HHL7IG\_CDS\_KNART\_R1\_D1\_2013JAN



HL7 Implementation Guide: Clinical Decision Support Knowledge Artifact Implementation Guide, Release 1

January 2013

**HL7 DSTU Ballot**

**Sponsored by:**

**Clinical Decision Support in collaboration with the Health and Human Services Standards and Interoperability Framework Health eDecisions Working Group**

Copyright © 2013 Health Level Seven International ® ALL RIGHTS RESERVED. The reproduction of this material in any form is strictly forbidden without the written permission of the publisher. HL7 and Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. Pat & TM Off**.**

Use of this material is governed by HL7's [**IP Compliance Policy**](http://www.hl7.org/legal/ippolicy.cfm?ref=nav).

Acknowledgments

The authors wish to recognize the S&I Framework Health eDecisions Initiative Workgroup and the HL7 CDS Workgroup for their contributions to this document.

Copyrights

This material includes SNOMED Clinical Terms ® (SNOMED CT®) which is used by permission of the International Health Terminology Standards Development Organization (IHTSDO). All rights reserved. SNOMED CT was originally created by The College of American Pathologists. "SNOMED ®" and "SNOMED CT ®" are registered trademarks of the IHTSDO.

This material contains content from LOINC® (http://loinc.org). The LOINC table, LOINC codes, and LOINC panels and forms file are copyright (c) 1995-2011, Regenstrief Institute, Inc. and the Logical Observation Identifiers Names and Codes (LOINC) Committee and available at no cost under the license at <http://loinc.org/terms-of-use>.

This material includes content used from the Clinical Decision Support Consortium (CDSC), specifically the CDSC L3 schema.

This material includes content used from Allscripts and the CREF schema, which is used with permission of Allscripts.

This material includes content from HL7 Arden Syntax, which is used with permission of HL7.

This material includes content used from the Guidelines Element Model (GEM) which is used by permission of Yale University.

This material includes content from the Agency for Healthcare Research and Quality (AHRQ) and its eRecs project.

Executive Summary

In support of the United States’ national objectives for healthcare reform, the Office of the National Coordinator for Health Information Technology (ONC) Standards and Interoperability (S&I) Framework has sponsored the development of harmonized interoperability specifications. These specifications are designed to support national health initiatives and healthcare priorities, including Meaningful Use, the Nationwide Health Information Network, and the ongoing mission to improve population health.

The nation is reaching a critical mass of Electronic Health Record systems (EHRs) that comply with data and vocabulary standards. Providers seeking to meaningfully use EHRs face a variety of challenging tasks. Those tasks include assessing needs, selecting and negotiating with a system vendor or reseller, implementing project management, and instituting workflow changes to improve clinical performance, control costs, and ultimately, improve outcomes. Additionally, many providers face the challenge of integration and interoperation with disparate systems. Many institutions use their own proprietary vocabularies and data models. Though it may offer some internal flexibility, it comes with a high, often hidden, long term maintenance cost.

In support of this wide deployment of EHRs, there is an opportunity to implement a health learning system that includes clinical decision support (CDS) and provides a broad range of benefits that can contribute towards improved health of individuals and the population as a whole (refer to “Digital Infrastructure for the Learning Health System: The Foundation for Continuous Improvement in Health and Health Care: Workshop Series Summary” listed in Appendix A – Referenced Documents).

The S&I Framework Health eDecisions Initiative (HeD) is developing a foundational specification, reusing much of the work currently done in CDS standardization, to enable the structuring and encoding of CDS content for use as “knowledge artifacts.” These artifacts can be used in support of many areas of the healthcare system, including quality and utilization measures, disease outbreaks, comparative effectiveness analysis, efficacy of drug treatments, and monitoring health trends. One of the key benefits of this proposed approach is the definition of a ‘lingua franca’ for the exchange of CDS knowledge and artifacts. Rather than having an un-scalable network of point-to-point communication channels, each with its own set of transformations, different organizations will only need to transform their content to an HeD-compatible format to communicate effectively with any other point in the network of providers that comprises today’s health care system. If the models and vocabularies are rich enough, in the future, some CDS vendors may opt to use HeD as an internal specification.

This implementation guide is developed in support of the HeD Artifact Sharing Use Case and is intended to assist implementers in the development of Clinical Decision Support (CDS) Knowledge Artifacts. The approach adopted in this implementation guide is designed to be flexible and reusable, and to provide a baseline for CDS vendors and CDS Knowledge Artifact implementers.

Table of Contents

[1 Introduction 8](#_Toc354654543)

[1.1 Design Approach and Rationale 8](#_Toc354654544)

[1.1.1 How a CDS Knowledge Artifact Works 10](#_Toc354654545)

[1.1.2 Lifecycle of a CDS knowledge artifact 10](#_Toc354654546)

[1.1.3 Extensibility of the HeD CDS Knowledge Artifact schema 11](#_Toc354654547)

[1.2 Audience 11](#_Toc354654548)

[1.2.1 Requisite Knowledge 11](#_Toc354654549)

[1.3 Scope of the Guide 12](#_Toc354654550)

[1.3.1 Contents of the Proposed Ballot 13](#_Toc354654551)

[1.4 Alignment to HeD Artifact Sharing Use Case 15](#_Toc354654552)

[1.4.1 Use Case Assumptions and Conditions 15](#_Toc354654553)

[1.4.2 Usage Conformance Testing Recommendations 15](#_Toc354654554)

[1.5 Organization of this Guide 16](#_Toc354654555)

[1.5.1 Definitions of Terms 16](#_Toc354654556)

[1.5.2 Definitions of Actors 18](#_Toc354654557)

[1.5.3 Conventions Used 19](#_Toc354654558)

[1.5.4 Schema Roadmap 23](#_Toc354654559)

[2 CDS Knowledge Artifact Types 24](#_Toc354654560)

[2.1 Conformance to the Health eDecisions Knowledge Artifact Specification 24](#_Toc354654561)

[2.2 HeD Knowledge Artifact 24](#_Toc354654562)

[2.2.1 Metadata Elements 24](#_Toc354654563)

[2.2.2 Def Elements 25](#_Toc354654564)

[2.2.3 Expression Elements 25](#_Toc354654565)

[2.2.4 Trigger Elements 25](#_Toc354654566)

[2.2.5 Condition Elements 25](#_Toc354654567)

[2.2.6 Behavior Elements 25](#_Toc354654568)

[2.2.7 Action Elements 25](#_Toc354654569)

[2.3 Event Condition Action (ECA) Rules 26](#_Toc354654570)

[2.3.1 Conceptual Overview 26](#_Toc354654571)

[2.3.2 ECA Rule CONFORMANCE REQUIREMENTS 27](#_Toc354654572)

[2.4 Order Sets 29](#_Toc354654573)

[2.4.1 Conceptual Overview 29](#_Toc354654574)

[2.4.2 Order Set Conformance Requirements 31](#_Toc354654575)

[2.5 Documentation Templates 33](#_Toc354654576)

[2.5.1 Conceptual Overview 33](#_Toc354654577)

[2.5.2 Documentation Template Conformance Requirements 35](#_Toc354654578)

[2.6 Value Sets and Vocabulary Codes 37](#_Toc354654579)

[2.6.1 Value Set Use Case 37](#_Toc354654580)

[2.6.2 Coded Data versus Free Text 37](#_Toc354654581)

[3 CDS Knowledge Artifact Components 38](#_Toc354654582)

[3.1 Metadata 38](#_Toc354654583)

[3.1.1 Clinical Data Models 38](#_Toc354654584)

[3.2 Actions 39](#_Toc354654585)

[3.2.1 Types of Actions 39](#_Toc354654586)

[3.2.2 Creating Action Sentence Expressions 41](#_Toc354654587)

[3.2.3 Organizing Actions 41](#_Toc354654588)

[3.2.4 Conditional Actions 42](#_Toc354654589)

[3.2.5 Behaviors 43](#_Toc354654590)

[3.3 Expressions 47](#_Toc354654591)

[3.3.1 Basic Values (Scalars) 47](#_Toc354654592)

[3.3.2 Comparison Operators 48](#_Toc354654593)

[3.3.3 Logical Operators 49](#_Toc354654594)

[3.3.4 Nullological Operators 49](#_Toc354654595)

[3.3.5 Conditional Operators 50](#_Toc354654596)

[3.3.6 Arithmetic Operators 51](#_Toc354654597)

[3.3.7 String Operators 52](#_Toc354654598)

[3.3.8 Date and Time Operators 53](#_Toc354654599)

[3.3.9 List Values 54](#_Toc354654600)

[3.3.10 Aggregate Operators 55](#_Toc354654601)

[3.3.11 Interval Values 56](#_Toc354654602)

[3.3.12 Structured Values 57](#_Toc354654603)

[3.3.13 Reusing Expressions 59](#_Toc354654604)

[3.3.14 External Data 59](#_Toc354654605)

[3.3.15 Parameters 62](#_Toc354654606)

[4 cds knowledge artifact components 64](#_Toc354654607)

[4.1 Knowledge Artifact Metadata 64](#_Toc354654608)

[4.2 Actions 65](#_Toc354654609)

[4.3 Conditions 68](#_Toc354654610)

[4.4 Supporting Evidence 68](#_Toc354654611)

[4.5 Supporting Reference 68](#_Toc354654612)

[4.6 Clinical Data Mapping 68](#_Toc354654613)

[4.7 Expression 68](#_Toc354654614)

[5 hed enumerations and vocabularies 69](#_Toc354654615)

[5.1 Datatypes.xsd Namespace Summary 69](#_Toc354654616)

[5.2 Datatypes.xsd Complex Type Summary 69](#_Toc354654617)

[5.3 Datatypes.xsd Simple Type Summary 70](#_Toc354654618)

[5.4 Datatypes.xsd Attribute Group Summary 71](#_Toc354654619)

[5.5 Datatypes.xsd Complex Data Types 71](#_Toc354654620)

[5.6 Datatypes.xsd Attribute Types 129](#_Toc354654621)

[5.7 KnowledgeDocument.xsd Namespace Summary 139](#_Toc354654622)

[5.8 KnowledgeDocument.xsd Global Element Summary 139](#_Toc354654623)

[5.9 KnowledgeDocument.xsd Complex Type Summary 140](#_Toc354654624)

[5.10 KnowledgeDocument.xsd SimpleType Summary 146](#_Toc354654625)

[5.11 KnowledgeDocument.xsd Elements 148](#_Toc354654626)

[5.12 KnowledgeDocument.xsd Complex Types 150](#_Toc354654627)

[5.13 KnowledgeDocument.xsd Simple Data Types 393](#_Toc354654628)

[5.14 Schema Datatypes.xsd 445](#_Toc354654629)

[5.15 Schema KnowledgeDocument.xsd 448](#_Toc354654630)

[5.16 Namespace Bindings 449](#_Toc354654631)

[6 Appendix A – Referenced documents 450](#_Toc354654632)

[7 appendix b - acronyms 453](#_Toc354654633)

[8 Appendix C – Definitions 454](#_Toc354654634)

[9 Appendix D – Language Implementation 455](#_Toc354654635)

[9.1 Clinical Data Retrieval in HeDS Artifacts 455](#_Toc354654636)

[9.1.1 Defining Clinical Data 455](#_Toc354654637)

[9.1.2 Conformance Levels 456](#_Toc354654638)

[9.1.3 Artifact Data Requirements 456](#_Toc354654639)

[9.2 Expression Language Conceptual Model 458](#_Toc354654640)

[9.2.1 Data Model 458](#_Toc354654641)

[9.2.2 Language Elements 459](#_Toc354654642)

[9.2.3 Semantic Validation 460](#_Toc354654643)

[9.2.4 Execution Model 460](#_Toc354654644)

[10 Appendix E – Examples 463](#_Toc354654645)

[10.1 FLACC Example 463](#_Toc354654646)

[10.2 RespiratoryOrder Example 471](#_Toc354654647)

[10.3 DopamineComplexIVOrderWithComplexLiteral Example 478](#_Toc354654648)

[10.4 HeartFailureAdmissionToMedSurgOrderSet Example 484](#_Toc354654649)

# Introduction

The S&I Framework is an approach adopted by ONC's Office of Standards & Interoperability to fulfill its charge of enabling harmonized interoperability specifications to support national health outcomes and healthcare priorities. The S&I Framework is a collaborative community of participants from the public and private sectors who are focused on providing the tools, services and guidance to facilitate the functional exchange of health information. More information about the S&I Framework can be found here: <http://siframework.org/>

The S&I Framework uses a set of integrated functions, processes, and tools that enable execution of specific value-creating initiatives. Each S&I Initiative focuses on a single, narrowly-scoped, broadly-applicable challenge. The goal of the Health eDecisions (HeD) Initiative is to identify, define and harmonize standards and specifications that facilitate the emergence of systems and services whereby shareable clinical decision support (CDS) can be implemented. Additional information about the HeD initiative and CDS, including a Project Charter, can be found here: <http://wiki.siframework.org/Health+eDecisions+Project+Charter+and+Members>

The HeD Initiative, based on stakeholder input and subject matter expert (SME) guidance, developed the HeD Artifact Sharing Use Case (HeD Use Case 1) to define the functional requirements for building a schema for the contents of three specific CDS Knowledge Artifact types – Event, Condition Action (ECA) Rules, Order Sets, and Documentation Templates. Consensus on Use Case 1 was achieved on Thursday, September 13, 2012, and the consensus approved Use Case can be found here: <http://wiki.siframework.org/Health+eDecisions+Use+Case>

Following consensus on the functional requirements outlined in HeD Use Case 1, the Harmonization phase kicked off. This included development of a Consensus Statement to formalize the direction and technical approach adopted by the initiative, and to serve as a roadmap during the definition of the technical artifacts and supporting documentation. The consensus statement for the HeD Initiative can be found here: <http://wiki.siframework.org/Health+eDecisions+Consensus+Statement>

Based off of the functional requirements outlined in the HeD Artifact Sharing Use Case, and the guidance outlined in the HeD Initiative Consensus Statement, the initiative harmonized a set of existing industry standards and specifications, and developed a CDS Knowledge Artifact schema, which represents each CDS Knowledge Artifact component in a standardized format. The Schema can be found here: <http://code.google.com/p/health-e-decisions/source/browse/#svn%2Fbranches%2Fv1.1%2Fsrc%2Fmain%2Fschema>

This Implementation Guide serves as a companion document for the CDS Knowledge Artifact schema, and includes additional background, contextual information, and detailed documentation and guidance in support schema implementation. In this introductory section, the approach and purpose for the CDS Knowledge Artifact Implementation Guide is defined and discussed. The specific approach to conceptually designing the CDS Knowledge Artifact is outlined, using existing implementation schemas and standards, and the lifecycle and extensibility of the CDS Knowledge Artifact are explored.

## Design Approach and Rationale

The specification defined in this implementation guide is based on a harmonization of existing standards and specifications as the primary mechanism for structuring knowledge artifacts. This guide is laid out to support the following implementation objectives:

1. To provide an overview and list of the standards/specifications upon which the Health eDecisions CDS Knowledge Artifact schema is built and explain how each contributes to the HeD CDS Knowledge Artifact schema.
2. To provide the rationale for why each standard/specification was chosen and to specify how they play together to solve this important problem.
3. To specify what piece of each standard/specification is used in the HeD CDS Knowledge Artifact schema and why.
4. To specify where and how the documentation for each supporting standard/specification may be obtained.
5. To explain the key concepts and components defined in this the guide and how they relate to one another.

The technical approach adopted by drawing on existing standards and specifications is designed to provide a catalog of components for generating knowledge artifacts, with a specific focus on defining the structure of the components and how they fit together. The intent of the CDS Knowledge Artifact specification is to provide a format for sharing. Because it is intended to provide an unambiguous mechanism for sharing, it may also be used as a format for processing and execution. This specification is not intended as a requirement for implementation, but as a vehicle for sharing CDS Knowledge Artifacts.

This implementation guide focuses not only on structure but also on semantics through the use of standard terminologies, value sets and taxonomies such as SNOMED-CT. The specifications highlight terminology bindings where defined and required.

This technical approach, of drawing on existing standards and specifications, is designed to support multiple goals for implementers:

1. **Maintenance**: Any subsequent change to components and supporting standards/specifications used for generating knowledge artifacts will be made in this single-source guide, which will then be republished and made available to the clinical decision support community.
2. **Translation**: This implementation guide is the natural home for transformation and mapping rules to convert HeD CDS Knowledge Artifact components into other relevant formats. Examples of the transformations supported through this guide include translations from the CDS Knowledge Artifact schema to the HL7 Order Sets DSTU and HL7 Arden Syntax. The methods of transformation and specific examples to support the transforms are included in this implementation guide.
3. **Modularity**: Those familiar with the CDS Knowledge Artifact Implementation Guide and its defined components are able to mix and match specific components needed to support the development and/or consumption of CDS knowledge artifacts. This lessens complexity of implementation and integration by supporting customization to the CDS vendor environment.
4. **Compatibility**: Different types of knowledge artifacts share common substructures that are addressed within one harmonized schema which can structurally map to multiple types of CDS knowledge artifacts, so that all EHR and CDS systems operate using the same set of components, constraints, terminologies, and value sets.
5. **Ease of Use:** The structure of this implementation guide is designed in a way that makes the process of implementing and integrating the CDS Knowledge Artifact easier, by providing enough constraint to ensure CDS Knowledge Artifacts are free of technical content errors, and ensure that the artifact blends easily into the user’s workflow.

The approach taken to develop the CDS Knowledge Artifact Implementation Guide has some similarity to existing approaches that have been used to develop balloted implementation guides, such as the approach used to develop a template library within the HL7 Implementation Guide for CDA® Release 2: IHE Health Story Consolidation, DSTU Release 1.1 (US Realm).

The CDS Knowledge Artifact Implementation Guide is an implementation specification structured as a series of layered constraints for each of the components used in the CDS Knowledge Artifact. This implementation guide adopts specific constraints defined in each of the harmonized schemas and specifications (see Section 2.5.3) and then adds constraints through conformance statements that further define and restrict the sequence and cardinality of components and the vocabularies for coded elements. These conformance statements are defined within the implementation guide at a high level to support initial piloting and adoption of the CDS Knowledge Artifact.

### How a CDS Knowledge Artifact Works

The foundation of each component is the CDS Knowledge Artifact schema, which represents each CDS Knowledge Artifact component in a standardized format for generation and consumption. The schema is a harmonized aggregation of multiple existing CDS standards and specifications.

Each of the three types of CDS Knowledge Artifacts detailed in this guide are comprised of reusable sets of data, called “components.” Each of the components defined in this schema represents an individual building block that can be used to assemble a CDS Knowledge Artifact. Each component is further broken down into “types” to enable further granularity when applying constraints. The modular approach is based on the concept of defining specific constraints for each of the components defined within the CDS Knowledge Artifact, to allow for implementation of components regardless of CDS vendor environment. The components are defined in detail in Section 4 of this guide, with Section5. covering detailed documentation of all types leveraged in constructing the schema.

This approach is based on several key technical decisions made by the Health eDecisions Initiative Workgroup, through the S&I Framework process, including the use of a harmonized schema that seeks to define a new standardized format for multiple types of CDS Knowledge Artifacts.

### Lifecycle of a CDS knowledge artifact

The CDS Knowledge Artifact is the structured and encoded format that this implementation guide supports. It is important to understand the lifecycle of a CDS Knowledge Artifact. Various actions may be carried out on a CDS Knowledge Artifact, each one resulting in a state change of the artifact.

The ovals in red show the different actions that may be performed on a CDS Knowledge Artifact – creation of the artifact (**Created**), pre-publication of the artifact (**Pre-Published**), publication of the artifact by a CDS Content Supplier (**Published**), review of the CDS Knowledge Artifact (**Reviewed**), replacement of an existing artifact with another artifact (**Superseded**) and retirement of the artifact from use (**Withdrawn**).

The rectangles in green show the different states that a CDS Knowledge Artifact goes through – creation prior to publication (**Draft**), in testing (**InTest**), active once published (**Active**), and retired (**Inactive**).

It should be noted that these states and actions apply to a particular version of a CDS Knowledge Artifact, and that any change to a CDS Knowledge Artifact version should result in a new version of that artifact.

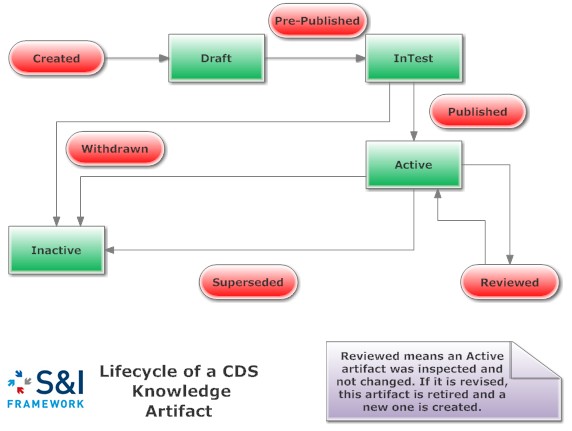


Figure 1 - CDS Knowledge Artifact Lifecycle

### Extensibility of the HeD CDS Knowledge Artifact schema

The CDS Knowledge Artifact types and components are designed to support extensibility by implementers, which allows for CDS vendors to employ flexibility when assembling the artifacts. The CDS Knowledge Artifact schema incorporates a base level of conformance, using conformance language as structured using guidance in Section 2.5.3.4, and then allows for flexibility by implementers in determining the additional attributes and values that may be applied to artifact components.

## Audience

The audience of this implementation guide includes, but is not limited to, software developers, CDS and EHR vendors, and other HIT implementer parties that are interested in developing and consuming CDS knowledge artifacts. This implementation guide also specifically covers consumers and integrators of the CDS Knowledge Artifact.

### Requisite Knowledge

This section includes pre-requisites for implementers and other users of the CDS Knowledge Artifact Implementation Guide. Specific prerequisites for CDS implementations using the CDS Knowledge Artifact Implementation Guide include the following (summarized into required information and optional information):

| Required to Know | Should Know |
| --- | --- |
| 1. Implementers must have a strong knowledge of the S&I Framework HeD Artifact Sharing Use Case and an understanding of Section 11 (Dataset Requirements) 2. Implementers must have a strong understanding in the use of XML, specifically XML Schema. | 1. Implementers should have a strong knowledge of the HL7 standards underlying this implementation guide, including: 2. The HL7 Virtual Medical Record (vMR) standard 3. The HL7 Arden Syntax standard 4. Implementers should reference existing documentation and schemas on the CDSC L3 schema (as noted in Appendix A – References) 5. Implementers should reference existing documentation from the Agency for Healthcare Research and Quality (AHRQ) and their eRecs project. 6. Implementers should reference existing documentation and schemas on the Allscripts CREF schema (as noted in Appendix A – References) 7. Implementers should reference existing documentation and schemas on the Guidelines Element Model (GEM) 8. Implementers should have a basic understanding of the following vocabularies/value sets: 9. CPT 10. CVX 11. NDF-RT 12. LOINC 13. SNOMED-CT |

Table 1 - Implementation Prerequisites - Audience

For those implementers without requisite knowledge in these areas, it is HIGHLY RECOMMENDED to read each of the documents defined in Appendix A – Referenced Documents. Note that the table above categorizes requisite knowledge for concepts covered in this implementation guide in its entirety – this categorization of prerequisites may not apply for those implementations where only a specific CDS Knowledge Artifact and/or specific components are being considered.

## Scope of the Guide

As defined in HeD Use Case 1, the scope of this implementation guide is limited to the CDS Knowledge Artifact and the different components that will be included in the knowledge artifact. This includes how to structure and encode the data elements for each of these components, and the structuring of the data elements into different knowledge artifact types. The three artifact types that are in scope are:

1. Event Condition Action (ECA) Rules
2. Order Sets
3. Documentation Templates

Specifically out of scope is the usage of the knowledge artifact with defined system behaviors, such as how to generate the artifact, how to export and import it, or how to populate a knowledge artifact using data from an existing system, such as an Electronic Health Record (EHR). Each of these system behaviors is specifically excluded from this implementation guide.

As part of the scope, validation of the knowledge artifact is included within scope, for high level conformance with the CDS Knowledge Artifact schema, which is the XML schema based off of the harmonized standards and specifications identified by HeD to satisfy Use Case 1 requirements. Conformance requirements are currently defined at a high level and are limited to the structure and encoding of the knowledge artifact. Additionally, schematron rules will be defined to express conformance constraints.

Wherever possible, specific validation rules are also included for terminologies, such as the use of SNOMED-CT within a CDS Knowledge Artifact.

### Contents of the Proposed Ballot

This table summarizes the materials included in the proposed ballot package for the S&I Framework Health eDecisions initiative CDS Knowledge Artifact submission. This ballot delivery package is intended for review by the HL7 community and other interested parties in the November-January timeframe:

| Filename | Description | Standards Applicability |
| --- | --- | --- |
| CDS Knowledge Artifact Implementation Guide | Implementation Guide | DSTU |
| action.xsd | Contains the action types needed for defining actions (covers Actions as defined in the HeD Artifact Sharing Use Case) | Informative |
| actor.xsd | Contains the Actor type, needed for all CDS Knowledge Artifacts | Informative |
| base.xsd | Contains the base types needed for assembling CDS Knowledge Artifacts | Informative |
| behavior.xsd | Contains the types needed for defining behaviors at the artifact, action, and group level | Informative |
| catalogItem.xsd | Contains the types needed to build Documentation templates | Informative |
| clinicalExpression.xsd | Contains the types associated with data mapping to the external models such as the vMR | Informative |
| condition.xsd | Contains the condition types, needed for ECA rules. | Informative |
| literalExpression.xsd | Contains the types needed to create literals of datatypes within expressions | Informative |
| datatypes.xsd | Supports the base datatypes needed for CDS Knowledge Artifacts   1. Imports ISO 21090 data types | Informative |
| expression.xsd | Contains the types needed for building Expressions (covers Expressions as defined in the HeD Artifact Sharing Use Case) | Informative |
| extdatatypes.xsd | Contains all the extensions to base ISO 21090 data types that are needed to support the HeD schema | Informative |
| knowledgeDocument.xsd | The main container for a CDS Knowledge Artifact | Informative |
| metadata.xsd | Contains the Metadata types for a CDS Knowledge Artifact (covers the Knowledge Artifact metadata as defined in the HeD Artifact Sharing Use Case) | Informative |
| xhtml1-strict.xsd | Per W3C, this schema defines the Second Edition of XHTML 1.0, a reformulation of HTML 4 as an XML 1.0 application, and three Document Type Definitions (DTDs) corresponding to the ones defined by HTML 4. The semantics of the elements and their attributes are defined in the W3C Recommendation for HTML 4. | Informative |
| HeDSchema.eap | Contains the CDS Knowledge Artifact schema and associated clinical data mappings in a UML model in a proprietary format | Informative |
| HeDSchema.xsi | Contains the CDS Knowledge Artifact schema and associated clinical data mappings in a UML model in a standard format | Informative |

Table 2 - Proposed Ballot Materials – CDS knowledge Artifact IG

## Alignment to HeD Artifact Sharing Use Case

The specific requirements implemented within this guide are focused on the structure, format, and encoding of a CDS Knowledge Artifact. These requirements are directly tied to the HeD Artifact Sharing Use Case (HeD Use Case 1) and as noted in Section 2.4.1 of this guide, a thorough understanding of the use case is expected for implementation.

Full material on the HeD Artifact Sharing Use Case can be found here:

<http://sibrowser.siframework.org/siclient/view?type=artifact&id=b3f1c2b0-626e-4c28-91fb-5c79e9d461bc&name=SIFramework_HeD_UC1_CDSArtifactSharing_v1.0.docx>

### Use Case Assumptions and Conditions

It is important for implementers to clearly understand the underlying CDS environmental assumptions defined in Section 5 of the HeD Artifact Sharing Use Case, to ensure that these assumptions align to the implementation environment in which CDS content will be exchanged using a knowledge artifact. Failure to meet any of these assumptions could impact implementation of the knowledge artifact.

### Usage Conformance Testing Recommendations

The following text is pre-adopted from the HL7 V2.7.1 Conformance (Chapter 2B, 2.B.7.5). Please refer to the base standard documentation for a full explanation of conformance concepts. Usage is described here as it introduces the revised approach to conditional element handling; upon successful ballot and publication this material will be replaced with a reference to the normative documentation.

#### System Requirements

This implementation guide is not focused on specific EHR and CDS system behaviors that may apply to the CDS Knowledge Artifact, such as the interaction between specific actors within the Use Case who may wish to search or import a CDS knowledge artifact.

Specific system requirements targeted in this implementation guide include the following:

|  |  |
| --- | --- |
| System Requirement | Description |
| Provides CDS Knowledge Artifact in Structured Format | The implementation guide defines how the knowledge artifact should be structured, NOT how the system actually generates the structure. |
| Provide metadata about CDS artifact in a standardized structured format | The implementation guide defines how the knowledge artifact metadata is applied to the different knowledge artifact types, and DOES NOT specify how systems should parse and interpret this metadata. |

Table 3 - CDS System Requirements Covered in this Guide

## Organization of this Guide

It is important for readers of this implementation guide to understand specific terms, actors, roles, and conventions used in this implementation guide.

### Definitions of Terms

Several terms are used throughout this document and a level of detailed technical understanding of healthcare standards is expected. It is extremely critical for the reader to review Appendix C - Definitions, to understand the specific acronyms and terms that are used in this implementation guide. In addition, the reader should be familiar with the terms defined in Appendix A of the HeD Artifact Sharing Use Case – wherever possible, this implementation guide reuses existing terms from that document and seeks to minimize the introduction of any new terms, including those listed in the table below.

| Term Name | Description of the Term and its Role in HeD Harmonized Schema |
| --- | --- |
| AHRQ eRecs | The AHRQ eRecs project is a source of recommendations that was used to inform the design of the expression language used within the CDS Knowledge Artifact schema. |
| Allscripts CREF | The Allscripts CREF schema is a foundational set of schemas that was used to inform the design of the design of the expression language used within the schema |
| CDS Knowledge Artifact | The CDS Knowledge Artifact (as defined in the HeD Artifact Sharing Use Case) is medical knowledge represented in a structured and encoded form to enable computer-based clinical decision support.  This implementation guide specifically is focused on the structure and encoding necessary to make the knowledge artifact available as CDS content. |
| CDS Knowledge Artifact Type | The CDS Knowledge Artifact Type represents the different types of CDS content that may constitute a CDS Knowledge Artifact. As defined in the HeD Artifact Sharing Use Case, the artifact type may consist of artifact specific data, metadata, and the components specific to that type of artifact.  This implementation guide specifically supports three initial knowledge artifact types:   1. Event Condition Action (ECA) Rules 2. Order Sets 3. Documentation Templates |
| CDSC L3 | The CDSC (CDS Consortium) L3 schema is foundational to the design and structure of the CDS Knowledge Artifact schema and is used as a source for a large number of the types defined in the schema. This schema has been closely aligned to the CDSC L3 schema. |
| Component | A component is a grouping of data elements within the CDS Knowledge Artifact. The structure of this implementation guide supports the use of a library of reusable components when developing a knowledge artifact. |
| Documentation Templates | A structured form for recording information on a patient into a set of pre-defined data slots. |
| Event Condition Action (ECA) Rules | A CDS knowledge artifact with the general syntax “on event, if condition is true, then do action.” |
| CDS Knowledge Artifact schema | The CDS Knowledge Artifact schema is the formal XML schema of the harmonization of multiple CDS standards and specifications adopted as the starting point for the CDS Knowledge Artifact structure, such as:   1. HL7 vMR 2. CDSC L3 3. Allscripts CREF 4. Arden Syntax 5. Yale GEM 6. AHRQ eRecs |
| HL7 Arden Syntax | The HL7 Arden Syntax standard is a primary source of knowledge that was used to inform the development of the expression language used within the CDS Knowledge Artifact schema. |
| HL7 vMR (Virtual Medical Record) | A Virtual Medical Record (vMR) for Clinical Decision Support (CDS) is a data model for representing clinical data relevant to CDS. The vMR encompasses data about a patient's demographics and clinical history, as well as CDS inferences about the patient (e.g. recommended clinical interventions).  The vMR standard plays two different roles in the HeD schema:   1. Expressions in a CDS Knowledge Artifact may refer, through the use of expressions, to an actual vMR record included as an ‘external data reference’ in the artifact. These are used for the purpose of deciding applicability of a portion of the artifact to a patient population. (For example, deciding the amount of insulin to give a diabetic patient is a function of the patient’s blood sugar levels.) 2. Actions in the CDS Knowledge Artifact (referred to as “orders” by many EHR systems) are represented using the same clinical data elements that are used in a virtual medical record. For example, the vMR “substance administration” type is used as the model to specify the medication to give a patient, including dose, frequency, PRN reasons, etc. |
| Order Sets | A pre-defined and approved group of orders related to a particular clinical condition (e.g., hypertension treatment and monitoring) or stage of care (e.g., hospital admission to Coronary Care Unit). Order sets are used within electronic health record systems as a checklist for physicians when treating a patient with a specific condition. An order set is a structured collection of orders presented to the physician in a computerized physician order entry system (CPOE). |
| Yale Guidelines Element Model (GEM) | The Yale Guidelines Element Model (GEM) is a supporting schema that is harmonized by the CDS Knowledge Artifact schema and provides additional background material for elements such as Knowledge Resources. GEM was a significant conceptual input to the CDSC L3 schema. |

Table 4 - Key Terms in this Guide

### Definitions of Actors

This implementation guide is specifically targeted to meeting the requirements of the following roles and actors (each of which are drawn from the HeD Artifact Sharing Use Case)

| Actor/Role from HeD Artifact Sharing Use Case | How Implementation Guide supports this role |
| --- | --- |
| CDS Knowledge Artifact Supplier (including CDS/EHR vendors and CDS content suppliers)   1. An organization/system that creates,collects and/or distributes CDS Knowledge Artifacts. | Provide the structure and encoding format needed to generate CDS content that conforms to the Health eDecisions CDS Knowledge Artifact schema.  NOTE: This implementation guide DOES NOT specify HOW the content supplier actually generates the CDS knowledge artifact, only the desired structure and encoding of that artifact. |
| CDS Knowledge Artifact Integrator (including CDS/EHR vendors and healthcare delivery systems that implement CDS systems)   1. An organization/person/system that imports, adapts, and maps CDS Knowledge artifacts to be embedded within a CDS system. | Provides a standardized format for vendors and implementers of EHR and CDS systems to adopt when creating and/or consuming CDS Content.  NOTE: CDS and EHR Vendors (as with all HeD Artifact Sharing actors) are given considerable flexibility when implementing the CDS Knowledge Artifact |

Table 6 – Roles and Actors Supported in this Guide

### Conventions Used

The conventions defined in this document are specifically drawn from other implementation guides and include common conventions adopted by HL7, IHE, ASTM, and ISO. This implementation guide adopts high-level conformance statements that apply to each complex type, element, and attribute defined within the HeD schema.

#### Use of Cardinality

Cardinality applies specifically to metadata and the data elements associated with the CDS Knowledge Artifact. The specific conventions for cardinality in this implementation guide are as follows:

| Cardinality | Explanation of Cardinality |
| --- | --- |
| 0..0 | The element is never present |
| 0..1 | The element MAY be omitted and has at most one occurrence |
| 1..1 | The element is present once and only once |
| 0..n | The element MAY be omitted or may repeat up to *n* times |
| 1..n | The element MUST appear at least once, and MAY repeat up to n times |
| 0..\* | The element MAY be omitted, or it MAY repeat an unlimited number of times |
| 1..\* | The element MUST appear at least once, and MAY repeat an unlimited number of times |
| m..n | The element MUST appear at least *m* times, and at most, *n* times |
| 2..2 | The element MUST appear two and only two times |
| 3..3 | The element MUST appear three and only three times |

Table 7 - Summary of Cardinality

#### Use of Versioning

Version control for this implementation guide and the associated schema files is enforced using the Google Code Repository that hosts the Health eDecisions/CDS Knowledge Artifact project. Each of the components included in the associated CDS Knowledge Artifact XML schema files are kept in this repository.

Versioning is of critical importance for this implementation guide due to the large number of XML schemas included in this implementation guide, and wherever necessary, the specific version of the XML schema being referenced in this implementation guide is noted. In all cases, the schema files hosted in the Google Code Repository are to be noted as the source of truth.

#### Use of References

Documentation and terms that appear throughout this document in ***bold italic*** text indicate a specific reference. Documents are referenced to indicate that implementers should refer to that documentation for final conformance language and other levels of guidance. An example is shown below:

For conformance language, please refer to the [***Conformance Implementation Manual***](http://wiki.hl7.org/index.php?title=Conformance_Implementation_Manual) for further details

Working code examples are also provided in this implementation guide to assist in understanding the CDS Knowledge Artifact schema. While this implementation guide is normative, examples are meant to be informative, and are provided for human readability. In all cases, the formal specification referred to by the example takes precedence.

#### Use of Conformance Language

Conformance language is defined within this implementation guide at a high level, to ensure alignment to the multiple standards/specifications which have been harmonized. The use of conformance language within this document is limited to further constraints or relaxation of constraints on existing standards/specifications. New conformance language that specifically deviates from the underlying standard/profile is avoided wherever possible. Also, in those instances where new metadata is being defined, specific constraints are offered. Implementers should refer to the CDS Knowledge Artifact schema for the source for all conformance statements and rules.

Conformance language is defined throughout this implementation guide using **BOLD** **CAPS** to denote the conformance criteria to be applied. The conformance language that is used in this implementation guide is drawn from RFC 2219, and the conformance matrix offered for use in this implementation guide is drawn from the HL7 Implementation Guide for CDA® Release 2: IHE Health Story Consolidation, Release 1:

1. **SHALL/MUST**: an absolute requirement for all implementations of the Knowledge Artifact
2. **SHALL NOT**: an absolute prohibition against inclusion for all implementations of the Knowledge Artifact
3. **SHOULD/SHOULD NOT**: A best practice or recommendation to be considered by implementers within the context of their requirements to implement the Knowledge Artifact; there may be valid reasons to ignore an item, but the full implications must be understood and carefully weighed before choosing a different course
4. **MAY**: This is truly optional language for an implementation; can be included or omitted as the implementer decides with no implications

The Consolidated Conformance Verb Matrix included as part of the HL7 Implementation Guide for CDA® Release 2: IHE Health Story Consolidation, Release 1 (shown below) summarizes how the different standards/profiles are used within the implementation guide, and also lists specific recommendations used in this implementation guide:

| RFC 2119 | HL7 | IHE | HITSP |
| --- | --- | --- | --- |
| SHALL  Absolute requirement of the specification | SHALL  Required/Mandatory | R (Required)  Element must be present but can be NULL. | R (Required)  Data elements must always be sent. A NULL can be sent. |
| SHALL NOT  Absolute prohibition of the specification | SHALL NOT  Not Required/Mandatory | - | - |
| SHOULD  Recommended  There may exist valid reasons in certain circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course. | SHOULD  Best Practice or Recommendation | R2 (Required if known)  The sending application must be able to demonstrate that it can send all required if known elements, unless it does not in fact gather that data. If the information cannot be transmitted, the data element contains a value indicating the reason for omission of the data. | R2 (Required if known)  If the sending application has data for the data element, it is REQUIRED to populate the data element. If the value is not known, the data element need not be sent. |
| SHOULD NOT  Not Recommended | SHOULD NOT  Not Recommended | - | - |
| MAY  Optional | MAY  Accepted/Permitted | O (Optional) | O (Optional) |
| - | - | C (Conditional)  A conditional data element is one that is required, required if known or optional depending upon other conditions. | C (Conditional)  Required to be sent when the conditions specified in the HITSP additional specifications column are true. |

Table 8 - Implementation Guide- Conformance Verb Matrix

The use of the word “recommendation” is also used throughout this implementation guide, especially in light of the initial level of development being done on the harmonization of CDS standards and specifications into a CDS Knowledge Artifact schema. A recommendation is used to offer implementers flexibility in their CDS environments by recommending an approach to be followed, while not constraining in any way the use of alternative options. Recommendations are primarily used in those areas where the S&I Framework requests further implementation feedback from implementers and pilot sites prior to defining conforming criteria.

Optionality is further defined for implementers for each of the metadata elements that were outlined within the HeD Artifact Sharing Use Case in Section 11, using IHE guidelines:

|  |  |
| --- | --- |
| Guideline Term | Definition |
| Required | Element must be present and CANNOT BE NULL (no NULL flavors allowed). |
| Required if Known | The sending system must be able to demonstrate that it can send all required elements within the CDS knowledge artifact, unless it does not gather that specific element or does not plan to include it. If the information cannot be transmitted in the CDS knowledge artifact, the data element contains a value indicating the reason for omission of the data from the knowledge artifact. |
| Optional | There is no need to include this element unless the implementer so desires. |
| Conditional | A conditional data element is one that is required, required if known or optional depending upon other conditions that may be present in the CDS environment.  Implementers have some latitude to apply conditions to specific metadata or other data elements within the knowledge artifact that do not apply to their environment. |

Table 9 - Data Element - Optionality Levels

### Schema Roadmap

The approach used to document the CDS Knowledge Artifact schema is very similar to the approach used in the development of the HL7 Version 3 Domain Analysis Model: Virtual Medical Record for Clinical Decision Support (vMR-CDS), Release 1. The schema has been developed as a set of XSD files that is documented within this implementation guide, together with the datatypes for each of the schema elements.

A Google Code Repository is being used to host the schemas, example files, and documentation associated with this Implementation Guide. The location of this repository is:

<http://code.google.com/p/health-e-decisions/>

Section 3 of this guide covers the different knowledge artifact types supported by the CDS Knowledge Artifact schema. Section 4 of this guide details the different components that may comprise an artifact type, and 5 covers the types defined in the CDS Knowledge Artifact schema.

| **Schema File Name** | **Supported Components** |
| --- | --- |
| metadata.xsd | Metadata |
| base.xsd | Supporting Evidence  Supporting Reference |
| expression.xsd | Expression |
| action.xsd | Action |

Table 10 - Alignment of Schema Files to CDS Knowledge Artifact components

For knowledge artifact types, a more relaxed level of constraint is applied. For components, a stricter level of constraint is applied (specific constraints and rules apply to elements and attributes to ensure that components within artifacts are well understood and can interoperate).

# CDS Knowledge Artifact Types

This implementation guide is designed to support each of the three CDS Knowledge Artifact types defined in the HeD Artifact Sharing Use Case. In this section, specific guidance on implementing each of these artifact types is defined, with specific emphasis on:

1. Describing a logical example of what the CDS Knowledge Artifact looks like
2. Defining the specific components of each knowledge artifact type
3. Identifying the required and optional components and their attributes for each knowledge artifact type – known as conformance statements

Each of the knowledge artifact types is structured in a format to allow for flexibility in implementation. This format allows for defining a base set of requirements for a specific artifact type, which may then be extended by implementers. Please refer to Appendix XX for details on each artifact type as well as its components and attributes.

## Conformance to the Health eDecisions Knowledge Artifact Specification

There are layers of constraints that must be met to conform to the HeD specification:

1. An artifact SHALL conform to the constraints specified in the HeD XML schema files. Thus, an HeD knowledge artifact must be well-formed and valid according to the HeD schema. These constraints are described in Chapter 5.
2. An artifact SHALL conform to the general constraints for HeD artifacts described in Chapter 3 (particularly Section 3.2) and Chapter 4. Many of these constraints are specified in the form of schematron rules that may be used as an optional tool for validation.
3. An artifact SHALL conform to the constraints described for its specific type. These constraints are described in
4. Section 3.3 for ECA Rule type artifacts
5. Section 3.4 for Order Set type artifacts
6. Section 3.5 for Documentation Template type artifacts
7. Each artifact SHALL also conform to the terminology specifications for its realm. The requirements for the US realm are described in Section/Appendix N.

## HeD Knowledge Artifact

An HeD Knowledge Artifact is represented in an XML file following the structure specified by its schema and conforming to the constraints specified within this document. The root element of the artifact is knowledgeDocument. A knowledgeDocument element contains a single ECA rule, an order set, or a documentation template. This section provides an overview of the HeD Knowledge Artifact and general constraints on the artifacts. Subsequent sections describe each artifact type in detail and enumerate constraints that are specific to the respective artifact type.

### Metadata Elements

Each knowledgeDocument contains a single metadata element. The metadata element supports knowledge management and the implementation of the artifact in a CDS system. The element contains information about the identity of the artifact (including its structure), its provenance, its lifecycle and status, and its applicability in the care context. The latter is specified in part in the applicability element. This element is used for indexing a knowledge artifact in a repository and as a guide to integrating the artifact into the appropriate system and workflow. This element is not used during the execution of the artifact. During execution, the conditions elements are used to determine if an artifact is applicable for the patient.

### Def Elements

The externalData element contains any number of def elements, each of which provides a declarative specification of the data that are needed by this artifact for execution. Each def element maps a name to a data element using an expression of type ClinicalRequest (e.g., the name lastBodyTemperature maps to a ObservationResult object with specified LOINC codes and was the one most recently measured). The def name can be used in other expressions (lastBodyTemperature > 102 degrees Fahrenheit) to refer to this data element. Mappings to clinical data must be specified using the VMR as the data model. This is explained in Section 4.x. The externalData may also contain any number of parameterDef elements that can be used to specify parameters that may be passed into the artifact when an artifact is called by another artifact or system. For example, an artifact for diabetes control may specify a parameter that allows control over the length of time since an A1c test has occurred.

### Expression Elements

The expressions element provides a mechanism to construct modular expressions that can improve reusability and readability of an artifact. That is expression elements defined here may be used elsewhere in the artifact where expressions are needed (e.g., in condition or inside other expressions). Each def element maps a name to an expression (e.g., the name elevatedLDL may be mapped to an expression, value of lastLDL > 100). The expression may be referenced by name in an expression of type ExpressionRef to construct more complex expressions (e.g., elevatedLDL and CHDRiskPercentAt10years > 20).

### Trigger Elements

The triggers element allows definition of events (e.g., new serum potassium result available) that cause the artifact to be executed. A more detailed description of how a trigger element is constructed is provided in Section 4.x.

### Condition Elements

The conditions elements contain logical criteria that evaluate to true or false and that control further execution of an artifact. The conditionRole element of a condition specifies how the criteria are to be used. In the current version of the schema, the only conditionRole is applicableScenario. When a condition with an applicableScenario role evaluates to true, the actions in the actionGroups element are executed. Conditions usually are based on the data items specified in the externalData element and represent the clinical situation (e.g., patient has diabetes and has not had a hemoglobin A1c element in the past six months).

### Behavior Elements

Behaviors control how an artifact is presented and how the users may interact with an artifact. At the level of the knowledgeDocument, no behavior types are included as part of the HeD specification. However, a supplier may define behavior types that are applicable to an artifact.

### Action Elements

The actionGroups element contains the “output” actions or the recommendations of the CDS. These may be in the form of messages (such as reminders), or structured clinical acts (e.g., a laboratory test order) that can be implemented via clinical systems such as a computerized provider entry system or a documentation system, or may create new events (e.g., declaration of a patient state such as failure to a treatment). Clinical actions must be specified using the VMR as the data model, as described in Section 4.x. The action elements may be nested inside actionGroup elements (e.g., all medication-related actions may be aggregated in one actionGroup element). To enable reuse and modularity, an actionGroup may be incorporated by reference from another CDS artifact (e.g., a ventilator protocol may be defined in an artifact that is reused in different order sets). Elements of type actionGroups can have behaviors associated with them. These behaviors specify, for example, how a user may select from a set of actions in the group, or whether an action is required or optional.

In the next chapter, the components of the HeD artifacts are described in more detail. The remaining sections of this chapter describe the three different types of artifacts currently supported by HeD.

NOT SURE YET IF NON-SPECIFIC (NOT APPLICABLE TO PARTICULAR TO AN ARTIFACT-TYPE) CONSTRAINTS SHOULD GO HERE OR IN THE NEXT CHAPTER.

## Event Condition Action (ECA) Rules

### Conceptual Overview

As defined in the HeD Artifact Sharing Use Case, an event condition action rule is an artifact with the general syntax “on event, if condition is true, then do action.” The event triggers the invocation of the rule. The condition is a logical test that, if satisfied or evaluates “true,” causes an action. The action part consists of a set of operations to execute. These actions may in turn cause further events to occur, which may in turn cause other ECA rules to fire.

A typical rule can be represented in the HeD Knowledge Artifact schema as shown in the diagram below:

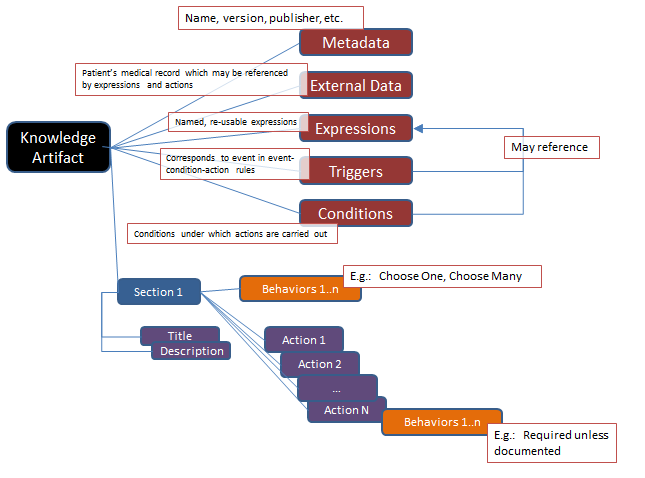


Figure 4 - ECA Rule – Conceptual Overview

A system with this rule in its rule repository activates the rule when the triggering events occur. At that point, the conditions are evaluated. Conditions may reference external data and may be composed of expressions defined in the expressions section. If the condition is true, the actions specified in the “action” part of the rule are executed (represented in Section 5.3 of the implementation guide – Working with Actions).

The action groups are the containers and organizers of the actions in an ECA rule. A rule typically has a single action group (top level section), but may have more. Conceptually, a set of actions in a rule could be considered a “mini order set” which is presented to a clinician at certain times and under certain conditions. As such, the actions may be structured hierarchically using action groups and behaviors to specify how the orders should be a shown to a provider, and to place restrictions on how a provider chooses from the available set of orders. It should be noted that this is just a conceptual example, and that not all actions are necessarily orders. For example, an action can be a creation of a new event that triggers another rule, a future encounter, or the creation of a state description of the patient.

The following example illustrates the conceptual structure of the rule:

Hemoglobin A1C Reminder for Patients with Poorly Controlled Diabetes

Adapted From: CDSC L3 Diabetes Mellitus Assessment Rule

Clinical Focus: Diabetes Mellitus

Trigger

1. Start of encounter in primary care clinic

Conditions

1. Patient is an adult, and
2. Patient has diabetes, and
3. Patient has not had a Hemoglobin A1C test in the last three months, and
4. Patient had an abnormal Hemoglobin A1C test sometime in the nine-month period before the last three months

Actions

1. Notify the provider
2. Order Hemoglobin A1C in the next three days

### ECA Rule CONFORMANCE REQUIREMENTS

The following sub-sections describe how to use the HeD schema for expressing a rule and specific conformance constraints for ECA rules. All constraints specified are relative to the root knowledgeDocument element.

#### Metadata

Constraint ECA-1: The value attribute of the artifactType element in the metadata element SHALL be specified as "Rule".

<sch:pattern name="ECA-1: Artifact type is Rule">

<sch:rule context="/hed:knowledgeDocument/hed:metadata/hed:artifactType">

<sch:assert test="./@value='Rule'">

The value attribute of artifact type must be 'Rule'

</sch:assert>

</sch:rule>

</sch:pattern>

#### External Data

No constraints specific to ECA rules are specified for the externalData element of a knowledgeDocument.

#### Expressions

No constraints specific to ECA rules are specified for the expressions element of a knowledgeDocument.

#### Triggers

Even though ECA rules require a trigger, triggers intentionally are left as optional in conformance requirements for ECA rules. This is because the triggering events are often specific to a particular implementation based on organizational workflows or policies. Thus, triggers may not always be appropriate to define in a sharable ECA rule Knowledge Artifact and it might be better to add these at the time the rule is integrated into the CDS system.

Thus, no constraints specific to ECA rules are specified for the triggers element of a knowledgeDocument.

Implementers **MAY** create additional types of Triggers that extend the existing types defined in this implementation guide. See Section XX on extending HeD enumerated types for details.

#### Conditions

An ECA Rule shall always define a condition that determines if the specified actions in the actionGroups element must be executed. The following constraint applies to conditions:

Constraint ECA-2: There SHALL be exactly one condition element specified in the conditions element whose conditionRole is “ApplicableScenario”.

<sch:pattern name="ECA-2: One condition of type ApplicableScenario is included">

<sch:rule context="/hed:knowledgeDocument">

<sch:assert test="count(hed:conditions/hed:condition[hed:conditionRole ='ApplicableScenario'])=1">

Exactly one condition of type ApplicableScenario must be present in a rule

</sch:assert>

</sch:rule>

</sch:pattern>

#### Behaviors

The current version of the HeD schema does not specify particular behaviors at the level of the rule. In fact, the behaviors defined in the HeD specification shall not be used within the behaviors element on the knowledgeDocument. However, CDS Knowledge Artifact suppliers MAY define ECA Rule behaviors that can be used at this level.

The following constraints are specified for behaviors at the ECA rule level:

Constraint ECA-3: The behaviors element SHALL NOT contain a behavior element of the types GroupSelectionBehavior, PrecheckBehavior, RequiredBehavior, or GroupOrganizationBehavior.

<sch:pattern name="ECA-3: None of the HeD predefined behaviors are used at the rule level">

<sch:rule context="/hed:knowledgeDocument/hed:behaviors/hed:behavior">

<sch:assert test="not(@xsi:type ='GroupSelectionBehavior' or

@xsi:type='PrecheckBehavior' or

@xsi:type='RequiredBehavior' or

@xsi:type='GroupOrganizationBehavior')">

A behavior of this type is not allowed at the level of the rule

</sch:assert>

</sch:rule>

</sch:pattern>

#### ActionGroups

No constraints specific to ECA rules are specified for the actionGroups element of a knowledgeDocument.

## Order Sets

### Conceptual Overview

As defined in the HeD Artifact Sharing Use Case, an order set is a pre-defined and approved group of orders related to a particular clinical condition (e.g., hypertension treatment and monitoring) or stage of care (e.g., hospital admission to Coronary Care Unit). An order set is used as a checklist for the clinician when managing a patient with a specific condition. It is a structured collection of orders (or actions in the HeD schema) relevant to that condition and presented to the clinician in a computerized provider order entry system (CPOE).

The actions in an order set are typically organized hierarchically, as a set of sections, sub-sections, etc., with the actions themselves at the very bottom of the structure. In the HeD Knowledge Artifact schema, we generalize the notion of “section” and call it a “group.” Each group and/or subgroup may have behavior indicators associated with it, e.g., the number of actions that can/should/must be selected from the group of actions.

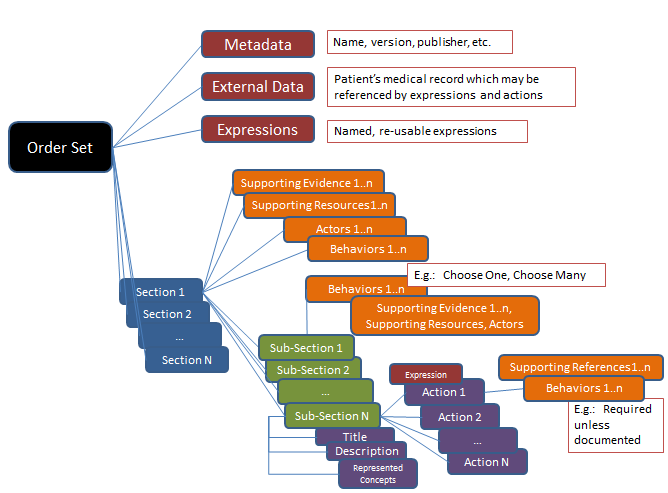


Figure 6 - Order Set – Conceptual Overview

A clinician chooses an order set from a library of order sets in the CPOE system based on metadata associated with the order set, and some information about the patient (typically just the diagnosis). The clinician then selects orders to be placed for a particular patient.

