explain the rationale for including in this specification.</i>

1586