The following example shows the a partial snippet of an order set that is organized according to the conceptual structure shown above:

Heart Failure Admission to Med/Surg (partial order set)

Venue: InPatient

Population: Adult

Section: General

1. ADHERE Risk Model (click here to link to evidence)
2. Heart Failure Survival Score (click here to link to evidence)

Section: Activity (CHOOSE ONE)

1. Ambulate
2. Bed rest

Section: Nursing Orders

Sub-section: Assessments (CHOOSE ANY)

1. Cardiac Monitor
2. Measure blood pressure; orthostatic

Sub-section: Interventions (CHOOSE ANY)

1. Elevate head of bed
2. Urinary catheter initiation/management

Section: Medications

Sub-section: Angiotensin-Converting Enzyme Inhibitors (CHOOSE ONE)

1. For patients with diastolic heart failure who are intolerant to an ACE inhibitor, consider the use of an ARB
2. For patients with systolic heart failure who are hemodynamically stable and are intolerant to an ACE inhibitor due to cough, use an ARB
3. captopril 6.25 milligram orally 3 times a day
4. lisinopril 2.5 milligram orally once a day

Sub-section: Beta-Blockers (CHOOSE ONE)

1. For patients with diastolic heart failure and a previous MI, use a beta-blocker
2. For patients with diastolic heart failure, consider the use of a beta-blocker
3. For patients with systolic heart failure who are hemodynamically stable, use beta-blocker therapy (eg, bisoprolol, carvedilol, metoprolol extended release)
4. bisoprolol 1.25 milligram orally once a day
5. carvedilol 3.125 milligram orally 2 times a day
6. carvedilol phosphate SR 10 mg multiphase 24 hr cap 1 capsule orally once a day
7. metoprolol succinate SR 25 mg 24 hr tab 0.5 tablet orally once a day

The HeD Knowledge Artifact schema can be used to construct an order set. Such an order set may be imported into the CPOE module of an electronic health record system.

### Order Set Conformance Requirements

The following sub-sections describe how to use the HeD schema for expressing an order set and specific conformance constraints for order sets. All constraints specified are relative to the root knowledgeDocument element.

#### Metadata

Constraint OS-1: The value attribute of the artifactType element in the metadata element SHALL be specified as "Order Set".

<sch:pattern name="OS-1: Artifact type is Order Set">

<sch:rule context="/hed:knowledgeDocument/hed:metadata/hed:artifactType">

<sch:assert test="./@value='Order Set'">

The value attribute of artifact type must be 'Order Set'

</sch:assert>

</sch:rule>

</sch:pattern>

#### External Data

External data are likely to be used only if an order set or its constituent actions have conditions or other expressions. Many order sets do not contain these, hence an order set may not include external data.

No constraints specific to order sets are specified for the externalData element of a knowledgeDocument.

#### Expressions

As noted above, order sets often do not contain any expressions. Expressions may be used when the order set or actions have patient-specific conditions.

No constraints specific to order sets are specified for the expressions element of a knowledgeDocument.

#### Triggers

No constraints specific to order sets are specified for the triggers element of a knowledgeDocument.

(Are triggers allowed?)

#### Conditions

The condition with type applicableScenario specifies that the order set **MAY** be used only if the condition expression evaluates to true at the point in time when the order set is used in the target system. If the desire is to express the clinical scenarios in which this order set is applicable without requiring a run-time evaluation of the condition, consider using the applicability elements of the metadata instead. This approach will allow the clinician to find an order set by browsing or searching for it in an order set repository.

No constraints specific to order sets are specified for the conditions element of a knowledgeDocument.

#### Behaviors

The current version of the HeD schema does not provide any behaviors applicable at the top level of the order set. Furthermore, none of the behaviors included as part of the HeD schema may be used here. Behaviors that apply at the action-level are described within the following section on actionGroups.

No constraints specific to order sets are specified for the behaviors element of a knowledgeDocument.

#### actionGroups

The actionGroups element contains the orders (represented as actions) which may be further organized into sub actionGroups. Behaviors may be specified at the level of the groups or the actions to specify the number of actions to be chosen, if an action should be prechecked (meaning it will be ordered, unless the clinician user unchecks that action), and whether an action must be executed. Actions, actionGroups, and behaviors are further described in Section 4.x.

The following constraints are defined for order sets:

Constraint OS-2: Order sets do not include functionality to modify or cancel an existing order. They also do not fire events. Accordingly, actions of type UpdateAction, RemoveAction FireEventAction, SHALL not be allowed in order sets.

<sch:pattern name="OS-2: Some action types are disallowed in order sets">

<sch:rule context="//hed:simpleAction">

<sch:assert test="not(@xsi:type ='FireAction' or

@xsi:type='UpdateAction' or

@xsi:type='RemoveAction')">

An action of this type is not allowed in an order set

</sch:assert>

</sch:rule>

</sch:pattern>

Constraint OS-3: Order sets SHALL only incorporate other artifacts via groupReference that are of type ‘Order Set’. A schematron rule is not available for this constraint.

Might need to add some constraints about behaviors here.

Upon completion of this implementation guide and prior to presentation to HL7 for final ballot, a mapping of the elements defined in the HL7 V3 Order Set DSTU will be provided, to show how the order set data elements defined in the CDS Knowledge Artifact schema map to that standard.

## Documentation Templates

### Conceptual Overview

As defined in the HeD Artifact Sharing Use Case, a documentation template is a structured form for recording information on a patient into a set of pre-defined data slots. These templates are used to guide structured data entry within an EHR or other clinical information system. Some types of clinical documents that can be represented via the Documentation Template artifacts are encounter summaries, procedure notes, patient-reported outcomes, and flowsheets.

A documentation template is a structured collection of what are known variously as documentation concepts, form elements, or observation items. Each documentation concept (the moniker used in the HeD schema) also can be thought of as a question to the user entering the data. Elements within the documentation concept guide and constrain the user’s responses, for example, a list from which to choose an answer, whether an answer is a number, a date, or some other type, and the cardinality of the answer.

Documentation concepts are contained in an action of type CollectInformationAction. This enables these concepts to be presented to the user conditionally (e.g., to ask questions appropriate to a patient’s gender or to ask questions based on other responses), to compute responses for a concept based on previous responses or data from an EHR score (e.g., a risk score), and to bind the responses into expressions that can drive logic elsewhere in the documentation template (e.g., ask questions conditionally as described above). More details on the use of CollectInformationAction and documentation concepts are provided in Section 4.x.

The documentation concepts in a template are typically organized hierarchically, into sections and sub-sections with the concepts themselves at the very bottom of the structure. As described in the overview of Order Sets, in HeD Knowledge Artifact schema these “sections” are called actionGroups. Similar to order sets, actionGroups in documentation templates may have behavior indicators associated with it, e.g., whether a documentation concept must have a response.

The figure below illustrates the conceptual structure of a documentation template.

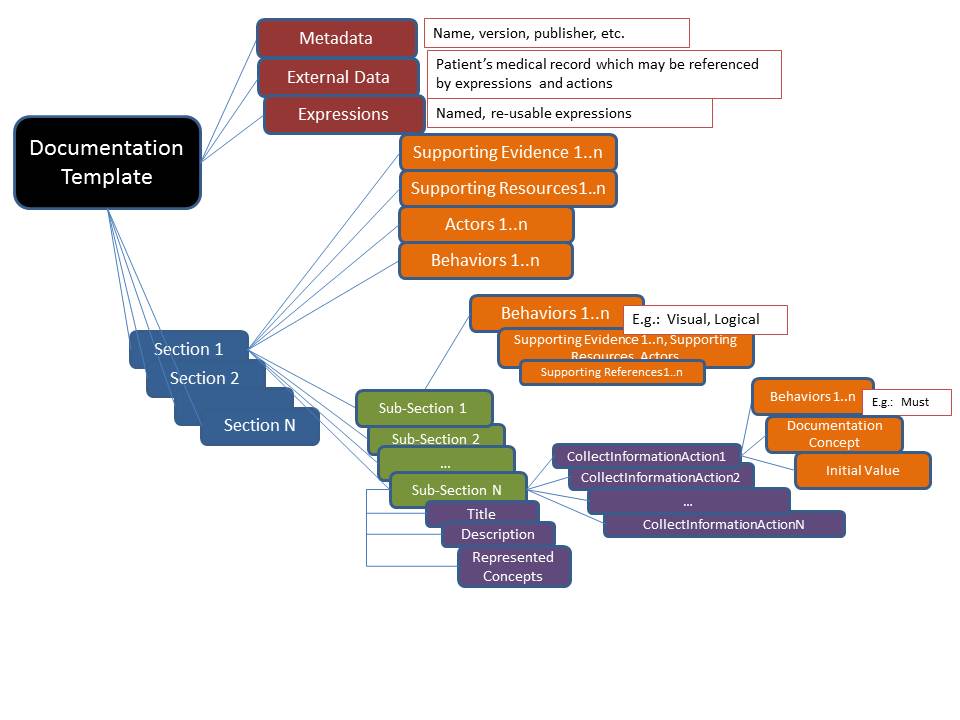


Figure 8 - Documentation Template - Conceptual Overview

The example below shows a documentation template organized according to the conceptual structure of the figure above.

History and Physical Exam for Headache

Venue: Office

Population: Adult

Section: Symptoms

1. Chief complaint (text, response required)
2. Duration (time interval)
3. Triggering factors (text)

Section: Physical Exam

Sub-section: Neurological Exam

1. Pupil (pick list: reacting to light, equal)
2. Tremors (boolean)

Sub-section: Vital signs

1. Heart rate (number)
2. Temperature (number)
3. Temperature location (pick list: oral, axillary, surface)

### Documentation Template Conformance Requirements

The following sub-sections describe how to use the HeD schema for expressing a documentation template and specific conformance constraints for documentation templates. All constraints specified are relative to the root knowledgeDocument element.

#### Metadata

Constraint DOC-1: The value attribute of the artifactType element in the metadata element SHALL be specified as "Documentation Template".

<sch:pattern name="DOC-1: Artifact type is Documentation Template">

<sch:rule context="/hed:knowledgeDocument/hed:metadata/hed:artifactType">

<sch:assert test="./@value='Documentation Template'">

The value attribute of artifact type must be 'Documentation Template'

</sch:assert>

</sch:rule>

</sch:pattern>

#### External Data

External data are likely to be used only if a documentation template has conditions or other expressions. If a documentation template does not contain these, then it may not include external data.

No constraints specific to documentation template are specified for the externalData element of a knowledgeDocument.

#### Expressions

Expressions may be used when the documentation template or the documentation concepts have patient-specific conditions.

No constraints specific to documentation templates are specified for the expressions element of a knowledgeDocument.

#### Triggers

No constraints specific to documentation templates are specified for the triggers element of a knowledgeDocument.

(Are triggers allowed?)

#### Conditions

The condition with type applicableScenario specifies that the documentation template **MAY** be used only if the condition expression evaluates to true at the point in time when the order set is used in the target system. As with order sets, if the desire is to express the clinical scenarios in which this documentation template is applicable without requiring a run-time evaluation of the condition, consider using the applicability elements of the metadata instead. This approach will allow the user to find a template by browsing or searching for it in a template repository.

No constraints specific to documentation template are specified for the conditions element of a knowledgeDocument.

#### Behaviors

The current version of the HeD schema does not provide any behaviors applicable at the top level of the documentation template. Furthermore, none of the behaviors included as part of the HeD schema may be used here.

No constraints specific to documentation template are specified for the behaviors element of a knowledgeDocument.

#### actionGroups

The actionGroups element contains the CollectInformationActions which contain the documentationConcept. The actionGroups element may also contain sub actionGroups or other documentation templates by references.

Behaviors may be specified at the level of the groups or the actions to specify the concepts to be documented. Actions, actionGroups, and behaviors are further described in Section 4.x.

The following constraints are defined for documentation templates:

Constraint DOC-2: Documentation concepts do not include functionality to cancel an existing order or remove records. Accordingly, an action of type RemoveAction SHALL NOT be included in documentation templates.

<sch:pattern name="DOC-2: Some action types are disallowed in documentation templates">

<sch:rule context="//hed:simpleAction">

<sch:assert test="not(@xsi:type =@xsi:type='RemoveAction')">

An action of this type is not allowed in a documentation template

</sch:assert>

</sch:rule>

</sch:pattern>

Constraint DOC-3: Documentation templates do not allow creation or modification of existing orders. Therefore “proposal” type elements from the VMR SHALL NOT be used within the actionSentence of a CreateAction or UpdateAction. A schematron rule is not available for this constraint.

Constraint DOC-4: Documentation templates SHALL only incorporate other artifacts via groupReference that are of type ‘Documentation Template’. A schematron rule is not available for this constraint.

Constraint DOC-5: Pre-check behavior for actions applies for clinical orders. Since documentation concepts do not include orders, precheck behaviors are disallowed. That is a CollectInformationAction SHALL NOT include a behavior of type Precheck.

<sch:pattern name="DOC-5: CollectInformationAction shall not incorporate precheck behavior">

<sch:rule context="//hed:simpleAction[@xsi:type='CollectInformationAction']/hed:behaviors/hed:behavior">

<sch:assert test="not(@xsi:type='PrecheckBehavior')">

An CollectInformationAction in a documentation template may not include a precheck behavior

</sch:assert>

</sch:rule>

</sch:pattern>

## Value Sets and Vocabulary Codes

The following sections explain guiding principles that Value Sets and Terminologies Work Group used in aligning vocabulary codes and value sets to CDS artifact data elements. The guiding principles for aligning coding systems and value sets to data elements are in line with vocabularies and value sets recommended by the CMS Blueprint for eMeasure Specifications. These specifications and full details of the value sets and vocabulary codes briefly described below can be found in the [HeD Value Sets and Terminology Implementation Guide](https://docs.google.com/a/esacinc.com/document/d/1JdjRxR2rQRioBXXxxiw9jpl6ZP5c2pDEbdW8xq5xjEc/edit).

### Value Set Use Case

If a value set is defined for HeD, then the value set must be used in order for end users to be considered conformant with HeD. Therefore, a valid use case should exist before a value set is defined for a data element.

### Coded Data versus Free Text

When there is general agreement across stakeholders regarding the semantic meaning of coded concepts (regardless of the actual descriptions) as well as in the importance of exchanging these defined concepts, it is worthwhile to encode a value set. An example is the concept of Frequency for substance administration. There would be unanimous agreement that 1x/day, 2x/day, 3x/day, etc. would make sense as coded values, even though there may be variation in the way these concepts are described (e.g., QD, BID, TID, QID).

On the other hand, if a shared consistency of the concepts that comprise a value set is not important, it may be better to NOT define a value set for that data element. An example is the concept of named departments within a hospital. Due to different departmental structures across healthcare organizations, it may not make sense to try to capture all of the different departmental concepts because of differences in granularity which may result in overlap of concepts, gaps for some organizations, and irrelevant coded values for others, resulting in little benefit from an attempt to determine a common set of agreed-upon representative concepts.

# CDS Knowledge Artifact Components

This chapter describes the components of a Knowledge Artifact, how they are used, and the constraints that apply to those components.

## Metadata

MET-1: The metadata element of the knowledgeDocument must specify the schemaIdentifier to have “urn:hl7-org:v3/hed” as the value of the root attribute, and “1” as the value of the version.

<sch:pattern name="MET-1: Schema identifier">

<sch:rule context="/hed:knowledgeDocument/hed:metadata/hed:schemaIdentifier">

<sch:assert test="./@root='urn:hl7-org:v3/hed' and ./@version='1'">

The schemaIdentifier root value must be ‘urn:hl7-org:v3/hed’ and the version must be '1'

</sch:assert>

</sch:rule>

</sch:pattern>

### Clinical Data Models

The HeD Schema specification itself does not reference any specific data model, and so can be used to create artifacts that capture logic expressed against any clinical data model. These data models are specified in the dataModels element of the artifact metadata.

NOTE: To enable sharing of artifacts, this implementation guide uses the vMR exclusively to represent clinical data.

For example, the following data models snippet specifies that the artifact will use the vMR:

<dataModels>

<modelReference>

<description value="Virtual Medical Record model" />

<referencedModel value="org.opencds.vmr.v1\_0.schema.vmr"/>

</modelReference>

</dataModels>

An artifact can reference any number of data models, so long as they are all defined within the dataModels element.

In addition to specifying the actual namespace of the data model within the dataModels element, the XML artifact must specify a local name for use in referencing the data model within the logic of the artifact. For example, the following namespace declaration in the root element specifies the local namespace for the vMR data model:

xmlns:vmr="org.opencds.vmr.v1\_0.schema.vmr"

With this namespace prefix defined, the types defined in the referenced schema can now be referenced within expression logic in the artifact. For example, the following snippet uses this local namespace prefix to define an external data request in terms of vMR types:

<def name="antithromboticNotPrescribedForDocumentedReason">

<expression xsi:type="ClinicalRequest" cardinality="Multiple"

dataType="vmr:ObservationResult" codeProperty="observationFocus.code"

dateProperty="observationEventTime">

<description>Patient reason or other reason for not prescribing an antithrombotic</description>

<codes xsi:type="List">

<element xsi:type="CodeLiteral" code="G8697"

codeSystem="2.16.840.1.113883.6.12" codeSystemName="CPT-4"

displayName="" />

</codes>

</expression>

</def>

This clinical request specifies that the type of data being requested is “vmr:ObservationResult”. During semantic verification, this type is resolved to the data model reference specified for the artifact.

For more information on specifying external data requirements for an artifact, please refer to the External Data section of the Expressions discussion in Section 4.4.14

## Actions

Actions are one of the core components of CDS that define the recommended operations to be executed in the specified clinical context. These can include creating orders for medications or diagnostic tests, collecting information, modifying or canceling an existing action, preventing a proposed action from occurring, or creating a new event that can trigger other CDS artifacts.

### Types of Actions

Actions are included in an artifact by adding an element simpleAction as a subElement to any element of type ActionGroup. The simpleAction element is of an abstract type called AtomicAction. The following concrete types of AtomicAction are defined in the schema and further described in Section XX:

1. CreateAction
2. RemoveAction
3. UpdateAction
4. FireEventAction
5. CollectInformationAction
6. DeclareResponseAction
7. MessageAction

#### Working with Clinical Data and Actions

The first three types of actions, respectively, create, remove, or update a clinical data object. These objects are specified as the actionSentence element of type Expression. The actionSentence expression returns a single object that is a subtype of ClinicalStatement from the VMR.

In the case of a createAction, the object returned by the actionSentence is a new one, i.e., one not representing existing data or orders from the patient records. The new object may be a proposal to carry out a new clinical action (such as starting a medication, or conducting a procedure). These are represented as VMR proposal statements. The new object may be an inference about the patient’s clinical state or a patient history element, which would be constructed as an object of the “Event” type from VMR. An actionSentence element is defined using an expression of type ObjectExpression, ObjectRedefine, or ComplexLiteral.

Table 11. Proposal STATEMENT types in the VMR

|  |
| --- |
| ProcedureProposal |
| SubstanceAdministrationProposal |
| AppointmentProposal |
| ObservationProposal |
| SupplyProposal |
| GoalProposal |

The actionSentence expression in removeAction and updateAction returns existing data or orders about the patient. These actionSentence objects will refer to an object that was specified in the externalData element, using its def element.

#### Communicating Messages

The MessageAction type may be used to communicate a message to a user. The message is constructed using an expression. The recipients of the message are specified as the actors of the MessageAction. The priority of the message is specified as the priority of the action and that is used by the CDS system to determine the channel to communicate the message based on the care context and the organization’s workflow.

Do we need the message to return a text string only?

#### Collecting Information

The CollectInformationAction type is used to obtain any information from the user. This type of action is most frequently used as part of the Documentation Template artifact type. An element called documentationConcept of type DocumentationItem specifies the details of the information to be obtained. The DocumentationItem type contains elements to describe the text to be displayed to the user, terminology codes, the data type and cardinality of the response, and the allowed range of values for the response.

The initialValue element of the CollectInformationAction type is used to optionally specify a value for the documentationConcept element that can be modified by the user. Since the initialValue is of type Expression, the value can be a simple literal value, a value computed from other responses entered by the user in the Documentation Template (e.g., when computing a risk score), a value derived from data in the patient records (e.g., question about use of anti-depressants might be responded from the medical record), or a combination of the above. An [example of a CollectInformationAction with an initialValue expression](#InitialValue_FLACC) is shown in Appendix E, FLACC Example. In this example, the value of the pain scale is computed based on user entered values for previous items.

In order to allow logic to be written that accesses the values entered by the user, the HeD Schema allows documentation items to be “bound” to a container. The container must be declared using a DeclareResponseAction. By default, the container is named “Responses”, but the schema allows any number of response containers to be created, so long as they are all uniquely named. The responseBinding element can then be used to designate a name for the value entered by the end-user. This name, once defined via a responseBinding element, will then be available as a property of the responses container. An [example of binding](#ResponseBinding_FLACC) is shown in Appendix E, FLACC Example. The collection name is specified using the DeclareResponseAction (the default name is “Responses”). An example of a [response container declaration](#DeclareResponseAction_FLACC) is shown in Appendix E, FLACC Example. The value of the property can then be used in an expression in some other action by referring to the name of the property within the response container. For example and as mentioned above, the value can be used to compute the initial value of another documentationConcept element, or as part of an actionSentence expression in createAction.

#### Creating Events

The FireEvent action can be used to create an event and place it in the working memory of the CDS system. The types of events that can be fired are specified in the enumeration DataEventType. A payload associated with the event is specified as an actionSentence. The event that is fired can be the trigger for another CDS artifact, as described in Section 4.x.

Need an example here – if we do not have any handy then let’s defer.

### Creating Action Sentence Expressions

Action sentences for new objects in the CreateAction, FireEvent, MessageEvent are created by constructing an object in a specified data model. The CDS Knowledge Artifact schema does not include a model of health data about a patient. However, this implementation guide specifies that HeD knowledge artifacts will use the vMR (instead the dependency is indirect) as the data model for a patient’s health data. There are two methods to specify action sentences using the vMR.

The first approach uses an expression of type ObjectExpression. In an ObjectExpression, the action sentence specifies the type of the Object (a concrete subtype of ClinicalStatement from the vMR) in the ObjectType attribute, and constraints on the values of properties of that type in the property elements. Since the constraints are specified as expressions, these can be literal values (5 mg), relative values (e.g., three days after surgery), or ranges (50 to 100 mg). An [example of ObjectExpression](#ObjectExpression_RespiratoryOrder) is shown in Appendix E, RespiratoryOrder.

An alternative approach is to use the ComplexLiteral expression. In this case, the vMR XML schema can be leveraged to specify the new object. The ComplexLiteral provides a more compact and simpler method to specify objects, when the properties can be specified as literal values, a common scenario in order sets. This is because the vMR schema does not allow ranges or constraints for many property values (e.g., the date of a Procedure). An [example of ComplexLiteral](#ComplexLiteral_Dopamine) is shown in Appendix E, DopamineComplexIVOrderWithComplexLiteral.

### Organizing Actions

Elements of the type ActionGroup can be used to organize the actions into logical and visual groupings. An ActionGroup type object incorporates its constituents in an element called subElements. There different types of constituents can be included:

1. An action (sub) group that is specified in place in this artifact (specified in the group element).
2. A reference to an action group. This functionally is a subgroup but resides in an artifact by itself (specified in the groupReference element).
3. Actions that are described in the previous sections (specified in the simpleAction element).

An action group may constitute a visual section of an order set (e.g., Medications section or Diets section) or of a documentation template (e.g., History of Present Illness section, Cardiovascular System Examination section) that are displayed with each other. The title of a visual group is the title or the heading of the section. An action group may also provide logical or functional relationship amongst the actions in a group (e.g., a set of orders which must be ordered all together or not at all). Such relationships are specified as behaviors. These are further described in Section 3.2.5.

Group references, specified by using the groupReference element, enable the construction of modular, reusable knowledge artifacts. A reusable group should be defined as a self-contained artifact in its own Knowledge Document. This group can then be used in another artifact by using the groupReference element and specifying the identifier from the metadata of the referenced order set. For example, a DVT Prophylaxis group may be defined as an order set artifact. This order set artifact may be included in Order Sets for clinical scenarios which call for the use of DVT prophylaxis. An [example of groupReference](#GroupReference_HeartFailure) is shown in Appendix E, HeartFailureAdmissionToMedSurgOrder. Similarly, a Smoking History documentation template artifact can be reused in a variety of documentation templates for visit notes based on different presenting conditions.

### Conditional Actions

Actions may have conditional expressions associated with them such that the action is executed only if given condition is true. Similarly, a group may have conditions associated with it, so that actions contained in it and its subgroups are executed only if the specified condition is true. This can be used to show orders selectively in an order set or to show appropriate documentation concepts in a documentation template (e.g., do not show prostate exam questions for female patients). The role type of these conditions is ApplicableScenario. An [example of applicableScenario](#ApplicableScenario_FLACC) is shown in Appendix E, HeartFailureAdmissionToMedSurgOrderSet.

Constraint ACT-1: An action or an action group MUST have at the most one condition with the condition role ApplicableScenario.

<sch:pattern name="ACT-1a: Actions have at most one condition with ApplicableScenario role">

<sch:rule context="//hed:simpleAction/hed:conditions">

<sch:assert test="count(hed:condition[hed:conditionRole ='ApplicableScenario'])&lt;=1">

Exactly one condition of type ApplicableScenario must be present in an action

</sch:assert>

</sch:rule>

</sch:pattern>

<sch:pattern name="ACT-1b: Action groups have at most one condition with ApplicableScenario role">

<sch:rule context="//hed:actionGroups/hed:conditions">

<sch:assert test="count(hed:condition[hed:conditionRole ='ApplicableScenario'])&lt;=1">

Exactly one condition of type ApplicableScenario must be present in an action group

</sch:assert>

</sch:rule>

</sch:pattern>

<sch:pattern name="ACT-1c: Action groups have at most one condition with ApplicableScenario role">

<sch:rule context="//hed:group/hed:conditions">

<sch:assert test="count(hed:condition[hed:conditionRole ='ApplicableScenario'])&lt;=1">

Exactly one condition of type ApplicableScenario must be present in an action group

</sch:assert>

</sch:rule>

</sch:pattern>

### Behaviors

Behaviors may be specified for groups and for actions. There are two types of behaviors which apply exclusively to groups: GroupSelectionBehavior and GroupOrganizationBehavior. The GroupSelectionBehavior specifies the number of items in the group that should be chosen by the recipient of the CDS. The selection options are listed in the enumeration GroupSelectionBehaviorType (e.g., all or none, exactly one, at most one).

The GroupOrganizationBehavior provides hints to the end user system to aid with display of the group’s subElements. The behavior is specified by selecting from the enumeration GroupOrganizationBehaviorType. A VisualGroup indicates a group of items displayed together, perhaps as a subsection with a title and description. A LogicalGroup indicates a group of items that are logically related, but do not need to be separated visually from other items. A SentenceGroup indicates group of items which share a common orderable (such as "aspirin"), but differ in the details of their administration (e.g., dose level, frequency, route of administration). The end user system may choose to separate out the common orderable items, and show the order details underneath, or could decide to show each item as a separate orderable in and of itself. An [example of GroupSelectionBehavior](#GroupSelectionBehavior_RespiratoryOrder) is shown in Appendix E, RespiratoryOrder. An [example of GroupOrganizationBehavior](#GroupOrganizationBehavior_HeartFailure) is shown in Appendix E, HeartFailureAdmissionToMedSurgOrderSet.

The following constraints are specified for group behaviors:

Constraint BHV-1: A behavior of type GroupSelectionBehavior MUST be specified as a behavior of an actionGroup element only. It MUST NOT be used with a simpleAction element.

<sch:pattern name="BHV-1: GroupSelectionBehavior can only be used with action groups">

<sch:rule context="//hed:behavior[@xsi:type='GroupSelectionBehavior']">

<sch:assert test="name(../..)='actionGroup'">

GroupSelectionBehavior MUST be specified under action groups only

</sch:assert>

</sch:rule>

</sch:pattern>

Constraint BHV-2: A behavior of type GroupSelectionBehavior MUST be specified as a behavior of an actionGroup element only. It MUST NOT be used with a simpleAction element.

<sch:pattern name="BHV-2: GroupOrganizationBehavior can only be used with action groups">

<sch:rule context="//hed:behavior[@xsi:type='GroupOrganizationBehavior']">

<sch:assert test="name(../..)='actionGroup'">

GroupOrganizationBehavior MUST be specified under action groups only

</sch:assert>

</sch:rule>

</sch:pattern>

Constraint BHV-4: An actionGroup element with a behavior of type GroupOrganizationBehavior and value of VisualGroup SHOULD have a title specified.

<sch:pattern name="BHV-4: An action group with GroupOrganizationBehavior of VisualGroup SHOULD have a non-empty title">

<sch:rule context="//hed:behavior[@xsi:type='GroupOrganizationBehavior' and hed:value='VisualGroup']">

<sch:report test="not(../../hed:title/@value)">

An action group with GroupOrganizationBehavior of VisualGroup MUST have a non-empty title

</sch:report>

</sch:rule>

</sch:pattern>

Constraint BHV-5: Any actions specified as a subElement in an actionGroup with behavior of type GroupOrganizationBehavior and value of SentenceGroup MUST NOT specify a behavior of type RequiredBehavior.

<sch:pattern name="BHV-5: Sub-elements of an action group with Group Organization Behavior of SentenceGroup MUST NOT specify Required Behavior.">

<sch:rule context="//hed:behavior[@xsi:type='GroupOrganizationBehavior' and hed:value='SentenceGroup']">

<sch:assert test="count(../../hed:subElements/hed:simpleAction/hed:behaviors/hed:behavior[@xsi:type='RequiredBehavior'])=0">

Sub-elements of an action group with Group Organization Behavior of SentenceGroup MUST NOT specify Required Behavior

</sch:assert>

</sch:rule>

</sch:pattern>

Constraint BHV-6: An actionGroup element with a behavior of type GroupOrganizationBehavior and value of SentenceGroup and also having a behavior of type GroupSelectionBehavior MUST specify the value of the latter as AtMostOne or ExactlyOne.

<sch:pattern name="BHV-6: An action group with Group Organization Behavior of SentenceGroup and a behavior of type GroupSelectionBehavior MUST specify the value of the latter as AtMostOne or ExactlyOne">

<sch:rule context="//hed:actionGroup/hed:behaviors/hed:behavior[@xsi:type='GroupOrganizationBehavior' and hed:value='SentenceGroup']">

<sch:assert test="../hed:behavior[@xsi:type='GroupSelectionBehavior' and (hed:value='AtMostOne' or hed:value='ExactlyOne')]">

An action group with Group Organization Behavior of SentenceGroup MUST specify a GroupSelectionBehavior of AtMostOne or ExactlyOne

</sch:assert>

</sch:rule>

</sch:pattern>

Constraint BHV-7: The subElement of an actionGroup element with a behavior of type GroupOrganizatioBehavior and value of SentenceGroup MUST only contain simpleAction elements.

<sch:pattern name="BHV-7: Sub-elements of an action group with Group Organization Behavior of SentenceGroup MUST be simple actions only">

<sch:rule context="//hed:actionGroup/hed:behaviors/hed:behavior[@xsi:type='GroupOrganizationBehavior' and hed:value='SentenceGroup']">

<sch:assert test="count(../../hed:subElements/hed:simpleAction) = count(../../hed:subElements/\*)">

Sub-elements of an action group with Group Organization Behavior of SentenceGroup MUST be simple actions only

</sch:assert>

</sch:rule>

</sch:pattern>

Constraint BHV-8: The subElement of an actionGroup element with a behavior of type GroupSelectionBehavior and value of AllOrNone or ExactlyOne or AtMostOne MUST NOT contain any actions or groups in its subElements that have a behavior of type RequiredBehavior and value of Must.

<sch:pattern name="BHV-8: Group Selection Behavior and sub-elements whose Required Behavior is Must">

<sch:rule context="//hed:behavior[@xsi:type='GroupSelectionBehavior' and (hed:value='AllOrNone' or hed:value='AtMostOne' or hed:value='ExactlyOne')]">

<sch:assert test="count(../../hed:subElements/\*/hed:behaviors/hed:behavior[@xsi:type='RequiredBehavior' and hed:value='Must'])=0">

An action group with Group Selection Behavior of AllOrNone, ExactlyOne, AtMostOne MUST NOT have any sub-elements whose Required Behavior is Must

</sch:assert>

</sch:rule>

</sch:pattern>

The RequiredBehavior is typically used at the action level, but may be used at the group level. RequiredBehavior specifies whether a given item or a group of items is optional, must be executed, or must be executed unless documentation is provided saying why it was not ordered. Such behavior assumes that a whole group of actions may be chosen as a single unit. An [example of RequiredBehavior](#RequiredBehavior_HeartFailure) is shown in Appendix E, HeartFailureAdmissionToMedSurgOrderSet.

The PrecheckBehavior type is used exclusively with Actions. This type of behavior indicates if an action should be checked or selection for execution when presented to the CDS recipient. This is a means for the CDS system to make it convenient for the user to execute that action or order, by saving the user the effort required to selection that action. This approach is used typically in order sets for orders that are either placed commonly or that the organization prefers (such as those that might be based on evidence). An [example of PrecheckBehavior](#PrecheckBehavior_HeartFailure) is shown in Appendix E, HeartFailureAdmissionToMedSurgOrderSet.

Constraint BHV-3: A behavior of type PrecheckBehavior MUST be specified as a behavior of a simpleAction element only. It MUST NOT be used with an actionGroup element.

<sch:pattern name="BHV-3: PrecheckBehavior can only be used with actions">

<sch:rule context="//hed:behavior[@xsi:type='PrecheckBehavior']">

<sch:assert test="name(../..)='simpleAction'">

PrecheckBehavior MUST be specified under actions only

</sch:assert>

</sch:rule>

</sch:pattern>

## Expressions

The HeD Schema expression component allows clinical decision support logic to be represented at various points within the artifact. The intent of this component is to ensure that expression logic within an artifact can be unambiguously specified so that it can be shared effectively.

Toward this end, the expressions within an HeD Schema artifact are represented at the syntax tree level. This form simplifies language processing tasks such as semantic verification and translation, while also allowing a human readable format that removes any potential ambiguity that would be introduced by a more prose-like representation.

This section provides several specific examples of logical expressions within HeD example artifacts to help guide content authors and implementers in expressing and understanding clinical decision support logic.

### Basic Values (Scalars)

Most expressions will at some point involve basic values such as integers and strings. For example, comparisons of patient age, or encounter time. These values are referred to as literals when they appear within an HeD expression, and there are several expression types defined that allow them to be represented.

The simplest expression for literals within HeD is the *Literal* expression. This expression can be used to select values of all the basic (scalar) types. For example:

<expression xsi:type="Literal" valueType="xsi:int" value="6"/>

<expression xsi:type="Literal" valueType="xsi:decimal" value="8.2"/>

<!-- NOTE: The date format for this literal is defined by the XSD standard -->

<expression xsi:type="Literal" valueType="xsi:date" value="2010-10-10"/>

<expression xsi:type="Literal" valueType="xsi:string" value="Patient is on antithrombotic."/>

In addition to the generic literal, the HeD expression schema defines literal expressions specifically for each supported type, including the most common ISO 21090 data types. For example:

<expression xsi:type="IntegerLiteral" value="20"/>

<expression xsi:type="StringLiteral" value="Patient is less than 20 years old."/>

<!-- NOTE: The date format for this literal is defined by the ISO-21090 standard -->

<expression xsi:type="TimestampIntervalLiteral" low="20120101" high="20121231"/>

<expression xsi:type="PhysicalQuantityLiteral" value="12" unit="mg"/>

<expression xsi:type="CodeLiteral"

code="G8697"

codeSystem="2.16.840.1.113883.6.12"

codeSystemName="CPT-4"

displayName="" />

The following table gives a complete listing of the Literal expressions available within the HeD Specification:

TODO: Link to reference:

|  |  |
| --- | --- |
| Expression | Description |
| Literal | Generic literal for scalar types. |
| AddressLiteral | Returns a value of type AD with the given attributes. |
| BooleanLiteral | Returns a value of type BL with the given attributes. |
| CodeLiteral | Returns a value of type CD with the given attributes. |
| CodedOrdinalLiteral | Returns a value of type CO with the given attributes. |
| SimpleCodeLiteral | Returns a value of type CS with the given attributes. |
| EntityNameLiteral | Returns a value of type EN with the given attributes. |
| IdentifierLiteral | Returns a value of type II with the given attributes. |
| IntegerLiteral | Returns a value of type INT with the given attributes. |
| IntegerIntervalLiteral | Returns a value of type IVL\_INT with the given attributes. |
| PhysicalQuantityIntervalLiteral | Returns a value of type IVL\_PQ with the given attributes. |
| RatioIntervalLiteral | Returns a value of type IVL\_RTO with the given attributes. |
| RealIntervalLiteral | Returns a value of type IVL\_REAL with the given attributes. |
| TimestampIntervalLiteral | Returns a value of type IVL\_TS with the given attributes. |
| PhysicalQuantityLiteral | Returns a value of type PQ with the given attributes. |
| RealLiteral | Returns a value of type Real with the given attributes. |
| RatioLiteral | Returns a value of type RTO with the given attributes. |
| StringLiteral | Returns a value of type ST with the given attributes. |
| UrlLiteral | Returns a value of type TEL with the given attributes. |
| TimestampLiteral | Returns a value of type TS with the given attributes. |
| PeriodLiteral | Returns a value of type PIVL\_TS with the given attributes. |

### Comparison Operators

The HeD Schema expression language defines a standard set of comparison operators for use with scalar values. Each comparison operator takes two arguments of the same type, and returns a boolean indicating the result of the comparison. Note that for comparison operators, if either or both operands evaluate to null, the result of the comparison is *unknown*, not false.

The following examples illustrate the use of comparison operators:

<condition xsi:type="Equal">

<operand xsi:type="Property" path="Status"/>

<operand xsi:type="Literal" valueType="xsi:string" value="Active"/>

</condition>

<when xsi:type="LessOrEqual">

<operand xsi:type="Property" path="demographics.age"/>

<operand xsi:type="IntegerLiteral" value="20"/>

</when>

The following table lists the comparison operators available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| Equal | Returns true if the operands are equal. |
| NotEqual | Returns true if the operands are not equal. |
| Less | Returns true if the first operand is less than the second operand. |
| LessOrEqual | Returns true if the first operand is less than or equal to the second operand. |
| Greater | Returns true if the first operand is greater than the second operand. |
| GreaterOrEqual | Returns true if the first operand is greater than or equal to the second operand. |

### Logical Operators

The HeD Schema defines logical operators that can be used to combine the results of logical expressions. And and Or can be used to combine any number of results, and Not can be used to invert the result of any expression.

Note that these operators are defined with 3-valued logic semantics, allowing the operators to deal consistently with missing information.

For example, in the following expression:

<condition xsi:type="And">

<operand xsi:type="LessOrEqual">

<operand xsi:type="Property" path="demographics.age"/>

<operand xsi:type="IntegerLiteral" value="20"/>

</operand>

<operand xsi:type="Equal">

<operand xsi:type="Property" path="status"/>

<operand xsi:type="Literal" valueType="xsi:string" value="Active"/>

</operand>

</condition>

If the patient’s age is not known, but the status is present and not equal to Active, the result of the And operation will be *false*. In contrast, if the patient’s age is not known, but the status *is* equal to Active, the result of the And operation is *unknown*. Only in the case where the patient’s Age is known to be less than or equal to 20, and the status is known to be Active will the And operation evaluate to *true*.

The following table lists the logical operators available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| And | Returns the logical conjunction of its operands. |
| Or | Returns the logical disjunction of its operands. |
| Not | Returns the logical negation of its operand. |

### Nullological Operators

The HeD Schema expression language defines several nullological operators that are useful for dealing with potential missing information. These are *IsNull*, *IfNull*, and *Coalesce*.

The following table lists the logical operators available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| IsNull | Returns true if the argument is *null*, false otherwise. |
| IfNull | Returns the first argument if it is not null, otherwise, returns the second argument. |
| Coalesce | Returns the first non-null argument, null if there are no non-null arguments. |

### Conditional Operators

The HeD Schema expression language defines several conditional expressions that can be used to return different values based on a condition, or set of conditions. These are the *Conditional* expression, and the *Case* expression.

The conditional expression allows a simple condition to be used to decide between one expression or another. For example:

<expression xsi:type="Conditional">

<condition xsi:type="LessOrEqual">

<operand xsi:type="Property" path="demographics.age"/>

<operand xsi:type="IntegerLiteral" value="20"/>

</condition>

<then xsi:type="StringLiteral" value="Patient is 20 years old or less."/>

<else xsi:type="StringLiteral" value="Patient is over 20 years old."/>

</expression>

The above expression will evaluate to the string *“Patient is 20 years old or less.”* if the patient age property is less than or equal to 20. Otherwise, the expression will evaluate to *“Patient is over 20 years old.”*

The case expression has two varieties, one that is equivalent to repeated conditionals, and one that allows a specific comparand to be identified and compared with each item to determine a result.

The following example illustrates the multi-condition variety:

<expression xsi:type="Case">

<caseItem>

<when xsi:type="LessOrEqual">

<operand xsi:type="Property" path="demographics.age"/>

<operand xsi:type="IntegerLiteral" value="20"/>

</when>

<then xsi:type="StringLiteral" value="Patient is 20 years old or less."/>

</caseItem>

<caseItem>

<when xsi:type="LessOrEqual">

<operand xsi:type="Property" path="demographics.age"/>

<operand xsi:type="IntegerLiteral" value="40"/>

</when>

<then xsi:type="StringLiteral" value="Patient is over 20, but not more than 40 years old."/>

</caseItem>

<else xsi:type="StringLiteral" value="Patient is over 40 years old."/>

</expression>

The following example illustrates a case expression using a comparand:

<expression xsi:type="Case">

<comparand xsi:type="Property" path="demographics.age"/>

<caseItem>

<when xsi:type="IntegerLiteral" value="10"/>

<then xsi:type="StringLiteral" value="Patient is 10 years old."/>

</caseItem>

<caseItem>

<when xsi:type="IntegerLiteral" value="20"/>

<then xsi:type="StringLiteral" value="Patient is 20."/>

</caseItem>

<else xsi:type="StringLiteral" value="Patient is neither 10 nor 20 years old."/>

</expression>

The following table lists the conditional operators available in the HeD Schema expression language:

TODO: Link to reference

| Expression | Description |
| --- | --- |
| Conditional | Allows for conditional evaluation between two expressions. |
| Case | Allows for multiple conditional expressions, or a comparand with multiple cases. |

### Arithmetic Operators

The HeD Schema expression language provides a complete set of arithmetic operators to allow for manipulation of integer and real values within artifacts. In general, these operators have the expected semantics for arithmetic operators.

Note that if an operand evaluates to null, the result of the operation is defined to be null. This provides consistent semantics when dealing with missing information.

The following examples illustrate the use of some common arithmetic operators:

<expression xsi:type="Add">

<operand xsi:type="IntegerLiteral" value="2"/>

<operand xsi:type="IntegerLiteral" value="2"/>

</expression>

<expression xsi:type="Multiply">

<operand xsi:type="IntegerLiteral" value="6"/>

<operand xsi:type="IntegerLiteral" value="9"/>

</expression>

<expression xsi:type="TruncatedDivide">

<operand xsi:type="IntegerLiteral" value="63"/>

<operand xsi:type="IntegerLiteral" value="2"/>

</expression>

The following table lists the arithmetic operators available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| Add | Performs numeric addition of its arguments. |
| Subtract | Performs numeric subtraction of its arguments. |
| Multiply | Performs numeric multiplication of its arguments. |
| Divide | Performs numeric division of its arguments. |
| TruncatedDivide | Performs integer division of its arguments. |
| Modulo | Computes the remainder of the division of its arguments. |
| Ceiling | Returns the first integer greater than or equal to its argument. |
| Floor | Returns the first integer less than or equal to its argument. |
| Abs | Returns the absolute value of its argument. |
| Negate | Returns the negative value of its argument. |
| Round | Returns the nearest numeric value to its argument, optionally specified to a number of decimal places for rounding. |
| Ln | Computes the natural logarithm of its argument. |
| Log | Computes the logarithm of its first argument, using the second argument as the base. |
| Power | Raises the first argument to the power given by the second argument. |
| Succ | Returns the successor of its argument. |
| Pred | Returns the predecessor of its argument. |

### String Operators

The HeD Schema expression language defines a set of string operators to allow for manipulation of string values within artifact definitions.

Indexes within strings are defined to be 1-based.

Note that except as noted within the documentation for each operator, if any argument evaluates to null, the result of the operation is also defined to be null.

The following examples illustrate some common string manipulation operators:

<expression xsi:type="Pos">

<pattern xsi:type="StringLiteral" value="abc"/>

<string xsi:type="StringLiteral" value="abcdefg"/>

</expression>

<expression xsi:type="Substring">

<stringToSub xsi:type="StringLiteral" value="abcdefg"/>

<startIndex xsi:type="IntegerLiteral" value="1"/>

<length xsi:type="IntegerLiteral" value="3"/>

</expression>

<expression xsi:type="Indexer">

<operand xsi:type="StringLiteral" value="abcdefg"></operand>

<index xsi:type="IntegerLiteral" value="1"/>

</expression>

The following table lists the string operators available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| Concat | Returns the concatenation of its arguments. |
| Combine | Combines a list of strings, optionally separating them with the given separator. |
| Split | Splits a string into a list of strings along a given separator. |
| Length | Returns the length of its argument. |
| Upper | Returns the upper case representation of its argument. |
| Lower | Returns the lower case representation of its argument. |
| Indexer | Returns the nth element of its argument. |
| Pos | Returns the starting position of a given pattern within a string. |
| Substring | Returns a substring of its argument. |

### Date and Time Operators

The HeD Schema expression language defines several operators for manipulating date and time values within HeD artifacts. These operators are defined using a common granularity type that allows the various granularities (e.g. day, month, week, hour, minute, second) of time to be manipulated.

Except as noted within the documentation for each operator, if any argument evaluates to null, the result of the operation is also defined to be null.

The following example constructs an interval of dates beginning 6 months before today, and ending today:

<dateRange xsi:type="Interval">

<begin xsi:type="DateAdd">

<date xsi:type="Today"/>

<granularity xsi:type="Literal" valueType="DateGranularity" value="Month"/>

<numberOfPeriods xsi:type="Literal" valueType="xsi:int" value="-6"/>

</begin>

<end xsi:type="Today"/>

</dateRange>

The following table lists the date and time operators available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| DateAdd | Adds a given number of periods to its argument. |
| DateDiff | Computes the number of periods between a starting and ending date. |
| DatePart | Returns a specified component of its argument. |
| Today | Returns the date (with no time component) of the start timestamp associated with the evaluation request. |
| Now | Returns the date and time of the start timestamp associated with the evaluation request. |
| Date | Constructs a date from its arguments. |
| Indexer | Returns the nth element of its argument. |
| Pos | Returns the starting position of a given pattern within a string. |
| Substring | Returns a substring of its argument. |

### List Values

The HeD Schema expression language allows for the expression and manipulation of lists of values of any type. The most basic list operation is the list selector:

<source xsi:type="List">

<element xsi:type="Value" valueType="xsi:int" value="1"/>

<element xsi:type="Value" valueType="xsi:int" value="2"/>

<element xsi:type="Value" valueType="xsi:int" value="3"/>

<element xsi:type="Value" valueType="xsi:int" value="4"/>

<element xsi:type="Value" valueType="xsi:int" value="5"/>

</source>

The above expression creates a list of integers with the elements *1, 2, 3, 4* and *5*.

TODO: Examples of union, intersect, difference, filter, foreach, isempty, and current.

The following table lists the list operators available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| List | Constructs a list from its arguments. |
| IsEmpty | Returns true if its argument contains no elements. |
| IsNotEmpty | Returns true if its argument contains any elements. |
| Equal | Returns true if its arguments have the same number of elements, and for each element considered in order, the elements are equal. |
| NotEqual | Returns true if its arguments are not equal. |
| Union | Returns a list containing all the elements of its arguments. |
| Difference | Returns a list containing only the elements in the first list that are not in the second list. |
| Intersect | Returns a list containing only the elements that are in all of its arguments. |
| Filter | Returns a list containing only the elements for which the given condition evaluates to true. |
| IndexOf | Returns the 1-based index of an element within the list, or 0 if the element is not present. |
| In | Returns true if the given element is in a given list. |
| Contains | Returns true if the given list contains a given element. |
| ProperIn | Returns true if every element in the first argument is in the second argument, and the second argument is strictly larger than the first. |
| ProperContains | Returns true if the first argument contains every element in the second argument, and the first argument is strictly larger than the second. |
| Sort | Returns a list with the same elements, sorted by the given sort criteria. |
| ForEach | Returns a list whose elements are determined by evaluated an expression for each element in its argument. |
| Distinct | Returns a list that contains the unique elements within its argument. |
| Current | Returns the contents of the current scope. |

### Aggregate Operators

For computing aggregate quantities, the HeD Schema expression language defines several aggregate operators. These operators perform computations on lists of values, either on the elements of the list directly, or on a specific property of each element in the list.

For example, the following invocation computes the sum of a list of integers:

<expression xsi:type="Sum">

<source xsi:type="List">

<element xsi:type="Value" valueType="xsi:int" value="1"/>

<element xsi:type="Value" valueType="xsi:int" value="2"/>

<element xsi:type="Value" valueType="xsi:int" value="3"/>

<element xsi:type="Value" valueType="xsi:int" value="4"/>

<element xsi:type="Value" valueType="xsi:int" value="5"/>

</source>

</expression>

Whereas the following example computes the sum of the *dose* property of the elements in the list returned by evaluated the *Medications* expression:

<expression xsi:type="Sum" path="Dose">

<source xsi:type="ExpressionRef" name="Medications"/>

</expression>

As another example, the following expression computes the average of 10 times the dose of the medications in the given list:

<expression xsi:type="Avg">

<source xsi:type="ForEach">

<source xsi:type="ExpressionRef" name="Medications"/>

<element xsi:type="Multiply">

<operand xsi:type="Property" path="dose"/>

<operand xsi:type="RealLiteral" value="10.0"/>

</element>

</source>

</expression>

Unless noted in the documentation for each operator, aggregate operators deal with missing information by excluding elements which have no value before performing the aggregation. In addition, an aggregate operation performed over an empty list is defined to return null, except as noted in the documentation for each operator (e.g. Count).

The following table lists the aggregate operators available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| Count | Returns the number of non-null elements in the source. |
| Sum | Computes the sum of non-null elements in the source. |
| Min | Returns the minimum element in the source. |
| Max | Returns the max element in the source. |
| Avg | Returns the average of the elements in the source. |
| AllTrue | Returns true if all the elements in source are true. |
| AnyTrue | Returns true if any element in source is true. |

### Interval Values

The HeD Schema expression language defines a complete set of operators for use in defining and manipulating interval values.

Constructing an interval is performed with the *Interval* expression:

<expression xsi:type="Interval" endOpen="true">

<begin xsi:type="Literal" valueType="xsi:Date" value="2010-10-10"/>

<end xsi:type="Literal" valueType="xsi:Date" value="2010-10-11"/>

</expression>

This expression returns an interval from October 10th, 2010, inclusive, to October 11th, 2010, exclusive.

The following table lists the interval operators available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| Equal | Returns true if the arguments are the same interval. |
| NotEqual | Returns true if the arguments are not the same interval. |
| Contains | Returns true if the interval contains the given point or interval. |
| ProperContains | Returns true if the first interval contains the second interval, and the first interval is strictly larger than the second. |
| Within | Returns true if the given element or interval is within an interval. |
| ProperIn | Returns true if the first interval is entirely contained within the second interval, and the first interval is strictly smaller than the second. |
| Before | Returns true if the first interval ends before the second one starts. |
| After | Returns true if the first interval starts after the second one ends. |
| Meets | Returns true if the first interval ends on the starting point of the second, or if the first interval starts on the ending point of the second. |
| Overlaps | Returns true if the first interval overlaps the second. |
| Union | Returns the interval that results from combining the arguments. |
| Difference | Returns the interval that results from subtracting the second interval from the first. |
| Length | Returns the length of the interval. |
| Begin | Returns the starting point of the interval. |
| End | Returns the ending point of the interval. |

### Structured Values

Structured values in the HeD Schema expression language are values with sets of named properties, each of which may have a value of any type. Structured values are most commonly used to represent clinical information such as encounters, problems, and procedures.

There are several operators that provide the ability to construct and manipulate structured values. The following example illustrates the creation of a simple structured value:

<actionSentence xsi:type="ObjectExpression"

objectType="vmr:SubstanceAdministrationProposal">

<description>Prescribe aspirin or other antithrombotic</description>

<property name="substance.substanceCode">

<value xsi:type="CodeLiteral"

code="2.16.840.1.113883.3.464.1003.196.12.1211"

codeSystem="National Committee for Quality Assurance"

displayName="Select a medication from this value set." />

</property>

</actionSentence>

In addition to constructing a value, a new structured value may be created by modifying the properties of an existing value:

<value xsi:type="ObjectRedefine">

<source xsi:type="ExpressionRef" name="PRNReasonRelatedObservation"/>

<property name="observationValue.physicalQuantity" xsi:type="PropertyExpression">

<value xsi:type="QuantityIntervalLiteral" lowIsInclusive="true" highIsInclusive="true">

<low xsi:type="PQ" value="200" unit="Milligram/Deciliter"/>

<high xsi:type="PQ" value="249" unit="Miligram/Deciliter"/>

</value>

</property>

</value>

The example above creates a new observation result based on the value of the *PRNReasonRelatedObservation* but with a new value for the *physicalQuantity* property. Note that this is also an expression evaluation, so there is no change in the value returned by *PRNReasonRelatedObservation*; subsequent evaluations of that expression will return the same original value.

To access properties of a structured value, use the *Property* expression. A property expression has a *path* attribute, and an optional *source* element, and a *value* element. The source element returns the structured value to be accessed. In some contexts, such as within a *Filter* expression, the source is implicit. If used outside such a context, a source must be provided.

The path attribute specifies a property path relative to that structured value. The property expression returns the value of the property specified by the property path. Property paths are allowed to include qualifiers to indicate that subproperties should be traversed. For example:

<expression xsi:type="Property" path="demographics.age.value">

<source xsi:type="ExpressionRef" name="Patient" />

</expression>

The above property expression accesses the *value* property of the *age* property of the *demographics* property of the structured value returned by the *Patient* expression.

The following table lists the structured value operators available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| ObjectExpression | Constructs a new structured value. |
| ObjectRedefine | Constructs a new structured value based on the values of an existing structured value, with specific properties given new values in the result. |
| Property | Returns the value of a specific property of a structured value. |

### Reusing Expressions

The HeD Schema expression language provides a mechanism for reusing expressions by declaring a named expression. This construct is similar to a function call with no parameters in a traditional imperative language.

The *ExpressionDef* type is used to define a named expression that can then be referenced within any HeD expression within the artifact:

<def name="PatientAge">

<expression xsi:type="Property" path="demographics.age.value">

<source xsi:type="ExpressionRef" name="Patient" />

</expression>

</def>

This example establishes the named expression *PatientAge*, which results in the age value of the patient; itself the result of evaluating the named expression *Patient*.

Note that circular expression references are not allowed, but that named expressions can be defined in any order, so long as the actual references do not result in a cycle.

The following table lists the expression definition components available in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| ExpressionDef | Defines a named expression that can be referenced by other expressions. |
| ExpressionRef | Returns the result of evaluating a named expression. |

### External Data

All access to external data within the HeD Schema expression language is performed through the use of *Request* expressions.

The base request expression defines the cardinality and data type of the request. These attributes determine the type of elements to be returned, as well as the whether the result will be a single value, or a list of values.

The type of the elements to be returned is specified with the *dataType* attribute of the *ClinicalRequest*, and must refer to the name of a type within a known data model specified in the *dataModels* element of the artifact metadata. For more information on specifying the data model, please refer to the Clinical Data Models section of the Metadata discussion in Section 4.1.1.

Note that *RequestBase* is an abstract type. This is intended to allow different types of requests to be introduced through extension. However, for the purposes of this implementation guide, the *ClinicalRequest* is the only request type that is defined. Furthermore, the HeD Schema places restrictions on where request expressions are allowed to appear within an artifact to improve readability of artifact definitions. The e*xternalData* element allows a list of *ExpressionDef* elements, each of which must contain one and only one *ClinicalRequest*. In addition, clinical requests may only appear within this external data element of the artifact. Further manipulation of these results must be performed elsewhere in the artifact, either in the *expressions* element (which allows any number of additional expressions to be defined), or directly within the logic of the artifact as appropriate.

In addition to the basic attributes defined on the base request, the *ClinicalRequest* introduces the ability to specify optional criteria for the request. The available criteria are intentionally restricted to the set of codes involved, and the date range involved. If these criteria are omitted, the request is interpreted to mean all data of that type.

NOTE: There is an implicit patient context assumed within HeD artifacts. Among other things, this implies that the relationships between clinical data (such as the patient and their associated encounters) are supplied by the implementation environment. This is an intentional simplifying assumption to avoid having to define those relationships explicitly within the artifact.

The following example illustrates a simple singleton request with no criteria:

<def name="Patient">

<expression xsi:type="ClinicalRequest" cardinality="Single"

dataType="vmr:EvaluatedPerson" isInitial="true" />

</def>

This definition (which must appear in the *externalData* section of the artifact) establishes the named expression *Patient* to return the singleton value of type *EvaluatedPerson*. Throughout the artifact, the patient information can now be accessed by referencing this expression. For example, the following definition (which must appear in the *expressions* section of the artifact) establishes the name *PatientAge* to refer specifically to the age property of the patient:

<def name="PatientAge">

<expression xsi:type="Property" path="demographics.age.value">

<source xsi:type="ExpressionRef" name="Patient" />

</expression>

</def>

The following example illustrates a simple multiple cardinality request:

<def name="antithromboticNotPrescribedForDocumentedReason">

<expression xsi:type="ClinicalRequest" cardinality="Multiple"

dataType="vmr:ObservationResult" codeProperty="observationFocus.code"

dateProperty="observationEventTime">

<description>Patient reason or other reason for not prescribing an antithrombotic</description>

<codes xsi:type="List">

<element xsi:type="CodeLiteral" code="G8697"

codeSystem="2.16.840.1.113883.6.12" codeSystemName="CPT-4"

displayName="" />

</codes>

</expression>

</def>

The above example defines *antithromboticNotPrescribedForDocumentedReason* to refer to all clinical data elements for the patient that are of type *ObservationResult* and that have an *observationFocus* of CPT-4:G8697. Because there is no date range criteria, this observation can be present at any time in the patient’s record.

As a final example, the following definition establishes *onAntiThrombotic* to refer to all clinical data elements for the patient that are of type *SubstanceAdministrationEvent*, had a substance code within the given value set, and that were administered within the past year.

<def name="onAntiThrombotic">

<expression xsi:type="ClinicalRequest" cardinality="Multiple"

dataType="vmr:SubstanceAdministrationEvent"

codeProperty="substanceAdministrationGeneralPurpose.code"

dateProperty="administrationTimeInterval.low" useValueSets="true">

<description>Patient prescribed antithrombotic within the past year</description>

<codes xsi:type="ValueSet" id="2.16.840.1.113883.3.464.1003.196.12.1211"

authority="National Committee for Quality Assurance" />

<dateRange xsi:type="Interval">

<begin xsi:type="DateAdd">

<date xsi:type="Today" />

<granularity xsi:type="Literal" valueType="DateGranularity"

value="Month" />

<numberOfPeriods xsi:type="Literal" valueType="xsi:int"

value="-12" />

</begin>

<end xsi:type="Today" />

</dateRange>

</expression>

</def>

In addition to specifying external data, the HeD Schema expression language defines several operators for referencing and working with terminology sets.

The following table lists the expressions relevant to defining external data and value sets in the HeD Schema expression language:

TODO: Link to reference

|  |  |
| --- | --- |
| Expression | Description |
| ClinicalRequest | Defines clinical data that will be used within the artifact. |
| ValueSet | Returns the list of codes for a value set. |
| InValueSet | Tests a code for membership in a value set. |
| Subsumes | Tests two codes for subsumption (i.e. whether the concept represented by one code is subsumed by the concept represented by another). |
| SetSubsumes | Computes the intersection of two sets of codes using the subsumption relationship. |

### Parameters

In addition to external data, the HeD Schema expression language provides a mechanism for defining parameters to an artifact. An artifact can define any number of parameters, each of which has a name, and a defined type, as well as an optional default value.

Parameter values, if any, are expected to be provided as part of the evaluation request, and can be accessed with a *ParameterRef* expression in any expression throughout the artifact.

The following example illustrates a parameter definition:

<parameterDef name="MonthsThreshold" xsi:type="xsi:int">

<default xsi:type="IntegerLiteral" value="6"/>

</parameterDef>

And this example illustrates the use of this parameter within an external data definition:

<expressionDef name="DiabetesDiagnoses">

<!-- Get Diabetes diagnoses within the last @MonthsThreshold months -->

<expression xsi:type="ClinicalRequest" dataType="vmr:Problem" cardinality="Multiple"

isInitial="true">

<codes xsi:type="ValueSet" id="DiabetesDiagnosisCodes"/>

<dateRange xsi:type="Interval">

<begin xsi:type="DateAdd">

<date xsi:type="Today"/>

<granularity xsi:type="Literal"

valueType="DateGranularity" value="Month"/>

<numberOfPeriods xsi:type="Negate">

<operand xsi:type="ParameterRef" name="MonthsThreshold"/>

</numberOfPeriods>

</begin>

<end xsi:type="Today"/>

</dateRange>

</expression>

</expressionDef>

The following table lists the expressions relevant to parameters in the HeD Schema expression language:

TODO: Link to reference

| Expression | Description |
| --- | --- |
| ParameterDef | Defines a parameter to the artifact. |
| ParameterRef | Returns the value of a parameter. |

# cds knowledge artifact components

I HAVE STARTED A NEW CHAPTER ABOVE RATHER THAN CREATING EDITING MARKS HERE.

Please refer to the supplemental materials submitted as part of the ballot; XML examples for each of the three CDS Knowledge Artifact types and related components are included therein.

## Knowledge Artifact Metadata

The current metadata model is intended to represent key knowledge artifact data independent of the containing components. The knowledge artifact metadata in this model describes the contents of the CDS Knowledge Artifact without specifically constraining this content.

To support the use of knowledge artifact metadata, the harmonized CDS Knowledge Artifact schema supports a complex type called **<Metadata>** which defines all the attributes for the knowledge artifact metadata. The table below shows how metadata (as specified in the HeD Artifact Sharing Use Case) is aligned to the HeD Schema:

| HeD Data Element | XPath | Optionality | Cardinality | Datatype |
| --- | --- | --- | --- | --- |
| Artifact Title | Title | Required | 1..\* |  |
| Artifact Description | Description | Optional | 0..1 |  |
| Artifact Identifier | Identifier | Required | 1..\* | II |
| Artifact Contributor | Contributor | Optional | 0..\* | Contribution |
| Related Resource (e.g. eMeasure reference, clinical quality measure reference, dependent artifacts, other versions) | KnowledgeResource | Required | 1..\* |  |
| Supporting Evidence | supportingEvidence | Optional | 0..\* | Evidence |
| Artifact Documentation | Documentation | Optional | 0..1 | ED |
| Publisher Name | Publisher | Optional | 0..\* | Organization |
| Publisher Contact Information | Publisher | Optional | 0..\* | Organization |
| Licensing, Usage, Restriction | usageTerms | Optional | 0..\* |  |
| Schema Version | schemaVersion | Required | 1..1 | INT |
| Schema Identifier | schemaIdentifier | Required | 1..1 | INT |
| Artifact Status | status | Required | 1..1 | ArtifactStatusType |
| Artifact History | eventHistory |  |  | ArtifactEvent |
| Category | category | Optional | 0..\* | CD |
| Key Terms | keyTerms | Optional | 0..\* | CD |

Table 18 - HeD Knowledge Artifact Metadata – Overview

For each CDS Knowledge Artifact type, a general set of constraints for the Knowledge Artifact metadata, will apply.

1. A CDS Knowledge Artifact **SHALL** contain a single (1) title element
2. A CDS Knowledge Artifact **SHALL** contain a single (1) description element
3. A CDS Knowledge Artifact **SHALL** contain one or more (1..\*) identifier elements that includes the VersionedIdentifier element.
4. A CDS Knowledge Artifact **MAY** contain zero or more documentation elements
5. A CDS Knowledge Artifact **SHALL** contain a single (1) schema version element
6. A CDS Knowledge Artifact **SHALL** contain a single (1) schema identifier element
7. A CDS Knowledge Artifact **SHALL** contain one or more (1..\*) publishers, which MAY be represented by a person or organization
8. A CDS Knowledge Artifact **SHALL** contain a single (1) status code.
9. A CDS Knowledge Artifact **SHOULD** use the HeDArtifactStatus value set
10. A CDS Knowledge Artifact **MAY** stipulate the use of MESH for specifying key terms to enable the search of knowledge artifacts.

## Actions

The action groups are the containers and organizers of the order set. An action group may constitute a visual section of the order set (e.g., Medications section or Diets section) or a non-visual section (a set of orders which must be ordered all together or not at all). Sections may contain subsections in the form of action groups or group reference sub-elements. The title of a visual group is the title or the heading of the section.

Group references enable the construction of modular, reusable order sets. A reusable group should be defined as a self-contained order set artifact in its own Knowledge Document. This group can then be used in another order set by using the group reference element and specifying the identifier of the order set. For example, a DVT Prophylaxis group may be defined as an order set artifact. This order set artifact may be included in Order Sets for clinical scenarios which call for the use of DVT prophylaxis.

Actions constitute the basic building blocks of CDS Knowledge Artifacts. Actions may have conditional expressions associated with them such that the action only appears in the CDS Knowledge Artifact if a given condition is true. Because the CDS Knowledge Artifact schema does not depend directly on the vMR (instead the dependency is indirect), actions are structured as expressions which specify vMR objects.

Actions may have conditional expressions associated with them such that the action only appears in the Order Set if a given condition is true. Because the CDS Knowledge Artifact schema this implementation guide specifically refers to does not depend directly on the vMR (instead the dependency is indirect), actions are structured as expressions that specify vMR objects. For example, if an order for *captopril 6.25 milligram orally 3 times a day is* present in an order set, a direct dependence on the vMR would produce the following XML fragment:

<SubstanceAdministrationProposal>

<id root=”SubAdminProp1”/>

<substanceCode code=”1998” codeSystem="2.16.840.1.113883.6.88" codeSystemName="RxNorm" displayName="captopril"/>

<doseQuantity lowValue=”6.25” lowUnit=”mg” highValue=”6.25” highUnit=”mg” lowIsInclusive=true highIsInclusive=true/>

<deliveryRoute code=”TBD” codeSystem="2.16.840.1.113883.6.88" codeSystemName="RxNorm" displayName="orally"/>

<dosingPeriod lowValue=”8” lowUnit=”h” highValue=”8” highUnit=”h” lowIsInclusive=true highIsInclusive=true/>

<dosingPeriodIntervalIsImportant value=”false”/>

</SubstanceAdministrationProposal>

However, in the CDS Knowledge Artifact schema, expressions are used to specify a vMR object thus:



The reason for the indirect dependence on vMR is to allow the HeD schema to be used in other contexts in which a different clinical data elements model would be appropriate.

Group references enable the construction of modular, reusable mini order sets. A reusable group should be defined as a self-contained artifact in its own Knowledge Document. This group can then be used in the actions section of a rule by using the group reference element and specifying the identifier of the artifact. For example, a DVT Prophylaxis group may be defined as an order set artifact. This order set artifact may be included in Rules which identify a patient at high risk for DVT.

To create an Action, specific elements may be used (which are documented in detail in Section 5.4 – Working with Actions - of this implementation guide)

1. An action **SHALL** contain 1 ActionMode element
2. An action **SHOULD** be constrained to those actions defined in the HeDActionModeType enumeration
3. An action **SHALL** contain 1 or more (1..\*) supporting evidence components
4. An action **SHALL** contain 1 or more (1..\*) supporting reference components
5. An action **SHALL** contain 1 urgency element
6. An action **MAY** contain 1 ActionSentence component
7. An action **MAY** contain a text alternative element instead of an ActionSentence Component
8. Behaviors may be defined at the group and sub-group levels. Such behaviors pertain to the set of actions included in the group / sub-group. There are two types of behaviors which apply to groups: group selection behavior and group organization behavior. Group selection behavior specifies the number of items in the group that may/must be chosen by the CDS System user. Group organization behavior provides hints to the end user system to aid with display. A visual group is meant to be displayed as a group of items, perhaps as a sub-section with a title / description. A logical group is a group of items that are logically related, but don't necessarily need to be separated visually from other items. A sentence group is a group of items which share a common orderable (such as "aspirin"), but differ in the details of their administration (dose level, frequency, route of administration, etc.). The end user system may choose to separate out the common orderable items, and show the order details underneath, or could decide to show each item as a separate orderable in and of itself.

In addition, there are two types of behaviors which are typically used at the action level, but may in fact be used at the group level: RequiredBehavior and SelectionFrequencyBehavior. RequiredBehavior specifies whether a given group of items is optional, must be ordered, or must be ordered unless documentation is provided saying why it wasn’t ordered. Such behavior assumes that a whole group of actions may be chosen as a single unit. Similarly, SelectionFrequencyBehavior also assumes that a whole group of actions may be chosen as a single unit, and gives an indication of whether the group is (or should be) a “MostFrequentlyOrdered” or “LessFrequentlyOrdered” group.

Behaviors may be defined for individual actions, and include two that are also defined for groups of orders. These are RequiredBehavior and SelectionFrequencyBehavior, and these work the same for an individual action as they do for groups.

## Conditions

Conditions are used to support ECA Rules and are defined in Section 5.5 of this implementation guide.

## Supporting Evidence

The attributes associated with an Evidence element are defined in Section 5.9 of this implementation guide.

1. The supportingEvidence element **MUST** be the only element to use the Evidence complex type in a CDS Knowledge Artifact.

## Supporting Reference

The Supporting Reference section of a CDS Knowledge Artifact are defined in Section 5.10 of this implementation guide.

## Clinical Data Mapping

Clinical data mapping occurs through the use of clinical expressions, such as ClinicalRequest, which contains all of the data types needed for specific clinical information in a CDS Knowledge Artifact. Clinical mapping data draws heavily from both the current vMR domain analysis model (specifically vMR extensions) and the CDSC L3 schema.

The vMR (Virtual Medical Record) was the chosen model for clinical data mapping by the HeD schema. Although the vMR is not specifically in scope for this implementation guide, a full version of the vMR data types and domain analysis model (DAM) are included in the HeDSchema.eap file that is part of this ballot package.

## Expression

The Expression element defines a base expression language for use with the CDS Knowledge Artifact. The language defined in the harmonized CDS Knowledge Artifact schema is closely aligned to the HL7 Arden Syntax and seeks to leverage many of its best practices and guidance. An expression within a CDS Knowledge Artifact is modeled to allow for customization using specific operators to compute logic that may be embedded in the artifact.

Each of the Expression types are specifically documented in Section 5.13 of this implementation guide. Additional background information, including the execution model, can be found in the Expressions document via the Google Code Repository: http://code.google.com/p/health-e-decisions/source/browse/branches/v1.0/src/main/docs/Expressions.docx

# hed enumerations and vocabularies

## Datatypes.xsd Namespace Summary

Namespace Summary

|  |
| --- |
| [**urn:hl7-org:v3/cdsdt**](#b5) |
| Targeting Schemas (1):  [datatypes.xsd](#b1291)  Targeting Components:  elements (5 local), [complexTypes](#b2) (25), [simpleTypes](#b3) (17), [attribute groups](#b4) (9) |

|  |
| --- |
| [**urn:hl7-org:v3/hed**](#b249) |
| Targeting Schemas (1):  [knowledgedocument.xsd](#b1292)  Targeting Components:  [elements](#b246) (2 global + 212 local), [complexTypes](#b247) (168), [simpleTypes](#b248) (54) |

|  |  |
| --- | --- |
| Schema Summary | |
| [datatypes.xsd](#b1291) | Introduction ------------ This XML document was originally developed in the course of development of the ISO/HL7 21090 standard (Healthcare Datatypes). |
| [knowledgedocument.xsd](#b1292) | This file allows organizations to extend the enumeration ArtifactLifeCycleEventType. |

Namespace "urn:hl7-org:v3/cdsdt"

Targeting Schemas (1):

[datatypes.xsd](#b1291)

Targeting Components:

elements (5 local), [complexTypes](#b2) (25), [simpleTypes](#b3) (17), [attribute groups](#b4) (9)

## Datatypes.xsd Complex Type Summary

| Complex Type Summary | |
| --- | --- |
| [**AD**](#b12) | Mailing and home or office addresses. |
| [**ADXP**](#b17) | A part with a type-tag signifying its role in the address. |
| [**ANY**](#b19) | Defines the basic properties of every data value. |
| [**BL**](#b24) | BL stands for the values of two-valued logic. |
| [**CD**](#b33) | A CD is a reference to a concept defined in an external code system, terminology, or ontology. |
| [**CO**](#b40) |  |
| [**CS**](#b45) | Coded data in its simplest form, where only the code is not predetermined. |
| [**EN**](#b52) | A name for a person, organization, place or thing. |
| [**ENXP**](#b58) | A part with a type code signifying the role of the part in the whole entity name, and qualifier codes for more detail about the name part type. |
| [**II**](#b64) | An identifier that uniquely identifies a thing or object. |
| [**INT**](#b69) | Integer numbers (-1,0,1,2, 100, 3398129, etc.) are precise numbers that are results of counting and enumerating. |
| [**IVL\_INT**](#b77) | A set of consecutive values of an ordered base datatype. |
| [**IVL\_PQ**](#b83) | A set of consecutive values of an ordered base datatype. |
| [***IVL\_QTY***](#b92) | A set of consecutive values of an ordered base datatype. |
| [**IVL\_REAL**](#b100) | A set of consecutive values of an ordered base datatype. |
| [**IVL\_RTO**](#b106) | A set of consecutive values of an ordered base datatype. |
| [**IVL\_TS**](#b112) | A set of consecutive values of an ordered base datatype. |
| [**PQ**](#b115) | A dimensioned quantity expressing the result of measuring. |
| [***QTY***](#b118) | The quantity datatype is an abstract generalization for all datatypes whose domain values has an order relation (less-or-equal) and where difference is defined in all of the datatype's totally ordered value subsets. |
| [**REAL**](#b123) | Fractional numbers. |
| [**RTO**](#b126) | A quantity constructed as the quotient of a numerator quantity divided by a denominator quantity. |
| [**ST**](#b131) | The character string datatype stands for text data, primarily intended for machine processing (e.g., sorting, querying, indexing, etc.) or direct display. |
| [**TEL**](#b139) | A locatable resource that is identified by a URI, such as a web page, a telephone number (voice, fax or some other resource mediated by telecommunication equipment), an e-mail address, or any other locatable resource that can be specified by a URL. |
| [**TS**](#b145) | A quantity specifying a point on the axis of natural time. |
| [**XP**](#b149) | A part of a name or address. |

## Datatypes.xsd Simple Type Summary

| Simple Type Summary | |
| --- | --- |
| [**AddressPartType**](#b152) |  |
| [**Code**](#b155) |  |
| [**Decimal**](#b158) |  |
| [**EntityNamePartQualifier**](#b161) |  |
| [**EntityNamePartType**](#b164) |  |
| [**EntityNameUse**](#b167) |  |
| [**PostalAddressUse**](#b170) |  |
| [**set\_EntityNamePartQualifier**](#b173) |  |
| [**set\_EntityNameUse**](#b176) |  |
| [**set\_PostalAddressUse**](#b179) |  |
| [**set\_TelecommunicationAddressUse**](#b182) |  |
| [**set\_TelecommunicationCapability**](#b185) |  |
| [**TelecommunicationAddressUse**](#b188) |  |
| [**TelecommunicationCapability**](#b191) |  |
| [**Uid**](#b194) |  |
| [**UncertaintyType**](#b197) |  |
| [**Uri**](#b200) |  |

## Datatypes.xsd Attribute Group Summary

| Attribute Group Summary | |
| --- | --- |
| [**ATTR\_PQ**](#b205) | Attribute group defining a PQ |
| [**ATTR\_PQ\_HIGH**](#b210) | Attribute group defining a IVL\_PQ high value |
| [**ATTR\_PQ\_LOW**](#b215) | Attribute group defining a IVL\_PQ low value |
| [**ATTR\_RTO**](#b220) | Attribute group defining a RTO numerator and denominator values |
| [**ATTR\_RTO\_HIGH**](#b225) | Attribute group defining a RTO numerator and denominator values |
| [**ATTR\_RTO\_LOW**](#b230) | Attribute group defining a RTO numerator and denominator values |
| [**ATTR\_TS**](#b235) | Attribute group defining a TS |
| [**ATTR\_TS\_HIGH**](#b240) | Attribute group defining a IVL\_TS high value |
| [**ATTR\_TS\_LOW**](#b245) | Attribute group defining a IVL\_TS low value |

## Datatypes.xsd Complex Data Types

complexType "AD"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 1 [attribute](#b8), 1 [element](#b10) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [use](#b8) | = | *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "PHYS" | "PST" | "TMP" | "ABC" | "IDE" | "SYL" | "SRCH" | "SNDX" | "PHON") | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [part](#b10)+ | |
| </...> | |

Content Model Elements (1):

[part](#b10) (in address) []

All Direct / Indirect Based Elements (1):

address

Annotation

Mailing and home or office addresses.  
  
AD is primarily used to communicate data that will allow printing mail labels, or that will allow a person to physically visit that address. The postal address datatype is not supposed to be a container for additional information that might be useful for finding geographic locations (e.g., GPS coordinates) or for performing epidemiological studies. Such additional information should be captured by other, more appropriate data structures.  
  
Addresses are essentially sequences of address parts, but add a "use" code and a valid time range for information about if and when the address can be used for a given purpose.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **AD** |

XML Source (w/o annotations (3))

<xs:complexType name="[**AD**](#b12)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:sequence>

<xs:element maxOccurs="unbounded" name="[**part**](#b10)" type="[**ADXP**](#b17)"/>

</xs:sequence>

<xs:attribute name="[**use**](#b8)" type="[**set\_PostalAddressUse**](#b179)" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b12) component only; 1/1)

 use

|  |  |
| --- | --- |
| Type: | [set\_PostalAddressUse](#b179) [] |
| Use: | optional |

A set of codes advising a system or user which address in a set of like addresses to select for a given purpose.  
An address without specific use code might be a default address useful for any purpose, but an address with a specific use code would be preferred for that respective purpose.  
If populated, the values contained in this attribute SHALL be taken from the HL7 PostalAddressUse code system.

Attribute Value

|  |
| --- |
| *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "PHYS" | "PST" | "TMP" | "ABC" | "IDE" | "SYL" | "SRCH" | "SNDX" | "PHON") |

Content Element Detail (all declarations; defined within [this](#b12) component only; 1/1)

 part

|  |  |
| --- | --- |
| Type: | [ADXP](#b17) [], empty content |

A sequence of address parts, such as street or post office Box, city, postal code, country, etc.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <part | |
|  | |  |  |  | | --- | --- | --- | | [type](#b15) | = | ("AL" | "ADL" | "UNID" | "UNIT" | "DAL" | "DINST" | "DINSTA" | "DINSTQ" | "DMOD" | "DMODID" | "SAL" | "BNR" | "BNN" | "BNS" | "STR" | "STB" | "STTYP" | "DIR" | "INT" | "CAR" | "CEN" | "CNT" | "CPA" | "CTY" | "DEL" | "POB" | "PRE" | "STA" | "ZIP" | "DPID") | | [value](#b147) | = | xs:string | |
| /> | |

complexType "ADXP"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b15) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b147) | = | xs:string | | [type](#b15) | = | ("AL" | "ADL" | "UNID" | "UNIT" | "DAL" | "DINST" | "DINSTA" | "DINSTQ" | "DMOD" | "DMODID" | "SAL" | "BNR" | "BNN" | "BNS" | "STR" | "STB" | "STTYP" | "DIR" | "INT" | "CAR" | "CEN" | "CNT" | "CPA" | "CTY" | "DEL" | "POB" | "PRE" | "STA" | "ZIP" | "DPID") | |
| /> | |

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [part](#b287) (defined in [AddressLiteral](#b289) complexType) [], | [part](#b10) (in address) [] |

Annotation

A part with a type-tag signifying its role in the address. Typical parts that exist in about every address are street, house number, or post box, postal code, city, country but other roles may be defined regionally, nationally, or on an enterprise level (e.g. in military addresses).

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [XP](#b149) [] (extension)  **ADXP** |

XML Source (w/o annotations (2))

<xs:complexType name="[**ADXP**](#b17)">

<xs:complexContent>

<xs:extension base="[**XP**](#b149)">

<xs:attribute name="[**type**](#b15)" type="[**AddressPartType**](#b152)" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b17) component only; 1/2)

 type

|  |  |
| --- | --- |
| Type: | [AddressPartType](#b152) [] |
| Use: | required |

Whether an address part names the street, city, country, postal code, post box, address line 1, etc.  
The value of this attribute SHALL be taken from the HL7 AddressPartType code system.

Attribute Value

|  |
| --- |
| *enumeration of* xs:string |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "AL" | - | Address Line : An address line is for either an additional locator, a delivery address or a street address. An address generally has only a delivery address line or a street address line, but not both | | "ADL" | - | Additional Locator : This can be a unit designator, such as apartment number, suite number, or floor. There may be several unit designators in an address (e.g., "3rd floor, Appt. 342"). This can also be a designator pointing away from the location, rather than specifying a smaller location within some larger one (e.g., Dutch "t.o." means "opposite to" for house boats located across the street facing houses) | | "UNID" | - | Unit Identifier : The number or name of a specific unit contained within a building or complex, as assigned by that building or complex | | "UNIT" | - | Unit Designator : Indicates the type of specific unit contained within a building or complex. E.g. Apartment, Floor | | "DAL" | - | Delivery Address Line : A delivery address line is frequently used instead of breaking out delivery mode, delivery installation, etc. An address generally has only a delivery address line or a street address line, but not both. | | "DINST" | - | Delivery Installation Type : Indicates the type of delivery installation (the facility to which the mail will be delivered prior to final shipping via the delivery mode.) Example: post office, letter carrier depot, community mail center, station, etc. | | "DINSTA" | - | Delivery Installation Area : The location of the delivery installation, usually a town or city, and is only required if the area is different from the municipality. Area to which mail delivery service is provided from any postal facility or service such as an individual letter carrier, rural route, or postal route | | "DINSTQ" | - | Delivery Installation Qualifier : A number, letter or name identifying a delivery installation. E.g., for Station A, the delivery installation qualifier would be 'A'. | | "DMOD" | - | Delivery Mode : Indicates the type of service offered, method of delivery. For example: post office box, rural route, general delivery, etc. | | "DMODID" | - | Delivery Mode Identifier: Represents the routing information such as a letter carrier route number. It is the identifying number of the designator (the box number or rural route number) | | "SAL" | - | Street Address Line : A street address line is frequently used instead of breaking out build number, street name, street type, etc. An address generally has only a delivery address line or a street address line, but not both. | | "BNR" | - | Building Number : The number of a building, house or lot alongside the street. Also known as "primary street number". This does not number the street but rather the building | | "BNN" | - | Building Number Numeric : The numeric portion of a building number | | "BNS" | - | Building Number Suffix : Any alphabetic character, fraction or other text that may appear after the numeric portion of a building number | | "STR" | - | Street Name : The name of the street, including the type | | "STB" | - | Street Name Base : The base name of a roadway or artery recognized by a municipality (excluding street type and direction) | | "STTYP" | - | Street Type : The designation given to the street. (e.g. Street, Avenue, Crescent, etc.) | | "DIR" | - | Direction : Direction (e.g., N, S, W, E) | | "INT" | - | Intersection : An intersection denotes that the actual address is located at or close to the intersection of two or more streets | | "CAR" | - | Care of : The name of the party who will take receipt at the specified address, and will take on responsibility for ensuring delivery to the target recipient.  Note: This is included only to support the convention of writing c/- address lines. This item is not appropriate for use when information is entrusted to one party on behalf of another in some significant way. | | "CEN" | - | Census Tract : A geographic sub-unit delineated for demographic purposes | | "CNT" | - | Country : Country | | "CPA" | - | County or Parish : A sub-unit of a state or province. (49 of the United States of America use the term "county;" Louisiana uses the term "parish") | | "CTY" | - | Municipality : The name of the city, town, village, or other community or delivery center | | "DEL" | - | Delimiter : Delimiters are printed without framing white space. If no value component is provided, the delimiter appears as a line break | | "POB" | - | Post box : A numbered box located in a post station | | "PRE" | - | Precinct : A subsection of a municipality | | "STA" | - | State or Province : A sub-unit of a country with limited sovereignty in a federally organized country | | "ZIP" | - | Postal Code : A postal code designating a region defined by the postal service | | "DPID" | - | Delivery Point Identifier : A value that uniquely identifies the postal address | |

complexType "ANY"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| XML Representation Summary |
| <.../> |

Known Direct Subtypes (15):

[AD](#b12) [], [BL](#b24) [], [CD](#b33) [], [CS](#b45) [], [EN](#b52) [], [II](#b64) [], [IVL\_INT](#b77) [], [IVL\_PQ](#b83) [], [*IVL\_QTY*](#b92) [], [IVL\_REAL](#b100) [], [IVL\_RTO](#b106) [], [IVL\_TS](#b112) [], [*QTY*](#b118) [], [ST](#b131) [], [TEL](#b139) []

Known Indirect Subtypes (9):

[CO](#b40) [], [INT](#b69) [], [PIVL\_TS](#b886) [], [PQ](#b115) [], [REAL](#b123) [], [RTO](#b126) [], [SXCM\_TS](#b1058) [], [TS](#b145) [], [VersionedIdentifier](#b1122) []

All Direct / Indirect Based Elements (45):

[actionId](#b260) [],

address,

[assertedRights](#b984) [],

category,

[citation](#b699) [],

[code](#b38) [],

code,

concept,

contact,

[description](#b271) (defined in [ActionGroup](#b275) complexType) [],

[description](#b698) (defined in [KnowledgeResource](#b701) complexType) [],

[description](#b448) (in coverage) [],

[description](#b512) (in [documentationConcept](#b392)) [],

[description](#b765) (in [metadata](#b684)) [],

[description](#b784) (in modelReference) [],

[displayText](#b511) (in [documentationConcept](#b392)) [],

displayText (in [item](#b729) defined in [ListConstraint](#b731) complexType),

[eventDateTime](#b311) [],

groupReference,

[high](#b90) [],

[identifier](#b680) (defined in [*ItemDefinition*](#b682) complexType) [],

identifier (in [identifiers](#b694) defined in [KnowledgeResource](#b701) complexType),

identifier (in [identifiers](#b759) in [metadata](#b684)),

itemCode,

[language](#b772) [],

[location](#b697) [],

[low](#b89) [],

[name](#b865) (defined in [Person](#b868) complexType) [],

[name](#b831) (in [affiliation](#b866)) [],

[period](#b884) (defined in [PIVL\_TS](#b886) complexType) [],

[period](#b860) (defined in [PeriodLiteral](#b862) complexType) [],

permissions (in [permissions](#b986) in rightsDeclaration),

[phase](#b883) (defined in [PIVL\_TS](#b886) complexType) [],

[phase](#b859) (defined in [PeriodLiteral](#b862) complexType) [],

[qualityOfEvidence](#b533) [],

[schemaIdentifier](#b761) [],

[strengthOfRecommendation](#b534) [],

templateId (in [templateIds](#b695) defined in [KnowledgeResource](#b701) complexType),

templateId (in [templateIds](#b762) in [metadata](#b684)),

term,

[textEquivalent](#b316) [],

[title](#b270) (defined in [ActionGroup](#b275) complexType) [],

[title](#b696) (defined in [KnowledgeResource](#b701) complexType) [],

[title](#b764) (in [metadata](#b684)) [],

[value](#b449) (in coverage) []

Annotation

Defines the basic properties of every data value. This is conceptually an abstract type, meaning that no proper value can be just a data value without belonging to any concrete type. Every public concrete type is a specialization of this general abstract DataValue type.  
  
This class is maintained despite the lack of attributes to maintain compatibility with the ISO 21090 data structure.

XML Source (w/o annotations (1))

<xs:complexType name="[**ANY**](#b19)"/>

complexType "BL"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b22) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b22) | = | xs:boolean | |
| /> | |

Annotation

BL stands for the values of two-valued logic. A BL value can be either true or false.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **BL** |

XML Source (w/o annotations (2))

<xs:complexType name="[**BL**](#b24)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attribute name="[**value**](#b22)" type="xs:boolean" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b24) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | required |

The value of the BL.

complexType "CD"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 5 [attributes](#b27) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [code](#b27) | = | xs:string | | [codeSystem](#b28) | = | xs:string | | [codeSystemName](#b29) | = | xs:string | | [displayName](#b30) | = | xs:string | | [originalText](#b31) | = | xs:string : "" | |
| /> | |

All Direct / Indirect Based Elements (10):

|  |  |
| --- | --- |
| category,  [code](#b38) [],  code,  concept,  itemCode, | [language](#b772) [],  [qualityOfEvidence](#b533) [],  [strengthOfRecommendation](#b534) [],  term,  [value](#b449) (in coverage) [] |

Annotation

A CD is a reference to a concept defined in an external code system, terminology, or ontology.  
  
A CD may also contain an original text or phrase that served as the basis of the coding.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **CD** |

XML Source (w/o annotations (6))

<xs:complexType name="[**CD**](#b33)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attribute name="[**code**](#b27)" type="[**Code**](#b155)" use="optional"/>

<xs:attribute name="[**codeSystem**](#b28)" type="[**Uid**](#b194)" use="optional"/>

<xs:attribute name="[**codeSystemName**](#b29)" type="xs:string" use="optional"/>

<xs:attribute name="[**displayName**](#b30)" type="xs:string" use="optional"/>

<xs:attribute default="" name="[**originalText**](#b31)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b33) component only; 5/5)

 code

|  |  |
| --- | --- |
| Type: | [Code](#b155) [] |
| Use: | optional |

The plain code symbol defined by the code system, or an expression in a syntax defined by the code system which describes the concept.  
Code SHALL be an exact match to a plain code symbol or expression defined by the code system. If the code system defines a code or expression that includes whitespace, the code SHALL include the whitespace. An expression can only be used where the codeSystem either defines an expression syntax, or there is a generally accepted syntax for the codeSystem. A code system may be defined that only defines an expression syntax with bindings to other code Systems for the elements of the expression.  
It is at the discretion of the interpreting system whether to check for an expression instead of a simple code and evaluate the expression instead of treating the expression as a code. In some cases, it may be unclear or ambiguous whether the code represents a single symbol or an expression. This usually arises where the code system defines an expression language and then defines pre-coordinated concepts with symbols which match their expression, e.g. UCUM. In other cases, it is safe to treat the expression as a symbol. There is no guarantee that this is always safe: the definitions of the codeSystem should always be consulted to determine how to handle potential expressions.

Attribute Value

|  |
| --- |
| xs:string |

 codeSystem

|  |  |
| --- | --- |
| Type: | [Uid](#b194) [] |
| Use: | optional |

The code system that defines the code, or if no code was found, the codeSystem in which no code was found.  
Code systems SHALL be referred to by a UID, which allows unambiguous reference to standard code systems and other local codesystems. Where either ISO or HL7 have assigned UID to code Systems, then these UIDs SHALL be used. Otherwise implementations SHALL use an appropriate ISO Object Identifier (OID) or UUID to construct a globally unique local coding system identifier.

Attribute Value

|  |
| --- |
| xs:string |

 codeSystemName

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

The common name of the coding system.  
  
The code system name has no computational value. codeSystemName can never modify the meaning of codeSystem and cannot exist without codeSystem.  
  
Information Processing Entities claiming direct or indirect conformance SHALL NOT functionally rely on codeSystemName. In addition, they MAY choose not to implement codeSystemName; but SHALL NOT reject instances because codeSystemName is present.  
  
Note: The purpose of a code system name is to assist an unaided human interpreter of a code value to interpret codeSystem.

 displayName

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

A name, title, or representation for the code or expression as it exists in the code system.  
If populated, the displayName SHALL be a valid human readable representation of the concept as defined by the code system at the time of data entry. The displayName SHALL conform to any rules defined by the codingSystem; if the codeSystem does not define a human representation for the code or expression, then none can be provided. displayName is included both as a courtesy to an unaided human interpreter of a code value and as a documentation of the name used to display the concept to the user. The display name has no functional meaning; it SHALL never exist without a code; and it SHALL never modify the meaning of the code. A display name may not be present if the code is an expression for which no display name has been assigned or can be derived. Information Processing Entities claiming direct or indirect conformance MAY choose not to implement displayName but SHALL NOT reject instances because displayName is present.  
Display names SHALL not alter the meaning of the code value. Therefore, display names SHOULD NOT be presented to the user on a receiving application system without ascertaining that the display name adequately represents the concept referred to by the code value. Communication SHALL NOT simply rely on the display name. The display name's main purpose is to support implementation debugging.

 originalText

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

The text as seen and/or selected by the user who entered the data which represents the intended meaning of the user. This attribute is equivalent to OriginalText.value in the ISO 21090 model.  
  
Note: Local implementations may influence what is required to represent that original text.  
  
Original text can be used in a structured user interface to capture what the user saw as a representation of the code on the data input screen, or in a situation where the user dictates or directly enters text, it is the text entered or uttered by the user.  
  
It is valid to use the CD datatype to store only the text that the user entered or uttered. In this situation, original text will exist without a code. In a situation where the code is assigned sometime after the text was entered, originalText is the text or phrase used as the basis for assigning the code.  
  
The original text SHALL be an excerpt of the relevant information in the original sources, rather than a pointer or exact reproduction. Thus the original text SHALL be represented in plain text form. In specific circumstances, when clearly descirbed the context of use, the originalText may be a reference to some other text artefact for which the resolution scope is clearly described.  
  
Values of type CD MAY have a original text despite not having a code. Any CD value with no code signifies a coding exception. In this case, originalText is a name or description of the concept that was not coded.

Attribute Value

|  |  |
| --- | --- |
| Default: | "" |

complexType "CO"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 1 [attribute](#b36), 1 [element](#b38) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b36) | = | xs:double | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [code](#b38)? | |
| </...> | |

Content Model Elements (1):

[code](#b38) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  [*QTY*](#b118) [] (extension)  **CO** |

XML Source

<xs:complexType name="[**CO**](#b40)">

<xs:complexContent>

<xs:extension base="[**QTY**](#b118)">

<xs:sequence>

<xs:element minOccurs="0" name="[**code**](#b38)" type="[**CD**](#b33)"/>

</xs:sequence>

<xs:attribute name="[**value**](#b36)" type="[**Decimal**](#b158)" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b40) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | optional |

Attribute Value

|  |
| --- |
| xs:double |

Content Element Detail (all declarations; defined within [this](#b40) component only; 1/1)

 code

|  |  |
| --- | --- |
| Type: | [CD](#b33) [], empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <code | |
|  | |  |  |  | | --- | --- | --- | | [code](#b27) | = | xs:string | | [codeSystem](#b28) | = | xs:string | | [codeSystemName](#b29) | = | xs:string | | [displayName](#b30) | = | xs:string | | [originalText](#b31) | = | xs:string : "" | |
| /> | |

complexType "CS"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b43) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [code](#b43) | = | xs:string | |
| /> | |

Annotation

Coded data in its simplest form, where only the code is not predetermined.  
  
The code system and code system version are implied and fixed by the context in which the CS value occurs.  
  
Due to its highly restricted functionality, CS SHALL only be used for simple structural attributes with highly controlled and stable terminologies where:  
- all codes come from a single code system  
- codes are not reused if their concept is deprecated  
- the publication and extensibility properties of the code system are well described and understood

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **CS** |

XML Source (w/o annotations (2))

<xs:complexType name="[**CS**](#b45)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attribute name="[**code**](#b43)" type="xs:string" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b45) component only; 1/1)

 code

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

The plain code symbol defined by the code system. If the code value is empty or null, then there is no code in the code system that represents the concept.  
Code SHALL only contain characters that are either a letter, a digit, or one of '.', '-', '\_' or ':'. Code systems that are used with CS SHALL NOT define code symbols or expression syntaxes that contain whitespace or any other characters not in this list.

complexType "EN"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 1 [attribute](#b48), 1 [element](#b50) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [use](#b48) | = | *list of* ("ABC" | "SYL" | "IDE" | "C" | "OR" | "T" | "I" | "P" | "ANON" | "A" | "R" | "OLD" | "DN" | "M" | "SRCH" | "PHON") | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [part](#b50)+ | |
| </...> | |

Content Model Elements (1):

[part](#b50) (in name defined in [Person](#b868) complexType) []

All Direct / Indirect Based Elements (1):

[name](#b865) (defined in [Person](#b868) complexType) []

Annotation

A name for a person, organization, place or thing.  
  
Examples: Jim Bob Walton, Jr., Health Level Seven, Inc., Lake Tahoe, etc. An entity name may be as simple as a character string or may consist of several entity name parts, such as, Jim, Bob, Walton, and Jr., Health Level Seven, and Inc.  
  
Entity names are essentially sequences of entity name parts, but add a "use" code.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **EN** |

XML Source (w/o annotations (3))

<xs:complexType name="[**EN**](#b52)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:sequence>

<xs:element maxOccurs="unbounded" name="[**part**](#b50)" type="[**ENXP**](#b58)"/>

</xs:sequence>

<xs:attribute name="[**use**](#b48)" type="[**set\_EntityNameUse**](#b176)" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b52) component only; 1/1)

 use

|  |  |
| --- | --- |
| Type: | [set\_EntityNameUse](#b176) [] |
| Use: | optional |

A set of codes advising a system or user which name in a set of names to select for a given purpose.  
A name without specific use code might be a default name useful for any purpose, but a name with a specific use code would be preferred for that respective purpose. Names SHOULD not be collected without at least one use code, but names MAY exist without use code, particularly for legacy data.  
If populated, the values contained in this attribute SHALL be taken from the HL7 EntityNameUse2 code system.

Attribute Value

|  |
| --- |
| *list of* ("ABC" | "SYL" | "IDE" | "C" | "OR" | "T" | "I" | "P" | "ANON" | "A" | "R" | "OLD" | "DN" | "M" | "SRCH" | "PHON") |

Content Element Detail (all declarations; defined within [this](#b52) component only; 1/1)

 part

|  |  |
| --- | --- |
| Type: | [ENXP](#b58) [], empty content |

A sequence of name parts, such as given name or family name, prefix, suffix, etc.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <part | |
|  | |  |  |  | | --- | --- | --- | | [qualifier](#b56) | = | *list of* ("LS" | "AC" | "NB" | "PR" | "HON" | "BR" | "AD" | "SP" | "MID" | "CL" | "IN" | "PFX" | "SFX") | | [type](#b55) | = | ("FAM" | "GIV" | "TITLE" | "DEL") | | [value](#b147) | = | xs:string | |
| /> | |

complexType "ENXP"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b55) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b147) | = | xs:string | | [type](#b55) | = | ("FAM" | "GIV" | "TITLE" | "DEL") | | [qualifier](#b56) | = | *list of* ("LS" | "AC" | "NB" | "PR" | "HON" | "BR" | "AD" | "SP" | "MID" | "CL" | "IN" | "PFX" | "SFX") | |
| /> | |

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [part](#b526) (defined in [EntityNameLiteral](#b528) complexType) [], | [part](#b50) (in [name](#b865) defined in [Person](#b868) complexType) [] |

Annotation

A part with a type code signifying the role of the part in the whole entity name, and qualifier codes for more detail about the name part type. (Typical name parts for person names are given names, and family names, titles, etc.).

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [XP](#b149) [] (extension)  **ENXP** |

XML Source (w/o annotations (3))

<xs:complexType name="[**ENXP**](#b58)">

<xs:complexContent>

<xs:extension base="[**XP**](#b149)">

<xs:attribute name="[**type**](#b55)" type="[**EntityNamePartType**](#b164)" use="required"/>

<xs:attribute name="[**qualifier**](#b56)" type="[**set\_EntityNamePartQualifier**](#b173)" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b58) component only; 2/3)

 type

|  |  |
| --- | --- |
| Type: | [EntityNamePartType](#b164) [] |
| Use: | required |

Indicates whether the name part is a given name, family name, prefix, suffix, etc.  
The value of this attribute SHALL be taken from the HL7 EntityNamePartType2 code system.

Attribute Value

|  |
| --- |
| *enumeration of* xs:string |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "FAM" | - | Family : Family name, this is the name that links to the genealogy. In some cultures (e.g. Eritrea) the family name of a son is the first name of his father | | "GIV" | - | Given: Given name. Note: don't call it "first name" since this given names do not always come first | | "TITLE" | - | Title : Part of the name that is acquired as a title due to academic, legal, employment or nobility status etc. Note: Title name parts include name parts that come after the name such as qualifications | | "DEL" | - | Delimiter : A delimiter has no meaning other than being literally printed in this name representation. A delimiter has no implicit leading and trailing white space | |

 qualifier

|  |  |
| --- | --- |
| Type: | [set\_EntityNamePartQualifier](#b173) [] |
| Use: | optional |

The qualifier is a set of codes each of which specifies a certain subcategory of the name part in addition to the main name part type.  
For example, a given name may be flagged as a nickname (CL), a family name may be a name acquired by marriage (SP) or a name from birth (BR).  
If populated, the values contained in this attribute SHALL be taken from the HL7 EntityNamePartQualifier2 code system.

Attribute Value

|  |
| --- |
| *list of* ("LS" | "AC" | "NB" | "PR" | "HON" | "BR" | "AD" | "SP" | "MID" | "CL" | "IN" | "PFX" | "SFX") |

complexType "II"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b61) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [root](#b61) | = | xs:string | | [extension](#b62) | = | xs:string : "" | |
| /> | |

Known Direct Subtypes (1):

[VersionedIdentifier](#b1122) []

All Direct / Indirect Based Elements (8):

[actionId](#b260) [],

groupReference,

[identifier](#b680) (defined in [*ItemDefinition*](#b682) complexType) [],

identifier (in [identifiers](#b694) defined in [KnowledgeResource](#b701) complexType),

identifier (in [identifiers](#b759) in [metadata](#b684)),

[schemaIdentifier](#b761) [],

templateId (in [templateIds](#b695) defined in [KnowledgeResource](#b701) complexType),

templateId (in [templateIds](#b762) in [metadata](#b684))

Annotation

An identifier that uniquely identifies a thing or object.  
  
Examples are object identifier for HL7 RIM objects, medical record number, order id, service catalog item id, Vehicle Identification Number (VIN), etc. Instance identifiers are usually defined based on ISO object identifiers.  
  
An identifier allows someone to select one record, object or thing from a set of candidates. Usually an identifier alone without any context is not usable. Identifiers are distinguished from concept descriptors as concept descriptors never identify an individual thing, although there may sometimes be an individual record or object that represents the concept.  
  
Information Processing Entities claiming direct or indirect conformance SHALL never assume that receiving applications can infer the identity of issuing authority or the type of the identifier from the identifier or components thereof.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **II** |

XML Source (w/o annotations (3))

<xs:complexType name="[**II**](#b64)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attribute name="[**root**](#b61)" type="[**Uid**](#b194)" use="required"/>

<xs:attribute default="" name="[**extension**](#b62)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b64) component only; 2/2)

 root

|  |  |
| --- | --- |
| Type: | [Uid](#b194) [] |
| Use: | required |

A unique identifier that guarantees the global uniqueness of the instance identifier.  
If root is populated, and there is no extension, then the root is a globally unique identifier in its own right. In the presence of a non-null extension, the root is the unique identifier for the "namespace" of the identifier in the extension. Note that this does NOT necessarily correlate with the organization that manages the issuing of the identifiers. A given organization may manage multiple identifier namespaces, and control over a given namespace may transfer from organization to organization over time while the root remains the same.  
This field can be either a DCE UUID, an Object Identifier (OID), or a special identifier taken from lists that may be published by ISO or HL7.  
Comparison of root values is always case sensitive. UUID's SHALL be represented in upper case, so UUID case should always be preserved.  
The root SHALL not be used to carry semantic meaning - all it does is ensure global computational uniqueness.

Attribute Value

|  |
| --- |
| xs:string |

 extension

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

A character string as a unique identifier within the scope of the identifier root.  
The root and extension scheme means that the concatenation of root and extension SHALL be a globally unique identifier for the item that this II value identifies.  
Some identifier schemes define certain style options to their code values. For example, the U.S. Social Security Number (SSN) is normally written with dashes that group the digits into a pattern "123-12-1234". However, the dashes are not meaningful and a SSN can also be represented as "123121234" without the dashes. In the case where identifier schemes provide for multiple representations, HL7 or ISO may make a ruling about which is the preferred form and document that ruling where that respective external identifier scheme is recognized.  
If no extension attribute is provided in a non-null II, then the root is the complete unique identifier.

Attribute Value

|  |  |
| --- | --- |
| Default: | "" |

complexType "INT"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b67) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b67) | = | xs:int | |
| /> | |

Annotation

Integer numbers (-1,0,1,2, 100, 3398129, etc.) are precise numbers that are results of counting and enumerating. Integer numbers are discrete, the set of integers is infinite but countable. No arbitrary limit is imposed on the range of integer numbers.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  [*QTY*](#b118) [] (extension)  **INT** |

XML Source (w/o annotations (2))

<xs:complexType name="[**INT**](#b69)">

<xs:complexContent>

<xs:extension base="[**QTY**](#b118)">

<xs:attribute name="[**value**](#b67)" type="xs:int" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b69) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | xs:int, predefined |
| Use: | required |

The value of the INT. Note that this specification imposes no limitations on the size of integer, but most implementations will map this to a 32 or 64 bit integer.

complexType "IVL\_INT"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 4 [attributes](#b72) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [low](#b72) | = | xs:int | | [high](#b73) | = | xs:int | | [lowIsInclusive](#b74) | = | xs:boolean | | [highIsInclusive](#b75) | = | xs:boolean | |
| /> | |

Annotation

A set of consecutive values of an ordered base datatype.  
  
Any ordered type can be the basis of an IVL; it does not matter whether the base type is discrete or continuous. If the base datatype is only partially ordered, all elements of the IVL must be elements of a totally ordered subset of the partially ordered datatype. For example, PQ is considered ordered. However the ordering of PQs is only partial; a total order is only defined among comparable quantities (quantities of the same physical dimension). While IVLs between 2 and 4 meter exists, there is no IVL between 2 meters and 4 seconds.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **IVL\_INT** |

XML Source (w/o annotations (5))

<xs:complexType name="[**IVL\_INT**](#b77)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attribute name="[**low**](#b72)" type="xs:int"/>

<xs:attribute name="[**high**](#b73)" type="xs:int"/>

<xs:attribute name="[**lowIsInclusive**](#b74)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b75)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b77) component only; 4/4)

 low

|  |  |
| --- | --- |
| Type: | xs:int, predefined |
| Use: | optional |

This is the low limit. If the low limit is not known, it may be null.  
The low limit SHALL NOT be positive infinity.

 high

|  |  |
| --- | --- |
| Type: | xs:int, predefined |
| Use: | optional |

This is the high limit. If the high limit is not known, it may be null.  
The high limit SHALL NOT be negative infinity, and SHALL be higher than the low limit if one exists.

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called lowIsClosed in the ISO 21090 specification.  
  
Whether low is included in the IVL (is closed) or excluded from the IVL (is open).

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called highIsClosed in the ISO 21090 specification.  
  
Whether high is included in the IVL (is closed) or excluded from the IVL (is open).

complexType "IVL\_PQ"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b80) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [lowValue](#b212) | = | xs:double | | [lowUnit](#b213) | = | xs:string | | [highValue](#b207) | = | xs:double | | [highUnit](#b208) | = | xs:string | | [lowIsInclusive](#b80) | = | xs:boolean | | [highIsInclusive](#b81) | = | xs:boolean | |
| /> | |

Annotation

A set of consecutive values of an ordered base datatype.  
  
Any ordered type can be the basis of an IVL; it does not matter whether the base type is discrete or continuous. If the base datatype is only partially ordered, all elements of the IVL must be elements of a totally ordered subset of the partially ordered datatype. For example, PQ is considered ordered. However the ordering of PQs is only partial; a total order is only defined among comparable quantities (quantities of the same physical dimension). While IVLs between 2 and 4 meter exists, there is no IVL between 2 meters and 4 seconds.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **IVL\_PQ** |

XML Source (w/o annotations (5))

<xs:complexType name="[**IVL\_PQ**](#b83)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attributeGroup ref="[**ATTR\_PQ\_LOW**](#b215)"/>

<xs:attributeGroup ref="[**ATTR\_PQ\_HIGH**](#b210)"/>

<xs:attribute name="[**lowIsInclusive**](#b80)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b81)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b83) component only; 2/6)

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called lowIsClosed in the ISO 21090 specification.  
  
Whether low is included in the IVL (is closed) or excluded from the IVL (is open).

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called highIsClosed in the ISO 21090 specification.  
  
Whether high is included in the IVL (is closed) or excluded from the IVL (is open).

complexType "IVL\_QTY"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definitions of 2 [attributes](#b86), 2 [elements](#b89) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [lowIsInclusive](#b86) | = | xs:boolean | | [highIsInclusive](#b87) | = | xs:boolean | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [low](#b89)?, [high](#b90)? | |
| </...> | |

Content Model Elements (2):

[high](#b90) [], [low](#b89) []

Annotation

A set of consecutive values of an ordered base datatype.  
  
Any ordered type can be the basis of an IVL; it does not matter whether the base type is discrete or continuous. If the base datatype is only partially ordered, all elements of the IVL must be elements of a totally ordered subset of the partially ordered datatype. For example, PQ is considered ordered. However the ordering of PQs is only partial; a total order is only defined among comparable quantities (quantities of the same physical dimension). While IVLs between 2 and 4 meter exists, there is no IVL between 2 meters and 4 seconds.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  ***IVL\_QTY*** |

XML Source (w/o annotations (5))

<xs:complexType abstract="true" name="[**IVL\_QTY**](#b92)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:sequence>

<xs:element minOccurs="0" name="[**low**](#b89)" type="[**QTY**](#b118)"/>

<xs:element minOccurs="0" name="[**high**](#b90)" type="[**QTY**](#b118)"/>

</xs:sequence>

<xs:attribute name="[**lowIsInclusive**](#b86)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b87)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b92) component only; 2/2)

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called lowIsClosed in the ISO 21090 specification.  
  
Whether low is included in the IVL (is closed) or excluded from the IVL (is open).

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called highIsClosed in the ISO 21090 specification.  
  
Whether high is included in the IVL (is closed) or excluded from the IVL (is open).

Content Element Detail (all declarations; defined within [this](#b92) component only; 2/2)

 low

|  |  |
| --- | --- |
| Type: | [*QTY*](#b118) [], empty content |

This is the low limit. If the low limit is not known, it may be null.  
The low limit SHALL NOT be positive infinity.

|  |
| --- |
| XML Representation Summary |
| <low/> |

 high

|  |  |
| --- | --- |
| Type: | [*QTY*](#b118) [], empty content |

This is the high limit. If the high limit is not known, it may be null.  
The high limit SHALL NOT be negative infinity, and SHALL be higher than the low limit if one exists.

|  |
| --- |
| XML Representation Summary |
| <high/> |

complexType "IVL\_REAL"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 4 [attributes](#b95) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [low](#b95) | = | xs:double | | [high](#b96) | = | xs:double | | [lowIsInclusive](#b97) | = | xs:boolean | | [highIsInclusive](#b98) | = | xs:boolean | |
| /> | |

Annotation

A set of consecutive values of an ordered base datatype.  
  
Any ordered type can be the basis of an IVL; it does not matter whether the base type is discrete or continuous. If the base datatype is only partially ordered, all elements of the IVL must be elements of a totally ordered subset of the partially ordered datatype. For example, PQ is considered ordered. However the ordering of PQs is only partial; a total order is only defined among comparable quantities (quantities of the same physical dimension). While IVLs between 2 and 4 meter exists, there is no IVL between 2 meters and 4 seconds.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **IVL\_REAL** |

XML Source (w/o annotations (5))

<xs:complexType name="[**IVL\_REAL**](#b100)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attribute name="[**low**](#b95)" type="[**Decimal**](#b158)" use="optional"/>

<xs:attribute name="[**high**](#b96)" type="[**Decimal**](#b158)" use="optional"/>

<xs:attribute name="[**lowIsInclusive**](#b97)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b98)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b100) component only; 4/4)

 low

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | optional |

This is the low limit. If the low limit is not known, it may be null.  
The low limit SHALL NOT be positive infinity.

Attribute Value

|  |
| --- |
| xs:double |

 high

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | optional |

This is the high limit. If the high limit is not known, it may be null.  
The high limit SHALL NOT be negative infinity, and SHALL be higher than the low limit if one exists.

Attribute Value

|  |
| --- |
| xs:double |

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called lowIsClosed in the ISO 21090 specification.  
  
Whether low is included in the IVL (is closed) or excluded from the IVL (is open).

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called highIsClosed in the ISO 21090 specification.  
  
Whether high is included in the IVL (is closed) or excluded from the IVL (is open).

complexType "IVL\_RTO"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b103) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [lowNumerator](#b227) | = | xs:double | | [lowDenominator](#b228) | = | xs:double | | [highNumerator](#b222) | = | xs:double | | [highDenominator](#b223) | = | xs:double | | [lowIsInclusive](#b103) | = | xs:boolean | | [highIsInclusive](#b104) | = | xs:boolean | |
| /> | |

Annotation

A set of consecutive values of an ordered base datatype.  
  
Any ordered type can be the basis of an IVL; it does not matter whether the base type is discrete or continuous. If the base datatype is only partially ordered, all elements of the IVL must be elements of a totally ordered subset of the partially ordered datatype. For example, PQ is considered ordered. However the ordering of PQs is only partial; a total order is only defined among comparable quantities (quantities of the same physical dimension). While IVLs between 2 and 4 meter exists, there is no IVL between 2 meters and 4 seconds.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **IVL\_RTO** |

XML Source (w/o annotations (5))

<xs:complexType name="[**IVL\_RTO**](#b106)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attributeGroup ref="[**ATTR\_RTO\_LOW**](#b230)"/>

<xs:attributeGroup ref="[**ATTR\_RTO\_HIGH**](#b225)"/>

<xs:attribute name="[**lowIsInclusive**](#b103)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b104)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b106) component only; 2/6)

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called lowIsClosed in the ISO 21090 specification.  
  
Whether low is included in the IVL (is closed) or excluded from the IVL (is open).

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called highIsClosed in the ISO 21090 specification.  
  
Whether high is included in the IVL (is closed) or excluded from the IVL (is open).

complexType "IVL\_TS"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b109) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [low](#b243) | = | xs:string | | [high](#b238) | = | xs:string | | [lowIsInclusive](#b109) | = | xs:boolean | | [highIsInclusive](#b110) | = | xs:boolean | |
| /> | |

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [phase](#b883) (defined in [PIVL\_TS](#b886) complexType) [], | [phase](#b859) (defined in [PeriodLiteral](#b862) complexType) [] |

Annotation

A set of consecutive values of an ordered base datatype.  
  
Any ordered type can be the basis of an IVL; it does not matter whether the base type is discrete or continuous. If the base datatype is only partially ordered, all elements of the IVL must be elements of a totally ordered subset of the partially ordered datatype. For example, PQ is considered ordered. However the ordering of PQs is only partial; a total order is only defined among comparable quantities (quantities of the same physical dimension). While IVLs between 2 and 4 meter exists, there is no IVL between 2 meters and 4 seconds.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **IVL\_TS** |

XML Source (w/o annotations (3))

<xs:complexType name="[**IVL\_TS**](#b112)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attributeGroup ref="[**ATTR\_TS\_LOW**](#b245)"/>

<xs:attributeGroup ref="[**ATTR\_TS\_HIGH**](#b240)"/>

<xs:attribute name="[**lowIsInclusive**](#b109)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b110)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b112) component only; 2/4)

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called lowIsClosed in the ISO 21090 specification.  
  
Whether low is included in the IVL (is closed) or excluded from the IVL (is open).

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

This attribute is called highIsClosed in the ISO 21090 specification.  
  
Whether high is included in the IVL (is closed) or excluded from the IVL (is open).

complexType "PQ"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b202) | = | xs:double | | [unit](#b203) | = | xs:string | |
| /> | |

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [period](#b884) (defined in [PIVL\_TS](#b886) complexType) [], | [period](#b860) (defined in [PeriodLiteral](#b862) complexType) [] |

Annotation

A dimensioned quantity expressing the result of measuring.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  [*QTY*](#b118) [] (extension)  **PQ** |

XML Source (w/o annotations (1))

<xs:complexType name="[**PQ**](#b115)">

<xs:complexContent>

<xs:extension base="[**QTY**](#b118)">

<xs:attributeGroup ref="[**ATTR\_PQ**](#b205)"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

complexType "QTY"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |

|  |
| --- |
| XML Representation Summary |
| <.../> |

Known Direct Subtypes (6):

[CO](#b40) [], [INT](#b69) [], [PQ](#b115) [], [REAL](#b123) [], [RTO](#b126) [], [TS](#b145) []

Known Indirect Subtypes (2):

[PIVL\_TS](#b886) [], [SXCM\_TS](#b1058) []

All Direct / Indirect Based Elements (5):

|  |  |
| --- | --- |
| [eventDateTime](#b311) [],  [high](#b90) [],  [low](#b89) [], | [period](#b884) (defined in [PIVL\_TS](#b886) complexType) [],  [period](#b860) (defined in [PeriodLiteral](#b862) complexType) [] |

Annotation

The quantity datatype is an abstract generalization for all datatypes whose domain values has an order relation (less-or-equal) and where difference is defined in all of the datatype's totally ordered value subsets.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  ***QTY*** |

XML Source (w/o annotations (1))

<xs:complexType abstract="true" name="[**QTY**](#b118)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)"/>

</xs:complexContent>

</xs:complexType>

complexType "REAL"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b121) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b121) | = | xs:double | |
| /> | |

Annotation

Fractional numbers. Typically used whenever quantities are measured, estimated, or computed from other real numbers. The typical representation is decimal, where the number of significant decimal digits is known as the precision.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  [*QTY*](#b118) [] (extension)  **REAL** |

XML Source (w/o annotations (2))

<xs:complexType name="[**REAL**](#b123)">

<xs:complexContent>

<xs:extension base="[**QTY**](#b118)">

<xs:attribute name="[**value**](#b121)" type="[**Decimal**](#b158)" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b123) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | required |

The value of the REAL.

Attribute Value

|  |
| --- |
| xs:double |

complexType "RTO"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [numerator](#b217) | = | xs:double | | [denominator](#b218) | = | xs:double | |
| /> | |

Annotation

A quantity constructed as the quotient of a numerator quantity divided by a denominator quantity.  
Common factors in the numerator and denominator are not automatically cancelled out.  
The RTO datatype supports titers (e.g., 1:128) and other quantities produced by laboratories that truly represent ratios. Ratios are not simply structured numerics, particularly blood pressure measurements (e.g. 120/60) are not ratios.  
Notes:  
1. Ratios are different from rational numbers, i.e., in ratios common factors in the numerator and denominator never cancel out. A ratio of two real or integer numbers is not automatically reduced to a real number. This datatype is not defined to generally represent rational numbers. It is used only if common factors in numerator and denominator are not supposed to cancel out. This is only rarely the case. For observation values, ratios occur almost exclusively with titers. In most other cases, REAL should be used instead of the RTO.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  [*QTY*](#b118) [] (extension)  **RTO** |

XML Source (w/o annotations (1))

<xs:complexType name="[**RTO**](#b126)">

<xs:complexContent>

<xs:extension base="[**QTY**](#b118)">

<xs:attributeGroup ref="[**ATTR\_RTO**](#b220)"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

complexType "ST"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b129) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

All Direct / Indirect Based Elements (16):

|  |  |
| --- | --- |
| [assertedRights](#b984) [],  [citation](#b699) [],  [description](#b271) (defined in [ActionGroup](#b275) complexType) [],  [description](#b698) (defined in [KnowledgeResource](#b701) complexType) [],  [description](#b448) (in coverage) [],  [description](#b512) (in [documentationConcept](#b392)) [],  [description](#b765) (in [metadata](#b684)) [],  [description](#b784) (in modelReference) [], | [displayText](#b511) (in [documentationConcept](#b392)) [],  displayText (in [item](#b729) defined in [ListConstraint](#b731) complexType),  [name](#b831) (in [affiliation](#b866)) [],  permissions (in [permissions](#b986) in rightsDeclaration),  [textEquivalent](#b316) [],  [title](#b270) (defined in [ActionGroup](#b275) complexType) [],  [title](#b696) (defined in [KnowledgeResource](#b701) complexType) [],  [title](#b764) (in [metadata](#b684)) [] |

Annotation

The character string datatype stands for text data, primarily intended for machine processing (e.g., sorting, querying, indexing, etc.) or direct display. Used for names, symbols, presentation and formal expressions.  
  
A ST SHALL have at least one character or else be null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **ST** |

XML Source (w/o annotations (2))

<xs:complexType name="[**ST**](#b131)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attribute name="[**value**](#b129)" type="xs:string" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b131) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

The actual content of the string.

complexType "TEL"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 4 [attributes](#b134) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b134) | = | xs:anyURI | | [use](#b135) | = | *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "TMP" | "AS" | "EC" | "MC" | "PG") | | [capabilities](#b136) | = | *list of* ("voice" | "fax" | "data" | "tty" | "sms") | | [useablePeriodOriginalText](#b137) | = | xs:string | |
| /> | |

All Direct / Indirect Based Elements (2):

contact, [location](#b697) []

Annotation

A locatable resource that is identified by a URI, such as a web page, a telephone number (voice, fax or some other resource mediated by telecommunication equipment), an e-mail address, or any other locatable resource that can be specified by a URL.  
  
The address is specified as a Universal Resource Locator (URL) qualified by time specification and use codes that help in deciding which address to use for a given time and purpose.  
  
The value attribute is constrained to be a uniform resource locator specified according to IETF RFCs 1738 and 2806 when used in this datatype.  
  
Note: The intent of this datatype is to be a locator, not an identifier; this datatype is used to refer to a locatable resource using a URL, and knowing the URL allows one to locate the object. However some use cases have arisen where a URI is used to refer to a locatable resource. Though this datatype allows for URIs to be used, the resource identified SHOULD always be locatable. A common use of locatable URIs is to refer to SOAP attachments.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  **TEL** |

XML Source (w/o annotations (5))

<xs:complexType name="[**TEL**](#b139)">

<xs:complexContent>

<xs:extension base="[**ANY**](#b19)">

<xs:attribute name="[**value**](#b134)" type="xs:anyURI" use="required"/>

<xs:attribute name="[**use**](#b135)" type="[**set\_TelecommunicationAddressUse**](#b182)" use="optional"/>

<xs:attribute name="[**capabilities**](#b136)" type="[**set\_TelecommunicationCapability**](#b185)" use="optional"/>

<xs:attribute name="[**useablePeriodOriginalText**](#b137)" type="xs:string"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b139) component only; 4/4)

 value

|  |  |
| --- | --- |
| Type: | xs:anyURI, predefined |
| Use: | required |

A uniform resource identifier specified according to IETF RFC 2396.  
The URI specifies the protocol and the contact point defined by that protocol for the resource.  
Examples: Notable uses of the telecommunication address datatype are for telephone and telefax numbers, e-mail addresses, Hypertext references, FTP references, etc.

 use

|  |  |
| --- | --- |
| Type: | [set\_TelecommunicationAddressUse](#b182) [] |
| Use: | optional |

One or more codes advising system or user which telecommunication address in a set of like addresses to select for a given telecommunication need.  
The telecommunication use code is not a complete classification for equipment types or locations. Its main purpose is to suggest or discourage the use of a particular telecommunication address. There are no easily defined rules that govern the selection of a telecommunication address. Conformance statements may clarify what rules may apply or how additional rules are applied.  
If populated, the values contained in this attribute SHALL be taken from the HL7 TelecommunicationAddressUse code system.

Attribute Value

|  |
| --- |
| *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "TMP" | "AS" | "EC" | "MC" | "PG") |

 capabilities

|  |  |
| --- | --- |
| Type: | [set\_TelecommunicationCapability](#b185) [] |
| Use: | optional |

One or more codes advising a system or user what telecommunication capabilities are known to be associated with the telecommunication address.  
If populated, the values contained in this attribute SHALL be taken from the HL7 TelecommunicationCapability code system.

Attribute Value

|  |
| --- |
| *list of* ("voice" | "fax" | "data" | "tty" | "sms") |

 useablePeriodOriginalText

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

This attribute is equivalent to the originalText.value within the useablePeriod attribute of this class in the ISO 21090 specification.  
  
The periods of time during which the telecommunication address can be used.  
  
For a telephone number, this can indicate the time of day in which the party can be reached on that telephone. For a web address, it may specify a time range in which the web content is promised to be available under the given address.

complexType "TS"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b143) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b143) | = | xs:string | |
| /> | |

Known Direct Subtypes (1):

[SXCM\_TS](#b1058) []

Known Indirect Subtypes (1):

[PIVL\_TS](#b886) []

All Direct / Indirect Based Elements (1):

[eventDateTime](#b311) []

Annotation

A quantity specifying a point on the axis of natural time. A point in time is most often represented as a calendar expression.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [ANY](#b19) [] (extension)  [*QTY*](#b118) [] (extension)  **TS** |

XML Source (w/o annotations (2))

<xs:complexType name="[**TS**](#b145)">

<xs:complexContent>

<xs:extension base="[**QTY**](#b118)">

<xs:attribute name="[**value**](#b143)" use="required">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:pattern value="[1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0-1][0-9])|(2[0-3]))(([0-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+\-](0[0-9]|1[0-3])([0-5][0-9]))?"/>

</xs:restriction>

</xs:simpleType>

</xs:attribute>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b145) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | [anonymous](#b142) simpleType (restriction of xs:string) [] |
| Use: | required |

The value of the TS. value is a string with the format "YYYY[MM[DD[HH[MM[SS[.U[U[U[U]]]]]]]]][+|-ZZzz]" that conforms to the constrained ISO 8601 defined in ISO 8824 (ASN.1) under clause 32 (generalized time). The format should be used to the degree of precision that is appropriate.

Attribute Value

|  |
| --- |
| xs:string |

|  |  |
| --- | --- |
| Pattern: | [1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0-1][0-9])|(2[0-3]))(([0‌-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+\-](0[0-9]|1[0-3])([0-5][0-9]))? |

Anonymous simpleType

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **simpleType** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | pattern: | [1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0‌-1][0-9])|(2[0-3]))(([0-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+‌\-](0[0-9]|1[0-3])([0-5][0-9]))? | |

complexType "XP"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b147) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b147) | = | xs:string | |
| /> | |

Known Direct Subtypes (2):

[ADXP](#b17) [], [ENXP](#b58) []

All Direct / Indirect Based Elements (4):

|  |  |
| --- | --- |
| [part](#b287) (defined in [AddressLiteral](#b289) complexType) [],  [part](#b526) (defined in [EntityNameLiteral](#b528) complexType) [], | [part](#b10) (in address) [],  [part](#b50) (in [name](#b865) defined in [Person](#b868) complexType) [] |

Annotation

A part of a name or address. Each part is a character string.

XML Source (w/o annotations (2))

<xs:complexType name="[**XP**](#b149)">

<xs:attribute name="[**value**](#b147)" type="xs:string" use="required"/>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b149) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

The actual string value of the part.

simpleType "AddressPartType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "AL" | - | Address Line : An address line is for either an additional locator, a delivery address or a street address. An address generally has only a delivery address line or a street address line, but not both | | "ADL" | - | Additional Locator : This can be a unit designator, such as apartment number, suite number, or floor. There may be several unit designators in an address (e.g., "3rd floor, Appt. 342"). This can also be a designator pointing away from the location, rather than specifying a smaller location within some larger one (e.g., Dutch "t.o." means "opposite to" for house boats located across the street facing houses) | | "UNID" | - | Unit Identifier : The number or name of a specific unit contained within a building or complex, as assigned by that building or complex | | "UNIT" | - | Unit Designator : Indicates the type of specific unit contained within a building or complex. E.g. Apartment, Floor | | "DAL" | - | Delivery Address Line : A delivery address line is frequently used instead of breaking out delivery mode, delivery installation, etc. An address generally has only a delivery address line or a street address line, but not both. | | "DINST" | - | Delivery Installation Type : Indicates the type of delivery installation (the facility to which the mail will be delivered prior to final shipping via the delivery mode.) Example: post office, letter carrier depot, community mail center, station, etc. | | "DINSTA" | - | Delivery Installation Area : The location of the delivery installation, usually a town or city, and is only required if the area is different from the municipality. Area to which mail delivery service is provided from any postal facility or service such as an individual letter carrier, rural route, or postal route | | "DINSTQ" | - | Delivery Installation Qualifier : A number, letter or name identifying a delivery installation. E.g., for Station A, the delivery installation qualifier would be 'A'. | | "DMOD" | - | Delivery Mode : Indicates the type of service offered, method of delivery. For example: post office box, rural route, general delivery, etc. | | "DMODID" | - | Delivery Mode Identifier: Represents the routing information such as a letter carrier route number. It is the identifying number of the designator (the box number or rural route number) | | "SAL" | - | Street Address Line : A street address line is frequently used instead of breaking out build number, street name, street type, etc. An address generally has only a delivery address line or a street address line, but not both. | | "BNR" | - | Building Number : The number of a building, house or lot alongside the street. Also known as "primary street number". This does not number the street but rather the building | | "BNN" | - | Building Number Numeric : The numeric portion of a building number | | "BNS" | - | Building Number Suffix : Any alphabetic character, fraction or other text that may appear after the numeric portion of a building number | | "STR" | - | Street Name : The name of the street, including the type | | "STB" | - | Street Name Base : The base name of a roadway or artery recognized by a municipality (excluding street type and direction) | | "STTYP" | - | Street Type : The designation given to the street. (e.g. Street, Avenue, Crescent, etc.) | | "DIR" | - | Direction : Direction (e.g., N, S, W, E) | | "INT" | - | Intersection : An intersection denotes that the actual address is located at or close to the intersection of two or more streets | | "CAR" | - | Care of : The name of the party who will take receipt at the specified address, and will take on responsibility for ensuring delivery to the target recipient.  Note: This is included only to support the convention of writing c/- address lines. This item is not appropriate for use when information is entrusted to one party on behalf of another in some significant way. | | "CEN" | - | Census Tract : A geographic sub-unit delineated for demographic purposes | | "CNT" | - | Country : Country | | "CPA" | - | County or Parish : A sub-unit of a state or province. (49 of the United States of America use the term "county;" Louisiana uses the term "parish") | | "CTY" | - | Municipality : The name of the city, town, village, or other community or delivery center | | "DEL" | - | Delimiter : Delimiters are printed without framing white space. If no value component is provided, the delimiter appears as a line break | | "POB" | - | Post box : A numbered box located in a post station | | "PRE" | - | Precinct : A subsection of a municipality | | "STA" | - | State or Province : A sub-unit of a country with limited sovereignty in a federally organized country | | "ZIP" | - | Postal Code : A postal code designating a region defined by the postal service | | "DPID" | - | Delivery Point Identifier : A value that uniquely identifies the postal address | |

All Direct / Indirect Based Attributes (1):

[ADXP](#b17)/@[type](#b15) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **AddressPartType** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "AL" | - | Address Line : An address line is for either an additional locator, a delivery address or a street address. An address generally has only a delivery address line or a street address line, but not both | | "ADL" | - | Additional Locator : This can be a unit designator, such as apartment number, suite number, or floor. There may be several unit designators in an address (e.g., "3rd floor, Appt. 342"). This can also be a designator pointing away from the location, rather than specifying a smaller location within some larger one (e.g., Dutch "t.o." means "opposite to" for house boats located across the street facing houses) | | "UNID" | - | Unit Identifier : The number or name of a specific unit contained within a building or complex, as assigned by that building or complex | | "UNIT" | - | Unit Designator : Indicates the type of specific unit contained within a building or complex. E.g. Apartment, Floor | | "DAL" | - | Delivery Address Line : A delivery address line is frequently used instead of breaking out delivery mode, delivery installation, etc. An address generally has only a delivery address line or a street address line, but not both. | | "DINST" | - | Delivery Installation Type : Indicates the type of delivery installation (the facility to which the mail will be delivered prior to final shipping via the delivery mode.) Example: post office, letter carrier depot, community mail center, station, etc. | | "DINSTA" | - | Delivery Installation Area : The location of the delivery installation, usually a town or city, and is only required if the area is different from the municipality. Area to which mail delivery service is provided from any postal facility or service such as an individual letter carrier, rural route, or postal route | | "DINSTQ" | - | Delivery Installation Qualifier : A number, letter or name identifying a delivery installation. E.g., for Station A, the delivery installation qualifier would be 'A'. | | "DMOD" | - | Delivery Mode : Indicates the type of service offered, method of delivery. For example: post office box, rural route, general delivery, etc. | | "DMODID" | - | Delivery Mode Identifier: Represents the routing information such as a letter carrier route number. It is the identifying number of the designator (the box number or rural route number) | | "SAL" | - | Street Address Line : A street address line is frequently used instead of breaking out build number, street name, street type, etc. An address generally has only a delivery address line or a street address line, but not both. | | "BNR" | - | Building Number : The number of a building, house or lot alongside the street. Also known as "primary street number". This does not number the street but rather the building | | "BNN" | - | Building Number Numeric : The numeric portion of a building number | | "BNS" | - | Building Number Suffix : Any alphabetic character, fraction or other text that may appear after the numeric portion of a building number | | "STR" | - | Street Name : The name of the street, including the type | | "STB" | - | Street Name Base : The base name of a roadway or artery recognized by a municipality (excluding street type and direction) | | "STTYP" | - | Street Type : The designation given to the street. (e.g. Street, Avenue, Crescent, etc.) | | "DIR" | - | Direction : Direction (e.g., N, S, W, E) | | "INT" | - | Intersection : An intersection denotes that the actual address is located at or close to the intersection of two or more streets | | "CAR" | - | Care of : The name of the party who will take receipt at the specified address, and will take on responsibility for ensuring delivery to the target recipient.  Note: This is included only to support the convention of writing c/- address lines. This item is not appropriate for use when information is entrusted to one party on behalf of another in some significant way. | | "CEN" | - | Census Tract : A geographic sub-unit delineated for demographic purposes | | "CNT" | - | Country : Country | | "CPA" | - | County or Parish : A sub-unit of a state or province. (49 of the United States of America use the term "county;" Louisiana uses the term "parish") | | "CTY" | - | Municipality : The name of the city, town, village, or other community or delivery center | | "DEL" | - | Delimiter : Delimiters are printed without framing white space. If no value component is provided, the delimiter appears as a line break | | "POB" | - | Post box : A numbered box located in a post station | | "PRE" | - | Precinct : A subsection of a municipality | | "STA" | - | State or Province : A sub-unit of a country with limited sovereignty in a federally organized country | | "ZIP" | - | Postal Code : A postal code designating a region defined by the postal service | | "DPID" | - | Delivery Point Identifier : A value that uniquely identifies the postal address | | |

XML Source (w/o annotations (30))

<xs:simpleType name="[**AddressPartType**](#b152)">

<xs:restriction base="xs:string">

<xs:enumeration value="AL"/>

<xs:enumeration value="ADL"/>

<xs:enumeration value="UNID"/>

<xs:enumeration value="UNIT"/>

<xs:enumeration value="DAL"/>

<xs:enumeration value="DINST"/>

<xs:enumeration value="DINSTA"/>

<xs:enumeration value="DINSTQ"/>

<xs:enumeration value="DMOD"/>

<xs:enumeration value="DMODID"/>

<xs:enumeration value="SAL"/>

<xs:enumeration value="BNR"/>

<xs:enumeration value="BNN"/>

<xs:enumeration value="BNS"/>

<xs:enumeration value="STR"/>

<xs:enumeration value="STB"/>

<xs:enumeration value="STTYP"/>

<xs:enumeration value="DIR"/>

<xs:enumeration value="INT"/>

<xs:enumeration value="CAR"/>

<xs:enumeration value="CEN"/>

<xs:enumeration value="CNT"/>

<xs:enumeration value="CPA"/>

<xs:enumeration value="CTY"/>

<xs:enumeration value="DEL"/>

<xs:enumeration value="POB"/>

<xs:enumeration value="PRE"/>

<xs:enumeration value="STA"/>

<xs:enumeration value="ZIP"/>

<xs:enumeration value="DPID"/>

</xs:restriction>

</xs:simpleType>

simpleType "Code"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| xs:string |

All Direct / Indirect Based Attributes (6):

|  |  |
| --- | --- |
| [ATTR\_PQ](#b205)/@[unit](#b203) [],  [ATTR\_PQ\_HIGH](#b210)/@[highUnit](#b208) [],  [ATTR\_PQ\_LOW](#b215)/@[lowUnit](#b213) [], | [CD](#b33)/@[code](#b27) [],  [CodeLiteral](#b389)/@[code](#b385) [],  [CodedOrdinalLiteral](#b381)/@[code](#b376) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **Code** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |

XML Source

<xs:simpleType name="[**Code**](#b155)">

<xs:restriction base="xs:string"/>

</xs:simpleType>

simpleType "Decimal"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| xs:double |

All Direct / Indirect Based Attributes (17):

|  |  |
| --- | --- |
| [ATTR\_PQ](#b205)/@[value](#b202) [],  [ATTR\_PQ\_HIGH](#b210)/@[highValue](#b207) [],  [ATTR\_PQ\_LOW](#b215)/@[lowValue](#b212) [],  [ATTR\_RTO](#b220)/@[denominator](#b218) [],  [ATTR\_RTO](#b220)/@[numerator](#b217) [],  [ATTR\_RTO\_HIGH](#b225)/@[highDenominator](#b223) [],  [ATTR\_RTO\_HIGH](#b225)/@[highNumerator](#b222) [],  [ATTR\_RTO\_LOW](#b230)/@[lowDenominator](#b228) [],  [ATTR\_RTO\_LOW](#b230)/@[lowNumerator](#b227) [], | [CO](#b40)/@[value](#b36) [],  [CodedOrdinalLiteral](#b381)/@[value](#b379) [],  [IVL\_REAL](#b100)/@[high](#b96) [],  [IVL\_REAL](#b100)/@[low](#b95) [],  [REAL](#b123)/@[value](#b121) [],  [RealIntervalLiteral](#b944)/@[high](#b940) [],  [RealIntervalLiteral](#b944)/@[low](#b939) [],  [RealLiteral](#b949)/@[value](#b947) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:double (restriction)  **Decimal** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:double |

XML Source

<xs:simpleType name="[**Decimal**](#b158)">

<xs:restriction base="xs:double"/>

</xs:simpleType>

simpleType "EntityNamePartQualifier"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "LS" | - | Legal Status : For organizations a suffix indicating the legal status, e.g., "Inc.", "Co.", "AG", "GmbH", "B.V." "S.A.", "Ltd." Etc. | | "AC" | - | Academic : Indicates that a prefix like "Dr." or a suffix like "M.D." or "Ph.D." is an academic title | | "NB" | - | Nobility : In Europe and Asia, there are still people with nobility titles (aristocrats). German "von" is generally a nobility title, not a mere voorvoegsel. Others are "Earl of" or "His Majesty King of..." etc. Rarely used nowadays, but some systems do keep track of this | | "PR" | - | Professional : Primarily in the British Imperial culture people tend to have an abbreviation of their professional organization as part of their credential suffices | | "HON" | - | Honorific : A honorific such as "The Right Honourable" or "Weledelgeleerde Heer". | | "BR" | - | Birth : A name that a person was given at birth or established as a consequence of adoption.  Note: this is not used for temporary names assigned at birth such as "Baby of Smith" - which is just a name with a use code of "TEMP". | | "AD" | - | Acquired : A name part a person acquired.  The name part may be acquired by adoption, or the person may have chosen to use the name part for some other reason.  Note: this differs from an Other/Psuedonym/Alias in that an acquired name part is acquired on a formal basis rather than an informal one (e.g. registered as part of the official name) | | "SP" | - | Spouse : The name assumed from the partner in a marital relationship. Usually the spouse's family name. Note that no inference about gender can be made from the existence of spouse names | | "MID" | - | Middle Name : Indicates that the name part is a middle name. Usage Notes: In general, the english 'middle name' concept is all of the given names after the first. This qualifier may be used to explicitly indicate which given names are considered to be middle names. The middle name qualifier may also be used with family names. This is a Scandinavian use case, matching the concept of "mellomnavn" / "mellannamn". Note that there are specific rules that indicate what names may be taken as a mellannamn in different Scandinavian countries | | "CL" | - | Callme : Callme is used to indicate which of the various name parts is used when interacting with the person | | "IN" | - | Initial : Indicates that a name part is just an initial. Initials do not imply a trailing period since this would not work with non-Latin scripts. Initials may consist of more than one letter, e.g., "Ph." could stand for "Philippe" or "Th." for "Thomas" | | "PFX" | - | Prefix : A prefix has a strong association to the immediately following name part. A prefix has no implicit trailing white space (it has implicit leading white space though). | | "SFX" | - | Suffix : A suffix has a strong association to the immediately preceding name part. A suffix has no implicit leading white space (it has implicit trailing white space though). | |

Known Direct Subtypes (1):

[set\_EntityNamePartQualifier](#b173) []

All Direct / Indirect Based Attributes (1):

[ENXP](#b58)/@[qualifier](#b56) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **EntityNamePartQualifier** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "LS" | - | Legal Status : For organizations a suffix indicating the legal status, e.g., "Inc.", "Co.", "AG", "GmbH", "B.V." "S.A.", "Ltd." Etc. | | "AC" | - | Academic : Indicates that a prefix like "Dr." or a suffix like "M.D." or "Ph.D." is an academic title | | "NB" | - | Nobility : In Europe and Asia, there are still people with nobility titles (aristocrats). German "von" is generally a nobility title, not a mere voorvoegsel. Others are "Earl of" or "His Majesty King of..." etc. Rarely used nowadays, but some systems do keep track of this | | "PR" | - | Professional : Primarily in the British Imperial culture people tend to have an abbreviation of their professional organization as part of their credential suffices | | "HON" | - | Honorific : A honorific such as "The Right Honourable" or "Weledelgeleerde Heer". | | "BR" | - | Birth : A name that a person was given at birth or established as a consequence of adoption.  Note: this is not used for temporary names assigned at birth such as "Baby of Smith" - which is just a name with a use code of "TEMP". | | "AD" | - | Acquired : A name part a person acquired.  The name part may be acquired by adoption, or the person may have chosen to use the name part for some other reason.  Note: this differs from an Other/Psuedonym/Alias in that an acquired name part is acquired on a formal basis rather than an informal one (e.g. registered as part of the official name) | | "SP" | - | Spouse : The name assumed from the partner in a marital relationship. Usually the spouse's family name. Note that no inference about gender can be made from the existence of spouse names | | "MID" | - | Middle Name : Indicates that the name part is a middle name. Usage Notes: In general, the english 'middle name' concept is all of the given names after the first. This qualifier may be used to explicitly indicate which given names are considered to be middle names. The middle name qualifier may also be used with family names. This is a Scandinavian use case, matching the concept of "mellomnavn" / "mellannamn". Note that there are specific rules that indicate what names may be taken as a mellannamn in different Scandinavian countries | | "CL" | - | Callme : Callme is used to indicate which of the various name parts is used when interacting with the person | | "IN" | - | Initial : Indicates that a name part is just an initial. Initials do not imply a trailing period since this would not work with non-Latin scripts. Initials may consist of more than one letter, e.g., "Ph." could stand for "Philippe" or "Th." for "Thomas" | | "PFX" | - | Prefix : A prefix has a strong association to the immediately following name part. A prefix has no implicit trailing white space (it has implicit leading white space though). | | "SFX" | - | Suffix : A suffix has a strong association to the immediately preceding name part. A suffix has no implicit leading white space (it has implicit trailing white space though). | | |

XML Source (w/o annotations (13))

<xs:simpleType name="[**EntityNamePartQualifier**](#b161)">

<xs:restriction base="xs:string">

<xs:enumeration value="LS"/>

<xs:enumeration value="AC"/>

<xs:enumeration value="NB"/>

<xs:enumeration value="PR"/>

<xs:enumeration value="HON"/>

<xs:enumeration value="BR"/>

<xs:enumeration value="AD"/>

<xs:enumeration value="SP"/>

<xs:enumeration value="MID"/>

<xs:enumeration value="CL"/>

<xs:enumeration value="IN"/>

<xs:enumeration value="PFX"/>

<xs:enumeration value="SFX"/>

</xs:restriction>

</xs:simpleType>

simpleType "EntityNamePartType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "FAM" | - | Family : Family name, this is the name that links to the genealogy. In some cultures (e.g. Eritrea) the family name of a son is the first name of his father | | "GIV" | - | Given: Given name. Note: don't call it "first name" since this given names do not always come first | | "TITLE" | - | Title : Part of the name that is acquired as a title due to academic, legal, employment or nobility status etc. Note: Title name parts include name parts that come after the name such as qualifications | | "DEL" | - | Delimiter : A delimiter has no meaning other than being literally printed in this name representation. A delimiter has no implicit leading and trailing white space | |

All Direct / Indirect Based Attributes (1):

[ENXP](#b58)/@[type](#b55) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **EntityNamePartType** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "FAM" | - | Family : Family name, this is the name that links to the genealogy. In some cultures (e.g. Eritrea) the family name of a son is the first name of his father | | "GIV" | - | Given: Given name. Note: don't call it "first name" since this given names do not always come first | | "TITLE" | - | Title : Part of the name that is acquired as a title due to academic, legal, employment or nobility status etc. Note: Title name parts include name parts that come after the name such as qualifications | | "DEL" | - | Delimiter : A delimiter has no meaning other than being literally printed in this name representation. A delimiter has no implicit leading and trailing white space | | |

XML Source (w/o annotations (5))

<xs:simpleType name="[**EntityNamePartType**](#b164)">

<xs:restriction base="xs:string">

<xs:enumeration value="FAM"/>

<xs:enumeration value="GIV"/>

<xs:enumeration value="TITLE"/>

<xs:enumeration value="DEL"/>

</xs:restriction>

</xs:simpleType>

simpleType "EntityNameUse"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "ABC" | - | Alphabetic : Alphabetic transcription of name (Japanese: romaji) | | "SYL" | - | Syllabic : Syllabic transcription of name (e.g., Japanese kana, Korean hangul) | | "IDE" | - | Ideographic : Ideographic representation of name (e.g., Japanese kanji, Chinese characters) | | "C" | - | Customary : Known as/conventional/the one you normally use | | "OR" | - | Official Registry Name : the formal name as registered in an official (government) registry, but which name might not be commonly used. May correspond to the concept of legal name | | "T" | - | Temporary : A temporary name. Note that a name valid time can provide more detailed information. This may also be used for temporary names assigned at birth or in emergency situations. | | "I" | - | Indigenous/Tribal : e.g. Chief Red Cloud | | "P" | - | Other/Pseudonym/Alias: A non-official name by which the person is sometimes known. (This may also be used to record informal names such as a nickname) | | "ANON" | - | Anonymous : Anonymous assigned name (used to protect a person's identity for privacy reasons) | | "A" | - | Business Name : A name used in a Professional or Business context .  Examples: Continuing to use a maiden name in a professional context, or using a stage performing name (some of these names are also pseudonyms) | | "R" | - | Religious : A name assumed as part of a religious vocation. e.g. Sister Mary Francis, Brother John | | "OLD" | - | No Longer in Use : This name is no longer in use (note: Names may also carry valid time ranges . This code is used to cover the situations where it is known that the name is no longer valid, but no particular time range for its use is known) | | "DN" | - | Do Not Use : This name should no longer be used when interacting with the person (i.e . in addition to no longer being used, the name should not be even mentioned when interacting with the person)  Note: applications are not required to compare names labeled "Do Not Use" and other names in order to eliminate name parts that are common between the other name and a name labeled "Do Not Use". | | "M" | - | Maiden Name : A name used prior to marriage.  Note that marriage naming customs vary greatly around the world. This name use is for use by applications that collect and store "maiden" names. Though the concept of maiden name is often gender specific, the use of this term is not gender specific. The use of this term does not imply any particular history for a person's name, nor should the maiden name be determined algorithmically | | "SRCH" | - | Search Type Uses: A name intended for use in searching or matching | | "PHON" | - | Phonetic : The name as understood by the data enterer, i.e. a close approximation of a phonetic spelling of the name, not based on a phonetic algorithm. | |

Known Direct Subtypes (1):

[set\_EntityNameUse](#b176) []

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [EN](#b52)/@[use](#b48) [], | [EntityNameLiteral](#b528)/@[use](#b524) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **EntityNameUse** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "ABC" | - | Alphabetic : Alphabetic transcription of name (Japanese: romaji) | | "SYL" | - | Syllabic : Syllabic transcription of name (e.g., Japanese kana, Korean hangul) | | "IDE" | - | Ideographic : Ideographic representation of name (e.g., Japanese kanji, Chinese characters) | | "C" | - | Customary : Known as/conventional/the one you normally use | | "OR" | - | Official Registry Name : the formal name as registered in an official (government) registry, but which name might not be commonly used. May correspond to the concept of legal name | | "T" | - | Temporary : A temporary name. Note that a name valid time can provide more detailed information. This may also be used for temporary names assigned at birth or in emergency situations. | | "I" | - | Indigenous/Tribal : e.g. Chief Red Cloud | | "P" | - | Other/Pseudonym/Alias: A non-official name by which the person is sometimes known. (This may also be used to record informal names such as a nickname) | | "ANON" | - | Anonymous : Anonymous assigned name (used to protect a person's identity for privacy reasons) | | "A" | - | Business Name : A name used in a Professional or Business context .  Examples: Continuing to use a maiden name in a professional context, or using a stage performing name (some of these names are also pseudonyms) | | "R" | - | Religious : A name assumed as part of a religious vocation. e.g. Sister Mary Francis, Brother John | | "OLD" | - | No Longer in Use : This name is no longer in use (note: Names may also carry valid time ranges . This code is used to cover the situations where it is known that the name is no longer valid, but no particular time range for its use is known) | | "DN" | - | Do Not Use : This name should no longer be used when interacting with the person (i.e . in addition to no longer being used, the name should not be even mentioned when interacting with the person)  Note: applications are not required to compare names labeled "Do Not Use" and other names in order to eliminate name parts that are common between the other name and a name labeled "Do Not Use". | | "M" | - | Maiden Name : A name used prior to marriage.  Note that marriage naming customs vary greatly around the world. This name use is for use by applications that collect and store "maiden" names. Though the concept of maiden name is often gender specific, the use of this term is not gender specific. The use of this term does not imply any particular history for a person's name, nor should the maiden name be determined algorithmically | | "SRCH" | - | Search Type Uses: A name intended for use in searching or matching | | "PHON" | - | Phonetic : The name as understood by the data enterer, i.e. a close approximation of a phonetic spelling of the name, not based on a phonetic algorithm. | | |

XML Source (w/o annotations (16))

<xs:simpleType name="[**EntityNameUse**](#b167)">

<xs:restriction base="xs:string">

<xs:enumeration value="ABC"/>

<xs:enumeration value="SYL"/>

<xs:enumeration value="IDE"/>

<xs:enumeration value="C"/>

<xs:enumeration value="OR"/>

<xs:enumeration value="T"/>

<xs:enumeration value="I"/>

<xs:enumeration value="P"/>

<xs:enumeration value="ANON"/>

<xs:enumeration value="A"/>

<xs:enumeration value="R"/>

<xs:enumeration value="OLD"/>

<xs:enumeration value="DN"/>

<xs:enumeration value="M"/>

<xs:enumeration value="SRCH"/>

<xs:enumeration value="PHON"/>

</xs:restriction>

</xs:simpleType>

simpleType "PostalAddressUse"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "H" | - | Home address : A communication address at a home, attempted contacts for business purposes might intrude privacy and chances are one will contact family or other household members instead of the person one wishes to call. Typically used with urgent cases, or if no other contacts are available | | "HP" | - | Primary home : The primary home, to reach a person after business hours | | "HV" | - | Vacation home : A vacation home, to reach a person while on vacation | | "WP" | - | Work place : An office address. First choice for business related contacts during business hours | | "DIR" | - | Direct : Indicates a work place address or telecommunication address that reaches the individual or organization directly without intermediaries. For phones, often referred to as a 'private line' | | "PUB" | - | Public : Indicates a work place address or telecommunication address that is a 'standard' address which may reach a reception service, mail-room, or other intermediary prior to the target entity | | "BAD" | - | Bad address : A flag indicating that the address is bad, in fact, useless | | "PHYS" | - | Physical Visit Address : Used primarily to visit an address | | "PST" | - | Postal Address : Used to send mail | | "TMP" | - | Temporary Address : A temporary address, may be good for visit or mailing. Note that an address history can provide more detailed information. | | "ABC" | - | Alphabetic : Alphabetic transcription of name (Japanese: romaji) | | "IDE" | - | Ideographic : Ideographic representation of name (e.g., Japanese kanji, Chinese characters) | | "SYL" | - | Syllabic : Syllabic transcription of name (e.g., Japanese kana, Korean hangul) | | "SRCH" | - | Search Type Uses : A name intended for use in searching or matching. | | "SNDX" | - | Soundex : An address spelled according to the SoundEx algorithm | | "PHON" | - | Phonetic : The address as understood by the data enterer, i.e. a close approximation of a phonetic spelling of the address, not based on a phonetic algorithm | |

Known Direct Subtypes (1):

[set\_PostalAddressUse](#b179) []

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [AD](#b12)/@[use](#b8) [], | [AddressLiteral](#b289)/@[use](#b285) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **PostalAddressUse** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "H" | - | Home address : A communication address at a home, attempted contacts for business purposes might intrude privacy and chances are one will contact family or other household members instead of the person one wishes to call. Typically used with urgent cases, or if no other contacts are available | | "HP" | - | Primary home : The primary home, to reach a person after business hours | | "HV" | - | Vacation home : A vacation home, to reach a person while on vacation | | "WP" | - | Work place : An office address. First choice for business related contacts during business hours | | "DIR" | - | Direct : Indicates a work place address or telecommunication address that reaches the individual or organization directly without intermediaries. For phones, often referred to as a 'private line' | | "PUB" | - | Public : Indicates a work place address or telecommunication address that is a 'standard' address which may reach a reception service, mail-room, or other intermediary prior to the target entity | | "BAD" | - | Bad address : A flag indicating that the address is bad, in fact, useless | | "PHYS" | - | Physical Visit Address : Used primarily to visit an address | | "PST" | - | Postal Address : Used to send mail | | "TMP" | - | Temporary Address : A temporary address, may be good for visit or mailing. Note that an address history can provide more detailed information. | | "ABC" | - | Alphabetic : Alphabetic transcription of name (Japanese: romaji) | | "IDE" | - | Ideographic : Ideographic representation of name (e.g., Japanese kanji, Chinese characters) | | "SYL" | - | Syllabic : Syllabic transcription of name (e.g., Japanese kana, Korean hangul) | | "SRCH" | - | Search Type Uses : A name intended for use in searching or matching. | | "SNDX" | - | Soundex : An address spelled according to the SoundEx algorithm | | "PHON" | - | Phonetic : The address as understood by the data enterer, i.e. a close approximation of a phonetic spelling of the address, not based on a phonetic algorithm | | |

XML Source (w/o annotations (17))

<xs:simpleType name="[**PostalAddressUse**](#b170)">

<xs:restriction base="xs:string">

<xs:enumeration value="H"/>

<xs:enumeration value="HP"/>

<xs:enumeration value="HV"/>

<xs:enumeration value="WP"/>

<xs:enumeration value="DIR"/>

<xs:enumeration value="PUB"/>

<xs:enumeration value="BAD"/>

<xs:enumeration value="PHYS"/>

<xs:enumeration value="PST"/>

<xs:enumeration value="TMP"/>

<xs:enumeration value="ABC"/>

<xs:enumeration value="IDE"/>

<xs:enumeration value="SYL"/>

<xs:enumeration value="SRCH"/>

<xs:enumeration value="SNDX"/>

<xs:enumeration value="PHON"/>

</xs:restriction>

</xs:simpleType>

simpleType "set\_EntityNamePartQualifier"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *list of* ("LS" | "AC" | "NB" | "PR" | "HON" | "BR" | "AD" | "SP" | "MID" | "CL" | "IN" | "PFX" | "SFX") |

All Direct / Indirect Based Attributes (1):

[ENXP](#b58)/@[qualifier](#b56) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [EntityNamePartQualifier](#b161) [] (list)  **set\_EntityNamePartQualifier** |

|  |  |
| --- | --- |
| Derivation: | list of [EntityNamePartQualifier](#b161) |

XML Source

<xs:simpleType name="[**set\_EntityNamePartQualifier**](#b173)">

<xs:list itemType="[**EntityNamePartQualifier**](#b161)"/>

</xs:simpleType>

simpleType "set\_EntityNameUse"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *list of* ("ABC" | "SYL" | "IDE" | "C" | "OR" | "T" | "I" | "P" | "ANON" | "A" | "R" | "OLD" | "DN" | "M" | "SRCH" | "PHON") |

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [EN](#b52)/@[use](#b48) [], | [EntityNameLiteral](#b528)/@[use](#b524) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [EntityNameUse](#b167) [] (list)  **set\_EntityNameUse** |

|  |  |
| --- | --- |
| Derivation: | list of [EntityNameUse](#b167) |

XML Source

<xs:simpleType name="[**set\_EntityNameUse**](#b176)">

<xs:list itemType="[**EntityNameUse**](#b167)"/>

</xs:simpleType>

simpleType "set\_PostalAddressUse"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "PHYS" | "PST" | "TMP" | "ABC" | "IDE" | "SYL" | "SRCH" | "SNDX" | "PHON") |

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [AD](#b12)/@[use](#b8) [], | [AddressLiteral](#b289)/@[use](#b285) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [PostalAddressUse](#b170) [] (list)  **set\_PostalAddressUse** |

|  |  |
| --- | --- |
| Derivation: | list of [PostalAddressUse](#b170) |

XML Source

<xs:simpleType name="[**set\_PostalAddressUse**](#b179)">

<xs:list itemType="[**PostalAddressUse**](#b170)"/>

</xs:simpleType>

simpleType "set\_TelecommunicationAddressUse"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "TMP" | "AS" | "EC" | "MC" | "PG") |

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [TEL](#b139)/@[use](#b135) [], | [UrlLiteral](#b1110)/@[use](#b1107) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [TelecommunicationAddressUse](#b188) [] (list)  **set\_TelecommunicationAddressUse** |

|  |  |
| --- | --- |
| Derivation: | list of [TelecommunicationAddressUse](#b188) |

XML Source

<xs:simpleType name="[**set\_TelecommunicationAddressUse**](#b182)">

<xs:list itemType="[**TelecommunicationAddressUse**](#b188)"/>

</xs:simpleType>

simpleType "set\_TelecommunicationCapability"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *list of* ("voice" | "fax" | "data" | "tty" | "sms") |

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [TEL](#b139)/@[capabilities](#b136) [], | [UrlLiteral](#b1110)/@[capabilities](#b1108) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [TelecommunicationCapability](#b191) [] (list)  **set\_TelecommunicationCapability** |

|  |  |
| --- | --- |
| Derivation: | list of [TelecommunicationCapability](#b191) |

XML Source

<xs:simpleType name="[**set\_TelecommunicationCapability**](#b185)">

<xs:list itemType="[**TelecommunicationCapability**](#b191)"/>

</xs:simpleType>

simpleType "TelecommunicationAddressUse"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "H" | - | Home address : A communication address at a home, attempted contacts for business purposes might intrude privacy and chances are one will contact family or other household members instead of the person one wishes to call. Typically used with urgent cases, or if no other contacts are available | | "HP" | - | Primary Home : The primary home, to reach a person after business hours | | "HV" | - | Vacation Home : A vacation home, to reach a person while on vacation | | "WP" | - | Work Place : An office address. First choice for business related contacts during business hours | | "DIR" | - | Direct : Indicates a work place address or telecommunication address that reaches the individual or organization directly without intermediaries. For phones, often referred to as a 'private line' | | "PUB" | - | Public : Indicates a work place address or telecommunication address that is a 'standard' address which may reach a reception service, mail-room, or other intermediary prior to the target entity | | "BAD" | - | Bad address : A flag indicating that the address is bad, in fact, useless | | "TMP" | - | Temporary address : A temporary address, may be good for visit or mailing. Note that an address history can provide more detailed information | | "AS" | - | Answering Service : An automated answering machine used for less urgent cases and if the main purpose of contact is to leave a message or access an automated announcement | | "EC" | - | Emergency Contact : A contact specifically designated to be used for emergencies. This is the first choice in emergencies, independent of any other use codes | | "MC" | - | Mobile Contact : A telecommunication device that moves and stays with its owner. May have characteristics of all other use codes, suitable for urgent matters, not the first choice for routine business | | "PG" | - | Pager: A paging device suitable to solicit a callback or to leave a very short message | |

Known Direct Subtypes (1):

[set\_TelecommunicationAddressUse](#b182) []

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [TEL](#b139)/@[use](#b135) [], | [UrlLiteral](#b1110)/@[use](#b1107) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **TelecommunicationAddressUse** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "H" | - | Home address : A communication address at a home, attempted contacts for business purposes might intrude privacy and chances are one will contact family or other household members instead of the person one wishes to call. Typically used with urgent cases, or if no other contacts are available | | "HP" | - | Primary Home : The primary home, to reach a person after business hours | | "HV" | - | Vacation Home : A vacation home, to reach a person while on vacation | | "WP" | - | Work Place : An office address. First choice for business related contacts during business hours | | "DIR" | - | Direct : Indicates a work place address or telecommunication address that reaches the individual or organization directly without intermediaries. For phones, often referred to as a 'private line' | | "PUB" | - | Public : Indicates a work place address or telecommunication address that is a 'standard' address which may reach a reception service, mail-room, or other intermediary prior to the target entity | | "BAD" | - | Bad address : A flag indicating that the address is bad, in fact, useless | | "TMP" | - | Temporary address : A temporary address, may be good for visit or mailing. Note that an address history can provide more detailed information | | "AS" | - | Answering Service : An automated answering machine used for less urgent cases and if the main purpose of contact is to leave a message or access an automated announcement | | "EC" | - | Emergency Contact : A contact specifically designated to be used for emergencies. This is the first choice in emergencies, independent of any other use codes | | "MC" | - | Mobile Contact : A telecommunication device that moves and stays with its owner. May have characteristics of all other use codes, suitable for urgent matters, not the first choice for routine business | | "PG" | - | Pager: A paging device suitable to solicit a callback or to leave a very short message | | |

XML Source (w/o annotations (13))

<xs:simpleType name="[**TelecommunicationAddressUse**](#b188)">

<xs:restriction base="xs:string">

<xs:enumeration value="H"/>

<xs:enumeration value="HP"/>

<xs:enumeration value="HV"/>

<xs:enumeration value="WP"/>

<xs:enumeration value="DIR"/>

<xs:enumeration value="PUB"/>

<xs:enumeration value="BAD"/>

<xs:enumeration value="TMP"/>

<xs:enumeration value="AS"/>

<xs:enumeration value="EC"/>

<xs:enumeration value="MC"/>

<xs:enumeration value="PG"/>

</xs:restriction>

</xs:simpleType>

simpleType "TelecommunicationCapability"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "voice" | - | Voice : This device can receive voice calls (i.e. talking to another person, or a recording device, or a voice activated computer) | | "fax" | - | Fax : This device can receive faxes. | | "data" | - | Data : This device can receive data calls (i.e. modem) | | "tty" | - | Text : This device is a text telephone. | | "sms" | - | SMS : This device can receive SMS messages | |

Known Direct Subtypes (1):

[set\_TelecommunicationCapability](#b185) []

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [TEL](#b139)/@[capabilities](#b136) [], | [UrlLiteral](#b1110)/@[capabilities](#b1108) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **TelecommunicationCapability** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "voice" | - | Voice : This device can receive voice calls (i.e. talking to another person, or a recording device, or a voice activated computer) | | "fax" | - | Fax : This device can receive faxes. | | "data" | - | Data : This device can receive data calls (i.e. modem) | | "tty" | - | Text : This device is a text telephone. | | "sms" | - | SMS : This device can receive SMS messages | | |

XML Source (w/o annotations (6))

<xs:simpleType name="[**TelecommunicationCapability**](#b191)">

<xs:restriction base="xs:string">

<xs:enumeration value="voice"/>

<xs:enumeration value="fax"/>

<xs:enumeration value="data"/>

<xs:enumeration value="tty"/>

<xs:enumeration value="sms"/>

</xs:restriction>

</xs:simpleType>

simpleType "Uid"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| xs:string |

All Direct / Indirect Based Attributes (5):

|  |  |
| --- | --- |
| [CD](#b33)/@[codeSystem](#b28) [],  [CodeLiteral](#b389)/@[codeSystem](#b384) [],  [CodedOrdinalLiteral](#b381)/@[codeSystem](#b375) [], | [II](#b64)/@[root](#b61) [],  [IdentifierLiteral](#b611)/@[root](#b608) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **Uid** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |

XML Source (w/o annotations (1))

<xs:simpleType name="[**Uid**](#b194)">

<xs:restriction base="xs:string"/>

</xs:simpleType>

simpleType "UncertaintyType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "U" | - | Uniform : The uniform distribution assigns a constant probability over the entire interval of possible outcomes, while all outcomes outside this interval are assumed to have zero probability. The width of this interval is 2 s v3. Thus, the uniform distribution assigns the probability densities f(x) = (2 s v3)-1 to values µ - s v3 = x = µ + s v3 and f(x) = 0 otherwise | | "N" | - | Normal (Gaussian) : This is the well-known bell-shaped normal distribution. Because of the central limit theorem, the normal distribution is the distribution of choice for an unbounded random variable that is an outcome of a combination of many stochastic processes. Even for values bounded on a single side (i.e. greater than 0) the normal distribution may be accurate enough if the mean is "far away" from the bound of the scale measured in terms of standard deviations | | "LN" | - | Log-Normal : The logarithmic normal distribution is used to transform skewed random variable X into a normally distributed random variable U = log X. The log-normal distribution can be specified with the properties mean µ and standard deviation s. Note however that mean µ and standard deviation s are the parameters of the raw value distribution, not the transformed parameters of the lognormal distribution that are conventionally referred to by the same letters. Those log-normal parameters µ log and slog relate to the mean µ and standard deviation s of the data value through slog2 = log (s2/µ2 + 1) and µlog = log µ - slog2/2 | | "G" | - | ? (gamma) : The gamma-distribution used for data that is skewed and bounded to the right, i.e. where the maximum of the distribution curve is located near the origin. The ?-distribution has two parameters a and ß. The relationship to mean µ and variance s2 is µ = a ß and s2 = a ß2 | | "E" | - | Exponential : Used for data that describes extinction. The exponential distribution is a special form of ?-distribution where a = 1, hence, the relationship to mean µ and variance s2 are µ = ß and s2 = ß2 | | "X2" | - | ? : Used to describe the sum of squares of random variables that occurs when a variance is estimated (rather than presumed) from the sample. The only parameter of the ?2-distribution is ?, so called the number of degrees of freedom (which is the number of independent parts in the sum). The ?2-distribution is a special type of ?-distribution with parameter a = ? /2 and ß = 2. Hence, µ = ? and s2 = 2 ? | | "T" | - | t (student) : Used to describe the quotient of a normal random variable and the square root of a ?2 random variable. The t-distribution has one parameter ?, the number of degrees of freedom. The relationship to mean µ and variance s2 are: µ = 0 and s2 = ? / (? - 2) | | "F" | - | F : Used to describe the quotient of two ?2 random variables. The F-distribution has two parameters ?1 and ?2, which are the numbers of degrees of freedom of the numerator and denominator variable respectively. The relationship to mean µ and variance s2 are: µ = ?2 / (?2 - 2) and s 2 = (2 ?22 (? 2 + ?1 - 2)) / (?1 (?2 - 2)2 (?2 - 4)) | | "B" | - | ?(beta) : The beta-distribution is used for data that is bounded on both sides and may or may not be skewed (e.g., occurs when probabilities are estimated.) Two parameters a and ß are available to adjust the curve. The mean µ and variance s2 relate as follows: µ = a / (a + ß) and (s2 = a ß/((a + ß)2 (a + ß + 1)) | |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **UncertaintyType** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "U" | - | Uniform : The uniform distribution assigns a constant probability over the entire interval of possible outcomes, while all outcomes outside this interval are assumed to have zero probability. The width of this interval is 2 s v3. Thus, the uniform distribution assigns the probability densities f(x) = (2 s v3)-1 to values µ - s v3 = x = µ + s v3 and f(x) = 0 otherwise | | "N" | - | Normal (Gaussian) : This is the well-known bell-shaped normal distribution. Because of the central limit theorem, the normal distribution is the distribution of choice for an unbounded random variable that is an outcome of a combination of many stochastic processes. Even for values bounded on a single side (i.e. greater than 0) the normal distribution may be accurate enough if the mean is "far away" from the bound of the scale measured in terms of standard deviations | | "LN" | - | Log-Normal : The logarithmic normal distribution is used to transform skewed random variable X into a normally distributed random variable U = log X. The log-normal distribution can be specified with the properties mean µ and standard deviation s. Note however that mean µ and standard deviation s are the parameters of the raw value distribution, not the transformed parameters of the lognormal distribution that are conventionally referred to by the same letters. Those log-normal parameters µ log and slog relate to the mean µ and standard deviation s of the data value through slog2 = log (s2/µ2 + 1) and µlog = log µ - slog2/2 | | "G" | - | ? (gamma) : The gamma-distribution used for data that is skewed and bounded to the right, i.e. where the maximum of the distribution curve is located near the origin. The ?-distribution has two parameters a and ß. The relationship to mean µ and variance s2 is µ = a ß and s2 = a ß2 | | "E" | - | Exponential : Used for data that describes extinction. The exponential distribution is a special form of ?-distribution where a = 1, hence, the relationship to mean µ and variance s2 are µ = ß and s2 = ß2 | | "X2" | - | ? : Used to describe the sum of squares of random variables that occurs when a variance is estimated (rather than presumed) from the sample. The only parameter of the ?2-distribution is ?, so called the number of degrees of freedom (which is the number of independent parts in the sum). The ?2-distribution is a special type of ?-distribution with parameter a = ? /2 and ß = 2. Hence, µ = ? and s2 = 2 ? | | "T" | - | t (student) : Used to describe the quotient of a normal random variable and the square root of a ?2 random variable. The t-distribution has one parameter ?, the number of degrees of freedom. The relationship to mean µ and variance s2 are: µ = 0 and s2 = ? / (? - 2) | | "F" | - | F : Used to describe the quotient of two ?2 random variables. The F-distribution has two parameters ?1 and ?2, which are the numbers of degrees of freedom of the numerator and denominator variable respectively. The relationship to mean µ and variance s2 are: µ = ?2 / (?2 - 2) and s 2 = (2 ?22 (? 2 + ?1 - 2)) / (?1 (?2 - 2)2 (?2 - 4)) | | "B" | - | ?(beta) : The beta-distribution is used for data that is bounded on both sides and may or may not be skewed (e.g., occurs when probabilities are estimated.) Two parameters a and ß are available to adjust the curve. The mean µ and variance s2 relate as follows: µ = a / (a + ß) and (s2 = a ß/((a + ß)2 (a + ß + 1)) | | |

XML Source (w/o annotations (9))

<xs:simpleType name="[**UncertaintyType**](#b197)">

<xs:restriction base="xs:string">

<xs:enumeration value="U"/>

<xs:enumeration value="N"/>

<xs:enumeration value="LN"/>

<xs:enumeration value="G"/>

<xs:enumeration value="E"/>

<xs:enumeration value="X2"/>

<xs:enumeration value="T"/>

<xs:enumeration value="F"/>

<xs:enumeration value="B"/>

</xs:restriction>

</xs:simpleType>

simpleType "Uri"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |

|  |
| --- |
| Simple Content Model |
| xs:anyURI |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:anyURI (restriction)  **Uri** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:anyURI |

XML Source (w/o annotations (1))

<xs:simpleType name="[**Uri**](#b200)">

<xs:restriction base="xs:anyURI"/>

</xs:simpleType>

## Datatypes.xsd Attribute Types

attributeGroup "ATTR\_PQ"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b202) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b202) | = | xs:double | | [unit](#b203) | = | xs:string | |
| ...> | |

Annotation

Attribute group defining a PQ

XML Source (w/o annotations (3))

<xs:attributeGroup name="[**ATTR\_PQ**](#b205)">

<xs:attribute name="[**value**](#b202)" type="[**Decimal**](#b158)" use="required"/>

<xs:attribute name="[**unit**](#b203)" type="[**Code**](#b155)" use="required"/>

</xs:attributeGroup>

Attribute Detail (all declarations; defined within [this](#b205) component only; 2/2)

 value

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | required |

The number which is multiplied by the unit to make the PQ.

Attribute Value

|  |
| --- |
| xs:double |

 unit

|  |  |
| --- | --- |
| Type: | [Code](#b155) [] |
| Use: | required |

The unit of measure specified in the Unified Code for Units of Measure (UCUM).  
UCUM defines two forms of expression, case sensitive and case insensitive. PQ uses the case sensitive codes. The codeSystem OID for the case sensitive form is 2.16.840.1.113883.6.8. The default value for unit is the UCUM code "1" (unity).  
Equality of physical quantities does not require the values and units to be equal independently. Value and unit is only how we represent physical quantities. For example, 1 m equals 100 cm. Although the units are different and the values are different, the physical quantities are equal. Therefore one should never expect a particular unit for a physical quantity but instead allow for automated conversion between different comparable units.  
The unit SHALL come from UCUM, which only specifies unambiguous measurement units. Sometimes it is not clear how some measurements in healthcare map to UCUM codes.  
Note: The general pattern for a measurement is value unit of Thing. In this scheme, the PQ represents the value and the unit, and the Thing is described by some coded concept that is linked to the PQ by the context of use. This maps obviously to some measurements, such as Patient Body Temperature of 37 Celsius, and 250 mg/day of Salicylate.  
However for some measurements that arise in healthcare, the scheme is not so obvious. Two classic examples are 5 Drinks of Beer, and 3 Acetaminophen tablets. At first glance it is tempting to classify these measurements like this: 5 drinks of Beer and 3 Acetaminophen tablets. The problem with this is that UCUM does not support units of "beer", "tablets" or "scoops".  
The reason for this is that neither tablets or scoops are proper units. What kind of tablets? How big is the glass? In these kinds of cases, the concept that appears to be a unit needs to further specified before interoperability is established. If a correct amount is required, then it is generally appropriate to specify an exact measurement with an appropriate UCUM unit. If this is not possible, then the concept is not part of the measurement. UCUM provides a unit called unity for use in these cases. The proper way to understand these measurements as 3 1 Acetaminophen tablets, where 1 is the UCUM unit for unity, and the Thing has a qualifier. The context of use will need to provide the extra qualifying information.

Attribute Value

|  |
| --- |
| xs:string |

attributeGroup "ATTR\_PQ\_HIGH"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b207) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [highValue](#b207) | = | xs:double | | [highUnit](#b208) | = | xs:string | |
| ...> | |

Annotation

Attribute group defining a IVL\_PQ high value

XML Source (w/o annotations (3))

<xs:attributeGroup name="[**ATTR\_PQ\_HIGH**](#b210)">

<xs:attribute name="[**highValue**](#b207)" type="[**Decimal**](#b158)" use="required"/>

<xs:attribute name="[**highUnit**](#b208)" type="[**Code**](#b155)" use="required"/>

</xs:attributeGroup>

Attribute Detail (all declarations; defined within [this](#b210) component only; 2/2)

 highValue

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | required |

The number which is multiplied by the unit to make the PQ.

Attribute Value

|  |
| --- |
| xs:double |

 highUnit

|  |  |
| --- | --- |
| Type: | [Code](#b155) [] |
| Use: | required |

The unit of measure specified in the Unified Code for Units of Measure (UCUM).  
UCUM defines two forms of expression, case sensitive and case insensitive. PQ uses the case sensitive codes. The codeSystem OID for the case sensitive form is 2.16.840.1.113883.6.8. The default value for unit is the UCUM code "1" (unity).  
Equality of physical quantities does not require the values and units to be equal independently. Value and unit is only how we represent physical quantities. For example, 1 m equals 100 cm. Although the units are different and the values are different, the physical quantities are equal. Therefore one should never expect a particular unit for a physical quantity but instead allow for automated conversion between different comparable units.  
The unit SHALL come from UCUM, which only specifies unambiguous measurement units. Sometimes it is not clear how some measurements in healthcare map to UCUM codes.  
Note: The general pattern for a measurement is value unit of Thing. In this scheme, the PQ represents the value and the unit, and the Thing is described by some coded concept that is linked to the PQ by the context of use. This maps obviously to some measurements, such as Patient Body Temperature of 37 Celsius, and 250 mg/day of Salicylate.  
However for some measurements that arise in healthcare, the scheme is not so obvious. Two classic examples are 5 Drinks of Beer, and 3 Acetaminophen tablets. At first glance it is tempting to classify these measurements like this: 5 drinks of Beer and 3 Acetaminophen tablets. The problem with this is that UCUM does not support units of "beer", "tablets" or "scoops".  
The reason for this is that neither tablets or scoops are proper units. What kind of tablets? How big is the glass? In these kinds of cases, the concept that appears to be a unit needs to further specified before interoperability is established. If a correct amount is required, then it is generally appropriate to specify an exact measurement with an appropriate UCUM unit. If this is not possible, then the concept is not part of the measurement. UCUM provides a unit called unity for use in these cases. The proper way to understand these measurements as 3 1 Acetaminophen tablets, where 1 is the UCUM unit for unity, and the Thing has a qualifier. The context of use will need to provide the extra qualifying information.

Attribute Value

|  |
| --- |
| xs:string |

attributeGroup "ATTR\_PQ\_LOW"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b212) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [lowValue](#b212) | = | xs:double | | [lowUnit](#b213) | = | xs:string | |
| ...> | |

Annotation

Attribute group defining a IVL\_PQ low value

XML Source (w/o annotations (3))

<xs:attributeGroup name="[**ATTR\_PQ\_LOW**](#b215)">

<xs:attribute name="[**lowValue**](#b212)" type="[**Decimal**](#b158)" use="required"/>

<xs:attribute name="[**lowUnit**](#b213)" type="[**Code**](#b155)" use="required"/>

</xs:attributeGroup>

Attribute Detail (all declarations; defined within [this](#b215) component only; 2/2)

 lowValue

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | required |

The number which is multiplied by the unit to make the PQ.

Attribute Value

|  |
| --- |
| xs:double |

 lowUnit

|  |  |
| --- | --- |
| Type: | [Code](#b155) [] |
| Use: | required |

The unit of measure specified in the Unified Code for Units of Measure (UCUM).  
UCUM defines two forms of expression, case sensitive and case insensitive. PQ uses the case sensitive codes. The codeSystem OID for the case sensitive form is 2.16.840.1.113883.6.8. The default value for unit is the UCUM code "1" (unity).  
Equality of physical quantities does not require the values and units to be equal independently. Value and unit is only how we represent physical quantities. For example, 1 m equals 100 cm. Although the units are different and the values are different, the physical quantities are equal. Therefore one should never expect a particular unit for a physical quantity but instead allow for automated conversion between different comparable units.  
The unit SHALL come from UCUM, which only specifies unambiguous measurement units. Sometimes it is not clear how some measurements in healthcare map to UCUM codes.  
Note: The general pattern for a measurement is value unit of Thing. In this scheme, the PQ represents the value and the unit, and the Thing is described by some coded concept that is linked to the PQ by the context of use. This maps obviously to some measurements, such as Patient Body Temperature of 37 Celsius, and 250 mg/day of Salicylate.  
However for some measurements that arise in healthcare, the scheme is not so obvious. Two classic examples are 5 Drinks of Beer, and 3 Acetaminophen tablets. At first glance it is tempting to classify these measurements like this: 5 drinks of Beer and 3 Acetaminophen tablets. The problem with this is that UCUM does not support units of "beer", "tablets" or "scoops".  
The reason for this is that neither tablets or scoops are proper units. What kind of tablets? How big is the glass? In these kinds of cases, the concept that appears to be a unit needs to further specified before interoperability is established. If a correct amount is required, then it is generally appropriate to specify an exact measurement with an appropriate UCUM unit. If this is not possible, then the concept is not part of the measurement. UCUM provides a unit called unity for use in these cases. The proper way to understand these measurements as 3 1 Acetaminophen tablets, where 1 is the UCUM unit for unity, and the Thing has a qualifier. The context of use will need to provide the extra qualifying information.

Attribute Value

|  |
| --- |
| xs:string |

attributeGroup "ATTR\_RTO"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b217) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [numerator](#b217) | = | xs:double | | [denominator](#b218) | = | xs:double | |
| ...> | |

Annotation

Attribute group defining a RTO numerator and denominator values

XML Source (w/o annotations (3))

<xs:attributeGroup name="[**ATTR\_RTO**](#b220)">

<xs:attribute name="[**numerator**](#b217)" type="[**Decimal**](#b158)" use="required"/>

<xs:attribute name="[**denominator**](#b218)" type="[**Decimal**](#b158)" use="required"/>

</xs:attributeGroup>

Attribute Detail (all declarations; defined within [this](#b220) component only; 2/2)

 numerator

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | required |

The quantity that is being divided in the ratio. This represents the numerator.expression.value from the ISO21090 datatypes.

Attribute Value

|  |
| --- |
| xs:double |

 denominator

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | required |

The quantity that divides the numerator in the ratio.  
The denominator SHALL not be zero. This represents the denominator.expression.value from the ISO 21090 datatypes.

Attribute Value

|  |
| --- |
| xs:double |

attributeGroup "ATTR\_RTO\_HIGH"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b222) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [highNumerator](#b222) | = | xs:double | | [highDenominator](#b223) | = | xs:double | |
| ...> | |

Annotation

Attribute group defining a RTO numerator and denominator values

XML Source (w/o annotations (3))

<xs:attributeGroup name="[**ATTR\_RTO\_HIGH**](#b225)">

<xs:attribute name="[**highNumerator**](#b222)" type="[**Decimal**](#b158)" use="required"/>

<xs:attribute name="[**highDenominator**](#b223)" type="[**Decimal**](#b158)" use="required"/>

</xs:attributeGroup>

Attribute Detail (all declarations; defined within [this](#b225) component only; 2/2)

 highNumerator

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | required |

The quantity that is being divided in the ratio. This represents the numerator.expression.value from the ISO21090 datatypes.

Attribute Value

|  |
| --- |
| xs:double |

 highDenominator

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | required |

The quantity that divides the numerator in the ratio.  
The denominator SHALL not be zero. This represents the denominator.expression.value from the ISO 21090 datatypes.

Attribute Value

|  |
| --- |
| xs:double |

attributeGroup "ATTR\_RTO\_LOW"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definitions of 2 [attributes](#b227) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [lowNumerator](#b227) | = | xs:double | | [lowDenominator](#b228) | = | xs:double | |
| ...> | |

Annotation

Attribute group defining a RTO numerator and denominator values

XML Source (w/o annotations (3))

<xs:attributeGroup name="[**ATTR\_RTO\_LOW**](#b230)">

<xs:attribute name="[**lowNumerator**](#b227)" type="[**Decimal**](#b158)" use="required"/>

<xs:attribute name="[**lowDenominator**](#b228)" type="[**Decimal**](#b158)" use="required"/>

</xs:attributeGroup>

Attribute Detail (all declarations; defined within [this](#b230) component only; 2/2)

 lowNumerator

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | required |

The quantity that is being divided in the ratio. This represents the numerator.expression.value from the ISO21090 datatypes.

Attribute Value

|  |
| --- |
| xs:double |

 lowDenominator

|  |  |
| --- | --- |
| Type: | [Decimal](#b158) [] |
| Use: | required |

The quantity that divides the numerator in the ratio.  
The denominator SHALL not be zero. This represents the denominator.expression.value from the ISO 21090 datatypes.

Attribute Value

|  |
| --- |
| xs:double |

attributeGroup "ATTR\_TS"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b233) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b233) | = | xs:string | |
| ...> | |

Annotation

Attribute group defining a TS

XML Source (w/o annotations (2))

<xs:attributeGroup name="[**ATTR\_TS**](#b235)">

<xs:attribute name="[**value**](#b233)" use="required">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:pattern value="[1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0-1][0-9])|(2[0-3]))(([0-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+\-](0[0-9]|1[0-3])([0-5][0-9]))?"/>

</xs:restriction>

</xs:simpleType>

</xs:attribute>

</xs:attributeGroup>

Attribute Detail (all declarations; defined within [this](#b235) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | [anonymous](#b232) simpleType (restriction of xs:string) [] |
| Use: | required |

The value of the TS. value is a string with the format "YYYY[MM[DD[HH[MM[SS[.U[U[U[U]]]]]]]]][+|-ZZzz]" that conforms to the constrained ISO 8601 defined in ISO 8824 (ASN.1) under clause 32 (generalized time). The format should be used to the degree of precision that is appropriate.

Attribute Value

|  |
| --- |
| xs:string |

|  |  |
| --- | --- |
| Pattern: | [1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0-1][0-9])|(2[0-3]))(([0‌-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+\-](0[0-9]|1[0-3])([0-5][0-9]))? |

Anonymous simpleType

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **simpleType** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | pattern: | [1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0‌-1][0-9])|(2[0-3]))(([0-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+‌\-](0[0-9]|1[0-3])([0-5][0-9]))? | |

attributeGroup "ATTR\_TS\_HIGH"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b238) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [high](#b238) | = | xs:string | |
| ...> | |

Annotation

Attribute group defining a IVL\_TS high value

XML Source (w/o annotations (2))

<xs:attributeGroup name="[**ATTR\_TS\_HIGH**](#b240)">

<xs:attribute name="[**high**](#b238)" use="optional">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:pattern value="[1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0-1][0-9])|(2[0-3]))(([0-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+\-](0[0-9]|1[0-3])([0-5][0-9]))?"/>

</xs:restriction>

</xs:simpleType>

</xs:attribute>

</xs:attributeGroup>

Attribute Detail (all declarations; defined within [this](#b240) component only; 1/1)

 high

|  |  |
| --- | --- |
| Type: | [anonymous](#b237) simpleType (restriction of xs:string) [] |
| Use: | optional |

This is the high limit. If the high limit is not known, it may be null.  
The high limit SHALL NOT be negative infinity, and SHALL be higher than the low limit if one exists.  
The value of the TS. value is a string with the format "YYYY[MM[DD[HH[MM[SS[.U[U[U[U]]]]]]]]][+|-ZZzz]" that conforms to the constrained ISO 8601 defined in ISO 8824 (ASN.1) under clause 32 (generalized time). The format should be used to the degree of precision that is appropriate.

Attribute Value

|  |
| --- |
| xs:string |

|  |  |
| --- | --- |
| Pattern: | [1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0-1][0-9])|(2[0-3]))(([0‌-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+\-](0[0-9]|1[0-3])([0-5][0-9]))? |

Anonymous simpleType

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **simpleType** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | pattern: | [1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0‌-1][0-9])|(2[0-3]))(([0-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+‌\-](0[0-9]|1[0-3])([0-5][0-9]))? | |

attributeGroup "ATTR\_TS\_LOW"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/cdsdt](#b5) |
| Includes: | definition of 1 [attribute](#b243) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [low](#b243) | = | xs:string | |
| ...> | |

Annotation

Attribute group defining a IVL\_TS low value

XML Source (w/o annotations (2))

<xs:attributeGroup name="[**ATTR\_TS\_LOW**](#b245)">

<xs:attribute name="[**low**](#b243)" use="optional">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:pattern value="[1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0-1][0-9])|(2[0-3]))(([0-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+\-](0[0-9]|1[0-3])([0-5][0-9]))?"/>

</xs:restriction>

</xs:simpleType>

</xs:attribute>

</xs:attributeGroup>

Attribute Detail (all declarations; defined within [this](#b245) component only; 1/1)

 low

|  |  |
| --- | --- |
| Type: | [anonymous](#b242) simpleType (restriction of xs:string) [] |
| Use: | optional |

This is the low limit. If the low limit is not known, it may be null.  
The low limit SHALL NOT be positive infinity.  
The value of the TS. value is a string with the format "YYYY[MM[DD[HH[MM[SS[.U[U[U[U]]]]]]]]][+|-ZZzz]" that conforms to the constrained ISO 8601 defined in ISO 8824 (ASN.1) under clause 32 (generalized time). The format should be used to the degree of precision that is appropriate.

Attribute Value

|  |
| --- |
| xs:string |

|  |  |
| --- | --- |
| Pattern: | [1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0-1][0-9])|(2[0-3]))(([0‌-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+\-](0[0-9]|1[0-3])([0-5][0-9]))? |

Anonymous simpleType

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **simpleType** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | pattern: | [1-2][0-9]{3,3}(((0[1-9])|(1[0-2]))(((0[1-9])|([1-2][0-9])|(3[0-1]))((([0‌-1][0-9])|(2[0-3]))(([0-5][0-9])(([0-5][0-9])(\.[0-9]{1,4})?)?)?)?)?)?([+‌\-](0[0-9]|1[0-3])([0-5][0-9]))? | |

## KnowledgeDocument.xsd Namespace Summary

Namespace "urn:hl7-org:v3/hed"

Targeting Schemas (1):

[knowledgedocument.xsd](#b1292)

Targeting Components:

[elements](#b246) (2 global + 212 local), [complexTypes](#b247) (168), [simpleTypes](#b248) (54)

## KnowledgeDocument.xsd Global Element Summary

|  |  |
| --- | --- |
| Global Element Summary | |
| [**itemDefinitions**](#b253) | A collection of item definitions   |  |  | | --- | --- | | Type: | anonymous complexType | |
| [**knowledgeDocument**](#b255) | knowledgeDocument represents a serialized Clinical Decision Support (CDS) knowledge artifact.   |  |  | | --- | --- | | Type: | [KnowledgeDocument](#b692) [] | |

## KnowledgeDocument.xsd Complex Type Summary

| **Complex Type Summary** | |
| --- | --- |
| [**Abs**](#b258) | The Abs operator returns the absolute value of its argument. |
| [***ActionBase***](#b267) | Actions are the output of the CDS system and represent the tasks that must be carried out by a human or a computer system. |
| [**ActionGroup**](#b275) | This type of action is used to organize a group of related actions into one container. |
| [**Actor**](#b279) | An actor is an entity responsible for the execution of an action. |
| [**Add**](#b282) | The Add operator performs numeric addition of its arguments. |
| [**AddressLiteral**](#b289) | Returns a value of type AD with the given attributes. |
| [**After**](#b292) | Returns true if the first interval starts after the second one ends. |
| [***AggregateExpression***](#b299) | Aggregate expressions perform operations on lists of data, either directly on a list of scalars, or indirectly on a list of objects, with a reference to a property present on each object in the list. |
| [**AllTrue**](#b302) | Returns true if all the elements in source are true. |
| [**And**](#b305) | The And operator returns the logical conjunction of its arguments. |
| [**AnyTrue**](#b308) | Returns true if any element in source is true. |
| [**ArtifactLifeCycleEvent**](#b313) | An event in the life cycle of an artifact. |
| [***AtomicAction***](#b318) | An action that is not further broken down into constituent actions. |
| [**Avg**](#b321) | Returns the average of the elements in source. |
| [**Before**](#b324) | Returns true if the first interval ends before the second one starts. |
| [**Begin**](#b327) | Returns the starting point of the interval operand. |
| [***Behavior***](#b329) | A behavior may be specified for a specific action or a group of actions. |
| [**Behaviors**](#b333) |  |
| [**BinaryExpression**](#b338) | The BinaryExpression type defines the abstract base type for all expressions that take two arguments. |
| [**BooleanLiteral**](#b343) | Returns a value of type BL with the given attributes. |
| [**Case**](#b350) | The case operator allows for multiple conditional expressions to be chained together in a single expression, rather than having to nest multiple Conditional operators. |
| [**CaseItem**](#b355) |  |
| [**Ceiling**](#b358) | The Ceiling operator returns the first integer greater than or equal to the argument. |
| [**ClinicalRequest**](#b369) | The clinical request expression defines clinical data that will be used by the artifact. |
| [**Coalesce**](#b372) | Returns the result of the first argument that evaluates is not null. |
| [**CodedOrdinalLiteral**](#b381) | Returns a value of type CO with the given attributes. |
| [**CodeLiteral**](#b389) | Returns a value of type CD with the given attributes. |
| [**CollectInformationAction**](#b396) | This action requests information from the actor. |
| [**Combine**](#b402) | The Combine operator combines a list of strings, optionally separating each string with the given separator. |
| [**ComplexLiteral**](#b407) | The ComplexLiteral expression allows an xml literal of any type to be included in an expression. |
| [**Concat**](#b410) | The Concat operator performs string concatenation of its arguments. |
| [**Condition**](#b415) | A condition specifies when a knowledge component is to be executed. |
| [**Conditional**](#b422) | The Conditional operator evaluates a condition, and returns the then argument if condition evaluates to true; otherwise the result of the else argument is returned. |
| [**Conditions**](#b426) | A collection of conditions that are used to define whether various aspects of the artifact, such as whether or not a particular action should be executed, or whether a particular order set item is applicable to a given patient. |
| [**Contains**](#b432) | Returns true if the given source contains the given element. |
| [**Contribution**](#b437) | A contribution is made by a specific contributor (organization, person, etc.), and was made in a particular way, as specified by the contributor's role. |
| [**Convert**](#b442) | Converts a value to a specific type. |
| [**Count**](#b445) | Returns the number of non-null elements in the source. |
| [**Coverage**](#b451) | Specifies various attributes of the patient population for whom and/or environment of care in which the CDS artifact is applicable. |
| [**CreateAction**](#b456) | A new action to be executed by a user or a computer system. |
| [**Current**](#b461) | The Current expression returns the value of the object currently in scope. |
| [**Date**](#b472) | Constructs a date/time value from the given components. |
| [**DateAdd**](#b479) | DateAdd adds numberOfPeriods date periods of the specified granularity to the given date. |
| [**DateDiff**](#b486) | DateDiff returns the number of granularity boundaries occurring between startDate and endDate. |
| [**DatePart**](#b492) | DatePart returns the granularity component of the given date. |
| [**DeclareResponseAction**](#b497) | The DeclareResponseAction provides a mechanism to declare a container for responses provided by the user in response to CollectInformationActions. |
| [**Difference**](#b500) | Returns the difference of the two operands. |
| [**Distinct**](#b505) | The Distinct operator takes a list of elements and returns a list containing only the unique elements within the input. |
| [**Divide**](#b508) | The Divide operator performs numeric division of its arguments. |
| [**DocumentationItem**](#b518) | An item type representing the definition of an individual item to be recorded in a structured clinical document. |
| [**End**](#b521) | Returns the ending point of the interval operand. |
| [**EntityNameLiteral**](#b528) | Returns a value of type EN with the given attributes. |
| [**Equal**](#b531) | The Equal operator returns true if the arguments are equal; and false otherwise. |
| [**Evidence**](#b537) | Reference to research on which the artifact is based. |
| [***Expression***](#b541) | The Expression type defines the abstract base type for all expressions used in the HeDS expression language. |
| [**ExpressionConstraint**](#b546) | A constraint specified in the form of an expression. |
| [**ExpressionDef**](#b552) | The ExpressionDef type defines an expression and an associated name that can be referenced by any expression in the artifact. |
| [**ExpressionRef**](#b557) | The ExpressionRef type defines an expression that references a previously defined NamedExpression. |
| [**Filter**](#b565) | The filter expression returns a list with only those elements in the source list for which the condition element evaluates to true. |
| [**FireEventAction**](#b571) | This action fires an event. |
| [**First**](#b578) | Returns the first element in a list. |
| [**Floor**](#b581) | The Floor operator returns the first integer less than or equal to the argument. |
| [**ForEach**](#b589) | The ForEach expression iterates over the list of elements in the source element, and returns a list with the same number of elements, where each element in the new list is the result of evaluating the element expression for each element in the source list. |
| [**Greater**](#b592) | Returns true if the first argument is greater than the second argument. |
| [**GreaterOrEqual**](#b595) | Returns true if the first argument is greater than or equal to the second argument. |
| [**GroupOrganizationBehavior**](#b600) | For a group of actions, specifies the organizational intent of the grouping. |
| [**GroupSelectionBehavior**](#b605) | For a group of actions, specifies the number of actions that may be chosen by an end user. |
| [**IdentifierLiteral**](#b611) | Returns a value of type II with the given attributes. |
| [**IfNull**](#b614) | If the first argument evaluates to null, returns the result of the second argument. |
| [**In**](#b620) | Returns true if the given element is in the source element. |
| [**Indexer**](#b626) | Returns the indexth element in a string or list. |
| [**IndexOf**](#b632) | IndexOf returns the 1-based index of the given element in the given source list. |
| [**InlineResource**](#b637) | An Inline Resource consists of both the resource reference information and the actual resource content/payload to be inserted inline. |
| [**IntegerIntervalLiteral**](#b645) | Returns a value of type IVL\_INT with the given attributes. |
| [**IntegerLiteral**](#b650) | Returns a value of type INT with the given attributes. |
| [**Intersect**](#b653) | Returns the intersection of the operands. |
| [**Interval**](#b662) | The Interval selector defines an interval value. |
| [**InValueSet**](#b669) | Returns true if the given code, or list of codes, is in the given value set. |
| [**IsEmpty**](#b672) | Returns true if the list contains no elements. |
| [**IsNotEmpty**](#b675) | Returns true if the list contains any elements. |
| [**IsNull**](#b678) | Returns true if the argument evaluates to null, false otherwise. |
| [***ItemDefinition***](#b682) | An item definition is the equivalent of an item in a data dictionary or a catalog in an electronic health record system. |
| [**KnowledgeDocument**](#b692) | A knowledge document is an instance of a CDS knowledge artifact such as a rule, an order set, or a documentation template |
| [**KnowledgeResource**](#b701) | KnowledgeResource specifies a reference to an associated resource of relevance to the artifact such as a guideline, a performance measure, another knowledge artifact, or a source of evidence for the artifact. |
| [**Last**](#b708) | Returns the last element in a list. |
| [**Length**](#b711) | Returns the length of its argument. |
| [**Less**](#b714) | Returns true if the first argument is less than the second argument. |
| [**LessOrEqual**](#b717) | Returns true if the first argument is less than or equal to the second argument. |
| [**List**](#b724) | The List selector returns a value of type List, whose elements are the result of evaluating the arguments to the List selector, in order. |
| [**ListConstraint**](#b731) | A constraint specifying that the value is from a list included here. |
| [**Literal**](#b737) | The Literal type defines a single scalar value. |
| [**Ln**](#b740) | The Ln operator computes the natural logarithm of its argument. |
| [**Log**](#b743) | The Log operator computes the logarithm of its first argument, using the second argument as the base. |
| [**Lower**](#b746) | Returns the lower case of its argument. |
| [**Max**](#b749) | Returns the maximum element in the source. |
| [**Meets**](#b752) | Returns true if the first interval ends on the starting point of the second, or if the first interval starts on the ending point of the second. |
| [**MessageAction**](#b757) | An action to deliver a message to a destination using an optionally specified channel. |
| [**Metadata**](#b779) | The container for all of the metadata associated with a CDS knowledge artifact. |
| [**Min**](#b782) | Returns the minimum element in the source. |
| [**ModelReference**](#b787) | A reference to some model by its Universal Resource Identifier. |
| [**Modulo**](#b790) | The Modulo operator computes the remainder of the division of its arguments. |
| [**Multiply**](#b793) | The Multiply operator performs numeric multiplication of its arguments. |
| [***NaryExpression***](#b798) | The Expression type defines an abstract base class for an expression that takes any number of arguments, including zero. |
| [**Negate**](#b801) | The Negate operator returns the negative of its argument. |
| [**Not**](#b804) | The Not operator returns the logical negation of its argument. |
| [**NotEqual**](#b807) | The NotEqual operator returns true if its argument are not the same value. |
| [**Now**](#b810) | Returns the date and time of the start timestamp associated with the evaluation request. |
| [**ObjectExpression**](#b817) | The ObjectExpression type allows objects of any type to be built up as an expression. |
| [**ObjectRedefine**](#b825) | The ObjectRedefine expression returns an object of the same type as the source argument, with the same values for each property, except the values of the properties listed in the property elements. |
| [**Or**](#b828) | The Or operator returns the logical disjunction of its arguments. |
| [**Organization**](#b833) | Identifies an organization, a corporation, an institution, or a government department that has relevance to the knowledge artifact. |
| [**Overlaps**](#b836) | Returns true if the first interval overlaps the second. |
| [**ParameterDef**](#b843) | ParameterDef defines a parameter that can be referenced by name anywhere within an expression. |
| [**ParameterRef**](#b848) | The ParameterRef expression allows the value of a parameter to be referenced as part of an expression. |
| [***Party***](#b853) | Party represents the abstract base type for entities that have addresses and contact information. |
| [**PeriodLiteral**](#b862) | Returns a value of type PIVL\_TS with the given attributes. |
| [**Person**](#b868) | Identifies a person who is associated with the knowledge artifact. |
| [**PhysicalQuantityIntervalLiteral**](#b874) | Returns a value of type IVL\_PQ with the given attributes. |
| [**PhysicalQuantityLiteral**](#b877) | Returns a value of type PQ with the given attributes. |
| [**PIVL\_TS**](#b886) | Note: because this type is defined as an extension of SXCM\_T, all of the attributes and elements accepted for T are also accepted by this definition. |
| [**Pos**](#b892) | Pos returns the 1-based index of the given pattern in the given string. |
| [**Power**](#b895) | The Power operator raises the first argument to the power given by the second argument. |
| [**PrecheckBehavior**](#b900) | For a particular action, specifies how often the action is expected to be selected in the particular context of the group containing that action. |
| [**Pred**](#b903) | The Pred operator returns the predecessor of the argument. |
| [**ProperContains**](#b906) | Returns true if the first operand contains the second, and is strictly larger. |
| [**ProperIn**](#b909) | Returns true if the first operand is in the second, and is strictly smaller. |
| [**Property**](#b917) | Returns the value of the property on source specified by the path attribute. |
| [**PropertyExpression**](#b923) | The PropertyExpression type is used within the ObjectLiteral type to provide the value of a specific property within an object literal expression. |
| [***RangeConstraint***](#b927) |  |
| [**RatioIntervalLiteral**](#b933) | Returns a value of type IVL\_RTO with the given attributes. |
| [**RatioLiteral**](#b936) | Returns a value of type RTO with the given attributes. |
| [**RealIntervalLiteral**](#b944) | Returns a value of type IVL\_REAL with the given attributes. |
| [**RealLiteral**](#b949) | Returns a value of type Real with the given attributes. |
| [**RemoveAction**](#b954) | This action removes another proposed action or an ongoing action. |
| [***RequestBase***](#b967) | The request expression defines the data that will be used by the artifact. |
| [**RequiredBehavior**](#b972) | For a single action, specifies what level of requiredness is associated with the action. |
| [**ResourceRelationshipReference**](#b977) | ResourceRelationshipReference defines the association between a resource and a resource set. |
| [**ResponseBinding**](#b982) | Defines the attributes required to specify a binding path for documentation item responses. |
| [**RightsDeclaration**](#b988) | This specifies the intellectual property rights associated with this CDS knowledge artifact, including who the rights holder is and what rights they assert. |
| [**Round**](#b994) | The Round operator returns the nearest integer to its argument. |
| [**SetSubsumes**](#b1000) | This operator returns the list of descendants that were subsumed by some code in the list of ancestors. |
| [**SimpleCodeLiteral**](#b1005) | Returns a value of type CS with the given attributes. |
| [**Sort**](#b1012) | Returns a list with all the elements in source, sorted by the given orderBy. |
| [**Split**](#b1018) | The Split operator splits a string into a list of strings using a separator. |
| [**StringLiteral**](#b1023) | Returns a value of type ST with the given attributes. |
| [**Substring**](#b1030) | Substring returns the string within stringToSub, beginning at the 1-based index startIndex, and consisting of length characters. |
| [**Subsumes**](#b1036) | This operator returns true if the operands were of the same code system, and the ancestor operand subsumed the descendant operand in the hierarchy of the code system. |
| [**Subtract**](#b1039) | The Subtract operator performs numeric subtraction of its arguments. |
| [**Succ**](#b1042) | The Succ operator returns the successor of the argument. |
| [**Sum**](#b1045) | Returns the sum of non-null elements in the source. |
| [**SupportingEvidence**](#b1049) | The evidence grade and the sources of evidence associated with this artifact. |
| [**SupportingResource**](#b1053) | Didactic or other informational resources associated with the artifact that can be provided to the CDS recipient. |
| [**SXCM\_TS**](#b1058) |  |
| [**TernaryExpression**](#b1063) | The TernaryExpression type defines the abstract base type for all expressions that take three arguments. |
| [**TimestampIntervalLiteral**](#b1069) | Returns a value of type IVL\_TS with the given attributes. |
| [**TimestampLiteral**](#b1072) | Returns a value of type TS with the given attributes. |
| [**Today**](#b1075) | Returns the date (with no time component) of the start timestamp associated with the evaluation request. |
| [**Trigger**](#b1080) |  |
| [**Triggers**](#b1084) |  |
| [**TruncatedDivide**](#b1087) | The TruncatedDivide operator performs integer division of its arguments. |
| [***UnaryExpression***](#b1092) | The UnaryExpression type defines the abstract base type for all expressions that take a single argument. |
| [**Union**](#b1095) | Returns the union of the operands. |
| [**UpdateAction**](#b1100) | This action changing the value of another existing action. |
| [**Upper**](#b1103) | Returns the upper case of its argument. |
| [**UrlLiteral**](#b1110) | Returns a value of type TEL with the given attributes. |
| [**ValueSet**](#b1117) | Returns a list of codes whose elements are defined by the given value set authority for the given value set id and version. |
| [**VersionedIdentifier**](#b1122) | VersionedIdentifier is composed of two parts: (1) an II identifier which identifies the set of all versions of a given resource. (2) the actual version of the instance of interest in this set. |
| [**Within**](#b1128) | Returns true if the given element is in the given interval. |

## KnowledgeDocument.xsd SimpleType Summary

| Simple Type Summary | |
| --- | --- |
| [**ArtifactLifeCycleEventType**](#b1131) | A version of an artifact may have different actions performed on it during the course of its life cycle. |
| [**ArtifactLifeCycleEventTypeCore**](#b1134) | A lifecycle event is applicable to a specific version of an artifact. |
| [**ArtifactLifeCycleEventTypeExt**](#b1137) | To add new items to the enumeration ArtifactLifeCycleEvent, comment out or delete the restriction on ArtifactLifeCycleEventTypeCore. |
| [**ArtifactStatusType**](#b1140) | A specific status is associated with each version of an artifact. |
| [**ArtifactStatusTypeCore**](#b1143) |  |
| [**ArtifactStatusTypeExt**](#b1146) | To add new items to the enumeration ArtifactStatus, comment out or delete the restriction on ArtifactStatusTypeCore. |
| [**ArtifactType**](#b1149) | Three types of artifacts are in scope for Health eDecisions Use Case #1: Event-condition-action rules, order sets and documentation templates. |
| [**ArtifactTypeCore**](#b1152) |  |
| [**ArtifactTypeExt**](#b1155) |  |
| [**CalendarCycle**](#b1158) | vocSet: D10684 (C-0-D10684-cpt) |
| [**CalendarCycleOneLetter**](#b1161) | abstDomain: V10701 (C-0-D10684-V10701-cpt) |
| [**CalendarCycleTwoLetter**](#b1164) | abstDomain: V10685 (C-0-D10684-V10685-cpt) |
| [**Cardinality**](#b1167) | Cardinality defines the expected cardinality of an element, single or multiple. |
| [**CompressionAlgorithm**](#b1170) | vocSet: D10620 (C-0-D10620-cpt) |
| [**ConditionRoleType**](#b1173) | The roles that a condition plays in the execution of a component. |
| [**ConditionRoleTypeCore**](#b1176) |  |
| [**ConditionRoleTypeExt**](#b1179) |  |
| [**ContributorType**](#b1182) | Enumeration of roles that contribute to the development and maintenance of a knowledge artifact. |
| [**ContributorTypeCore**](#b1185) |  |
| [**ContributorTypeExt**](#b1188) |  |
| [**CoverageType**](#b1191) | Specifies clinical metadata that can be used to retrieve, index and/or categorize the knowledge artifact. |
| [**CoverageTypeCore**](#b1194) |  |
| [**CoverageTypeExt**](#b1197) |  |
| [**DataEventType**](#b1200) | Enumeration of types of events related to access, creation, removal, or update of data. |
| [**DataEventTypeCore**](#b1203) |  |
| [**DataEventTypeExt**](#b1206) |  |
| [**DateGranularity**](#b1209) | DateGranularity specifies the granularities available for temporal operations such as DateAdd, DateDiff, and DatePart. |
| [**EventType**](#b1212) | An enumeration of event types. |
| [**EventTypeCore**](#b1215) |  |
| [**EventTypeExt**](#b1218) |  |
| [**GregorianCalendarCycle**](#b1221) | abstDomain: V10758 (C-0-D10684-V10685-V10758-cpt) |
| [**GroupOrganizationBehaviorType**](#b1224) | Defines organization behavior of a group: gives the reason why the items are grouped together. |
| [**GroupOrganizationBehaviorTypeCore**](#b1227) |  |
| [**GroupOrganizationBehaviorTypeExt**](#b1230) |  |
| [**GroupSelectionBehaviorType**](#b1233) | Defines selection behavior of a group: specifies the number of selectable items in the group that may be selected by the end user when the items of the group are displayed. |
| [**GroupSelectionBehaviorTypeCore**](#b1236) |  |
| [**GroupSelectionBehaviorTypeExt**](#b1239) |  |
| [**PrecheckBehaviorType**](#b1242) | Defines selection frequency behavior for an action or group; i.e., for most frequently selected items, the end-user system may provide convenience options in the UI (such as pre-selection) in order to (1) communicate to the end user what the most frequently selected item is, or should, be in a particular context, and (2) save the end user time. |
| [**PrecheckBehaviorTypeCore**](#b1245) |  |
| [**PrecheckBehaviorTypeExt**](#b1248) |  |
| [**RangeConstraintType**](#b1251) | The enumeration of different types of range constraints on values. |
| [**RangeConstraintTypeCore**](#b1254) |  |
| [**RangeConstraintTypeExt**](#b1257) |  |
| [**RequestCardinality**](#b1260) | RequestCardinality defines the expected cardinality of the request, single or multiple. |
| [**RequiredBehaviorType**](#b1263) | Defines requiredness behavior for selecting an action or an action group; i.e., whether the action or action group is required or optional. |
| [**RequiredBehaviorTypeCore**](#b1266) |  |
| [**RequiredBehaviorTypeExt**](#b1269) |  |
| [**ResourceRelationshipType**](#b1272) | A specific status is associated with each version of an artifact. |
| [**ResourceRelationshipTypeCore**](#b1275) |  |
| [**ResourceRelationshipTypeExt**](#b1278) | To add new items to the enumeration ArtifactStatus, comment out or delete the restriction on ArtifactStatusTypeCore. |
| [**SetOperator**](#b1281) |  |
| [**ValueType**](#b1284) | A specification of a constraint on the range of values for an item. |
| [**ValueTypeCore**](#b1287) |  |
| [**ValueTypeExt**](#b1290) |  |

## KnowledgeDocument.xsd Elements

element <itemDefinitions> (global)

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Type: | anonymous complexType |
| Includes: | definition of 1 [element](#b251) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <itemDefinitions> | |
|  | |  |  | | --- | --- | | *Content:* | [item](#b251)\* | |
| </itemDefinitions> | |

Content model elements (1):

[item](#b251) (in [itemDefinitions](#b253)) []

Annotation

A collection of item definitions

XML Source (w/o annotations (1))

<xs:element name="[**itemDefinitions**](#b253)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" minOccurs="0" name="[**item**](#b251)" type="[**ItemDefinition**](#b682)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

Content Element Detail (all declarations; defined within [this](#b253) component only; 1/1)

 item

|  |  |
| --- | --- |
| Type: | [*ItemDefinition*](#b682) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <item> | |
|  | |  |  | | --- | --- | | *Content:* | [identifier](#b680)? | |
| </item> | |

element <knowledgeDocument> (global)

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Type: | [KnowledgeDocument](#b692) [] |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <knowledgeDocument> | |
|  | |  |  | | --- | --- | | *Content:* | [metadata](#b684), [externalData](#b685)?, [expressions](#b686)?, [triggers](#b687)?, [conditions](#b688)?, [behaviors](#b689)?, [actionGroups](#b690) | |
| </knowledgeDocument> | |

Content model elements (7):

|  |  |
| --- | --- |
| [actionGroups](#b690) [],  [behaviors](#b689) (in [knowledgeDocument](#b255)) [],  [conditions](#b688) (in [knowledgeDocument](#b255)) [],  [expressions](#b686) [], | [externalData](#b685) [],  [metadata](#b684) [],  [triggers](#b687) [] |

Annotation

knowledgeDocument represents a serialized Clinical Decision Support (CDS) knowledge artifact. It is intended to define a general serialization format for an Order Set, a Event-Condition-Action (ECA) Rule, and a Document Template. Given the general intended purpose for this schema, it is important to note that the serialization of a given CDS artifact is defined by both this schema, in conjunction with a template defining artifact-specific constraints. For additional information on these artifact-specific constraints, please refer to the Implementation Guide.

XML Source (w/o annotations (1))

<xs:element name="[**knowledgeDocument**](#b255)" type="[**KnowledgeDocument**](#b692)"/>

## KnowledgeDocument.xsd Complex Types

complexType "Abs"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

The Abs operator returns the absolute value of its argument.  
If the argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Abs** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Abs**](#b258)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "ActionBase"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definitions of 6 [elements](#b260) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)? | |
| </...> | |

Content Model Elements (6):

|  |  |
| --- | --- |
| [actionId](#b260) [],  [actors](#b263) [],  [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [], | [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [],  [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [],  [supportingResources](#b262) [] |

Known Direct Subtypes (2):

[ActionGroup](#b275) [], [*AtomicAction*](#b318) []

Known Indirect Subtypes (7):

[CollectInformationAction](#b396) [], [CreateAction](#b456) [], [DeclareResponseAction](#b497) [], [FireEventAction](#b571) [], [MessageAction](#b757) [], [RemoveAction](#b954) [], [UpdateAction](#b1100) []

All Direct / Indirect Based Elements (3):

[actionGroups](#b690) [], group, simpleAction

Annotation

Actions are the output of the CDS system and  
represent the tasks that must be carried out by a human or a  
computer system.

XML Source (w/o annotations (7))

<xs:complexType abstract="true" name="[**ActionBase**](#b267)">

<xs:sequence minOccurs="0">

<xs:element maxOccurs="1" minOccurs="0" name="[**actionId**](#b260)" type="[**dt:II**](#b64)"/>

<xs:element minOccurs="0" name="[**supportingEvidence**](#b261)" type="[**SupportingEvidence**](#b1049)"/>

<xs:element minOccurs="0" name="[**supportingResources**](#b262)" type="[**SupportingResource**](#b1053)"/>

<!-- TODO -->

<xs:element maxOccurs="1" minOccurs="0" name="[**actors**](#b263)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="actor" type="[**Actor**](#b279)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**behaviors**](#b264)" type="[**Behaviors**](#b333)"/>

<!-- Additional conditions specific to the action execution -->

<xs:element maxOccurs="1" minOccurs="0" name="[**conditions**](#b265)" type="[**Conditions**](#b426)"/>

<!-- <xs:element name="applicableScenario" type="Expression" minOccurs="0"  
maxOccurs="1"> <xs:annotation> <xs:documentation>The clinical (sub) scenario  
in which this action is applicable. This scenario is additive to any scenarios  
specified in the containers of this action such as action groups or the knowledge  
document. In other words, this scenario does not override the scenario constraints  
specified in the container elements. </xs:documentation> </xs:annotation>  
</xs:element> -->

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b267) component only; 6/6)

 actionId

|  |  |
| --- | --- |
| Type: | [dt:II](#b64) [], empty content |

An identifier for the action. The identifier must  
be unique within the scope of the artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <actionId | |
|  | |  |  |  | | --- | --- | --- | | [extension](#b62) | = | xs:string : "" | | [root](#b61) | = | xs:string | |
| /> | |

 supportingEvidence

|  |  |
| --- | --- |
| Type: | [SupportingEvidence](#b1049) [], complex content |

The evidence grade and the sources of evidence  
associated with this artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <supportingEvidence> | |
|  | |  |  | | --- | --- | | *Content:* | [evidence](#b1047)+ | |
| </supportingEvidence> | |

 supportingResources

|  |  |
| --- | --- |
| Type: | [SupportingResource](#b1053) [], complex content |

Didactic or other informational resources  
associated with the action that can be provided to the CDS  
recipient. Information resources can include inline text  
commentary and links to web resources.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <supportingResources> | |
|  | |  |  | | --- | --- | | *Content:* | [resource](#b1051)+ | |
| </supportingResources> | |

 actors

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <actors> | |
|  | |  |  | | --- | --- | | *Content:* | actor+ | |
| </actors> | |

 behaviors

|  |  |
| --- | --- |
| Type: | [Behaviors](#b333) [], complex content |

The behaviors associated with how the action is  
presented and executed. The semantics and the validity of  
behaviors for actions are described elsewhere.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <behaviors> | |
|  | |  |  | | --- | --- | | *Content:* | [behavior](#b331)+ | |
| </behaviors> | |

 conditions

|  |  |
| --- | --- |
| Type: | [Conditions](#b426) [], complex content |

The conditions section lists all conditions that  
pertain to the action. Conditions define the logic that determine  
the applicability of the action in the given context, any  
precondition or post condition, and/or any inclusion and exclusion  
criteria for the given action.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <conditions> | |
|  | |  |  | | --- | --- | | *Content:* | [condition](#b424)+ | |
| </conditions> | |

complexType "ActionGroup"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 4 [elements](#b270) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)?, [title](#b270)?, [description](#b271)?, [representedConcepts](#b272)?, [subElements](#b273) | |
| </...> | |

Content Model Elements (10):

|  |  |
| --- | --- |
| [actionId](#b260) [],  [actors](#b263) [],  [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [],  [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [],  [description](#b271) (defined in [ActionGroup](#b275) complexType) [], | [representedConcepts](#b272) [],  [subElements](#b273) [],  [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [],  [supportingResources](#b262) [],  [title](#b270) (defined in [ActionGroup](#b275) complexType) [] |

All Direct / Indirect Based Elements (2):

[actionGroups](#b690) [], group

Annotation

This type of action is used to organize a group of  
related actions into one container.  
  
The semantics of how the group's  
subelements interact with which each  
other and how the subelements  
might be presented are specified in  
the group behavior.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*ActionBase*](#b267) [] (extension)  **ActionGroup** |

XML Source (w/o annotations (8))

<xs:complexType name="[**ActionGroup**](#b275)">

<xs:complexContent>

<xs:extension base="[**ActionBase**](#b267)">

<xs:sequence>

<xs:element minOccurs="0" name="[**title**](#b270)" type="[**dt:ST**](#b131)"/>

<xs:element minOccurs="0" name="[**description**](#b271)" type="[**dt:ST**](#b131)"/>

<xs:element minOccurs="0" name="[**representedConcepts**](#b272)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="concept" type="[**dt:CD**](#b33)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="[**subElements**](#b273)">

<xs:complexType>

<xs:sequence>

<xs:choice maxOccurs="unbounded">

<xs:element name="simpleAction" type="[**AtomicAction**](#b318)"/>

<xs:element name="group" type="[**ActionGroup**](#b275)"/>

<xs:element name="groupReference" type="[**VersionedIdentifier**](#b1122)"/>

</xs:choice>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b275) component only; 4/10)

 title

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

A brief title that is shown to the user of the  
artifact, i.e., the recipient of the CDS

|  |  |
| --- | --- |
| XML Representation Summary | |
| <title | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 description

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

A lengthier description of the action group  
that can  
be displayed to the user or the recipient of the CDS

|  |  |
| --- | --- |
| XML Representation Summary | |
| <description | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 representedConcepts

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

The concept(s) represented by this action group. For instance, a section may represent a group of beta-blockers, a composite orderable such as an insulin sliding scale, or a set of order sentences for a particular orderable (e.g., Tylenol).

|  |  |
| --- | --- |
| XML Representation Summary | |
| <representedConcepts> | |
|  | |  |  | | --- | --- | | *Content:* | concept+ | |
| </representedConcepts> | |

 subElements

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

The constituent elements of the group can be of  
different types including subgroups, simple or atomic actions,  
and subgroups embedded by reference.  
  
While the group allows  
artifacts of different types to be mixed and  
matched in any  
combination, a particular type of artifact might  
further restrict  
the combinations. For example, an artifact type  
might require  
subelements of a particular group to be either  
groups or simple  
actions; elements of both types cannot exist in  
the group.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <subElements> | |
|  | |  |  | | --- | --- | | *Content:* | (simpleAction | group | groupReference)+ | |
| </subElements> | |

complexType "Actor"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b277) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [actor](#b277) | |
| </...> | |

Content Model Elements (1):

[actor](#b277) (in actor in actors) []

All Direct / Indirect Based Elements (1):

actor (in [actors](#b263))

Annotation

An actor is an entity responsible for the execution of an action.

XML Source (w/o annotations (2))

<xs:complexType name="[**Actor**](#b279)">

<xs:sequence>

<xs:element name="[**actor**](#b277)" type="[**ExpressionDef**](#b552)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b279) component only; 1/1)

 actor

|  |  |
| --- | --- |
| Type: | [ExpressionDef](#b552) [], complex content |

The expression must evaluate to one of the following:  
1. CD or a List of CD. In this case, the actor defines the role or roles of entities to execute the action.  
2. II or a List of II. In this case, the actor defines entities, such as a provider or a team.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <actor | |
|  | |  |  |  | | --- | --- | --- | | [name](#b548) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [expression](#b550) | |
| </actor> | |

complexType "Add"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

The Add operator performs numeric addition of its arguments.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Add** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Add**](#b282)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "AddressLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b285), 1 [element](#b287) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [use](#b285) | = | *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "PHYS" | "PST" | "TMP" | "ABC" | "IDE" | "SYL" | "SRCH" | "SNDX" | "PHON") | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [part](#b287)+ | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [part](#b287) (defined in [AddressLiteral](#b289) complexType) [] |

Annotation

Returns a value of type AD with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **AddressLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**AddressLiteral**](#b289)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="unbounded" name="[**part**](#b287)" type="[**dt:ADXP**](#b17)"/>

</xs:sequence>

<xs:attribute name="[**use**](#b285)" type="[**dt:set\_PostalAddressUse**](#b179)" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b289) component only; 1/1)

 use

|  |  |
| --- | --- |
| Type: | [dt:set\_PostalAddressUse](#b179) [] |
| Use: | optional |

Attribute Value

|  |
| --- |
| *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "PHYS" | "PST" | "TMP" | "ABC" | "IDE" | "SYL" | "SRCH" | "SNDX" | "PHON") |

Content Element Detail (all declarations; defined within [this](#b289) component only; 1/2)

 part

|  |  |
| --- | --- |
| Type: | [dt:ADXP](#b17) [], empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <part | |
|  | |  |  |  | | --- | --- | --- | | [type](#b15) | = | ("AL" | "ADL" | "UNID" | "UNIT" | "DAL" | "DINST" | "DINSTA" | "DINSTQ" | "DMOD" | "DMODID" | "SAL" | "BNR" | "BNN" | "BNS" | "STR" | "STB" | "STTYP" | "DIR" | "INT" | "CAR" | "CEN" | "CNT" | "CPA" | "CTY" | "DEL" | "POB" | "PRE" | "STA" | "ZIP" | "DPID") | | [value](#b147) | = | xs:string | |
| /> | |

complexType "After"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns true if the first interval starts after the second one ends.  
In other words, if the starting point of the first interval is greater  
than the ending point of the second interval.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **After** |

XML Source (w/o annotations (1))

<xs:complexType name="[**After**](#b292)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "AggregateExpression"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definitions of 1 [attribute](#b295), 1 [element](#b297) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [path](#b295) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b297) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b297) (defined in [*AggregateExpression*](#b299) complexType) [] |

Known Direct Subtypes (7):

[AllTrue](#b302) [], [AnyTrue](#b308) [], [Avg](#b321) [], [Count](#b445) [], [Max](#b749) [], [Min](#b782) [], [Sum](#b1045) []

Annotation

Aggregate expressions perform operations on lists of data,  
either directly on a list of scalars, or indirectly on a  
list of objects, with a reference to a property present  
on each object in the list.  
  
Aggregate expressions deal with missing information by  
excluding missing values from consideration before performing  
the aggregated operation. For example, in a Sum over Dose, any  
instance of Medication with no value for Dose would be ignored.  
  
An aggregate operation performed over an empty list is defined  
to return null, except as noted in the documentation for each  
operator (Count, AllTrue, and AnyTrue are the exceptions).

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  ***AggregateExpression*** |

XML Source (w/o annotations (1))

<xs:complexType abstract="true" name="[**AggregateExpression**](#b299)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**source**](#b297)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**path**](#b295)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b299) component only; 1/1)

 path

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b299) component only; 1/2)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

complexType "AllTrue"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [path](#b295) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b297) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b297) (defined in [*AggregateExpression*](#b299) complexType) [] |

Annotation

Returns true if all the elements in source are true.  
  
If a path is specified, elements with no value for the  
property specified by the path are ignored.  
  
If source contains no non-null elements, true is returned.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*AggregateExpression*](#b299) [] (extension)  **AllTrue** |

XML Source (w/o annotations (1))

<xs:complexType name="[**AllTrue**](#b302)">

<xs:complexContent>

<xs:extension base="[**AggregateExpression**](#b299)"/>

</xs:complexContent>

</xs:complexType>

complexType "And"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b796)\* | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b796) (defined in [*NaryExpression*](#b798) complexType) [] |

Annotation

The And operator returns the logical conjunction of its arguments. Note that this  
operator is defined as n-ary, allowing any number of arguments. The result  
of And with no arguments is defined to be false. The result of an And with  
a single argument is defined to be the result of the argument. The result of  
and with two arguments is defined using 3-valued logic semantics. This means  
that if either argument is false, the result is false; if both arguments  
are true, the result is true; otherwise, the result is null. The result  
of more than two arguments is defined as successive invocations of And.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*NaryExpression*](#b798) [] (extension)  **And** |

XML Source (w/o annotations (1))

<xs:complexType name="[**And**](#b305)">

<xs:complexContent>

<xs:extension base="[**NaryExpression**](#b798)"/>

</xs:complexContent>

</xs:complexType>

complexType "AnyTrue"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [path](#b295) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b297) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b297) (defined in [*AggregateExpression*](#b299) complexType) [] |

Annotation

Returns true if any element in source is true.  
  
If a path is specified, elements with no value for the  
property specified by the path are ignored.  
  
If source contains no non-null elements, false is returned.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*AggregateExpression*](#b299) [] (extension)  **AnyTrue** |

XML Source (w/o annotations (1))

<xs:complexType name="[**AnyTrue**](#b308)">

<xs:complexContent>

<xs:extension base="[**AggregateExpression**](#b299)"/>

</xs:complexContent>

</xs:complexType>

complexType "ArtifactLifeCycleEvent"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b310) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [eventType](#b310), [eventDateTime](#b311) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [eventDateTime](#b311) [], | [eventType](#b310) (in artifactLifeCycleEvent) [] |

All Direct / Indirect Based Elements (1):

artifactLifeCycleEvent

Annotation

An event in the life cycle of an artifact. Both the  
type of event are specified, as well as the point in time in which  
that event took place.

XML Source (w/o annotations (1))

<xs:complexType name="[**ArtifactLifeCycleEvent**](#b313)">

<xs:sequence>

<xs:element name="[**eventType**](#b310)">

<xs:complexType>

<xs:attribute name="value" type="[**ArtifactLifeCycleEventType**](#b1131)" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="[**eventDateTime**](#b311)" type="[**dt:TS**](#b145)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b313) component only; 2/2)

 eventType

|  |  |
| --- | --- |
| Type: | anonymous complexType, empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <eventType | |
|  | |  |  |  | | --- | --- | --- | | value | = | (("Created" | "Pre-published" | "Published" | "Reviewed" | "Withdrawn" | "Superseded") | ("Created" | "Pre-published" | "Published" | "Reviewed" | "Withdrawn" | "Superseded")) | |
| /> | |

 eventDateTime

|  |  |
| --- | --- |
| Type: | [dt:TS](#b145) [], empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <eventDateTime | |
|  | |  |  |  | | --- | --- | --- | | [value](#b143) | = | xs:string | |
| /> | |

complexType "AtomicAction"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definition of 1 [element](#b316) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)?, [textEquivalent](#b316)? | |
| </...> | |

Content Model Elements (7):

|  |  |
| --- | --- |
| [actionId](#b260) [],  [actors](#b263) [],  [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [],  [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [], | [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [],  [supportingResources](#b262) [],  [textEquivalent](#b316) [] |

Known Direct Subtypes (7):

[CollectInformationAction](#b396) [], [CreateAction](#b456) [], [DeclareResponseAction](#b497) [], [FireEventAction](#b571) [], [MessageAction](#b757) [], [RemoveAction](#b954) [], [UpdateAction](#b1100) []

All Direct / Indirect Based Elements (1):

simpleAction

Annotation

An action that is not further broken down into  
constituent actions.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*ActionBase*](#b267) [] (extension)  ***AtomicAction*** |

XML Source (w/o annotations (2))

<xs:complexType abstract="true" name="[**AtomicAction**](#b318)">

<xs:complexContent>

<xs:extension base="[**ActionBase**](#b267)">

<xs:sequence>

<xs:element minOccurs="0" name="[**textEquivalent**](#b316)" type="[**dt:ST**](#b131)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b318) component only; 1/7)

 textEquivalent

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

A brief textual description of the action that  
summarizes the action

|  |  |
| --- | --- |
| XML Representation Summary | |
| <textEquivalent | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

complexType "Avg"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [path](#b295) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b297) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b297) (defined in [*AggregateExpression*](#b299) complexType) [] |

Annotation

Returns the average of the elements in source.  
  
If a path is specified, elements with no value for the  
property specified by the path are ignored.  
  
If source contains no non-null elements, null is returned.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*AggregateExpression*](#b299) [] (extension)  **Avg** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Avg**](#b321)">

<xs:complexContent>

<xs:extension base="[**AggregateExpression**](#b299)"/>

</xs:complexContent>

</xs:complexType>

complexType "Before"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns true if the first interval ends before the second one starts.  
In other words, if the ending point of the first interval is less than  
the starting point of the second interval.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Before** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Before**](#b324)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "Begin"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

Returns the starting point of the interval operand.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Begin** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Begin**](#b327)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "Behavior"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |

|  |
| --- |
| XML Representation Summary |
| <.../> |

Known Direct Subtypes (4):

[GroupOrganizationBehavior](#b600) [], [GroupSelectionBehavior](#b605) [], [PrecheckBehavior](#b900) [], [RequiredBehavior](#b972) []

All Direct / Indirect Based Elements (1):

[behavior](#b331) []

Annotation

A behavior may be specified for a specific action or a group of actions. This is the base type for all Behaviors.

XML Source (w/o annotations (1))

<xs:complexType abstract="true" name="[**Behavior**](#b329)"/>

complexType "Behaviors"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b331) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [behavior](#b331)+ | |
| </...> | |

Content Model Elements (1):

[behavior](#b331) []

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [], | [behaviors](#b689) (in [knowledgeDocument](#b255)) [] |

XML Source

<xs:complexType name="[**Behaviors**](#b333)">

<xs:sequence>

<xs:element maxOccurs="unbounded" name="[**behavior**](#b331)" type="[**Behavior**](#b329)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b333) component only; 1/1)

 behavior

|  |  |
| --- | --- |
| Type: | [*Behavior*](#b329) [], empty content |

|  |
| --- |
| XML Representation Summary |
| <behavior/> |

complexType "BinaryExpression"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b336) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Known Direct Subtypes (22):

[Add](#b282) [], [After](#b292) [], [Before](#b324) [], [Difference](#b500) [], [Divide](#b508) [], [Equal](#b531) [], [Greater](#b592) [], [GreaterOrEqual](#b595) [], [IfNull](#b614) [], [Less](#b714) [], [LessOrEqual](#b717) [], [Log](#b743) [], [Meets](#b752) [], [Modulo](#b790) [], [Multiply](#b793) [], [NotEqual](#b807) [], [Overlaps](#b836) [], [Power](#b895) [], [ProperContains](#b906) [], [ProperIn](#b909) [], [Subtract](#b1039) [], [TruncatedDivide](#b1087) []

Annotation

The BinaryExpression type defines the abstract base type for all expressions  
that take two arguments.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **BinaryExpression** |

XML Source (w/o annotations (1))

<xs:complexType name="[**BinaryExpression**](#b338)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="2" minOccurs="2" name="[**operand**](#b336)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b338) component only; 1/2)

 operand

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <operand> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </operand> | |

complexType "BooleanLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b341) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b341) | = | xs:boolean | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type BL with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **BooleanLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**BooleanLiteral**](#b343)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**value**](#b341)" type="xs:boolean" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b343) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | required |

complexType "Case"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [elements](#b346) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [comparand](#b346)?, [caseItem](#b347)+, [else](#b348) | |
| </...> | |

Content Model Elements (4):

|  |  |
| --- | --- |
| [caseItem](#b347) [],  [comparand](#b346) [], | [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [else](#b348) (defined in [Case](#b350) complexType) [] |

Annotation

The case operator allows for multiple conditional expressions to  
be chained together in a single expression, rather than having to  
nest multiple Conditional operators.  
  
In addition, the comparand operand provides a variant on the case that  
allows a single value to be compared in each conditional.  
  
If a comparand is not provided, the type of each when element of the  
caseItems within the Case is expected to be boolean.  
  
If a comparand is provided, the type of each when element of the  
caseItems within the Case is expected to be of the same type as the comparand.  
  
An else element must always be provided.  
  
The type of all then elements, and the else element, must be the same is,  
and is the resulting type of the Case operator.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Case** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Case**](#b350)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**comparand**](#b346)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="unbounded" minOccurs="1" name="[**caseItem**](#b347)" type="[**CaseItem**](#b355)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**else**](#b348)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b350) component only; 3/4)

 comparand

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <comparand> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </comparand> | |

 caseItem

|  |  |
| --- | --- |
| Type: | [CaseItem](#b355) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <caseItem> | |
|  | |  |  | | --- | --- | | *Content:* | [when](#b352), [then](#b353) | |
| </caseItem> | |

 else

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <else> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </else> | |

complexType "CaseItem"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b352) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [when](#b352), [then](#b353) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [then](#b353) (in caseItem) [], | [when](#b352) [] |

All Direct / Indirect Based Elements (1):

[caseItem](#b347) []

XML Source

<xs:complexType name="[**CaseItem**](#b355)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**when**](#b352)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**then**](#b353)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b355) component only; 2/2)

 when

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <when> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </when> | |

 then

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <then> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </then> | |

complexType "Ceiling"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

The Ceiling operator returns the first integer greater than or  
equal to the argument. If the argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Ceiling** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Ceiling**](#b358)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "ClinicalRequest"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 4 [attributes](#b361), 2 [elements](#b366) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [scope](#b957) | = | xs:string | | [cardinality](#b958) | = | ("Single" | "Multiple") | | [dataType](#b959) | = | xs:QName | | [templateId](#b960) | = | xs:string | | [idProperty](#b961) | = | xs:string | | [triggerType](#b962) | = | (("DataElementAdded" | "DataElementModified" | "DataElementRemoved" | "DataElementAccessed" | "DataElementAccessEnded") | ("DataElementAdded" | "DataElementModified" | "DataElementRemoved" | "DataElementAccessed" | "DataElementAccessEnded")) | | [isInitial](#b963) | = | xs:boolean : "true" | | [codeProperty](#b361) | = | xs:string | | [dateProperty](#b362) | = | xs:string | | [useValueSets](#b363) | = | xs:boolean : "false" | | [useSubsumption](#b364) | = | xs:boolean : "false" | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [timeOffset](#b965)?, [codes](#b366)?, [dateRange](#b367)? | |
| </...> | |

Content Model Elements (4):

|  |  |
| --- | --- |
| [codes](#b366) (defined in [ClinicalRequest](#b369) complexType) [],  [dateRange](#b367) [], | [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [timeOffset](#b965) [] |

Annotation

The clinical request expression defines clinical data that will be used by the artifact.  
  
The codes element optionally allows a set of codes to be provided. The codes list restricts  
the set of clinical statements returned to only those clinical statements that matched some  
code in the set.  
  
The dateRange element optionally allows a date range to be provided. The clinical statements  
returned would be only those clinical statements whose date fell within the range specified.  
  
The codeProperty attribute optionally specifies which property of the model contains the Code  
or Codes for the clinical statement.  
  
The dateProperty attribute optionally specifies which property of the model contains the  
clinically relevant date for the clinical statement.  
  
These properties (codeProperty and dateProperty) could potentially be specified elsewhere,  
rather than on each Request expression, but allowing them at the request expression level  
gives the most flexibility.  
  
The useValueSets attribute determines whether references to value sets in the Codes element  
will be expanded, or left as value set references for the purposes of communicating across  
the data boundary.  
  
The useSubsumption attribute determines whether subsumption should be used to determine  
whether or not a given data item should be included in the result.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*RequestBase*](#b967) [] (extension)  **ClinicalRequest** |

XML Source (w/o annotations (1))

<xs:complexType name="[**ClinicalRequest**](#b369)">

<xs:complexContent>

<xs:extension base="[**RequestBase**](#b967)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**codes**](#b366)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**dateRange**](#b367)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**codeProperty**](#b361)" type="xs:string" use="optional"/>

<xs:attribute name="[**dateProperty**](#b362)" type="xs:string" use="optional"/>

<xs:attribute default="false" name="[**useValueSets**](#b363)" type="xs:boolean" use="optional"/>

<xs:attribute default="false" name="[**useSubsumption**](#b364)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b369) component only; 4/11)

 codeProperty

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 dateProperty

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 useValueSets

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

Attribute Value

|  |  |
| --- | --- |
| Default: | "false" |

 useSubsumption

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

Attribute Value

|  |  |
| --- | --- |
| Default: | "false" |

Content Element Detail (all declarations; defined within [this](#b369) component only; 2/4)

 codes

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <codes> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </codes> | |

 dateRange

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <dateRange> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </dateRange> | |

complexType "Coalesce"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b796)\* | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b796) (defined in [*NaryExpression*](#b798) complexType) [] |

Annotation

Returns the result of the first argument that evaluates is not null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*NaryExpression*](#b798) [] (extension)  **Coalesce** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Coalesce**](#b372)">

<xs:complexContent>

<xs:extension base="[**NaryExpression**](#b798)"/>

</xs:complexContent>

</xs:complexType>

complexType "CodedOrdinalLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 5 [attributes](#b375) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [codeSystem](#b375) | = | xs:string | | [code](#b376) | = | xs:string | | [codeSystemName](#b377) | = | xs:string | | [displayName](#b378) | = | xs:string | | [value](#b379) | = | xs:double | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type CO with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **CodedOrdinalLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**CodedOrdinalLiteral**](#b381)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**codeSystem**](#b375)" type="[**dt:Uid**](#b194)" use="required"/>

<xs:attribute name="[**code**](#b376)" type="[**dt:Code**](#b155)" use="required"/>

<xs:attribute name="[**codeSystemName**](#b377)" type="xs:string" use="optional"/>

<xs:attribute name="[**displayName**](#b378)" type="xs:string" use="optional"/>

<xs:attribute name="[**value**](#b379)" type="[**dt:Decimal**](#b158)" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b381) component only; 5/5)

 codeSystem

|  |  |
| --- | --- |
| Type: | [dt:Uid](#b194) [] |
| Use: | required |

Attribute Value

|  |
| --- |
| xs:string |

 code

|  |  |
| --- | --- |
| Type: | [dt:Code](#b155) [] |
| Use: | required |

Attribute Value

|  |
| --- |
| xs:string |

 codeSystemName

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 displayName

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 value

|  |  |
| --- | --- |
| Type: | [dt:Decimal](#b158) [] |
| Use: | required |

Attribute Value

|  |
| --- |
| xs:double |

complexType "CodeLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 4 [attributes](#b384) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [codeSystem](#b384) | = | xs:string | | [code](#b385) | = | xs:string | | [codeSystemName](#b386) | = | xs:string | | [displayName](#b387) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type CD with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **CodeLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**CodeLiteral**](#b389)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**codeSystem**](#b384)" type="[**dt:Uid**](#b194)" use="required"/>

<xs:attribute name="[**code**](#b385)" type="[**dt:Code**](#b155)" use="required"/>

<xs:attribute name="[**codeSystemName**](#b386)" type="xs:string" use="optional"/>

<xs:attribute name="[**displayName**](#b387)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b389) component only; 4/4)

 codeSystem

|  |  |
| --- | --- |
| Type: | [dt:Uid](#b194) [] |
| Use: | required |

Attribute Value

|  |
| --- |
| xs:string |

 code

|  |  |
| --- | --- |
| Type: | [dt:Code](#b155) [] |
| Use: | required |

Attribute Value

|  |
| --- |
| xs:string |

 codeSystemName

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 displayName

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

complexType "CollectInformationAction"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [elements](#b392) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)?, [textEquivalent](#b316)?, [documentationConcept](#b392), [initialValue](#b393)?, [responseBinding](#b394)? | |
| </...> | |

Content Model Elements (10):

|  |  |
| --- | --- |
| [actionId](#b260) [],  [actors](#b263) [],  [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [],  [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [],  [documentationConcept](#b392) [], | [initialValue](#b393) [],  [responseBinding](#b394) [],  [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [],  [supportingResources](#b262) [],  [textEquivalent](#b316) [] |

Annotation

This action requests information from the actor. The  
information request is specified as a DocumentationItem.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*ActionBase*](#b267) [] (extension)  [*AtomicAction*](#b318) [] (extension)  **CollectInformationAction** |

XML Source (w/o annotations (4))

<xs:complexType name="[**CollectInformationAction**](#b396)">

<xs:complexContent>

<xs:extension base="[**AtomicAction**](#b318)">

<xs:sequence>

<xs:element name="[**documentationConcept**](#b392)" type="[**DocumentationItem**](#b518)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**initialValue**](#b393)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**responseBinding**](#b394)" type="[**ResponseBinding**](#b982)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b396) component only; 3/10)

 documentationConcept

|  |  |
| --- | --- |
| Type: | [DocumentationItem](#b518) [], complex content |

This provides a specification of the  
information to be collected from the user.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <documentationConcept> | |
|  | |  |  | | --- | --- | | *Content:* | [identifier](#b680)?, [displayText](#b511), [description](#b512)?, [itemCodes](#b513)?, [responseDataType](#b514), [responseCardinality](#b515)?, [responseRange](#b516)\* | |
| </documentationConcept> | |

 initialValue

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

An expression to compute an initial value for  
the documentation concept. The initial value could be computed  
from previous data about the patient available via expressions  
specified in the external data

|  |  |
| --- | --- |
| XML Representation Summary | |
| <initialValue> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </initialValue> | |

 responseBinding

|  |  |
| --- | --- |
| Type: | [ResponseBinding](#b982) [], empty content |

Defines the response binding for the documentation item.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <responseBinding | |
|  | |  |  |  | | --- | --- | --- | | [container](#b979) | = | xs:string : "Responses" | | [property](#b980) | = | xs:string | |
| /> | |

complexType "Combine"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b399) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b399), [separator](#b400)? | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [separator](#b400) (defined in [Combine](#b402) complexType) [], | [source](#b399) (defined in [Combine](#b402) complexType) [] |

Annotation

The Combine operator combines a list of strings,  
optionally separating each string with the given  
separator.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Combine** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Combine**](#b402)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**source**](#b399)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**separator**](#b400)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b402) component only; 2/3)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

 separator

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <separator> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </separator> | |

complexType "ComplexLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b405) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [value](#b405) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [value](#b405) (defined in [ComplexLiteral](#b407) complexType) [] |

Annotation

The ComplexLiteral expression allows an xml literal of any type to be  
included in an expression.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **ComplexLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**ComplexLiteral**](#b407)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**value**](#b405)" type="xs:anyType"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b407) component only; 1/2)

 value

|  |  |
| --- | --- |
| Type: | xs:anyType, any content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <value> | |
|  | ... |
| </value> | |

complexType "Concat"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b796)\* | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b796) (defined in [*NaryExpression*](#b798) complexType) [] |

Annotation

The Concat operator performs string concatenation  
of its arguments.  
If any argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*NaryExpression*](#b798) [] (extension)  **Concat** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Concat**](#b410)">

<xs:complexContent>

<xs:extension base="[**NaryExpression**](#b798)"/>

</xs:complexContent>

</xs:complexType>

complexType "Condition"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b412) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [logic](#b412), [conditionRole](#b413) | |
| </...> | |

Content Model Elements (2):

[conditionRole](#b413) [], [logic](#b412) []

All Direct / Indirect Based Elements (1):

[condition](#b424) (defined in [Conditions](#b426) complexType) []

Annotation

A condition specifies when a knowledge component is  
to be executed. For example, an ECA rule uses an ApplicableScenario  
condition to determine whether or not the action described by  
the artifact should be executed.

XML Source (w/o annotations (3))

<xs:complexType name="[**Condition**](#b415)">

<xs:sequence>

<xs:element name="[**logic**](#b412)" type="[**Expression**](#b541)"/>

<xs:element name="[**conditionRole**](#b413)" type="[**ConditionRoleType**](#b1173)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b415) component only; 2/2)

 logic

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

The logic specification of the condition. Often,  
though not necessarily, the logic is an expression about patient  
data. The expression must evaluate to a Boolean value.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <logic> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </logic> | |

 conditionRole

|  |  |
| --- | --- |
| Type: | [ConditionRoleType](#b1173) [], simple content |

The role determines when to evaluate the  
expression and how to proceed based on the expression results.  
Different artifact types use different types of conditions to  
control various aspects of the artifact. See the condition role  
type enumeration documentation for more discussion.

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <conditionRole> | | |
|  | *Content:* | { "ApplicableScenario" | "ApplicableScenario" } |
| </conditionRole> | | |

complexType "Conditional"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [elements](#b418) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [condition](#b418), [then](#b419), [else](#b420) | |
| </...> | |

Content Model Elements (4):

|  |  |
| --- | --- |
| [condition](#b418) (defined in [Conditional](#b422) complexType) [],  [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [else](#b420) (defined in [Conditional](#b422) complexType) [],  [then](#b419) (defined in [Conditional](#b422) complexType) [] |

Annotation

The Conditional operator evaluates a condition, and returns the then  
argument if condition evaluates to true; otherwise the result  
of the else argument is returned.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Conditional** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Conditional**](#b422)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**condition**](#b418)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**then**](#b419)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**else**](#b420)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b422) component only; 3/4)

 condition

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <condition> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </condition> | |

 then

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <then> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </then> | |

 else

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <else> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </else> | |

complexType "Conditions"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b424) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [condition](#b424)+ | |
| </...> | |

Content Model Elements (1):

[condition](#b424) (defined in [Conditions](#b426) complexType) []

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [], | [conditions](#b688) (in [knowledgeDocument](#b255)) [] |

Annotation

A collection of conditions that are used to  
define whether various aspects of the artifact, such as  
whether or not a particular action should be executed, or  
whether a particular order set item is applicable to a  
given patient.

XML Source (w/o annotations (1))

<xs:complexType name="[**Conditions**](#b426)">

<xs:sequence>

<xs:element maxOccurs="unbounded" name="[**condition**](#b424)" type="[**Condition**](#b415)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b426) component only; 1/1)

 condition

|  |  |
| --- | --- |
| Type: | [Condition](#b415) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <condition> | |
|  | |  |  | | --- | --- | | *Content:* | [logic](#b412), [conditionRole](#b413) | |
| </condition> | |

complexType "Contains"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b429) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b429), [element](#b430) | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [element](#b430) (defined in [Contains](#b432) complexType) [], | [source](#b429) (defined in [Contains](#b432) complexType) [] |

Annotation

Returns true if the given source contains the given element.  
There are four overloads of this operator:  
List, Scalar : The type of the scalar must be the same as the element type of the list.  
List, List : The element type of both lists must be the same.  
Interval, Scalar : The type of the scalar must be the same as the point type of the interval.  
Interval, Interval : The point type of both intervals must be the same.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Contains** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Contains**](#b432)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**source**](#b429)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**element**](#b430)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b432) component only; 2/3)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

 element

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <element> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </element> | |

complexType "Contribution"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b434) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [contributor](#b434), [role](#b435) | |
| </...> | |

Content Model Elements (2):

[contributor](#b434) [], [role](#b435) []

All Direct / Indirect Based Elements (1):

contribution

Annotation

A contribution is made by a specific contributor  
(organization, person, etc.), and was made in a particular way, as  
specified by the contributor's role. For example, a contributor may  
have been an author, or may have been a reviewer.

XML Source (w/o annotations (1))

<xs:complexType name="[**Contribution**](#b437)">

<xs:sequence>

<xs:element name="[**contributor**](#b434)" type="[**Party**](#b853)"/>

<xs:element name="[**role**](#b435)">

<xs:complexType>

<xs:attribute name="value" type="[**ContributorType**](#b1182)"/>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b437) component only; 2/2)

 contributor

|  |  |
| --- | --- |
| Type: | [*Party*](#b853) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <contributor> | |
|  | |  |  | | --- | --- | | *Content:* | [addresses](#b850)?, [contacts](#b851)? | |
| </contributor> | |

 role

|  |  |
| --- | --- |
| Type: | anonymous complexType, empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <role | |
|  | |  |  |  | | --- | --- | --- | | value | = | (("Author" | "Editor" | "Endorser" | "Reviewer") | ("Author" | "Editor" | "Endorser" | "Reviewer")) | |
| /> | |

complexType "Convert"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b440) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [toType](#b440) | = | xs:QName | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

Converts a value to a specific type.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Convert** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Convert**](#b442)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)">

<xs:attribute name="[**toType**](#b440)" type="xs:QName" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b442) component only; 1/1)

 toType

|  |  |
| --- | --- |
| Type: | xs:QName, predefined |
| Use: | required |

complexType "Count"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [path](#b295) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b297) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b297) (defined in [*AggregateExpression*](#b299) complexType) [] |

Annotation

Returns the number of non-null elements in the source.  
  
If a path is specified, the count returns the number of  
elements that have a value for the property specified by the path.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*AggregateExpression*](#b299) [] (extension)  **Count** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Count**](#b445)">

<xs:complexContent>

<xs:extension base="[**AggregateExpression**](#b299)"/>

</xs:complexContent>

</xs:complexType>

complexType "Coverage"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [elements](#b447) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [focus](#b447), [description](#b448)?, [value](#b449)? | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b448) (in coverage) [],  [focus](#b447) [], | [value](#b449) (in coverage) [] |

All Direct / Indirect Based Elements (1):

coverage

Annotation

Specifies various attributes of the patient  
population for whom and/or environment of care in which the CDS  
artifact is applicable.

XML Source (w/o annotations (1))

<xs:complexType name="[**Coverage**](#b451)">

<xs:sequence>

<xs:element name="[**focus**](#b447)">

<xs:complexType>

<xs:attribute name="value" type="[**CoverageType**](#b1191)" use="required"/>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**description**](#b448)" type="[**dt:ST**](#b131)"/>

<xs:element minOccurs="0" name="[**value**](#b449)" type="[**dt:CD**](#b33)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b451) component only; 3/3)

 focus

|  |  |
| --- | --- |
| Type: | anonymous complexType, empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <focus | |
|  | |  |  |  | | --- | --- | --- | | value | = | (("PatientGender" | "PatientAgeGroup" | "ClinicalFocus" | "TargetUser" | "WorkflowSetting" | "WorkflowTask" | "ClinicalVenue") | ("PatientGender" | "PatientAgeGroup" | "ClinicalFocus" | "TargetUser" | "WorkflowSetting" | "WorkflowTask" | "ClinicalVenue")) | |
| /> | |

 description

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <description | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 value

|  |  |
| --- | --- |
| Type: | [dt:CD](#b33) [], empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <value | |
|  | |  |  |  | | --- | --- | --- | | [code](#b27) | = | xs:string | | [codeSystem](#b28) | = | xs:string | | [codeSystemName](#b29) | = | xs:string | | [displayName](#b30) | = | xs:string | | [originalText](#b31) | = | xs:string : "" | |
| /> | |

complexType "CreateAction"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b454) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)?, [textEquivalent](#b316)?, [actionSentence](#b454)? | |
| </...> | |

Content Model Elements (8):

|  |  |
| --- | --- |
| [actionId](#b260) [],  [actionSentence](#b454) (defined in [CreateAction](#b456) complexType) [],  [actors](#b263) [],  [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [], | [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [],  [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [],  [supportingResources](#b262) [],  [textEquivalent](#b316) [] |

Annotation

A new action to be executed by a user or a computer  
system. The sentence provides the details of the action to be  
executed.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*ActionBase*](#b267) [] (extension)  [*AtomicAction*](#b318) [] (extension)  **CreateAction** |

XML Source (w/o annotations (2))

<xs:complexType name="[**CreateAction**](#b456)">

<xs:complexContent>

<xs:extension base="[**AtomicAction**](#b318)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**actionSentence**](#b454)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b456) component only; 1/8)

 actionSentence

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

The parameters of the action that is to be  
executed. For example, an action may be to order a medication.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <actionSentence> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </actionSentence> | |

complexType "Current"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b459) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [scope](#b459) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

The Current expression returns the value of the object  
currently in scope. For example, within an ForEach  
expression, this returns the current element being  
considered in the iteration.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Current** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Current**](#b461)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**scope**](#b459)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b461) component only; 1/1)

 scope

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

complexType "Date"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 7 [elements](#b464) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [year](#b464), [month](#b465), [day](#b466), [hour](#b467)?, [minute](#b468)?, [second](#b469)?, [millisecond](#b470)? | |
| </...> | |

Content Model Elements (8):

|  |  |
| --- | --- |
| [day](#b466) [],  [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [hour](#b467) [],  [millisecond](#b470) [], | [minute](#b468) [],  [month](#b465) [],  [second](#b469) [],  [year](#b464) [] |

Annotation

Constructs a date/time value from the given components.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Date** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Date**](#b472)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**year**](#b464)" type="xs:int"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**month**](#b465)" type="xs:int"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**day**](#b466)" type="xs:int"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**hour**](#b467)" type="xs:int"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**minute**](#b468)" type="xs:int"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**second**](#b469)" type="xs:int"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**millisecond**](#b470)" type="xs:double"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b472) component only; 7/8)

 year

|  |  |
| --- | --- |
| Type: | xs:int, predefined, simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <year> | | |
|  | *Content:* | { xs:int } |
| </year> | | |

 month

|  |  |
| --- | --- |
| Type: | xs:int, predefined, simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <month> | | |
|  | *Content:* | { xs:int } |
| </month> | | |

 day

|  |  |
| --- | --- |
| Type: | xs:int, predefined, simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <day> | | |
|  | *Content:* | { xs:int } |
| </day> | | |

 hour

|  |  |
| --- | --- |
| Type: | xs:int, predefined, simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <hour> | | |
|  | *Content:* | { xs:int } |
| </hour> | | |

 minute

|  |  |
| --- | --- |
| Type: | xs:int, predefined, simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <minute> | | |
|  | *Content:* | { xs:int } |
| </minute> | | |

 second

|  |  |
| --- | --- |
| Type: | xs:int, predefined, simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <second> | | |
|  | *Content:* | { xs:int } |
| </second> | | |

 millisecond

|  |  |
| --- | --- |
| Type: | xs:double, predefined, simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <millisecond> | | |
|  | *Content:* | { xs:double } |
| </millisecond> | | |

complexType "DateAdd"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [elements](#b475) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [date](#b475), [granularity](#b476), [numberOfPeriods](#b477) | |
| </...> | |

Content Model Elements (4):

|  |  |
| --- | --- |
| [date](#b475) (defined in [DateAdd](#b479) complexType) [],  [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [granularity](#b476) (defined in [DateAdd](#b479) complexType) [],  [numberOfPeriods](#b477) [] |

Annotation

DateAdd adds numberOfPeriods date periods of the specified granularity to the  
given date.  
Note that this is different than adding an Interval to a date time, because for  
operations on granularities such as month and year, the interval is not well-defined  
due to varying month and year lengths. As a result, DateAdd is used to provide  
well-defined and consistent semantics for date arithmetic involving months and years.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **DateAdd** |

XML Source (w/o annotations (1))

<xs:complexType name="[**DateAdd**](#b479)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**date**](#b475)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**granularity**](#b476)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**numberOfPeriods**](#b477)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b479) component only; 3/4)

 date

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <date> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </date> | |

 granularity

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <granularity> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </granularity> | |

 numberOfPeriods

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <numberOfPeriods> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </numberOfPeriods> | |

complexType "DateDiff"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [elements](#b482) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [startDate](#b482), [endDate](#b483), [granularity](#b484) | |
| </...> | |

Content Model Elements (4):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [endDate](#b483) [], | [granularity](#b484) (defined in [DateDiff](#b486) complexType) [],  [startDate](#b482) [] |

Annotation

DateDiff returns the number of granularity boundaries occurring between  
startDate and endDate.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **DateDiff** |

XML Source (w/o annotations (1))

<xs:complexType name="[**DateDiff**](#b486)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**startDate**](#b482)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**endDate**](#b483)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**granularity**](#b484)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b486) component only; 3/4)

 startDate

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <startDate> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </startDate> | |

 endDate

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <endDate> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </endDate> | |

 granularity

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <granularity> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </granularity> | |

complexType "DatePart"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b489) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [date](#b489), [granularity](#b490) | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [date](#b489) (defined in [DatePart](#b492) complexType) [],  [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [granularity](#b490) (defined in [DatePart](#b492) complexType) [] |

Annotation

DatePart returns the granularity component of the given date.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **DatePart** |

XML Source (w/o annotations (1))

<xs:complexType name="[**DatePart**](#b492)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**date**](#b489)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**granularity**](#b490)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b492) component only; 2/3)

 date

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <date> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </date> | |

 granularity

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <granularity> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </granularity> | |

complexType "DeclareResponseAction"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b495) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [Name](#b495) | = | xs:string : "Responses" | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)?, [textEquivalent](#b316)? | |
| </...> | |

Content Model Elements (7):

|  |  |
| --- | --- |
| [actionId](#b260) [],  [actors](#b263) [],  [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [],  [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [], | [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [],  [supportingResources](#b262) [],  [textEquivalent](#b316) [] |

Annotation

The DeclareResponseAction provides a mechanism to declare a container for responses provided by the user  
in response to CollectInformationActions. The intended semantics are to provide a container that can be  
used to access responses within expressions used subsequently in an artifact.  
  
The DeclareResponseAction creates a named container within the Parameters scope of the artifact, and expressions  
may access the contents of a response using a ParameterRef expression.  
  
The container is expected to be a collection of name-value pairs, and the intended semantics are to allow the  
Property expression to be used, in connection with a ParameterRef expression as the source, to retrieve the current  
value for a property.  
  
The CollectInformationAction contains a responseBinding attribute that specifies the name of the container, and the  
name of the property to be used to store the response value.  
  
If no Name attribute is provided, the response container will be named Responses.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*ActionBase*](#b267) [] (extension)  [*AtomicAction*](#b318) [] (extension)  **DeclareResponseAction** |

XML Source (w/o annotations (1))

<xs:complexType name="[**DeclareResponseAction**](#b497)">

<xs:complexContent>

<xs:extension base="[**AtomicAction**](#b318)">

<xs:attribute default="Responses" name="[**Name**](#b495)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b497) component only; 1/1)

 Name

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Attribute Value

|  |  |
| --- | --- |
| Default: | "Responses" |

complexType "Difference"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns the difference of the two operands.  
This operator has two overloads:  
List, List  
Interval, Interval  
  
For the list overload, this operator returns a list with  
the elements that appear in the first operand, that do not  
appear in the second operand.  
  
For the interval overload, this operator returns the portion  
of the first interval that does not overlap with the second.  
Note that if the operands do not overlap, or if the second  
operand is completely contained within the first, this operator  
returns null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Difference** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Difference**](#b500)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "Distinct"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b503) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b503) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b503) (defined in [Distinct](#b505) complexType) [] |

Annotation

The Distinct operator takes a list of elements and returns a list containing  
only the unique elements within the input. For example, given the list of  
integers { 1, 1, 1, 2, 2, 3, 4, 4 }, the result of Distinct would be { 1, 2, 3, 4 }.  
  
The operator uses equality comparison semantics as defined in the Equal  
operator.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Distinct** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Distinct**](#b505)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element name="[**source**](#b503)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b505) component only; 1/2)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

complexType "Divide"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

The Divide operator performs numeric division of its arguments.  
Note that the result of Divide is a decimal, even if its arguments  
are integers.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Divide** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Divide**](#b508)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "DocumentationItem"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 6 [elements](#b511) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [identifier](#b680)?, [displayText](#b511), [description](#b512)?, [itemCodes](#b513)?, [responseDataType](#b514), [responseCardinality](#b515)?, [responseRange](#b516)\* | |
| </...> | |

Content Model Elements (7):

|  |  |
| --- | --- |
| [description](#b512) (in documentationConcept) [],  [displayText](#b511) (in documentationConcept) [],  [identifier](#b680) (defined in [*ItemDefinition*](#b682) complexType) [],  [itemCodes](#b513) [], | [responseCardinality](#b515) [],  [responseDataType](#b514) [],  [responseRange](#b516) [] |

All Direct / Indirect Based Elements (1):

[documentationConcept](#b392) []

Annotation

An item type representing the definition of an  
individual item to be recorded in a structured clinical document.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*ItemDefinition*](#b682) [] (extension)  **DocumentationItem** |

XML Source (w/o annotations (8))

<xs:complexType name="[**DocumentationItem**](#b518)">

<xs:complexContent>

<xs:extension base="[**ItemDefinition**](#b682)">

<xs:sequence>

<xs:element name="[**displayText**](#b511)" type="[**dt:ST**](#b131)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**description**](#b512)" type="[**dt:ST**](#b131)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**itemCodes**](#b513)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" minOccurs="0" name="itemCode" type="[**dt:CD**](#b33)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="[**responseDataType**](#b514)" type="[**ValueType**](#b1284)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**responseCardinality**](#b515)" type="[**Cardinality**](#b1167)"/>

<xs:element maxOccurs="unbounded" minOccurs="0" name="[**responseRange**](#b516)" type="[**RangeConstraint**](#b927)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b518) component only; 6/7)

 displayText

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

Brief text or title for this documentation item  
that is the caption displayed to the user performing the  
documentation.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <displayText | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 description

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

A lengthier description of this documentation  
item that is displayed to the user.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <description | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 itemCodes

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

A collection of codes for concepts that are the  
equivalents of this documentation item.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <itemCodes> | |
|  | |  |  | | --- | --- | | *Content:* | itemCode\* | |
| </itemCodes> | |

 responseDataType

|  |  |
| --- | --- |
| Type: | [ValueType](#b1284) [], simple content |

The data type of the value or the response that  
is entered by the user.

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <responseDataType> | | |
|  | *Content:* | { ("Boolean" | "Code" | "CodedOrdinal" | "EntityName" | "Identifier" | "Integer" | "IntegerInterval" | "Period" | "PhysicalQuantity" | "PhysicalQuantityInterval" | "QuantityInterval" | "Ratio" | "RatioInterval" | "Real" | "RealInterval" | "SimpleCode" | "String" | "Timestamp" | "TimestampInterval" | "URL") | ("Boolean" | "Code" | "CodedOrdinal" | "EntityName" | "Identifier" | "Integer" | "IntegerInterval" | "Period" | "PhysicalQuantity" | "PhysicalQuantityInterval" | "QuantityInterval" | "Ratio" | "RatioInterval" | "Real" | "RealInterval" | "SimpleCode" | "String" | "Timestamp" | "TimestampInterval" | "URL") } |
| </responseDataType> | | |

 responseCardinality

|  |  |
| --- | --- |
| Type: | [Cardinality](#b1167) [], simple content |

The number of allowed responses, single or  
multiple.

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <responseCardinality> | | |
|  | *Content:* | { *enumeration of* xs:string } |
| </responseCardinality> | | |

Simple Content

|  |  |
| --- | --- |
| Enumeration: | "Single", "Multiple" |

 responseRange

|  |  |
| --- | --- |
| Type: | [*RangeConstraint*](#b927) [], complex content |

The constraints (within the responseDataType)  
on the values that may be entered by the user

|  |  |
| --- | --- |
| XML Representation Summary | |
| <responseRange> | |
|  | |  |  | | --- | --- | | *Content:* | [constraintType](#b925) | |
| </responseRange> | |

complexType "End"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

Returns the ending point of the interval operand.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **End** |

XML Source (w/o annotations (1))

<xs:complexType name="[**End**](#b521)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "EntityNameLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b524), 1 [element](#b526) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [use](#b524) | = | *list of* ("ABC" | "SYL" | "IDE" | "C" | "OR" | "T" | "I" | "P" | "ANON" | "A" | "R" | "OLD" | "DN" | "M" | "SRCH" | "PHON") | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [part](#b526)+ | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [part](#b526) (defined in [EntityNameLiteral](#b528) complexType) [] |

Annotation

Returns a value of type EN with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **EntityNameLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**EntityNameLiteral**](#b528)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="unbounded" name="[**part**](#b526)" type="[**dt:ENXP**](#b58)"/>

</xs:sequence>

<xs:attribute name="[**use**](#b524)" type="[**dt:set\_EntityNameUse**](#b176)" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b528) component only; 1/1)

 use

|  |  |
| --- | --- |
| Type: | [dt:set\_EntityNameUse](#b176) [] |
| Use: | optional |

Attribute Value

|  |
| --- |
| *list of* ("ABC" | "SYL" | "IDE" | "C" | "OR" | "T" | "I" | "P" | "ANON" | "A" | "R" | "OLD" | "DN" | "M" | "SRCH" | "PHON") |

Content Element Detail (all declarations; defined within [this](#b528) component only; 1/2)

 part

|  |  |
| --- | --- |
| Type: | [dt:ENXP](#b58) [], empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <part | |
|  | |  |  |  | | --- | --- | --- | | [qualifier](#b56) | = | *list of* ("LS" | "AC" | "NB" | "PR" | "HON" | "BR" | "AD" | "SP" | "MID" | "CL" | "IN" | "PFX" | "SFX") | | [type](#b55) | = | ("FAM" | "GIV" | "TITLE" | "DEL") | | [value](#b147) | = | xs:string | |
| /> | |

complexType "Equal"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

The Equal operator returns true if the arguments are equal; and false  
otherwise. Equality semantics are defined to be value-based. This means  
for scalars that equality returns true if and only if the result of  
each argument evaluates to the same value. For object types, this means  
that equality returns true if and only if the objects are of the same  
type, and the values for all properties are the same. For list types,  
this means that equality returns true if and only if the lists contain  
elements of the same type, have the same number of elements, and for  
each element in the lists, the elements are equal using the same semantics.  
For interval types, equality returns true if and only if the intervals  
are over the same point type, and they have the same value for the  
beginning and ending points of the interval. If either argument  
is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Equal** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Equal**](#b531)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "Evidence"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [elements](#b533) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [qualityOfEvidence](#b533)?, [strengthOfRecommendation](#b534)?, [resources](#b535)? | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [qualityOfEvidence](#b533) [],  [resources](#b535) (in evidence) [], | [strengthOfRecommendation](#b534) [] |

All Direct / Indirect Based Elements (1):

[evidence](#b1047) []

Annotation

Reference to research on which the artifact is  
based. This evidence can be 'graded' depending on its quality and  
pedigree and the strength of the recommendations it makes.

XML Source (w/o annotations (4))

<xs:complexType name="[**Evidence**](#b537)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**qualityOfEvidence**](#b533)" type="[**dt:CD**](#b33)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**strengthOfRecommendation**](#b534)" type="[**dt:CD**](#b33)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**resources**](#b535)" type="[**SupportingResource**](#b1053)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b537) component only; 3/3)

 qualityOfEvidence

|  |  |
| --- | --- |
| Type: | [dt:CD](#b33) [], empty content |

The quality of the evidence associated with this  
artifact. The codeSystem attribute specifies the quality scale  
used to grade this evidence source while the code specifies the  
actual quality score (represented as a coded value) associated  
with this evidence reference. CodeSystemName specifies the name of  
the scale. DisplayName specifies the display name of the coded  
value (the score).

|  |  |
| --- | --- |
| XML Representation Summary | |
| <qualityOfEvidence | |
|  | |  |  |  | | --- | --- | --- | | [code](#b27) | = | xs:string | | [codeSystem](#b28) | = | xs:string | | [codeSystemName](#b29) | = | xs:string | | [displayName](#b30) | = | xs:string | | [originalText](#b31) | = | xs:string : "" | |
| /> | |

 strengthOfRecommendation

|  |  |
| --- | --- |
| Type: | [dt:CD](#b33) [], empty content |

The strength of the recommendation assigned to  
this reference. The codeSystem attribute specifies the scale used  
to grade this evidence source while the code specifies the actual  
score (represented as a coded value) for the strength of the  
evidence. CodeSystemName specifies the name of the scale.  
DisplayName specifies the display name of the coded value (the  
score).

|  |  |
| --- | --- |
| XML Representation Summary | |
| <strengthOfRecommendation | |
|  | |  |  |  | | --- | --- | --- | | [code](#b27) | = | xs:string | | [codeSystem](#b28) | = | xs:string | | [codeSystemName](#b29) | = | xs:string | | [displayName](#b30) | = | xs:string | | [originalText](#b31) | = | xs:string : "" | |
| /> | |

 resources

|  |  |
| --- | --- |
| Type: | [SupportingResource](#b1053) [], complex content |

The set of resource references associated with  
the evidence.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <resources> | |
|  | |  |  | | --- | --- | | *Content:* | [resource](#b1051)+ | |
| </resources> | |

complexType "Expression"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definition of 1 [element](#b539) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Known Direct Subtypes (64):

[AddressLiteral](#b289) [], [*AggregateExpression*](#b299) [], [BinaryExpression](#b338) [], [BooleanLiteral](#b343) [], [Case](#b350) [], [CodeLiteral](#b389) [], [CodedOrdinalLiteral](#b381) [], [Combine](#b402) [], [ComplexLiteral](#b407) [], [Conditional](#b422) [], [Contains](#b432) [], [Current](#b461) [], [Date](#b472) [], [DateAdd](#b479) [], [DateDiff](#b486) [], [DatePart](#b492) [], [Distinct](#b505) [], [EntityNameLiteral](#b528) [], [ExpressionRef](#b557) [], [Filter](#b565) [], [First](#b578) [], [ForEach](#b589) [], [IdentifierLiteral](#b611) [], [In](#b620) [], [InValueSet](#b669) [], [IndexOf](#b632) [], [Indexer](#b626) [], [IntegerIntervalLiteral](#b645) [], [IntegerLiteral](#b650) [], [Interval](#b662) [], [Last](#b708) [], [List](#b724) [], [Literal](#b737) [], [*NaryExpression*](#b798) [], [Now](#b810) [], [ObjectExpression](#b817) [], [ObjectRedefine](#b825) [], [ParameterRef](#b848) [], [PeriodLiteral](#b862) [], [PhysicalQuantityIntervalLiteral](#b874) [], [PhysicalQuantityLiteral](#b877) [], [Pos](#b892) [], [Property](#b917) [], [RatioIntervalLiteral](#b933) [], [RatioLiteral](#b936) [], [RealIntervalLiteral](#b944) [], [RealLiteral](#b949) [], [*RequestBase*](#b967) [], [Round](#b994) [], [SetSubsumes](#b1000) [], [SimpleCodeLiteral](#b1005) [], [Sort](#b1012) [], [Split](#b1018) [], [StringLiteral](#b1023) [], [Substring](#b1030) [], [Subsumes](#b1036) [], [TernaryExpression](#b1063) [], [TimestampIntervalLiteral](#b1069) [], [TimestampLiteral](#b1072) [], [Today](#b1075) [], [*UnaryExpression*](#b1092) [], [UrlLiteral](#b1110) [], [ValueSet](#b1117) [], [Within](#b1128) []

Known Indirect Subtypes (53):

[Abs](#b258) [], [Add](#b282) [], [After](#b292) [], [AllTrue](#b302) [], [And](#b305) [], [AnyTrue](#b308) [], [Avg](#b321) [], [Before](#b324) [], [Begin](#b327) [], [Ceiling](#b358) [], [ClinicalRequest](#b369) [], [Coalesce](#b372) [], [Concat](#b410) [], [Convert](#b442) [], [Count](#b445) [], [Difference](#b500) [], [Divide](#b508) [], [End](#b521) [], [Equal](#b531) [], [Floor](#b581) [], [Greater](#b592) [], [GreaterOrEqual](#b595) [], [IfNull](#b614) [], [Intersect](#b653) [], [IsEmpty](#b672) [], [IsNotEmpty](#b675) [], [IsNull](#b678) [], [Length](#b711) [], [Less](#b714) [], [LessOrEqual](#b717) [], [Ln](#b740) [], [Log](#b743) [], [Lower](#b746) [], [Max](#b749) [], [Meets](#b752) [], [Min](#b782) [], [Modulo](#b790) [], [Multiply](#b793) [], [Negate](#b801) [], [Not](#b804) [], [NotEqual](#b807) [], [Or](#b828) [], [Overlaps](#b836) [], [Power](#b895) [], [Pred](#b903) [], [ProperContains](#b906) [], [ProperIn](#b909) [], [Subtract](#b1039) [], [Succ](#b1042) [], [Sum](#b1045) [], [TruncatedDivide](#b1087) [], [Union](#b1095) [], [Upper](#b1103) []

All Direct / Indirect Based Elements (71):

[actionSentence](#b454) (defined in [CreateAction](#b456) complexType) [],

[actionSentence](#b569) (defined in [FireEventAction](#b571) complexType) [],

[actionSentence](#b952) (defined in [RemoveAction](#b954) complexType) [],

[actionSentence](#b1098) (defined in [UpdateAction](#b1100) complexType) [],

[ancestor](#b1033) [],

[ancestors](#b997) [],

[begin](#b659) [],

[codes](#b366) (defined in [ClinicalRequest](#b369) complexType) [],

[collection](#b618) [],

[comparand](#b346) [],

[condition](#b418) (defined in [Conditional](#b422) complexType) [],

[condition](#b563) (defined in [Filter](#b565) complexType) [],

[constraint](#b544) [],

[date](#b475) (defined in [DateAdd](#b479) complexType) [],

[date](#b489) (defined in [DatePart](#b492) complexType) [],

[dateRange](#b367) [],

[default](#b841) [],

[descendent](#b1034) [],

[descendents](#b998) [],

[element](#b430) (defined in [Contains](#b432) complexType) [],

[element](#b587) (defined in [ForEach](#b589) complexType) [],

[element](#b617) (defined in [In](#b620) complexType) [],

[element](#b630) (defined in [IndexOf](#b632) complexType) [],

[element](#b722) (defined in [List](#b724) complexType) [],

[element](#b1125) (defined in [Within](#b1128) complexType) [],

[else](#b348) (defined in [Case](#b350) complexType) [],

[else](#b420) (defined in [Conditional](#b422) complexType) [],

[end](#b660) [],

[endDate](#b483) [],

[expression](#b550) (defined in [ExpressionDef](#b552) complexType) [],

[expression](#b1078) (in [trigger](#b1082)) [],

[granularity](#b476) (defined in [DateAdd](#b479) complexType) [],

[granularity](#b484) (defined in [DateDiff](#b486) complexType) [],

[granularity](#b490) (defined in [DatePart](#b492) complexType) [],

[index](#b624) [],

[initialValue](#b393) [],

[interval](#b1126) [],

[logic](#b412) [],

[message](#b755) [],

[numberOfPeriods](#b477) [],

[operand](#b336) (defined in [BinaryExpression](#b338) complexType) [],

[operand](#b623) (defined in [Indexer](#b626) complexType) [],

[operand](#b796) (defined in [*NaryExpression*](#b798) complexType) [],

[operand](#b991) (defined in [Round](#b994) complexType) [],

[operand](#b1061) (defined in [TernaryExpression](#b1063) complexType) [],

[operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [],

[pattern](#b889) [],

[precision](#b992) [],

[separator](#b400) (defined in [Combine](#b402) complexType) [],

[separator](#b1016) (defined in [Split](#b1018) complexType) [],

[source](#b297) (defined in [*AggregateExpression*](#b299) complexType) [],

[source](#b399) (defined in [Combine](#b402) complexType) [],

[source](#b429) (defined in [Contains](#b432) complexType) [],

[source](#b503) (defined in [Distinct](#b505) complexType) [],

[source](#b562) (defined in [Filter](#b565) complexType) [],

[source](#b576) (defined in [First](#b578) complexType) [],

[source](#b586) (defined in [ForEach](#b589) complexType) [],

[source](#b629) (defined in [IndexOf](#b632) complexType) [],

[source](#b706) (defined in [Last](#b708) complexType) [],

[source](#b915) (defined in [Property](#b917) complexType) [],

[source](#b1010) (defined in [Sort](#b1012) complexType) [],

[source](#b822) (in [actionSentence](#b1098) defined in [UpdateAction](#b1100) complexType) [],

[startDate](#b482) [],

[string](#b890) [],

[stringToSplit](#b1015) [],

[then](#b419) (defined in [Conditional](#b422) complexType) [],

[then](#b353) (in [caseItem](#b347)) [],

[timeOffset](#b965) [],

[value](#b921) (defined in [PropertyExpression](#b923) complexType) [],

value (in [item](#b729) defined in [ListConstraint](#b731) complexType),

[when](#b352) []

Annotation

The Expression type defines the abstract base type for all expressions used in the  
HeDS expression language.

XML Source (w/o annotations (1))

<xs:complexType abstract="true" name="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**description**](#b539)" type="xs:string"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b541) component only; 1/1)

 description

|  |  |
| --- | --- |
| Type: | xs:string, predefined, simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <description> | | |
|  | *Content:* | { xs:string } |
| </description> | | |

complexType "ExpressionConstraint"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b544) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [constraintType](#b925), [constraint](#b544) | |
| </...> | |

Content Model Elements (2):

[constraint](#b544) [], [constraintType](#b925) []

Annotation

A constraint specified in the form of an  
expression. The constraint type and the constraint expression  
combine together to specify the full constraint. For example,  
the constraint type specifies that the lower bound is being specified  
and the expression provides the lower bound value.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*RangeConstraint*](#b927) [] (extension)  **ExpressionConstraint** |

XML Source (w/o annotations (2))

<xs:complexType name="[**ExpressionConstraint**](#b546)">

<xs:complexContent>

<xs:extension base="[**RangeConstraint**](#b927)">

<xs:sequence>

<xs:element name="[**constraint**](#b544)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b546) component only; 1/2)

 constraint

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <constraint> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </constraint> | |

complexType "ExpressionDef"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b548), 1 [element](#b550) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [name](#b548) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [expression](#b550) | |
| </...> | |

Content Model Elements (1):

[expression](#b550) (defined in [ExpressionDef](#b552) complexType) []

All Direct / Indirect Based Elements (3):

|  |  |
| --- | --- |
| [actor](#b277) (in actor in [actors](#b263)) [],  def (in [expressions](#b686)), | def (in [externalData](#b685)) |

Annotation

The ExpressionDef type defines an expression and an associated  
name that can be referenced by any expression in the artifact.  
The name must be unique within the artifact.

XML Source (w/o annotations (1))

<xs:complexType name="[**ExpressionDef**](#b552)">

<xs:sequence>

<xs:element name="[**expression**](#b550)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**name**](#b548)" type="xs:string"/>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b552) component only; 1/1)

 name

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b552) component only; 1/1)

 expression

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <expression> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </expression> | |

complexType "ExpressionRef"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b555) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [name](#b555) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

The ExpressionRef type defines an expression that references a  
previously defined NamedExpression. The result of evaluating an  
ExpressionReference is the result of evaluating the referenced  
NamedExpression.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **ExpressionRef** |

XML Source (w/o annotations (1))

<xs:complexType name="[**ExpressionRef**](#b557)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**name**](#b555)" type="xs:string"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b557) component only; 1/1)

 name

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

complexType "Filter"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b560), 2 [elements](#b562) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [scope](#b560) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b562), [condition](#b563) | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [condition](#b563) (defined in [Filter](#b565) complexType) [],  [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b562) (defined in [Filter](#b565) complexType) [] |

Annotation

The filter expression returns a list with only those elements  
in the source list for which the condition element evaluates  
to true.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Filter** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Filter**](#b565)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**source**](#b562)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**condition**](#b563)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**scope**](#b560)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b565) component only; 1/1)

 scope

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b565) component only; 2/3)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

 condition

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <condition> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </condition> | |

complexType "FireEventAction"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b568) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)?, [textEquivalent](#b316)?, [eventType](#b568), [actionSentence](#b569)? | |
| </...> | |

Content Model Elements (9):

|  |  |
| --- | --- |
| [actionId](#b260) [],  [actionSentence](#b569) (defined in [FireEventAction](#b571) complexType) [],  [actors](#b263) [],  [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [],  [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [], | [eventType](#b568) (defined in [FireEventAction](#b571) complexType) [],  [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [],  [supportingResources](#b262) [],  [textEquivalent](#b316) [] |

Annotation

This action fires an event. The event can serve as  
a trigger to another artifact.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*ActionBase*](#b267) [] (extension)  [*AtomicAction*](#b318) [] (extension)  **FireEventAction** |

XML Source (w/o annotations (3))

<xs:complexType name="[**FireEventAction**](#b571)">

<xs:complexContent>

<xs:extension base="[**AtomicAction**](#b318)">

<xs:sequence>

<xs:element name="[**eventType**](#b568)" type="[**EventType**](#b1212)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**actionSentence**](#b569)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b571) component only; 2/9)

 eventType

|  |  |
| --- | --- |
| Type: | [EventType](#b1212) [], simple content |

The type of the event that is fired.

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <eventType> | | |
|  | *Content:* | { ("DataEvent" | "PeriodicEvent") | ("DataEvent" | "PeriodicEvent") } |
| </eventType> | | |

 actionSentence

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

The action sentence is the payload of the  
event. Another artifact receives this payload as an input.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <actionSentence> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </actionSentence> | |

complexType "First"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b574), 1 [element](#b576) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [orderBy](#b574) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b576) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b576) (defined in [First](#b578) complexType) [] |

Annotation

Returns the first element in a list. If the order  
by attribute is specified, the list is sorted by that  
ordering prior to returning the first element.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **First** |

XML Source (w/o annotations (1))

<xs:complexType name="[**First**](#b578)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element name="[**source**](#b576)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**orderBy**](#b574)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b578) component only; 1/1)

 orderBy

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b578) component only; 1/2)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

complexType "Floor"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

The Floor operator returns the first integer less than or  
equal to the argument. If the argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Floor** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Floor**](#b581)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "ForEach"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b584), 2 [elements](#b586) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [scope](#b584) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b586), [element](#b587) | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [element](#b587) (defined in [ForEach](#b589) complexType) [], | [source](#b586) (defined in [ForEach](#b589) complexType) [] |

Annotation

The ForEach expression iterates over the list of elements  
in the source element, and returns a list with the same  
number of elements, where each element in the new list is  
the result of evaluating the element expression for  
each element in the source list.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **ForEach** |

XML Source (w/o annotations (1))

<xs:complexType name="[**ForEach**](#b589)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**source**](#b586)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**element**](#b587)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**scope**](#b584)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b589) component only; 1/1)

 scope

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b589) component only; 2/3)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

 element

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <element> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </element> | |

complexType "Greater"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns true if the first argument is greater than the second argument.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Greater** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Greater**](#b592)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "GreaterOrEqual"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns true if the first argument is greater than or equal to the second argument.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **GreaterOrEqual** |

XML Source (w/o annotations (1))

<xs:complexType name="[**GreaterOrEqual**](#b595)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "GroupOrganizationBehavior"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b598) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [value](#b598) | |
| </...> | |

Content Model Elements (1):

[value](#b598) (defined in [GroupOrganizationBehavior](#b600) complexType) []

Annotation

For a group of actions, specifies the organizational intent of the grouping. This is meant to provide a hint to the system which displays the group of actions to an end user.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Behavior*](#b329) [] (extension)  **GroupOrganizationBehavior** |

XML Source (w/o annotations (1))

<xs:complexType name="[**GroupOrganizationBehavior**](#b600)">

<xs:complexContent>

<xs:extension base="[**Behavior**](#b329)">

<xs:sequence>

<xs:element name="[**value**](#b598)" type="[**GroupOrganizationBehaviorType**](#b1224)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b600) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | [GroupOrganizationBehaviorType](#b1224) [], simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <value> | | |
|  | *Content:* | { ("VisualGroup" | "LogicalGroup" | "SentenceGroup") | ("VisualGroup" | "LogicalGroup" | "SentenceGroup") } |
| </value> | | |

complexType "GroupSelectionBehavior"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b603) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [value](#b603) | |
| </...> | |

Content Model Elements (1):

[value](#b603) (defined in [GroupSelectionBehavior](#b605) complexType) []

Annotation

For a group of actions, specifies the number of actions that may be chosen by an end user.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Behavior*](#b329) [] (extension)  **GroupSelectionBehavior** |

XML Source (w/o annotations (1))

<xs:complexType name="[**GroupSelectionBehavior**](#b605)">

<xs:complexContent>

<xs:extension base="[**Behavior**](#b329)">

<xs:sequence>

<xs:element name="[**value**](#b603)" type="[**GroupSelectionBehaviorType**](#b1233)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b605) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | [GroupSelectionBehaviorType](#b1233) [], simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <value> | | |
|  | *Content:* | { ("Any" | "All" | "AllOrNone" | "ExactlyOne" | "AtMostOne" | "OneOrMore") | ("Any" | "All" | "AllOrNone" | "ExactlyOne" | "AtMostOne" | "OneOrMore") } |
| </value> | | |

complexType "IdentifierLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b608) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [root](#b608) | = | xs:string | | [extension](#b609) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type II with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **IdentifierLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**IdentifierLiteral**](#b611)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**root**](#b608)" type="[**dt:Uid**](#b194)" use="required"/>

<xs:attribute name="[**extension**](#b609)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b611) component only; 2/2)

 root

|  |  |
| --- | --- |
| Type: | [dt:Uid](#b194) [] |
| Use: | required |

Attribute Value

|  |
| --- |
| xs:string |

 extension

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

complexType "IfNull"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

If the first argument evaluates to null, returns the result of the  
second argument. Otherwise, returns the result of the first argument.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **IfNull** |

XML Source (w/o annotations (1))

<xs:complexType name="[**IfNull**](#b614)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "In"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b617) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [element](#b617), [collection](#b618) | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [collection](#b618) [],  [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [element](#b617) (defined in [In](#b620) complexType) [] |

Annotation

Returns true if the given element is in the source element.  
There are four overloads of this operator:  
Scalar, List : The type of the scalar must be the same as the element type of the list.  
List, List : The element type of both lists must be the same.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **In** |

XML Source (w/o annotations (1))

<xs:complexType name="[**In**](#b620)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**element**](#b617)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**collection**](#b618)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b620) component only; 2/3)

 element

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <element> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </element> | |

 collection

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <collection> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </collection> | |

complexType "Indexer"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b623) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b623), [index](#b624) | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [index](#b624) [], | [operand](#b623) (defined in [Indexer](#b626) complexType) [] |

Annotation

Returns the indexth element in a string or list.  
Indexes in strings and lists are defined to be one-based.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Indexer** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Indexer**](#b626)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**operand**](#b623)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**index**](#b624)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b626) component only; 2/3)

 operand

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <operand> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </operand> | |

 index

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <index> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </index> | |

complexType "IndexOf"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b629) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b629), [element](#b630) | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [element](#b630) (defined in [IndexOf](#b632) complexType) [], | [source](#b629) (defined in [IndexOf](#b632) complexType) [] |

Annotation

IndexOf returns the 1-based index of the given element in the given source list.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **IndexOf** |

XML Source (w/o annotations (1))

<xs:complexType name="[**IndexOf**](#b632)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**source**](#b629)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**element**](#b630)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b632) component only; 2/3)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

 element

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <element> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </element> | |

complexType "InlineResource"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b635) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [identifiers](#b694)?, [templateIds](#b695)?, [title](#b696)?, [location](#b697)?, [description](#b698)?, [citation](#b699)?, [content](#b635)? | |
| </...> | |

Content Model Elements (7):

|  |  |
| --- | --- |
| [citation](#b699) [],  [content](#b635) [],  [description](#b698) (defined in [KnowledgeResource](#b701) complexType) [],  [identifiers](#b694) (defined in [KnowledgeResource](#b701) complexType) [], | [location](#b697) [],  [templateIds](#b695) (defined in [KnowledgeResource](#b701) complexType) [],  [title](#b696) (defined in [KnowledgeResource](#b701) complexType) [] |

All Direct / Indirect Based Elements (1):

[documentation](#b766) []

Annotation

An Inline Resource consists of both the resource  
reference information and the actual resource content/payload to be  
inserted inline. The content of the document must be represented in  
valid xhtml format within the content/div node.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [KnowledgeResource](#b701) [] (extension)  **InlineResource** |

XML Source (w/o annotations (2))

<xs:complexType name="[**InlineResource**](#b637)">

<xs:complexContent>

<xs:extension base="[**KnowledgeResource**](#b701)">

<xs:sequence>

<xs:element minOccurs="0" name="[**content**](#b635)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b637) component only; 1/7)

 content

|  |  |
| --- | --- |
| Type: | xs:anyType, any content |

The document content in xhtml format.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <content> | |
|  | ... |
| </content> | |

complexType "IntegerIntervalLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 4 [attributes](#b640) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [low](#b640) | = | xs:int | | [high](#b641) | = | xs:int | | [lowIsInclusive](#b642) | = | xs:boolean | | [highIsInclusive](#b643) | = | xs:boolean | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type IVL\_INT with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **IntegerIntervalLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**IntegerIntervalLiteral**](#b645)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**low**](#b640)" type="xs:int"/>

<xs:attribute name="[**high**](#b641)" type="xs:int"/>

<xs:attribute name="[**lowIsInclusive**](#b642)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b643)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b645) component only; 4/4)

 low

|  |  |
| --- | --- |
| Type: | xs:int, predefined |
| Use: | optional |

 high

|  |  |
| --- | --- |
| Type: | xs:int, predefined |
| Use: | optional |

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

complexType "IntegerLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b648) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b648) | = | xs:int | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type INT with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **IntegerLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**IntegerLiteral**](#b650)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**value**](#b648)" type="xs:int" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b650) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | xs:int, predefined |
| Use: | required |

complexType "Intersect"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b796)\* | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b796) (defined in [*NaryExpression*](#b798) complexType) [] |

Annotation

Returns the intersection of the operands.  
This operator has two overloads:  
List  
Interval  
  
For the list overload, this operator returns a list with  
the elements that appear in the lists.  
  
For the interval overload, this operator returns the interval  
that defines the overlapping portion of the operands. Note that  
if the operands do not overlap, this operator returns null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*NaryExpression*](#b798) [] (extension)  **Intersect** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Intersect**](#b653)">

<xs:complexContent>

<xs:extension base="[**NaryExpression**](#b798)"/>

</xs:complexContent>

</xs:complexType>

complexType "Interval"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b656), 2 [elements](#b659) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [beginOpen](#b656) | = | xs:boolean : "false" | | [endOpen](#b657) | = | xs:boolean : "false" | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [begin](#b659)?, [end](#b660)? | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [begin](#b659) [],  [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [end](#b660) [] |

Annotation

The Interval selector defines an interval value. An interval must be defined  
on values that support comparison, as well as successor and predecessor values.  
The beginning and ending of the interval may each be defined as open or closed.  
The default is closed, indicating an inclusive interval.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Interval** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Interval**](#b662)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**begin**](#b659)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**end**](#b660)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute default="false" name="[**beginOpen**](#b656)" type="xs:boolean" use="optional"/>

<xs:attribute default="false" name="[**endOpen**](#b657)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b662) component only; 2/2)

 beginOpen

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

Attribute Value

|  |  |
| --- | --- |
| Default: | "false" |

 endOpen

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

Attribute Value

|  |  |
| --- | --- |
| Default: | "false" |

Content Element Detail (all declarations; defined within [this](#b662) component only; 2/3)

 begin

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <begin> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </begin> | |

 end

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <end> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </end> | |

complexType "InValueSet"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [attributes](#b665) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [id](#b665) | = | xs:string | | [version](#b666) | = | xs:string | | [authority](#b667) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns true if the given code, or list of codes, is in the  
given value set.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **InValueSet** |

XML Source (w/o annotations (1))

<xs:complexType name="[**InValueSet**](#b669)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**id**](#b665)" type="xs:string" use="required"/>

<xs:attribute name="[**version**](#b666)" type="xs:string" use="optional"/>

<xs:attribute name="[**authority**](#b667)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b669) component only; 3/3)

 id

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

 version

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 authority

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

complexType "IsEmpty"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

Returns true if the list contains no elements.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **IsEmpty** |

XML Source (w/o annotations (1))

<xs:complexType name="[**IsEmpty**](#b672)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "IsNotEmpty"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

Returns true if the list contains any elements.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **IsNotEmpty** |

XML Source (w/o annotations (1))

<xs:complexType name="[**IsNotEmpty**](#b675)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "IsNull"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

Returns true if the argument evaluates to null, false otherwise.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **IsNull** |

XML Source (w/o annotations (1))

<xs:complexType name="[**IsNull**](#b678)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "ItemDefinition"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definition of 1 [element](#b680) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [identifier](#b680)? | |
| </...> | |

Content Model Elements (1):

[identifier](#b680) (defined in [*ItemDefinition*](#b682) complexType) []

Known Direct Subtypes (1):

[DocumentationItem](#b518) []

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [documentationConcept](#b392) [], | [item](#b251) (in [itemDefinitions](#b253)) [] |

Annotation

An item definition is the equivalent of an item in  
a data dictionary or a catalog in an electronic health record  
system.

XML Source (w/o annotations (2))

<xs:complexType abstract="true" name="[**ItemDefinition**](#b682)">

<xs:sequence>

<xs:element minOccurs="0" name="[**identifier**](#b680)" type="[**VersionedIdentifier**](#b1122)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b682) component only; 1/1)

 identifier

|  |  |
| --- | --- |
| Type: | [VersionedIdentifier](#b1122) [], empty content |

The identifier for the item.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <identifier | |
|  | |  |  |  | | --- | --- | --- | | [extension](#b62) | = | xs:string : "" | | [root](#b61) | = | xs:string | | [version](#b1120) | = | xs:string | |
| /> | |

complexType "KnowledgeDocument"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 7 [elements](#b684) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [metadata](#b684), [externalData](#b685)?, [expressions](#b686)?, [triggers](#b687)?, [conditions](#b688)?, [behaviors](#b689)?, [actionGroups](#b690) | |
| </...> | |

Content Model Elements (7):

|  |  |
| --- | --- |
| [actionGroups](#b690) [],  [behaviors](#b689) (in [knowledgeDocument](#b255)) [],  [conditions](#b688) (in [knowledgeDocument](#b255)) [],  [expressions](#b686) [], | [externalData](#b685) [],  [metadata](#b684) [],  [triggers](#b687) [] |

All Direct / Indirect Based Elements (1):

[knowledgeDocument](#b255) []

Annotation

A knowledge document is an instance of a CDS  
knowledge artifact such as a rule, an order set, or a documentation  
template

XML Source (w/o annotations (10))

<xs:complexType name="[**KnowledgeDocument**](#b692)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**metadata**](#b684)" type="[**Metadata**](#b779)"/>

<!-- Probably should constrain using schematron the expression type to  
Request -->

<xs:element maxOccurs="1" minOccurs="0" name="[**externalData**](#b685)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" minOccurs="0" name="parameter" type="[**ParameterDef**](#b843)"/>

<xs:element maxOccurs="unbounded" minOccurs="0" name="def" type="[**ExpressionDef**](#b552)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element maxOccurs="1" minOccurs="0" name="[**expressions**](#b686)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" minOccurs="0" name="def" type="[**ExpressionDef**](#b552)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element maxOccurs="1" minOccurs="0" name="[**triggers**](#b687)" type="[**Triggers**](#b1084)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**conditions**](#b688)" type="[**Conditions**](#b426)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**behaviors**](#b689)" type="[**Behaviors**](#b333)"/>

<!-- actions -->

<xs:element maxOccurs="1" minOccurs="1" name="[**actionGroups**](#b690)" type="[**ActionGroup**](#b275)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b692) component only; 7/7)

 metadata

|  |  |
| --- | --- |
| Type: | [Metadata](#b779) [], complex content |

The metadata section of the knowledge document defines the core metadata associated with this CDS knowledge artifact such as (1) the unique identifier for this artifact, (2) the unique identifier for its associated template(s), (3) the title and description of the artifact, (4) the status and history of the artifact, (5) any relevant entities associated with this artifact, and (6) information needed to categorize and retrieve the artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <metadata> | |
|  | |  |  | | --- | --- | | *Content:* | [identifiers](#b759), [artifactType](#b760), [schemaIdentifier](#b761), [templateIds](#b762)?, [dataModels](#b763)?, [title](#b764), [description](#b765)?, [documentation](#b766)?, [relatedResources](#b767)?, [supportingEvidence](#b768)?, [applicability](#b769)?, [keyTerms](#b770)?, [categories](#b771)?, [language](#b772)?, [status](#b773), [eventHistory](#b774)?, [contributions](#b775)?, [publishers](#b776)?, [usageTerms](#b777)? | |
| </metadata> | |

 externalData

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

The externalData section allows a CDS artifact author to define 'named expressions' to fetch information from an external source and bind this information to the 'context' of the knowledge artifact for later reference by the logic modules (e.g., the condition for the knowledge artifact or actions). It is the responsibility of the implementation to determine the nature of this boundary and how to fetch this information. For instance, one may write an expression to retrieve from a patient vMR the age of a patient or a list of clinical problems whose problem code are contained in a given ICD-9 value set. The age value and the list of patient problems may then be used in the 'condition' section of the same knowledge artifact to determine the applicability of the knowledge document to the given patient.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <externalData> | |
|  | |  |  | | --- | --- | | *Content:* | parameter\*, def\* | |
| </externalData> | |

 expressions

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

The expressions section allows a CDS artifact author to define 'named expressions' that can be referenced anywhere within expressions in the artifact. This allows expression logic to be reused, as well as to be organized for readability and maintainability.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <expressions> | |
|  | |  |  | | --- | --- | | *Content:* | def\* | |
| </expressions> | |

 triggers

|  |  |
| --- | --- |
| Type: | [Triggers](#b1084) [], complex content |

The triggers section defines the list of all triggers that 'activate' or 'trigger' the CDS knowledge artifact. For instance, opening a patient record may trigger a rule to execute if the conditions of the rule are met.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <triggers> | |
|  | |  |  | | --- | --- | | *Content:* | [trigger](#b1082)+ | |
| </triggers> | |

 conditions

|  |  |
| --- | --- |
| Type: | [Conditions](#b426) [], complex content |

The conditions section lists all conditions that pertain to the knowledge artifact. Conditions define the logic that determine the applicability of the artifact in the given context, any precondition or post condition, and/or any inclusion and exclusion criteria for the given CDS artifact. Conditions are structured as expressions to be evaluated in the target system.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <conditions> | |
|  | |  |  | | --- | --- | | *Content:* | [condition](#b424)+ | |
| </conditions> | |

 behaviors

|  |  |
| --- | --- |
| Type: | [Behaviors](#b333) [], complex content |

The behaviors section defines the set of behaviors for this knowledge document. While there are no artifact-level behaviors defined at this time, this element is included as a point of extension, should it be needed.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <behaviors> | |
|  | |  |  | | --- | --- | | *Content:* | [behavior](#b331)+ | |
| </behaviors> | |

 actionGroups

|  |  |
| --- | --- |
| Type: | [ActionGroup](#b275) [], complex content |

The actionGroups is essentially the top-level container for the groups of actions that make up a given knowledge document. This container defines the main content of the knowledge artifact: logical grouping constructs such as the clinical sections and orderables in an order set, the tasks to be performed by a rule, or the sections and menu choices that make up a document template.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <actionGroups> | |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)?, [title](#b270)?, [description](#b271)?, [representedConcepts](#b272)?, [subElements](#b273) | |
| </actionGroups> | |

complexType "KnowledgeResource"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 6 [elements](#b694) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [identifiers](#b694)?, [templateIds](#b695)?, [title](#b696)?, [location](#b697)?, [description](#b698)?, [citation](#b699)? | |
| </...> | |

Content Model Elements (6):

|  |  |
| --- | --- |
| [citation](#b699) [],  [description](#b698) (defined in [KnowledgeResource](#b701) complexType) [],  [identifiers](#b694) (defined in [KnowledgeResource](#b701) complexType) [], | [location](#b697) [],  [templateIds](#b695) (defined in [KnowledgeResource](#b701) complexType) [],  [title](#b696) (defined in [KnowledgeResource](#b701) complexType) [] |

Known Direct Subtypes (1):

[InlineResource](#b637) []

All Direct / Indirect Based Elements (3):

|  |  |
| --- | --- |
| [documentation](#b766) [],  [resource](#b1051) (defined in [SupportingResource](#b1053) complexType) [], | resource (in [resources](#b975) in relatedResource) |

Annotation

KnowledgeResource specifies a reference to an  
associated resource of relevance to the artifact such as a  
guideline, a performance measure, another knowledge artifact, or a  
source of evidence for the artifact.

XML Source (w/o annotations (9))

<xs:complexType name="[**KnowledgeResource**](#b701)">

<xs:sequence>

<xs:element minOccurs="0" name="[**identifiers**](#b694)">

<xs:complexType>

<xs:sequence>

<xs:element name="identifier" type="[**VersionedIdentifier**](#b1122)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**templateIds**](#b695)">

<xs:complexType>

<xs:sequence>

<xs:element name="templateId" type="[**dt:II**](#b64)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element maxOccurs="1" minOccurs="0" name="[**title**](#b696)" type="[**dt:ST**](#b131)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**location**](#b697)" type="[**dt:TEL**](#b139)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**description**](#b698)" type="[**dt:ST**](#b131)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**citation**](#b699)" type="[**dt:ST**](#b131)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b701) component only; 6/6)

 identifiers

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

The set of unique identifiers for this resource.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <identifiers> | |
|  | |  |  | | --- | --- | | *Content:* | identifier | |
| </identifiers> | |

 templateIds

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

The set of unique identifiers for the templates  
associated with this resource.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <templateIds> | |
|  | |  |  | | --- | --- | | *Content:* | templateId | |
| </templateIds> | |

 title

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

The title of the document

|  |  |
| --- | --- |
| XML Representation Summary | |
| <title | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 location

|  |  |
| --- | --- |
| Type: | [dt:TEL](#b139) [], empty content |

The URL of the given resource.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <location | |
|  | |  |  |  | | --- | --- | --- | | [capabilities](#b136) | = | *list of* ("voice" | "fax" | "data" | "tty" | "sms") | | [use](#b135) | = | *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "TMP" | "AS" | "EC" | "MC" | "PG") | | [useablePeriodOriginalText](#b137) | = | xs:string | | [value](#b134) | = | xs:anyURI | |
| /> | |

 description

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

A short textual description of the resource.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <description | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 citation

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

The resource citation.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <citation | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

complexType "Last"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b704), 1 [element](#b706) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [orderBy](#b704) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b706) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b706) (defined in [Last](#b708) complexType) [] |

Annotation

Returns the last element in a list. If the order by  
attribute is specified, the list is sorted by that  
ordering prior to returning the last element.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Last** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Last**](#b708)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element name="[**source**](#b706)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**orderBy**](#b704)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b708) component only; 1/1)

 orderBy

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b708) component only; 1/2)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

complexType "Length"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

Returns the length of its argument. For strings, the length is the number of  
characters in the string. For intervals, the length is defined as the ending  
point minus the beginning point.  
If the argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Length** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Length**](#b711)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "Less"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns true if the first argument is less than the second argument.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Less** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Less**](#b714)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "LessOrEqual"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns true if the first argument is less than or equal to the second argument.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **LessOrEqual** |

XML Source (w/o annotations (1))

<xs:complexType name="[**LessOrEqual**](#b717)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "List"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b720), 1 [element](#b722) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [key](#b720) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [element](#b722)\* | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [element](#b722) (defined in [List](#b724) complexType) [] |

Annotation

The List selector returns a value of type List, whose elements are the result  
of evaluating the arguments to the List selector, in order.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **List** |

XML Source (w/o annotations (1))

<xs:complexType name="[**List**](#b724)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="unbounded" minOccurs="0" name="[**element**](#b722)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**key**](#b720)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b724) component only; 1/1)

 key

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b724) component only; 1/2)

 element

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <element> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </element> | |

complexType "ListConstraint"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b727), 1 [element](#b729) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [strictSelection](#b727) | = | xs:boolean | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [constraintType](#b925), [item](#b729)+ | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [constraintType](#b925) [], | [item](#b729) (defined in [ListConstraint](#b731) complexType) [] |

Annotation

A constraint specifying that the value is from a  
list included here.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*RangeConstraint*](#b927) [] (extension)  **ListConstraint** |

XML Source (w/o annotations (7))

<xs:complexType name="[**ListConstraint**](#b731)">

<xs:complexContent>

<xs:extension base="[**RangeConstraint**](#b927)">

<xs:sequence>

<xs:element maxOccurs="unbounded" minOccurs="1" name="[**item**](#b729)">

<xs:complexType>

<xs:sequence>

<xs:element name="value" type="[**Expression**](#b541)"/>

<xs:element minOccurs="0" name="codes">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" minOccurs="0" name="code" type="[**dt:CD**](#b33)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="displayText" type="[**dt:ST**](#b131)"/>

<!-- <xs:element name="numericEquivalent" type="xs:decimal" maxOccurs="1"  
minOccurs="0" > <xs:annotation> <xs:documentation>An equivalent numeric value  
for the field. This is used when the documentation item is part of a set  
of items that are used to compute a score</xs:documentation> </xs:annotation></xs:element> -->

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

<xs:attribute name="[**strictSelection**](#b727)" type="xs:boolean"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b731) component only; 1/1)

 strictSelection

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

If strictSelection is set to true, the value  
entered by the user must be  
an item from the list. If this is set  
to false, the value may not be restricted to this list.

Content Element Detail (all declarations; defined within [this](#b731) component only; 1/2)

 item

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

An item in the list of possible values

|  |  |
| --- | --- |
| XML Representation Summary | |
| <item> | |
|  | |  |  | | --- | --- | | *Content:* | value, codes?, displayText? | |
| </item> | |

complexType "Literal"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b734) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [valueType](#b734) | = | xs:QName | | [value](#b735) | = | xs:anySimpleType | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

The Literal type defines a single scalar value. For example, the literal 5,  
the boolean value true or the string "Code".

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Literal** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Literal**](#b737)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**valueType**](#b734)" type="xs:QName" use="required"/>

<xs:attribute name="[**value**](#b735)" type="xs:anySimpleType" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b737) component only; 2/2)

 valueType

|  |  |
| --- | --- |
| Type: | xs:QName, predefined |
| Use: | required |

 value

|  |  |
| --- | --- |
| Type: | xs:anySimpleType, predefined |
| Use: | required |

complexType "Ln"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

The Ln operator computes the natural logarithm of its argument.  
If the argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Ln** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Ln**](#b740)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "Log"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

The Log operator computes the logarithm of its first argument,  
using the second argument as the base.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Log** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Log**](#b743)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "Lower"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

Returns the lower case of its argument.  
If the argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Lower** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Lower**](#b746)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "Max"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [path](#b295) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b297) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b297) (defined in [*AggregateExpression*](#b299) complexType) [] |

Annotation

Returns the maximum element in the source.  
  
If a path is specified, elements with no value for the  
property specified by the path are ignored.  
  
If source contains no non-null elements, null is returned.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*AggregateExpression*](#b299) [] (extension)  **Max** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Max**](#b749)">

<xs:complexContent>

<xs:extension base="[**AggregateExpression**](#b299)"/>

</xs:complexContent>

</xs:complexType>

complexType "Meets"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns true if the first interval ends on the starting point of the second,  
or if the first interval starts on the ending point of the second. In other  
words, if the ending point of the first interval is equal to the starting  
point of the second, or if the starting point of the first interval is  
equal to the ending point of the second.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Meets** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Meets**](#b752)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "MessageAction"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b755) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)?, [textEquivalent](#b316)?, [message](#b755) | |
| </...> | |

Content Model Elements (8):

|  |  |
| --- | --- |
| [actionId](#b260) [],  [actors](#b263) [],  [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [],  [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [], | [message](#b755) [],  [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [],  [supportingResources](#b262) [],  [textEquivalent](#b316) [] |

Annotation

An action to deliver a message to a destination  
using an optionally specified channel.  
The recipient is specified  
using the actor element.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*ActionBase*](#b267) [] (extension)  [*AtomicAction*](#b318) [] (extension)  **MessageAction** |

XML Source (w/o annotations (2))

<xs:complexType name="[**MessageAction**](#b757)">

<xs:complexContent>

<xs:extension base="[**AtomicAction**](#b318)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**message**](#b755)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b757) component only; 1/8)

 message

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

The expression resolves to a message to be  
delivered.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <message> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </message> | |

complexType "Metadata"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 19 [elements](#b759) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [identifiers](#b759), [artifactType](#b760), [schemaIdentifier](#b761), [templateIds](#b762)?, [dataModels](#b763)?, [title](#b764), [description](#b765)?, [documentation](#b766)?, [relatedResources](#b767)?, [supportingEvidence](#b768)?, [applicability](#b769)?, [keyTerms](#b770)?, [categories](#b771)?, [language](#b772)?, [status](#b773), [eventHistory](#b774)?, [contributions](#b775)?, [publishers](#b776)?, [usageTerms](#b777)? | |
| </...> | |

Content Model Elements (19):

|  |  |
| --- | --- |
| [applicability](#b769) [],  [artifactType](#b760) [],  [categories](#b771) [],  [contributions](#b775) [],  [dataModels](#b763) [],  [description](#b765) (in metadata) [],  [documentation](#b766) [],  [eventHistory](#b774) [],  [identifiers](#b759) (in metadata) [],  [keyTerms](#b770) [], | [language](#b772) [],  [publishers](#b776) [],  [relatedResources](#b767) [],  [schemaIdentifier](#b761) [],  [status](#b773) [],  [supportingEvidence](#b768) (in metadata) [],  [templateIds](#b762) (in metadata) [],  [title](#b764) (in metadata) [],  [usageTerms](#b777) [] |

All Direct / Indirect Based Elements (1):

[metadata](#b684) []

Annotation

The container for all of the metadata associated  
with a CDS knowledge artifact. Ideally, the metadata for artifacts  
is provided independently by the publisher for determining which  
artifact to retrieve.

XML Source (w/o annotations (17))

<xs:complexType name="[**Metadata**](#b779)">

<xs:sequence>

<xs:element name="[**identifiers**](#b759)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="identifier" type="[**VersionedIdentifier**](#b1122)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="[**artifactType**](#b760)">

<xs:complexType>

<xs:attribute name="value" type="[**ArtifactType**](#b1149)" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="[**schemaIdentifier**](#b761)" type="[**VersionedIdentifier**](#b1122)"/>

<xs:element minOccurs="0" name="[**templateIds**](#b762)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="templateId" type="[**VersionedIdentifier**](#b1122)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**dataModels**](#b763)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="modelReference" type="[**ModelReference**](#b787)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="[**title**](#b764)" type="[**dt:ST**](#b131)"/>

<xs:element minOccurs="0" name="[**description**](#b765)" type="[**dt:ST**](#b131)"/>

<xs:element minOccurs="0" name="[**documentation**](#b766)" type="[**InlineResource**](#b637)"/>

<xs:element minOccurs="0" name="[**relatedResources**](#b767)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="relatedResource" type="[**ResourceRelationshipReference**](#b977)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**supportingEvidence**](#b768)" type="[**SupportingEvidence**](#b1049)"/>

<xs:element minOccurs="0" name="[**applicability**](#b769)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="coverage" type="[**Coverage**](#b451)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**keyTerms**](#b770)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="term" type="[**dt:CD**](#b33)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**categories**](#b771)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="category" type="[**dt:CD**](#b33)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**language**](#b772)" type="[**dt:CD**](#b33)"/>

<xs:element name="[**status**](#b773)">

<xs:complexType>

<xs:attribute name="value" type="[**ArtifactStatusType**](#b1140)" use="required"/>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**eventHistory**](#b774)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="artifactLifeCycleEvent" type="[**ArtifactLifeCycleEvent**](#b313)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**contributions**](#b775)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="contribution" type="[**Contribution**](#b437)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**publishers**](#b776)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="publisher" type="[**Party**](#b853)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**usageTerms**](#b777)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="rightsDeclaration" type="[**RightsDeclaration**](#b988)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b779) component only; 19/19)

 identifiers

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

Each version of a CDS knowledge artifact may have more than one identifier associated with it. Note that each identifier must be globally unique in the universe of CDS knowledge artifacts in which a particular artifact resides.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <identifiers> | |
|  | |  |  | | --- | --- | | *Content:* | identifier+ | |
| </identifiers> | |

 artifactType

|  |  |
| --- | --- |
| Type: | anonymous complexType, empty content |

Currently three types of artifacts are in scope for Health eDecisions Use Case #1: order sets, event-condition-action rules, and documentation templates. Additional types will be added in future revisions of the standard.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <artifactType | |
|  | |  |  |  | | --- | --- | --- | | value | = | (("Rule" | "Order Set" | "Documentation Template") | ("Rule" | "Order Set" | "Documentation Template")) | |
| /> | |

 schemaIdentifier

|  |  |
| --- | --- |
| Type: | [VersionedIdentifier](#b1122) [], empty content |

This is the identifier of the XML schema (and its version) which governs the structure of this CDS Knowledge Artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <schemaIdentifier | |
|  | |  |  |  | | --- | --- | --- | | [extension](#b62) | = | xs:string : "" | | [root](#b61) | = | xs:string | | [version](#b1120) | = | xs:string | |
| /> | |

 templateIds

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

These are the identifiers of templates which  
further constrain the structure of this knowledge artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <templateIds> | |
|  | |  |  | | --- | --- | | *Content:* | templateId+ | |
| </templateIds> | |

 dataModels

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

Set of data models referenced in the Expression  
objects in this knowledge artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <dataModels> | |
|  | |  |  | | --- | --- | | *Content:* | modelReference+ | |
| </dataModels> | |

 title

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <title | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 description

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <description | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 documentation

|  |  |
| --- | --- |
| Type: | [InlineResource](#b637) [], complex content |

Documentation for this knowledge reference may  
consist of a reference to an external resource; the documentation  
may also be included in-line if desired.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <documentation> | |
|  | |  |  | | --- | --- | | *Content:* | [identifiers](#b694)?, [templateIds](#b695)?, [title](#b696)?, [location](#b697)?, [description](#b698)?, [citation](#b699)?, [content](#b635)? | |
| </documentation> | |

 relatedResources

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

A set of resources related to this artifact,  
along with an indication of the type of relationship. An artifact  
may be derived from or depend on other artifacts, along with other  
types of relationships. See the Artifact Lifecycle diagram in the  
Implementation Guide for more information.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <relatedResources> | |
|  | |  |  | | --- | --- | | *Content:* | relatedResource+ | |
| </relatedResources> | |

 supportingEvidence

|  |  |
| --- | --- |
| Type: | [SupportingEvidence](#b1049) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <supportingEvidence> | |
|  | |  |  | | --- | --- | | *Content:* | [evidence](#b1047)+ | |
| </supportingEvidence> | |

 applicability

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

Specifies the conditions under which this  
artifact is applicable.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <applicability> | |
|  | |  |  | | --- | --- | | *Content:* | coverage+ | |
| </applicability> | |

 keyTerms

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

Provides a list of coded key terms that pertain to this artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <keyTerms> | |
|  | |  |  | | --- | --- | | *Content:* | term+ | |
| </keyTerms> | |

 categories

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

Provides a list of coded categories to which this  
artifact belongs.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <categories> | |
|  | |  |  | | --- | --- | | *Content:* | category+ | |
| </categories> | |

 language

|  |  |
| --- | --- |
| Type: | [dt:CD](#b33) [], empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <language | |
|  | |  |  |  | | --- | --- | --- | | [code](#b27) | = | xs:string | | [codeSystem](#b28) | = | xs:string | | [codeSystemName](#b29) | = | xs:string | | [displayName](#b30) | = | xs:string | | [originalText](#b31) | = | xs:string : "" | |
| /> | |

 status

|  |  |
| --- | --- |
| Type: | anonymous complexType, empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <status | |
|  | |  |  |  | | --- | --- | --- | | value | = | (("Draft" | "InTest" | "Active" | "Inactive") | ("Draft" | "InTest" | "Active" | "Inactive")) | |
| /> | |

 eventHistory

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

This is the history of events which have occurred  
for this particular version of the artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <eventHistory> | |
|  | |  |  | | --- | --- | | *Content:* | artifactLifeCycleEvent+ | |
| </eventHistory> | |

 contributions

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

Includes a list of people and/or organizations  
who have contributed to the development of this artifact.  
Contributions are not necessarily tied to specific versions of the  
artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <contributions> | |
|  | |  |  | | --- | --- | | *Content:* | contribution+ | |
| </contributions> | |

 publishers

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

The set of people and/or organizations who  
publish the artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <publishers> | |
|  | |  |  | | --- | --- | | *Content:* | publisher+ | |
| </publishers> | |

 usageTerms

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

This is the set of rights reserved by the person  
or organization holding the rights to this artifact, along with  
the set of permissions granted to consumers.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <usageTerms> | |
|  | |  |  | | --- | --- | | *Content:* | rightsDeclaration+ | |
| </usageTerms> | |

complexType "Min"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [path](#b295) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b297) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b297) (defined in [*AggregateExpression*](#b299) complexType) [] |

Annotation

Returns the minimum element in the source.  
  
If a path is specified, elements with no value for the  
property specified by the path are ignored.  
  
If source contains no non-null elements, null is returned.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*AggregateExpression*](#b299) [] (extension)  **Min** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Min**](#b782)">

<xs:complexContent>

<xs:extension base="[**AggregateExpression**](#b299)"/>

</xs:complexContent>

</xs:complexType>

complexType "ModelReference"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b784) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b784), [referencedModel](#b785) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b784) (in modelReference) [], | [referencedModel](#b785) [] |

All Direct / Indirect Based Elements (1):

modelReference

Annotation

A reference to some model by its Universal Resource Identifier.

XML Source (w/o annotations (4))

<xs:complexType name="[**ModelReference**](#b787)">

<xs:sequence>

<xs:element name="[**description**](#b784)" type="[**dt:ST**](#b131)"/>

<xs:element name="[**referencedModel**](#b785)">

<xs:complexType>

<xs:attribute name="value" type="xs:anyURI"/>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b787) component only; 2/2)

 description

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

The object is assigned a name that can be used in  
this artifact.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <description | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 referencedModel

|  |  |
| --- | --- |
| Type: | anonymous complexType, empty content |

The identifier of the object that is being  
referenced.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <referencedModel | |
|  | |  |  |  | | --- | --- | --- | | value | = | xs:anyURI | |
| /> | |

complexType "Modulo"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

The Modulo operator computes the remainder of the division of its  
arguments. If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Modulo** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Modulo**](#b790)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "Multiply"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

The Multiply operator performs numeric multiplication of its arguments.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Multiply** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Multiply**](#b793)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "NaryExpression"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definition of 1 [element](#b796) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b796)\* | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b796) (defined in [*NaryExpression*](#b798) complexType) [] |

Known Direct Subtypes (6):

[And](#b305) [], [Coalesce](#b372) [], [Concat](#b410) [], [Intersect](#b653) [], [Or](#b828) [], [Union](#b1095) []

Annotation

The Expression type defines an abstract base class for an expression  
that takes any number of arguments, including zero.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  ***NaryExpression*** |

XML Source (w/o annotations (1))

<xs:complexType abstract="true" name="[**NaryExpression**](#b798)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="unbounded" minOccurs="0" name="[**operand**](#b796)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b798) component only; 1/2)

 operand

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <operand> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </operand> | |

complexType "Negate"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

The Negate operator returns the negative of its argument.  
If the argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Negate** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Negate**](#b801)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "Not"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

The Not operator returns the logical negation of its argument. If the argument  
is true, the result is false; if the argument is false, the result is true;  
otherwise, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Not** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Not**](#b804)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "NotEqual"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

The NotEqual operator returns true if its argument are not the  
same value.  
  
The NotEqual operator is a shorthand for invocation of logical  
negation of the Equal operator.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **NotEqual** |

XML Source (w/o annotations (1))

<xs:complexType name="[**NotEqual**](#b807)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "Now"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns the date and time of the start timestamp associated with the  
evaluation request. Now is defined in this way for two reasons:  
1) The operation will always return the same value within any given evaluation,  
ensuring that the result of an expression containing Now will always return the  
same result.  
2) The operation will return the timestamp associated with the evaluation  
request, allowing the evaluation to be performed with the same timezone  
information as the data delivered with the evaluation request.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Now** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Now**](#b810)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)"/>

</xs:complexContent>

</xs:complexType>

complexType "ObjectExpression"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b813), 1 [element](#b815) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [objectType](#b813) | = | xs:QName | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [property](#b815)\* | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [property](#b815) (defined in [ObjectExpression](#b817) complexType) [] |

Annotation

The ObjectExpression type allows objects of any type to be built  
up as an expression. The objectType attribute specifies the type of  
the object being built, and the list of property elements specify  
the values for the properties of the object. Note that the value  
of a property may be any expression, including another ObjectLiteral.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **ObjectExpression** |

XML Source (w/o annotations (1))

<xs:complexType name="[**ObjectExpression**](#b817)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="unbounded" minOccurs="0" name="[**property**](#b815)" type="[**PropertyExpression**](#b923)"/>

</xs:sequence>

<xs:attribute name="[**objectType**](#b813)" type="xs:QName"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b817) component only; 1/1)

 objectType

|  |  |
| --- | --- |
| Type: | xs:QName, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b817) component only; 1/2)

 property

|  |  |
| --- | --- |
| Type: | [PropertyExpression](#b923) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <property | |
|  | |  |  |  | | --- | --- | --- | | [name](#b919) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [value](#b921) | |
| </property> | |

complexType "ObjectRedefine"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b820), 2 [elements](#b822) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [scope](#b820) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b822)?, [property](#b823)+ | |
| </...> | |

Content Model Elements (3):

[description](#b539) (defined in [*Expression*](#b541) complexType) [],

[property](#b823) (in actionSentence defined in [UpdateAction](#b1100) complexType) [],

[source](#b822) (in actionSentence defined in [UpdateAction](#b1100) complexType) []

All Direct / Indirect Based Elements (1):

[actionSentence](#b1098) (defined in [UpdateAction](#b1100) complexType) []

Annotation

The ObjectRedefine expression returns an object of the same type as the  
source argument, with the same values for each property, except the values  
of the properties listed in the property elements. For those properties,  
the values will be set based on the expressions in those elements. This  
operator allows an object to be "copied" with new values specified only  
for a given list of properties.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **ObjectRedefine** |

XML Source (w/o annotations (1))

<xs:complexType name="[**ObjectRedefine**](#b825)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**source**](#b822)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="unbounded" minOccurs="1" name="[**property**](#b823)" type="[**PropertyExpression**](#b923)"/>

</xs:sequence>

<xs:attribute name="[**scope**](#b820)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b825) component only; 1/1)

 scope

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b825) component only; 2/3)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

 property

|  |  |
| --- | --- |
| Type: | [PropertyExpression](#b923) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <property | |
|  | |  |  |  | | --- | --- | --- | | [name](#b919) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [value](#b921) | |
| </property> | |

complexType "Or"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b796)\* | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b796) (defined in [*NaryExpression*](#b798) complexType) [] |

Annotation

The Or operator returns the logical disjunction of its arguments. Note that  
this operator is defined as n-ary, allowing any number of arguments. The result  
of Or with no arguments is defined to be true. The result of Or with a single  
argument is defined to be the result of the argument. The result of Or with two  
arguments is defined using 3-valued logic semantics. This means that if either  
argument is true, the result is true; if both arguments are false, the result  
is false; otherwise, the result is null. The result of more than two arguments  
is defined as successive invocations of Or.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*NaryExpression*](#b798) [] (extension)  **Or** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Or**](#b828)">

<xs:complexContent>

<xs:extension base="[**NaryExpression**](#b798)"/>

</xs:complexContent>

</xs:complexType>

complexType "Organization"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b831) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [addresses](#b850)?, [contacts](#b851)?, [name](#b831) | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [addresses](#b850) [],  [contacts](#b851) [], | [name](#b831) (in affiliation) [] |

All Direct / Indirect Based Elements (1):

[affiliation](#b866) []

Annotation

Identifies an organization, a corporation, an institution, or a government department that has relevance to the knowledge artifact. Note that organization extends Party by adding a name attribute of type ST.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Party*](#b853) [] (extension)  **Organization** |

XML Source (w/o annotations (2))

<xs:complexType name="[**Organization**](#b833)">

<xs:complexContent>

<xs:extension base="[**Party**](#b853)">

<xs:sequence>

<xs:element name="[**name**](#b831)" type="[**dt:ST**](#b131)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b833) component only; 1/3)

 name

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

The name of the organization.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <name | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

complexType "Overlaps"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns true if the first interval overlaps the second. In other  
words, if the ending point of the first interval is greater than  
or equal to the starting point of the second interval, or if the  
starting point of the first interval is less than or equal to  
the ending point of the second interval.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Overlaps** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Overlaps**](#b836)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "ParameterDef"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b838), 1 [element](#b841) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [name](#b838) | = | xs:string | | [parameterType](#b839) | = | xs:QName | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [default](#b841)? | |
| </...> | |

Content Model Elements (1):

[default](#b841) []

All Direct / Indirect Based Elements (1):

parameter

Annotation

ParameterDef defines a parameter that can be referenced by name  
anywhere within an expression. Parameters are defined at the artifact  
level, and may be provided as part of the payload for an evaluation  
request. If no parameter value is provided, the default element is  
used to provide the value for the parameter. If no parameter or  
default is provided, the parameter is defined to be null.

XML Source (w/o annotations (1))

<xs:complexType name="[**ParameterDef**](#b843)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**default**](#b841)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**name**](#b838)" type="xs:string"/>

<xs:attribute name="[**parameterType**](#b839)" type="xs:QName"/>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b843) component only; 2/2)

 name

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 parameterType

|  |  |
| --- | --- |
| Type: | xs:QName, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b843) component only; 1/1)

 default

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <default> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </default> | |

complexType "ParameterRef"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b846) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [name](#b846) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

The ParameterRef expression allows the value of a parameter to be referenced  
as part of an expression.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **ParameterRef** |

XML Source (w/o annotations (1))

<xs:complexType name="[**ParameterRef**](#b848)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**name**](#b846)" type="xs:string"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b848) component only; 1/1)

 name

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

complexType "Party"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definitions of 2 [elements](#b850) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [addresses](#b850)?, [contacts](#b851)? | |
| </...> | |

Content Model Elements (2):

[addresses](#b850) [], [contacts](#b851) []

Known Direct Subtypes (2):

[Organization](#b833) [], [Person](#b868) []

All Direct / Indirect Based Elements (4):

[affiliation](#b866) [], [contributor](#b434) [], publisher, [rightsHolder](#b985) []

Annotation

Party represents the abstract base type for entities that have addresses and contact information. It is intended to be extended and specialized by the Person and Organization concept. Note that Party allows for polymorphism using the xsi:type construct. For instance, by defining a 'contributor' to be of type 'Party', one allows the contributor to be either a person or a company (its derived types).

XML Source (w/o annotations (5))

<xs:complexType abstract="true" name="[**Party**](#b853)">

<xs:sequence>

<xs:element minOccurs="0" name="[**addresses**](#b850)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="address" type="[**dt:AD**](#b12)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element minOccurs="0" name="[**contacts**](#b851)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="contact" type="[**dt:TEL**](#b139)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b853) component only; 2/2)

 addresses

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

The set of addresses associated with this entity.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <addresses> | |
|  | |  |  | | --- | --- | | *Content:* | address+ | |
| </addresses> | |

 contacts

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

The set of contact information associated with this entity.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <contacts> | |
|  | |  |  | | --- | --- | | *Content:* | contact+ | |
| </contacts> | |

complexType "PeriodLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b856), 2 [elements](#b859) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b233) | = | xs:string | | [alignment](#b856) | = | (("D" | "H" | "J" | "M" | "N" | "S" | "W" | "Y") | (xs:string | ("CD" | "CH" | "CM" | "CN" | "CS" | "CW" | "CY" | "DM" | "DW" | "DY" | "HD" | "MY" | "NH" | "SN" | "WY"))) | | [institutionSpecified](#b857) | = | xs:boolean : "false" | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [phase](#b859)?, [period](#b860)? | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [period](#b860) (defined in [PeriodLiteral](#b862) complexType) [], | [phase](#b859) (defined in [PeriodLiteral](#b862) complexType) [] |

Annotation

Returns a value of type PIVL\_TS with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **PeriodLiteral** |

XML Source (w/o annotations (5))

<xs:complexType name="[**PeriodLiteral**](#b862)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**phase**](#b859)" type="[**dt:IVL\_TS**](#b112)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**period**](#b860)" type="[**dt:PQ**](#b115)"/>

</xs:sequence>

<xs:attributeGroup ref="[**dt:ATTR\_TS**](#b235)"/>

<xs:attribute name="[**alignment**](#b856)" type="[**CalendarCycle**](#b1158)" use="optional"/>

<xs:attribute default="false" name="[**institutionSpecified**](#b857)" type="xs:boolean" use="optional"/>

</xs:extension>

<!-- changed from bl to xs:boolean since bl was removed as a type  
Aziz Boxwala  
-->

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b862) component only; 2/3)

 alignment

|  |  |
| --- | --- |
| Type: | [CalendarCycle](#b1158) [] |
| Use: | optional |

Specifies if and how the repetitions are aligned to  
the cycles of the underlying calendar (e.g., to  
distinguish every 30 days from "the 5th of every  
month".) A non-aligned periodic interval recurs  
independently from the calendar. An aligned periodic  
interval is synchronized with the calendar.

Attribute Value

|  |
| --- |
| ("D" | "H" | "J" | "M" | "N" | "S" | "W" | "Y") | (xs:string | ("CD" | "CH" | "CM" | "CN" | "CS" | "CW" | "CY" | "DM" | "DW" | "DY" | "HD" | "MY" | "NH" | "SN" | "WY")) |

 institutionSpecified

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

Indicates whether the exact timing is up to the party  
executing the schedule (e.g., to distinguish "every 8  
hours" from "3 times a day".)

Attribute Value

|  |  |
| --- | --- |
| Default: | "false" |

Content Element Detail (all declarations; defined within [this](#b862) component only; 2/3)

 phase

|  |  |
| --- | --- |
| Type: | [dt:IVL\_TS](#b112) [], empty content |

A prototype of the repeating interval specifying the  
duration of each occurrence and anchors the periodic  
interval sequence at a certain point in time.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <phase | |
|  | |  |  |  | | --- | --- | --- | | [high](#b238) | = | xs:string | | [highIsInclusive](#b110) | = | xs:boolean | | [low](#b243) | = | xs:string | | [lowIsInclusive](#b109) | = | xs:boolean | |
| /> | |

 period

|  |  |
| --- | --- |
| Type: | [dt:PQ](#b115) [], empty content |

A time duration specifying a reciprocal measure of  
the frequency at which the periodic interval repeats.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <period | |
|  | |  |  |  | | --- | --- | --- | | [unit](#b203) | = | xs:string | | [value](#b202) | = | xs:double | |
| /> | |

complexType "Person"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b865) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [addresses](#b850)?, [contacts](#b851)?, [name](#b865), [affiliation](#b866)? | |
| </...> | |

Content Model Elements (4):

|  |  |
| --- | --- |
| [addresses](#b850) [],  [affiliation](#b866) [], | [contacts](#b851) [],  [name](#b865) (defined in [Person](#b868) complexType) [] |

Annotation

Identifies a person who is associated with the knowledge artifact. A person may be a contributor, a rights holder, a publisher, and so on. Person extends party by adding a person name attribute and an affiliation. Note, Person.name should be constrained to be of type EN.PN

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Party*](#b853) [] (extension)  **Person** |

XML Source (w/o annotations (3))

<xs:complexType name="[**Person**](#b868)">

<xs:complexContent>

<xs:extension base="[**Party**](#b853)">

<xs:sequence>

<xs:element name="[**name**](#b865)" type="[**dt:EN**](#b52)"/>

<xs:element minOccurs="0" name="[**affiliation**](#b866)" type="[**Organization**](#b833)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b868) component only; 2/4)

 name

|  |  |
| --- | --- |
| Type: | [dt:EN](#b52) [], complex content |

The name of the person.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <name | |
|  | |  |  |  | | --- | --- | --- | | [use](#b48) | = | *list of* ("ABC" | "SYL" | "IDE" | "C" | "OR" | "T" | "I" | "P" | "ANON" | "A" | "R" | "OLD" | "DN" | "M" | "SRCH" | "PHON") | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [dt:part](#b50)+ | |
| </name> | |

 affiliation

|  |  |
| --- | --- |
| Type: | [Organization](#b833) [], complex content |

The organizational affiliation for this person.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <affiliation> | |
|  | |  |  | | --- | --- | | *Content:* | [addresses](#b850)?, [contacts](#b851)?, [name](#b831) | |
| </affiliation> | |

complexType "PhysicalQuantityIntervalLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b871) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [lowValue](#b212) | = | xs:double | | [lowUnit](#b213) | = | xs:string | | [highValue](#b207) | = | xs:double | | [highUnit](#b208) | = | xs:string | | [lowIsInclusive](#b871) | = | xs:boolean | | [highIsInclusive](#b872) | = | xs:boolean | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type IVL\_PQ with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **PhysicalQuantityIntervalLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**PhysicalQuantityIntervalLiteral**](#b874)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attributeGroup ref="[**dt:ATTR\_PQ\_LOW**](#b215)"/>

<xs:attributeGroup ref="[**dt:ATTR\_PQ\_HIGH**](#b210)"/>

<xs:attribute name="[**lowIsInclusive**](#b871)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b872)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b874) component only; 2/6)

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

complexType "PhysicalQuantityLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b202) | = | xs:double | | [unit](#b203) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type PQ with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **PhysicalQuantityLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**PhysicalQuantityLiteral**](#b877)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attributeGroup ref="[**dt:ATTR\_PQ**](#b205)"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

complexType "PIVL\_TS"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b880), 2 [elements](#b883) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b143) | = | xs:string | | [operator](#b1056) | = | ("I" | "U" | "D") : "I" | | [alignment](#b880) | = | (("D" | "H" | "J" | "M" | "N" | "S" | "W" | "Y") | (xs:string | ("CD" | "CH" | "CM" | "CN" | "CS" | "CW" | "CY" | "DM" | "DW" | "DY" | "HD" | "MY" | "NH" | "SN" | "WY"))) | | [institutionSpecified](#b881) | = | xs:boolean : "false" | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [phase](#b883)?, [period](#b884)? | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [period](#b884) (defined in [PIVL\_TS](#b886) complexType) [], | [phase](#b883) (defined in [PIVL\_TS](#b886) complexType) [] |

Annotation

Note: because this type is defined as an extension of SXCM\_T,  
all of the attributes and elements accepted for T are also  
accepted by this definition. However, they are NOT allowed  
by the normative description of this type. Unfortunately,  
we cannot write a general purpose schematron contraints to  
provide that extra validation, thus applications must be  
aware that instance (fragments) that pass validation with  
this might might still not be legal.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [dt:ANY](#b19) [] (extension)  [*dt:QTY*](#b118) [] (extension)  [dt:TS](#b145) [] (extension)  [SXCM\_TS](#b1058) [] (extension)  **PIVL\_TS** |

XML Source (w/o annotations (5))

<xs:complexType name="[**PIVL\_TS**](#b886)">

<xs:complexContent>

<xs:extension base="[**SXCM\_TS**](#b1058)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**phase**](#b883)" type="[**dt:IVL\_TS**](#b112)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**period**](#b884)" type="[**dt:PQ**](#b115)"/>

</xs:sequence>

<xs:attribute name="[**alignment**](#b880)" type="[**CalendarCycle**](#b1158)" use="optional"/>

<xs:attribute default="false" name="[**institutionSpecified**](#b881)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b886) component only; 2/4)

 alignment

|  |  |
| --- | --- |
| Type: | [CalendarCycle](#b1158) [] |
| Use: | optional |

Specifies if and how the repetitions are aligned to  
the cycles of the underlying calendar (e.g., to  
distinguish every 30 days from "the 5th of every  
month".) A non-aligned periodic interval recurs  
independently from the calendar. An aligned periodic  
interval is synchronized with the calendar.

Attribute Value

|  |
| --- |
| ("D" | "H" | "J" | "M" | "N" | "S" | "W" | "Y") | (xs:string | ("CD" | "CH" | "CM" | "CN" | "CS" | "CW" | "CY" | "DM" | "DW" | "DY" | "HD" | "MY" | "NH" | "SN" | "WY")) |

 institutionSpecified

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

Indicates whether the exact timing is up to the party  
executing the schedule (e.g., to distinguish "every 8  
hours" from "3 times a day".)

Attribute Value

|  |  |
| --- | --- |
| Default: | "false" |

Content Element Detail (all declarations; defined within [this](#b886) component only; 2/2)

 phase

|  |  |
| --- | --- |
| Type: | [dt:IVL\_TS](#b112) [], empty content |

A prototype of the repeating interval specifying the  
duration of each occurrence and anchors the periodic  
interval sequence at a certain point in time.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <phase | |
|  | |  |  |  | | --- | --- | --- | | [high](#b238) | = | xs:string | | [highIsInclusive](#b110) | = | xs:boolean | | [low](#b243) | = | xs:string | | [lowIsInclusive](#b109) | = | xs:boolean | |
| /> | |

 period

|  |  |
| --- | --- |
| Type: | [dt:PQ](#b115) [], empty content |

A time duration specifying a reciprocal measure of  
the frequency at which the periodic interval repeats.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <period | |
|  | |  |  |  | | --- | --- | --- | | [unit](#b203) | = | xs:string | | [value](#b202) | = | xs:double | |
| /> | |

complexType "Pos"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b889) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [pattern](#b889), [string](#b890) | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [pattern](#b889) [], | [string](#b890) [] |

Annotation

Pos returns the 1-based index of the given pattern in the given string.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Pos** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Pos**](#b892)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**pattern**](#b889)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**string**](#b890)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b892) component only; 2/3)

 pattern

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <pattern> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </pattern> | |

 string

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <string> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </string> | |

complexType "Power"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

The Power operator raises the first argument to the power  
given by the second argument.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Power** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Power**](#b895)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "PrecheckBehavior"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b898) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [value](#b898) | |
| </...> | |

Content Model Elements (1):

[value](#b898) (defined in [PrecheckBehavior](#b900) complexType) []

Annotation

For a particular action, specifies how often the action is expected to be selected in the particular context of the group containing that action. In general, depending on the group selection behavior, there may be zero, one or more actions which are frequently selected. This setting can serve as a hint to the system that displays the action to the end user: some systems will pre-select those actions which are (or should be) most frequently selected.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Behavior*](#b329) [] (extension)  **PrecheckBehavior** |

XML Source (w/o annotations (1))

<xs:complexType name="[**PrecheckBehavior**](#b900)">

<xs:complexContent>

<xs:extension base="[**Behavior**](#b329)">

<xs:sequence>

<xs:element name="[**value**](#b898)" type="[**PrecheckBehaviorType**](#b1242)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b900) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | [PrecheckBehaviorType](#b1242) [], simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <value> | | |
|  | *Content:* | { ("Yes" | "No") | ("Yes" | "No") } |
| </value> | | |

complexType "Pred"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

The Pred operator returns the predecessor of the argument.  
For example, the predecessor of 2 is 1.  
If the argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Pred** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Pred**](#b903)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "ProperContains"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns true if the first operand contains the second, and is strictly larger.  
There are two overloads of this operator:  
List, List : The element type of both lists must be the same.  
Interval, Interval : The point type of both intervals must be the same.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **ProperContains** |

XML Source (w/o annotations (1))

<xs:complexType name="[**ProperContains**](#b906)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "ProperIn"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

Returns true if the first operand is in the second, and is strictly smaller.  
There are two overloads of this operator:  
List, List : The element type of both lists must be the same.  
Interval, Interval : The point type of both intervals must be the same.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **ProperIn** |

XML Source (w/o annotations (1))

<xs:complexType name="[**ProperIn**](#b909)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "Property"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b912), 1 [element](#b915) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [path](#b912) | = | xs:string | | [scope](#b913) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b915)? | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b915) (defined in [Property](#b917) complexType) [] |

Annotation

Returns the value of the property on source specified by the  
path attribute.  
  
If the path attribute contains qualifiers, each qualifier is  
traversed to obtain the actual value.  
  
If a scope is specified, the name is used to resolve the scope  
in which the path will be resolved. Scopes can be named by operators  
such as Filter and ForEach.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Property** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Property**](#b917)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**source**](#b915)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**path**](#b912)" type="xs:string" use="required"/>

<xs:attribute name="[**scope**](#b913)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b917) component only; 2/2)

 path

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

 scope

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Content Element Detail (all declarations; defined within [this](#b917) component only; 1/2)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

complexType "PropertyExpression"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b919), 1 [element](#b921) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [name](#b919) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [value](#b921) | |
| </...> | |

Content Model Elements (1):

[value](#b921) (defined in [PropertyExpression](#b923) complexType) []

All Direct / Indirect Based Elements (2):

[property](#b815) (defined in [ObjectExpression](#b817) complexType) [],

[property](#b823) (in [actionSentence](#b1098) defined in [UpdateAction](#b1100) complexType) []

Annotation

The PropertyExpression type is used within the ObjectLiteral type to  
provide the value of a specific property within an object literal  
expression.

XML Source (w/o annotations (1))

<xs:complexType name="[**PropertyExpression**](#b923)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**value**](#b921)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**name**](#b919)" type="xs:string" use="required"/>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b923) component only; 1/1)

 name

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

Content Element Detail (all declarations; defined within [this](#b923) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <value> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </value> | |

complexType "RangeConstraint"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definition of 1 [element](#b925) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [constraintType](#b925) | |
| </...> | |

Content Model Elements (1):

[constraintType](#b925) []

Known Direct Subtypes (2):

[ExpressionConstraint](#b546) [], [ListConstraint](#b731) []

All Direct / Indirect Based Elements (1):

[responseRange](#b516) []

XML Source (w/o annotations (1))

<xs:complexType abstract="true" name="[**RangeConstraint**](#b927)">

<xs:sequence>

<xs:element name="[**constraintType**](#b925)" type="[**RangeConstraintType**](#b1251)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b927) component only; 1/1)

 constraintType

|  |  |
| --- | --- |
| Type: | [RangeConstraintType](#b1251) [], simple content |

The constraint type defines how the value range  
is being constrained. For example, the constraint type may  
indicate the lower bound of the range.

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <constraintType> | | |
|  | *Content:* | { ("Minimum" | "Maximum" | "List" | "Component") | ("Minimum" | "Maximum" | "List" | "Component") } |
| </constraintType> | | |

complexType "RatioIntervalLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b930) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [lowNumerator](#b227) | = | xs:double | | [lowDenominator](#b228) | = | xs:double | | [highNumerator](#b222) | = | xs:double | | [highDenominator](#b223) | = | xs:double | | [lowIsInclusive](#b930) | = | xs:boolean | | [highIsInclusive](#b931) | = | xs:boolean | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type IVL\_RTO with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **RatioIntervalLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**RatioIntervalLiteral**](#b933)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attributeGroup ref="[**dt:ATTR\_RTO\_LOW**](#b230)"/>

<xs:attributeGroup ref="[**dt:ATTR\_RTO\_HIGH**](#b225)"/>

<xs:attribute name="[**lowIsInclusive**](#b930)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b931)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b933) component only; 2/6)

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

complexType "RatioLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [numerator](#b217) | = | xs:double | | [denominator](#b218) | = | xs:double | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type RTO with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **RatioLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**RatioLiteral**](#b936)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attributeGroup ref="[**dt:ATTR\_RTO**](#b220)"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

complexType "RealIntervalLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 4 [attributes](#b939) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [low](#b939) | = | xs:double | | [high](#b940) | = | xs:double | | [lowIsInclusive](#b941) | = | xs:boolean | | [highIsInclusive](#b942) | = | xs:boolean | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type IVL\_REAL with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **RealIntervalLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**RealIntervalLiteral**](#b944)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**low**](#b939)" type="[**dt:Decimal**](#b158)"/>

<xs:attribute name="[**high**](#b940)" type="[**dt:Decimal**](#b158)"/>

<xs:attribute name="[**lowIsInclusive**](#b941)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b942)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b944) component only; 4/4)

 low

|  |  |
| --- | --- |
| Type: | [dt:Decimal](#b158) [] |
| Use: | optional |

Attribute Value

|  |
| --- |
| xs:double |

 high

|  |  |
| --- | --- |
| Type: | [dt:Decimal](#b158) [] |
| Use: | optional |

Attribute Value

|  |
| --- |
| xs:double |

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

complexType "RealLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b947) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b947) | = | xs:double | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type Real with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **RealLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**RealLiteral**](#b949)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**value**](#b947)" type="[**dt:Decimal**](#b158)" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b949) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | [dt:Decimal](#b158) [] |
| Use: | required |

Attribute Value

|  |
| --- |
| xs:double |

complexType "RemoveAction"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b952) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)?, [textEquivalent](#b316)?, [actionSentence](#b952) | |
| </...> | |

Content Model Elements (8):

|  |  |
| --- | --- |
| [actionId](#b260) [],  [actionSentence](#b952) (defined in [RemoveAction](#b954) complexType) [],  [actors](#b263) [],  [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [], | [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [],  [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [],  [supportingResources](#b262) [],  [textEquivalent](#b316) [] |

Annotation

This action removes another proposed action or an  
ongoing action.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*ActionBase*](#b267) [] (extension)  [*AtomicAction*](#b318) [] (extension)  **RemoveAction** |

XML Source (w/o annotations (2))

<xs:complexType name="[**RemoveAction**](#b954)">

<xs:complexContent>

<xs:extension base="[**AtomicAction**](#b318)">

<xs:sequence>

<xs:element name="[**actionSentence**](#b952)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b954) component only; 1/8)

 actionSentence

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

The expression must resolve to the action that  
is being removed.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <actionSentence> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </actionSentence> | |

complexType "RequestBase"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definitions of 7 [attributes](#b957), 1 [element](#b965) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [scope](#b957) | = | xs:string | | [cardinality](#b958) | = | ("Single" | "Multiple") | | [dataType](#b959) | = | xs:QName | | [templateId](#b960) | = | xs:string | | [idProperty](#b961) | = | xs:string | | [triggerType](#b962) | = | (("DataElementAdded" | "DataElementModified" | "DataElementRemoved" | "DataElementAccessed" | "DataElementAccessEnded") | ("DataElementAdded" | "DataElementModified" | "DataElementRemoved" | "DataElementAccessed" | "DataElementAccessEnded")) | | [isInitial](#b963) | = | xs:boolean : "true" | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [timeOffset](#b965)? | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [timeOffset](#b965) [] |

Known Direct Subtypes (1):

[ClinicalRequest](#b369) []

Annotation

The request expression defines the data that will be used by the artifact.  
  
The result of a request is defined to return the same data for subsequent invocations  
within the same evaluation request. This means in particular that patient data updates made  
during the evaluation request are not visible to the artifact. In effect, the patient data  
is a snapshot of the data as of the start of the evaluation. This ensures strict deterministic  
and functional behavior of the artifact, and allows the implementation engine freedom to  
cache intermediate results in order to improve performance.  
  
The dataType attribute specifies the type of clinical data being requested.  
  
The templateId attribute specifies an optional template to be used. If specified, the request  
is defined to return only objects that conform to the template.  
  
The idProperty attribute specifies which property of the model contains the Id for the clinical  
statement.  
  
The triggerType attribute optionally specifies whether this request should only include  
data that was accessed or updated as a result of some event that is triggering the evaluation  
of the artifact. This attribute is only valid on a request that is specified as the request  
for a data changed event trigger.  
  
The isInitial attribute determines whether the request is part of the initial data requirements  
for the artifact.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  ***RequestBase*** |

XML Source (w/o annotations (1))

<xs:complexType abstract="true" name="[**RequestBase**](#b967)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**timeOffset**](#b965)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**scope**](#b957)" type="xs:string" use="optional"/>

<xs:attribute name="[**cardinality**](#b958)" type="[**RequestCardinality**](#b1260)" use="required"/>

<xs:attribute name="[**dataType**](#b959)" type="xs:QName" use="required"/>

<xs:attribute name="[**templateId**](#b960)" type="xs:string" use="optional"/>

<xs:attribute name="[**idProperty**](#b961)" type="xs:string" use="optional"/>

<xs:attribute name="[**triggerType**](#b962)" type="[**DataEventType**](#b1200)" use="optional"/>

<xs:attribute default="true" name="[**isInitial**](#b963)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b967) component only; 7/7)

 scope

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 cardinality

|  |  |
| --- | --- |
| Type: | [RequestCardinality](#b1260) [] |
| Use: | required |

Attribute Value

|  |
| --- |
| *enumeration of* xs:string |

|  |  |
| --- | --- |
| Enumeration: | "Single", "Multiple" |

 dataType

|  |  |
| --- | --- |
| Type: | xs:QName, predefined |
| Use: | required |

 templateId

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 idProperty

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 triggerType

|  |  |
| --- | --- |
| Type: | [DataEventType](#b1200) [] |
| Use: | optional |

Attribute Value

|  |
| --- |
| ("DataElementAdded" | "DataElementModified" | "DataElementRemoved" | "DataElementAccessed" | "DataElementAccessEnded") | ("DataElementAdded" | "DataElementModified" | "DataElementRemoved" | "DataElementAccessed" | "DataElementAccessEnded") |

 isInitial

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

Attribute Value

|  |  |
| --- | --- |
| Default: | "true" |

Content Element Detail (all declarations; defined within [this](#b967) component only; 1/2)

 timeOffset

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <timeOffset> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </timeOffset> | |

complexType "RequiredBehavior"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b970) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [value](#b970) | |
| </...> | |

Content Model Elements (1):

[value](#b970) (defined in [RequiredBehavior](#b972) complexType) []

Annotation

For a single action, specifies what level of requiredness is associated with the action.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Behavior*](#b329) [] (extension)  **RequiredBehavior** |

XML Source (w/o annotations (1))

<xs:complexType name="[**RequiredBehavior**](#b972)">

<xs:complexContent>

<xs:extension base="[**Behavior**](#b329)">

<xs:sequence>

<xs:element name="[**value**](#b970)" type="[**RequiredBehaviorType**](#b1263)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b972) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | [RequiredBehaviorType](#b1263) [], simple content |

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <value> | | |
|  | *Content:* | { ("Must" | "Could" | "MustUnlessDocumented") | ("Must" | "Could" | "MustUnlessDocumented") } |
| </value> | | |

complexType "ResourceRelationshipReference"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b974) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [relationship](#b974), [resources](#b975) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [relationship](#b974) [], | [resources](#b975) (in relatedResource) [] |

All Direct / Indirect Based Elements (1):

relatedResource

Annotation

ResourceRelationshipReference defines the  
association between a resource and a resource set. If one consider  
such relationship as a triple (subject-predicate-object or  
node-arc-node) with the source entity as the subject of the triple,  
then ResourceRelationshipType defines the predicate and the object  
of this relationship.

XML Source (w/o annotations (4))

<xs:complexType name="[**ResourceRelationshipReference**](#b977)">

<xs:sequence>

<xs:element name="[**relationship**](#b974)">

<xs:complexType>

<xs:attribute name="value" type="[**ResourceRelationshipType**](#b1272)" use="required"/>

</xs:complexType>

</xs:element>

<xs:element minOccurs="1" name="[**resources**](#b975)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="resource" type="[**KnowledgeResource**](#b701)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b977) component only; 2/2)

 relationship

|  |  |
| --- | --- |
| Type: | anonymous complexType, empty content |

Relationship defines the association between two  
resources - i.e., the predicate/arc of a triple. The object of the  
relationship is often a pre-existing resource that predates the  
subject of the relationship. This predicate can be one of six  
types: (1) AdaptedFrom - the subject was adapted from the object  
of the relationship. For instance, a diabetes visit order set may  
be adapted from the HbA1c reminder (2) AssociatedResource - the  
object is associated with the subject. For instance, an HbA1c  
reminder may be associated with an HbA1c eMeasure (3) DependsOn -  
the subject depends on the object. For instance, an HbA1c reminder  
may depend on a Diabetes Value Set (4) DerivedFrom - the subject  
was derived from the object. For instance, an HbA1c reminder rule  
may be derived from a Diabetes Guideline (5) SimilarTo - the  
subject and object are similar. For instance, an HbA1c reminder  
may be similar to an LDL reminder and (6) VersionOf - the subject  
is a version of the object (and vice-versa). For instance, the  
HbA1c reminder v2 may be a version of the HbA1c reminder v1.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <relationship | |
|  | |  |  |  | | --- | --- | --- | | value | = | (("AdaptedFrom" | "AssociatedResource" | "DependsOn" | "DerivedFrom" | "SimilarTo" | "VersionOf") | ("AdaptedFrom" | "AssociatedResource" | "DependsOn" | "DerivedFrom" | "SimilarTo" | "VersionOf")) | |
| /> | |

 resources

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

The set of resources associated with the subject  
of this relationship.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <resources> | |
|  | |  |  | | --- | --- | | *Content:* | resource+ | |
| </resources> | |

complexType "ResponseBinding"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b979) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [container](#b979) | = | xs:string : "Responses" | | [property](#b980) | = | xs:string | |
| /> | |

All Direct / Indirect Based Elements (1):

[responseBinding](#b394) []

Annotation

Defines the attributes required to specify a binding path for documentation item responses.  
  
The container attribute specifies the name of the response container that will be used. If  
no container attribute is provided, the default container name of Responses will be used.  
  
The property attribute specifies the name of the property within the container that will be  
used to store the user response value.

XML Source (w/o annotations (1))

<xs:complexType name="[**ResponseBinding**](#b982)">

<xs:attribute default="Responses" name="[**container**](#b979)" type="xs:string" use="optional"/>

<xs:attribute name="[**property**](#b980)" type="xs:string" use="required"/>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b982) component only; 2/2)

 container

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

Attribute Value

|  |  |
| --- | --- |
| Default: | "Responses" |

 property

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

complexType "RightsDeclaration"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [elements](#b984) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [assertedRights](#b984), [rightsHolder](#b985), [permissions](#b986)? | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [assertedRights](#b984) [],  [permissions](#b986) (in rightsDeclaration) [], | [rightsHolder](#b985) [] |

All Direct / Indirect Based Elements (1):

rightsDeclaration

Annotation

This specifies the intellectual property rights  
associated with this CDS knowledge artifact, including who the  
rights holder is and what rights they assert. It also specifies what  
permissions are granted for usage. The asserted rights and  
persmissions are specified as a free-form text string.

XML Source (w/o annotations (1))

<xs:complexType name="[**RightsDeclaration**](#b988)">

<xs:sequence>

<xs:element name="[**assertedRights**](#b984)" type="[**dt:ST**](#b131)"/>

<xs:element name="[**rightsHolder**](#b985)" type="[**Party**](#b853)"/>

<xs:element minOccurs="0" name="[**permissions**](#b986)">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="permissions" type="[**dt:ST**](#b131)"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b988) component only; 3/3)

 assertedRights

|  |  |
| --- | --- |
| Type: | [dt:ST](#b131) [], empty content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <assertedRights | |
|  | |  |  |  | | --- | --- | --- | | [value](#b129) | = | xs:string | |
| /> | |

 rightsHolder

|  |  |
| --- | --- |
| Type: | [*Party*](#b853) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <rightsHolder> | |
|  | |  |  | | --- | --- | | *Content:* | [addresses](#b850)?, [contacts](#b851)? | |
| </rightsHolder> | |

 permissions

|  |  |
| --- | --- |
| Type: | anonymous complexType, complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <permissions> | |
|  | |  |  | | --- | --- | | *Content:* | permissions+ | |
| </permissions> | |

complexType "Round"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b991) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b991), [precision](#b992)? | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [operand](#b991) (defined in [Round](#b994) complexType) [], | [precision](#b992) [] |

Annotation

The Round operator returns the nearest integer to its  
argument. The semantics of round are defined as a traditional  
round, meaning that a decimal value of 0.5 or higher will round  
to 1. If the argument is null, the result is null.  
If specified, precision determines the decimal place at which  
the rounding will occur. If precision is not specified or null, 0 is assumed.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Round** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Round**](#b994)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**operand**](#b991)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**precision**](#b992)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b994) component only; 2/3)

 operand

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <operand> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </operand> | |

 precision

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <precision> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </precision> | |

complexType "SetSubsumes"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b997) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [ancestors](#b997)?, [descendents](#b998)? | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [ancestors](#b997) [],  [descendents](#b998) [], | [description](#b539) (defined in [*Expression*](#b541) complexType) [] |

Annotation

This operator returns the list of descendants that were subsumed by some code  
in the list of ancestors.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **SetSubsumes** |

XML Source (w/o annotations (1))

<xs:complexType name="[**SetSubsumes**](#b1000)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**ancestors**](#b997)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**descendents**](#b998)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1000) component only; 2/3)

 ancestors

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <ancestors> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </ancestors> | |

 descendents

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <descendents> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </descendents> | |

complexType "SimpleCodeLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b1003) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [code](#b1003) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type CS with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **SimpleCodeLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**SimpleCodeLiteral**](#b1005)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**code**](#b1003)" type="xs:string" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b1005) component only; 1/1)

 code

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

complexType "Sort"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 1 [attribute](#b1008), 1 [element](#b1010) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [orderBy](#b1008) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b1010) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b1010) (defined in [Sort](#b1012) complexType) [] |

Annotation

Returns a list with all the elements in source, sorted by the given orderBy.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Sort** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Sort**](#b1012)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element name="[**source**](#b1010)" type="[**Expression**](#b541)"/>

</xs:sequence>

<xs:attribute name="[**orderBy**](#b1008)" type="xs:string" use="required"/>

</xs:extension>

<!-- TODO: Model ordering more explicitly? Multi-column sort? Ascending/Descending? -->

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b1012) component only; 1/1)

 orderBy

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

Content Element Detail (all declarations; defined within [this](#b1012) component only; 1/2)

 source

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <source> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </source> | |

complexType "Split"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b1015) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [stringToSplit](#b1015), [separator](#b1016)? | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [separator](#b1016) (defined in [Split](#b1018) complexType) [], | [stringToSplit](#b1015) [] |

Annotation

The Split operator splits a string into a list of strings using a  
separator. If the stringToSplit argument is null, the result is null.  
If the stringToSplit argument does not contain any appearances of the  
separator, the result is a list of strings containing one element that  
is value of the stringToSplit argument.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Split** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Split**](#b1018)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**stringToSplit**](#b1015)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**separator**](#b1016)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1018) component only; 2/3)

 stringToSplit

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <stringToSplit> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </stringToSplit> | |

 separator

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <separator> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </separator> | |

complexType "StringLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b1021) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b1021) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type ST with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **StringLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**StringLiteral**](#b1023)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**value**](#b1021)" type="xs:string" use="required"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b1023) component only; 1/1)

 value

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

complexType "Substring"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [elements](#b1026) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [stringToSub](#b1026), [startIndex](#b1027), [length](#b1028)? | |
| </...> | |

Content Model Elements (4):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [length](#b1028) [], | [startIndex](#b1027) [],  [stringToSub](#b1026) [] |

Annotation

Substring returns the string within stringToSub, beginning at the 1-based index  
startIndex, and consisting of length characters. If length is ommitted, the  
substring returned starts at startIndex and continues to the end of stringToSub.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Substring** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Substring**](#b1030)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**stringToSub**](#b1026)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**startIndex**](#b1027)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**length**](#b1028)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1030) component only; 3/4)

 stringToSub

|  |  |
| --- | --- |
| Type: | xs:anyType, any content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <stringToSub> | |
|  | ... |
| </stringToSub> | |

 startIndex

|  |  |
| --- | --- |
| Type: | xs:anyType, any content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <startIndex> | |
|  | ... |
| </startIndex> | |

 length

|  |  |
| --- | --- |
| Type: | xs:anyType, any content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <length> | |
|  | ... |
| </length> | |

complexType "Subsumes"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b1033) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [ancestor](#b1033)?, [descendent](#b1034)? | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [ancestor](#b1033) [],  [descendent](#b1034) [], | [description](#b539) (defined in [*Expression*](#b541) complexType) [] |

Annotation

This operator returns true if the operands were of the same code system,  
and the ancestor operand subsumed the descendant operand in the hierarchy of  
the code system. If the codes are the same code, the operator returns true.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Subsumes** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Subsumes**](#b1036)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="0" name="[**ancestor**](#b1033)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="0" name="[**descendent**](#b1034)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1036) component only; 2/3)

 ancestor

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <ancestor> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </ancestor> | |

 descendent

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <descendent> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </descendent> | |

complexType "Subtract"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

The Subtract operator performs numeric subtraction of its arguments.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **Subtract** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Subtract**](#b1039)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "Succ"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

The Succ operator returns the successor of the argument.  
For example, the successor of 1 is 2.  
If the argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Succ** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Succ**](#b1042)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "Sum"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [path](#b295) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b297) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [source](#b297) (defined in [*AggregateExpression*](#b299) complexType) [] |

Annotation

Returns the sum of non-null elements in the source.  
  
If a path is specified, elements with no value for the  
property specified by the path are ignored.  
  
If source contains no non-null elements, null is returned.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*AggregateExpression*](#b299) [] (extension)  **Sum** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Sum**](#b1045)">

<xs:complexContent>

<xs:extension base="[**AggregateExpression**](#b299)"/>

</xs:complexContent>

</xs:complexType>

complexType "SupportingEvidence"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b1047) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [evidence](#b1047)+ | |
| </...> | |

Content Model Elements (1):

[evidence](#b1047) []

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [], | [supportingEvidence](#b768) (in [metadata](#b684)) [] |

Annotation

The evidence grade and the sources of evidence  
associated with this artifact.

XML Source (w/o annotations (2))

<xs:complexType name="[**SupportingEvidence**](#b1049)">

<xs:sequence>

<xs:element maxOccurs="unbounded" name="[**evidence**](#b1047)" type="[**Evidence**](#b537)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1049) component only; 1/1)

 evidence

|  |  |
| --- | --- |
| Type: | [Evidence](#b537) [], complex content |

A single evidence reference.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <evidence> | |
|  | |  |  | | --- | --- | | *Content:* | [qualityOfEvidence](#b533)?, [strengthOfRecommendation](#b534)?, [resources](#b535)? | |
| </evidence> | |

complexType "SupportingResource"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b1051) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [resource](#b1051)+ | |
| </...> | |

Content Model Elements (1):

[resource](#b1051) (defined in [SupportingResource](#b1053) complexType) []

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [resources](#b535) (in [evidence](#b1047)) [], | [supportingResources](#b262) [] |

Annotation

Didactic or other informational resources  
associated with the artifact that can be provided to the CDS  
recipient. Information resources can include inline text commentary  
and links to web resources.Note, supporting resources excludes  
supporting evidence. For supporting evidence, use  
SupportingEvidence.

XML Source (w/o annotations (2))

<xs:complexType name="[**SupportingResource**](#b1053)">

<xs:sequence>

<xs:element maxOccurs="unbounded" name="[**resource**](#b1051)" type="[**KnowledgeResource**](#b701)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1053) component only; 1/1)

 resource

|  |  |
| --- | --- |
| Type: | [KnowledgeResource](#b701) [], complex content |

An individual supporting resource.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <resource> | |
|  | |  |  | | --- | --- | | *Content:* | [identifiers](#b694)?, [templateIds](#b695)?, [title](#b696)?, [location](#b697)?, [description](#b698)?, [citation](#b699)? | |
| </resource> | |

complexType "SXCM\_TS"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b1056) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b143) | = | xs:string | | [operator](#b1056) | = | ("I" | "U" | "D") : "I" | |
| /> | |

Known Direct Subtypes (1):

[PIVL\_TS](#b886) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [dt:ANY](#b19) [] (extension)  [*dt:QTY*](#b118) [] (extension)  [dt:TS](#b145) [] (extension)  **SXCM\_TS** |

XML Source (w/o annotations (1))

<xs:complexType name="[**SXCM\_TS**](#b1058)">

<xs:complexContent>

<xs:extension base="[**dt:TS**](#b145)">

<xs:attribute default="I" name="[**operator**](#b1056)" type="[**SetOperator**](#b1281)" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b1058) component only; 1/2)

 operator

|  |  |
| --- | --- |
| Type: | [SetOperator](#b1281) [] |
| Use: | optional |

A code specifying whether the set component is included  
(union) or excluded (set-difference) from the set, or  
other set operations with the current set component and  
the set as constructed from the representation stream  
up to the current point.

Attribute Value

|  |
| --- |
| *enumeration of* xs:string |

|  |  |
| --- | --- |
| Enumeration: | "I", "U", "D" |
| Default: | "I" |

complexType "TernaryExpression"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b1061) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1061)[3..3] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1061) (defined in [TernaryExpression](#b1063) complexType) [] |

Annotation

The TernaryExpression type defines the abstract base type for all expressions  
that take three arguments.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **TernaryExpression** |

XML Source (w/o annotations (1))

<xs:complexType name="[**TernaryExpression**](#b1063)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="3" minOccurs="3" name="[**operand**](#b1061)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1063) component only; 1/2)

 operand

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <operand> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </operand> | |

complexType "TimestampIntervalLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [attributes](#b1066) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [low](#b243) | = | xs:string | | [high](#b238) | = | xs:string | | [lowIsInclusive](#b1066) | = | xs:boolean | | [highIsInclusive](#b1067) | = | xs:boolean | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type IVL\_TS with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **TimestampIntervalLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**TimestampIntervalLiteral**](#b1069)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attributeGroup ref="[**dt:ATTR\_TS\_LOW**](#b245)"/>

<xs:attributeGroup ref="[**dt:ATTR\_TS\_HIGH**](#b240)"/>

<xs:attribute name="[**lowIsInclusive**](#b1066)" type="xs:boolean" use="optional"/>

<xs:attribute name="[**highIsInclusive**](#b1067)" type="xs:boolean" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b1069) component only; 2/4)

 lowIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

 highIsInclusive

|  |  |
| --- | --- |
| Type: | xs:boolean, predefined |
| Use: | optional |

complexType "TimestampLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b233) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type TS with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **TimestampLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**TimestampLiteral**](#b1072)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attributeGroup ref="[**dt:ATTR\_TS**](#b235)"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

complexType "Today"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns the date (with no time component) of the start timestamp associated with  
the evaluation request. See the Now operator for more information on the rationale  
for defining the Today operator in this way.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Today** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Today**](#b1075)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)"/>

</xs:complexContent>

</xs:complexType>

complexType "Trigger"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b1077) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [eventType](#b1077), [expression](#b1078)? | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [eventType](#b1077) (in trigger) [], | [expression](#b1078) (in trigger) [] |

All Direct / Indirect Based Elements (1):

[trigger](#b1082) []

XML Source (w/o annotations (2))

<xs:complexType name="[**Trigger**](#b1080)">

<xs:sequence>

<xs:element name="[**eventType**](#b1077)" type="[**EventType**](#b1212)"/>

<xs:element minOccurs="0" name="[**expression**](#b1078)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1080) component only; 2/2)

 eventType

|  |  |
| --- | --- |
| Type: | [EventType](#b1212) [], simple content |

The event type can be one of either a DataEvent or a PeriodicEvent. A DataEvent is an event that is triggered by the value, presence, or absence of a particular data item. For instance, a data event may be triggered by a new serum potassium result below 3.5, a new appointment event to a Primary Care Clinic, or a new susbstance administration proposal for dabigatran. A periodic event is an event that is triggered on a regular basis. For instance, every day at midnight.

|  |  |  |
| --- | --- | --- |
| XML Representation Summary | | |
| <eventType> | | |
|  | *Content:* | { ("DataEvent" | "PeriodicEvent") | ("DataEvent" | "PeriodicEvent") } |
| </eventType> | | |

 expression

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

The event expression that must be met for the trigger to activate the record.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <expression> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </expression> | |

complexType "Triggers"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b1082) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [trigger](#b1082)+ | |
| </...> | |

Content Model Elements (1):

[trigger](#b1082) []

All Direct / Indirect Based Elements (1):

[triggers](#b687) []

XML Source (w/o annotations (1))

<xs:complexType name="[**Triggers**](#b1084)">

<xs:sequence>

<xs:element maxOccurs="unbounded" name="[**trigger**](#b1082)" type="[**Trigger**](#b1080)"/>

</xs:sequence>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1084) component only; 1/1)

 trigger

|  |  |
| --- | --- |
| Type: | [Trigger](#b1080) [], complex content |

The trigger element represents an event that 'triggers' the knowledge artifact. For instance, 'evaluate this artifact whenever a new Problem with a Diabetes Diagnosis code is added to the patient's record' or 'Evaluate this artifact every week day at 10:00 PM'. A trigger can model either a data event or a periodic event.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <trigger> | |
|  | |  |  | | --- | --- | | *Content:* | [eventType](#b1077), [expression](#b1078)? | |
| </trigger> | |

complexType "TruncatedDivide"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b336)[2..2] | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b336) (defined in [BinaryExpression](#b338) complexType) [] |

Annotation

The TruncatedDivide operator performs integer division of its arguments.  
If either argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [BinaryExpression](#b338) [] (extension)  **TruncatedDivide** |

XML Source (w/o annotations (1))

<xs:complexType name="[**TruncatedDivide**](#b1087)">

<xs:complexContent>

<xs:extension base="[**BinaryExpression**](#b338)"/>

</xs:complexContent>

</xs:complexType>

complexType "UnaryExpression"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Abstract: | (cannot be assigned directly to elements used in instance XML documents) |
| Includes: | definition of 1 [element](#b1090) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Known Direct Subtypes (17):

[Abs](#b258) [], [Begin](#b327) [], [Ceiling](#b358) [], [Convert](#b442) [], [End](#b521) [], [Floor](#b581) [], [IsEmpty](#b672) [], [IsNotEmpty](#b675) [], [IsNull](#b678) [], [Length](#b711) [], [Ln](#b740) [], [Lower](#b746) [], [Negate](#b801) [], [Not](#b804) [], [Pred](#b903) [], [Succ](#b1042) [], [Upper](#b1103) []

Annotation

The UnaryExpression type defines the abstract base type for all expressions  
that take a single argument.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  ***UnaryExpression*** |

XML Source (w/o annotations (1))

<xs:complexType abstract="true" name="[**UnaryExpression**](#b1092)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**operand**](#b1090)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1092) component only; 1/2)

 operand

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <operand> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </operand> | |

complexType "Union"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b796)\* | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b796) (defined in [*NaryExpression*](#b798) complexType) [] |

Annotation

Returns the union of the operands.  
This operator has two overloads:  
List  
Interval  
  
For the list overload, this operator returns a list with all  
elements from all operands.  
  
For the interval overload, this operator returns the interval  
that starts at the earliest starting point in the operands, and  
ends at the latest starting point in the operands. Note that if  
the operands do not overlap, this operator returns null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*NaryExpression*](#b798) [] (extension)  **Union** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Union**](#b1095)">

<xs:complexContent>

<xs:extension base="[**NaryExpression**](#b798)"/>

</xs:complexContent>

</xs:complexType>

complexType "UpdateAction"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [element](#b1098) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | ([actionId](#b260)?, [supportingEvidence](#b261)?, [supportingResources](#b262)?, [actors](#b263)?, [behaviors](#b264)?, [conditions](#b265)?)?, [textEquivalent](#b316)?, [actionSentence](#b1098) | |
| </...> | |

Content Model Elements (8):

|  |  |
| --- | --- |
| [actionId](#b260) [],  [actionSentence](#b1098) (defined in [UpdateAction](#b1100) complexType) [],  [actors](#b263) [],  [behaviors](#b264) (defined in [*ActionBase*](#b267) complexType) [], | [conditions](#b265) (defined in [*ActionBase*](#b267) complexType) [],  [supportingEvidence](#b261) (defined in [*ActionBase*](#b267) complexType) [],  [supportingResources](#b262) [],  [textEquivalent](#b316) [] |

Annotation

This action changing the value of another  
existing  
action. The action being modified may be a proposed  
action (e.g., an  
medication being prescribed by a clinician) or be  
an ongoing action  
(e.g., an existing prescription). In these cases,  
a modification can  
be used to change the dose of the medication. It  
may also be used to  
discontinue a medication by changing the stop  
date.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*ActionBase*](#b267) [] (extension)  [*AtomicAction*](#b318) [] (extension)  **UpdateAction** |

XML Source (w/o annotations (2))

<xs:complexType name="[**UpdateAction**](#b1100)">

<xs:complexContent>

<xs:extension base="[**AtomicAction**](#b318)">

<xs:sequence>

<xs:element name="[**actionSentence**](#b1098)" type="[**ObjectRedefine**](#b825)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1100) component only; 1/8)

 actionSentence

|  |  |
| --- | --- |
| Type: | [ObjectRedefine](#b825) [], complex content |

The modification to the action. This is  
specified by modifying the property of an existing action using  
the ObjectRedefine expression.

|  |  |
| --- | --- |
| XML Representation Summary | |
| <actionSentence | |
|  | |  |  |  | | --- | --- | --- | | [scope](#b820) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [source](#b822)?, [property](#b823)+ | |
| </actionSentence> | |

complexType "Upper"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [operand](#b1090) | |
| </...> | |

Content Model Elements (2):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [], | [operand](#b1090) (defined in [*UnaryExpression*](#b1092) complexType) [] |

Annotation

Returns the upper case of its argument.  
If the argument is null, the result is null.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  [*UnaryExpression*](#b1092) [] (extension)  **Upper** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Upper**](#b1103)">

<xs:complexContent>

<xs:extension base="[**UnaryExpression**](#b1092)"/>

</xs:complexContent>

</xs:complexType>

complexType "UrlLiteral"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [attributes](#b1106) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [value](#b1106) | = | xs:anyURI | | [use](#b1107) | = | *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "TMP" | "AS" | "EC" | "MC" | "PG") | | [capabilities](#b1108) | = | *list of* ("voice" | "fax" | "data" | "tty" | "sms") | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a value of type TEL with the given attributes.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **UrlLiteral** |

XML Source (w/o annotations (1))

<xs:complexType name="[**UrlLiteral**](#b1110)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**value**](#b1106)" type="xs:anyURI" use="required"/>

<xs:attribute name="[**use**](#b1107)" type="[**dt:set\_TelecommunicationAddressUse**](#b182)" use="optional"/>

<xs:attribute name="[**capabilities**](#b1108)" type="[**dt:set\_TelecommunicationCapability**](#b185)" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b1110) component only; 3/3)

 value

|  |  |
| --- | --- |
| Type: | xs:anyURI, predefined |
| Use: | required |

 use

|  |  |
| --- | --- |
| Type: | [dt:set\_TelecommunicationAddressUse](#b182) [] |
| Use: | optional |

Attribute Value

|  |
| --- |
| *list of* ("H" | "HP" | "HV" | "WP" | "DIR" | "PUB" | "BAD" | "TMP" | "AS" | "EC" | "MC" | "PG") |

 capabilities

|  |  |
| --- | --- |
| Type: | [dt:set\_TelecommunicationCapability](#b185) [] |
| Use: | optional |

Attribute Value

|  |
| --- |
| *list of* ("voice" | "fax" | "data" | "tty" | "sms") |

complexType "ValueSet"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 3 [attributes](#b1113) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [id](#b1113) | = | xs:string | | [version](#b1114) | = | xs:string | | [authority](#b1115) | = | xs:string | |
|  | > |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </...> | |

Content Model Elements (1):

[description](#b539) (defined in [*Expression*](#b541) complexType) []

Annotation

Returns a list of codes whose elements are defined by the  
given value set authority for the given value set id and version.  
  
If version is an asterisk (\*), the current version of the value  
set is returned.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **ValueSet** |

XML Source (w/o annotations (1))

<xs:complexType name="[**ValueSet**](#b1117)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:attribute name="[**id**](#b1113)" type="xs:string" use="required"/>

<xs:attribute name="[**version**](#b1114)" type="xs:string" use="optional"/>

<xs:attribute name="[**authority**](#b1115)" type="xs:string" use="optional"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b1117) component only; 3/3)

 id

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | required |

 version

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

 authority

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

complexType "VersionedIdentifier"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definition of 1 [attribute](#b1120) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <... | |
|  | |  |  |  | | --- | --- | --- | | [root](#b61) | = | xs:string | | [extension](#b62) | = | xs:string : "" | | [version](#b1120) | = | xs:string | |
| /> | |

All Direct / Indirect Based Elements (6):

groupReference,

[identifier](#b680) (defined in [*ItemDefinition*](#b682) complexType) [],

identifier (in [identifiers](#b694) defined in [KnowledgeResource](#b701) complexType),

identifier (in [identifiers](#b759) in [metadata](#b684)),

[schemaIdentifier](#b761) [],

templateId (in [templateIds](#b762) in [metadata](#b684))

Annotation

VersionedIdentifier is composed of two parts: (1)  
an II identifier which identifies the set of all versions of a given  
resource. (2) the actual version of the instance of interest in this  
set. The VersionedIdentifier therefore points to an individual  
'versioned' instance of a resource such as the third version of a  
reminder rule.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [dt:ANY](#b19) [] (extension)  [dt:II](#b64) [] (extension)  **VersionedIdentifier** |

XML Source (w/o annotations (1))

<xs:complexType name="[**VersionedIdentifier**](#b1122)">

<xs:complexContent>

<xs:extension base="[**dt:II**](#b64)">

<xs:attribute name="[**version**](#b1120)" type="xs:string"/>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Attribute Detail (all declarations; defined within [this](#b1122) component only; 1/3)

 version

|  |  |
| --- | --- |
| Type: | xs:string, predefined |
| Use: | optional |

complexType "Within"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |
| Includes: | definitions of 2 [elements](#b1125) |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <...> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)?, [element](#b1125), [interval](#b1126) | |
| </...> | |

Content Model Elements (3):

|  |  |
| --- | --- |
| [description](#b539) (defined in [*Expression*](#b541) complexType) [],  [element](#b1125) (defined in [Within](#b1128) complexType) [], | [interval](#b1126) [] |

Annotation

Returns true if the given element is in the given interval.  
There are two overloads of this operator:  
Scalar, Interval : The type of the scalar must be the same as the point type of the interval.  
Interval, Interval : The point type of both intervals must be the same.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  [*Expression*](#b541) [] (extension)  **Within** |

XML Source (w/o annotations (1))

<xs:complexType name="[**Within**](#b1128)">

<xs:complexContent>

<xs:extension base="[**Expression**](#b541)">

<xs:sequence>

<xs:element maxOccurs="1" minOccurs="1" name="[**element**](#b1125)" type="[**Expression**](#b541)"/>

<xs:element maxOccurs="1" minOccurs="1" name="[**interval**](#b1126)" type="[**Expression**](#b541)"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

Content Element Detail (all declarations; defined within [this](#b1128) component only; 2/3)

 element

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <element> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </element> | |

 interval

|  |  |
| --- | --- |
| Type: | [*Expression*](#b541) [], complex content |

|  |  |
| --- | --- |
| XML Representation Summary | |
| <interval> | |
|  | |  |  | | --- | --- | | *Content:* | [description](#b539)? | |
| </interval> | |

## KnowledgeDocument.xsd Simple Data Types

simpleType "ArtifactLifeCycleEventType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("Created" | "Pre-published" | "Published" | "Reviewed" | "Withdrawn" | "Superseded") | ("Created" | "Pre-published" | "Published" | "Reviewed" | "Withdrawn" | "Superseded") |

All Direct / Indirect Based Attributes (1):

[eventType](#b310) (in artifactLifeCycleEvent)/@value

Annotation

A version of an artifact may have different actions performed on it during the course of its life cycle. Each action is considered an artifact life cycle event, and may be recorded in the history of that version of the artifact.  
  
See the Implementation Guide for a state-transition diagram showing the legal transitions from each state; each transition is equivalent to an event.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([ArtifactLifeCycleEventTypeCore](#b1134) | [ArtifactLifeCycleEventTypeExt](#b1137))  **ArtifactLifeCycleEventType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [ArtifactLifeCycleEventTypeCore](#b1134)
2. [ArtifactLifeCycleEventTypeExt](#b1137)

XML Source (w/o annotations (1))

<xs:simpleType name="[**ArtifactLifeCycleEventType**](#b1131)">

<xs:union memberTypes="[**ArtifactLifeCycleEventTypeCore**](#b1134) [**ArtifactLifeCycleEventTypeExt**](#b1137)"/>

</xs:simpleType>

simpleType "ArtifactLifeCycleEventTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Created" | - | The date/time when a particular version of an artifact is created may be recorded in the history of the artifact-version. | | "Pre-published" | - | An artifact may be made available to consumers in a pre-publication state for public commentary, etc., or may simply be made available for review internal to the publisher. This event can be recorded in the artifact-version's history. | | "Published" | | | | "Reviewed" | - | A specific version of artifact may be reviewed by parties internal or external to the publisher, and this can be recorded in the lifecycle of the artifact-version. | | "Withdrawn" | - | A specific version of an artifact may be withdrawn by a publisher for various reasons. | | "Superseded" | - | A specific version of an artifact may be superseded by another version of that same artifact, or by another artifact altogether. | |

Known Direct Subtypes (2):

[ArtifactLifeCycleEventType](#b1131) [], [ArtifactLifeCycleEventTypeExt](#b1137) []

All Direct / Indirect Based Attributes (1):

[eventType](#b310) (in artifactLifeCycleEvent)/@value

Annotation

A lifecycle event is applicable to a specific version of an artifact.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **ArtifactLifeCycleEventTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "Created" | - | The date/time when a particular version of an artifact is created may be recorded in the history of the artifact-version. | | "Pre-published" | - | An artifact may be made available to consumers in a pre-publication state for public commentary, etc., or may simply be made available for review internal to the publisher. This event can be recorded in the artifact-version's history. | | "Published" | | | | "Reviewed" | - | A specific version of artifact may be reviewed by parties internal or external to the publisher, and this can be recorded in the lifecycle of the artifact-version. | | "Withdrawn" | - | A specific version of an artifact may be withdrawn by a publisher for various reasons. | | "Superseded" | - | A specific version of an artifact may be superseded by another version of that same artifact, or by another artifact altogether. | | |

XML Source (w/o annotations (6))

<xs:simpleType name="[**ArtifactLifeCycleEventTypeCore**](#b1134)">

<xs:restriction base="xs:string">

<xs:enumeration value="Created"/>

<xs:enumeration value="Pre-published"/>

<xs:enumeration value="Published"/>

<xs:enumeration value="Reviewed"/>

<xs:enumeration value="Withdrawn"/>

<xs:enumeration value="Superseded"/>

</xs:restriction>

</xs:simpleType>

simpleType "ArtifactLifeCycleEventTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Created" | - | The date/time when a particular version of an artifact is created may be recorded in the history of the artifact-version. | | "Pre-published" | - | An artifact may be made available to consumers in a pre-publication state for public commentary, etc., or may simply be made available for review internal to the publisher. This event can be recorded in the artifact-version's history. | | "Published" | | | | "Reviewed" | - | A specific version of artifact may be reviewed by parties internal or external to the publisher, and this can be recorded in the lifecycle of the artifact-version. | | "Withdrawn" | - | A specific version of an artifact may be withdrawn by a publisher for various reasons. | | "Superseded" | - | A specific version of an artifact may be superseded by another version of that same artifact, or by another artifact altogether. | |

Known Direct Subtypes (1):

[ArtifactLifeCycleEventType](#b1131) []

All Direct / Indirect Based Attributes (1):

[eventType](#b310) (in artifactLifeCycleEvent)/@value

Annotation

To add new items to the enumeration ArtifactLifeCycleEvent, comment out or delete the restriction on ArtifactLifeCycleEventTypeCore.(Optionally, also remove  
the directive above to include the artifactlifecycleeventtypecore.xsd).  
Now, add a new restriction such as on string and define new enumerated types. See the commented out example below

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [ArtifactLifeCycleEventTypeCore](#b1134) [] (restriction)  **ArtifactLifeCycleEventTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [ArtifactLifeCycleEventTypeCore](#b1134) |

XML Source (w/o annotations (1))

<xs:simpleType name="[**ArtifactLifeCycleEventTypeExt**](#b1137)">

<xs:restriction base="[**ArtifactLifeCycleEventTypeCore**](#b1134)"/>

<!--  
<xs:restriction base="xs:string">  
<xs:enumeration value="Erased"/>  
<xs:enumeration value="Restored"/>  
</xs:restriction>  
-->

</xs:simpleType>

simpleType "ArtifactStatusType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("Draft" | "InTest" | "Active" | "Inactive") | ("Draft" | "InTest" | "Active" | "Inactive") |

All Direct / Indirect Based Attributes (1):

[status](#b773)/@value

Annotation

A specific status is associated with each version of an artifact.  
  
See the Implementation Guide for a state-transition diagram showing the legal transitions from each state; each state is equivalent to particular status.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([ArtifactStatusTypeCore](#b1143) | [ArtifactStatusTypeExt](#b1146))  **ArtifactStatusType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [ArtifactStatusTypeCore](#b1143)
2. [ArtifactStatusTypeExt](#b1146)

XML Source (w/o annotations (1))

<xs:simpleType name="[**ArtifactStatusType**](#b1140)">

<xs:union memberTypes="[**ArtifactStatusTypeCore**](#b1143) [**ArtifactStatusTypeExt**](#b1146)"/>

</xs:simpleType>

simpleType "ArtifactStatusTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Enumeration: | |  | | --- | | "Draft" | | "InTest" | | "Active" | | "Inactive" | |

Known Direct Subtypes (2):

[ArtifactStatusType](#b1140) [], [ArtifactStatusTypeExt](#b1146) []

All Direct / Indirect Based Attributes (1):

[status](#b773)/@value

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **ArtifactStatusTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | enumeration: | |  | | --- | | "Draft" | | "InTest" | | "Active" | | "Inactive" | | |

XML Source (w/o annotations (1))

<xs:simpleType name="[**ArtifactStatusTypeCore**](#b1143)">

<xs:restriction base="xs:string">

<xs:enumeration value="Draft"/>

<xs:enumeration value="InTest"/>

<xs:enumeration value="Active"/>

<xs:enumeration value="Inactive"/>

</xs:restriction>

</xs:simpleType>

simpleType "ArtifactStatusTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Enumeration: | |  | | --- | | "Draft" | | "InTest" | | "Active" | | "Inactive" | |

Known Direct Subtypes (1):

[ArtifactStatusType](#b1140) []

All Direct / Indirect Based Attributes (1):

[status](#b773)/@value

Annotation

To add new items to the enumeration ArtifactStatus, comment out or delete the restriction on ArtifactStatusTypeCore.(Optionally, also remove  
the directive above to include the artifactstatustypecore.xsd).  
Now, add a new restriction such as on string and define new enumerated types. See the commented out example below. !!! BKM get example from ArtifactLifeCycleEventTypeExt

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [ArtifactStatusTypeCore](#b1143) [] (restriction)  **ArtifactStatusTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [ArtifactStatusTypeCore](#b1143) |

XML Source (w/o annotations (1))

<xs:simpleType name="[**ArtifactStatusTypeExt**](#b1146)">

<xs:restriction base="[**ArtifactStatusTypeCore**](#b1143)"/>

</xs:simpleType>

simpleType "ArtifactType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("Rule" | "Order Set" | "Documentation Template") | ("Rule" | "Order Set" | "Documentation Template") |

All Direct / Indirect Based Attributes (1):

[artifactType](#b760)/@value

Annotation

Three types of artifacts are in scope for Health eDecisions Use Case #1: Event-condition-action rules, order sets and documentation templates. In addition, a CDS artifact may also be a value set. !!! Get more info from Aziz about this ... !!!

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([ArtifactTypeCore](#b1152) | [ArtifactTypeExt](#b1155))  **ArtifactType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [ArtifactTypeCore](#b1152)
2. [ArtifactTypeExt](#b1155)

XML Source (w/o annotations (1))

<xs:simpleType name="[**ArtifactType**](#b1149)">

<xs:union memberTypes="[**ArtifactTypeCore**](#b1152) [**ArtifactTypeExt**](#b1155)"/>

</xs:simpleType>

simpleType "ArtifactTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Rule" | - | Denotes an Event-Condition-Action Rule. An ECA rule has a specific structure that is a restriction on the general structure of a CDS Knowledge Artifact. See the Implementation Guide for the constraints applicable to ECA rules. | | "Order Set" | - | Denotes an Order Set for use by physicians at the point of care. An Order Set has a specific structure that is a restriction on the general structure of a CDS Knowledge Artifact. See the Implementation Guide for the constraints applicable to Order Sets. | | "Documentation Template" | - | Denotes a Documentation Template for use by care providers. A Documentation Template has a specific structure that is a restriction on the general structure of a CDS Knowledge Artifact. See the Implementation Guide for the constraints applicable to Documentation Templates. | |

Known Direct Subtypes (2):

[ArtifactType](#b1149) [], [ArtifactTypeExt](#b1155) []

All Direct / Indirect Based Attributes (1):

[artifactType](#b760)/@value

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **ArtifactTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "Rule" | - | Denotes an Event-Condition-Action Rule. An ECA rule has a specific structure that is a restriction on the general structure of a CDS Knowledge Artifact. See the Implementation Guide for the constraints applicable to ECA rules. | | "Order Set" | - | Denotes an Order Set for use by physicians at the point of care. An Order Set has a specific structure that is a restriction on the general structure of a CDS Knowledge Artifact. See the Implementation Guide for the constraints applicable to Order Sets. | | "Documentation Template" | - | Denotes a Documentation Template for use by care providers. A Documentation Template has a specific structure that is a restriction on the general structure of a CDS Knowledge Artifact. See the Implementation Guide for the constraints applicable to Documentation Templates. | | |

XML Source (w/o annotations (3))

<xs:simpleType name="[**ArtifactTypeCore**](#b1152)">

<xs:restriction base="xs:string">

<xs:enumeration value="Rule"/>

<xs:enumeration value="Order Set"/>

<xs:enumeration value="Documentation Template"/>

</xs:restriction>

</xs:simpleType>

simpleType "ArtifactTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Rule" | - | Denotes an Event-Condition-Action Rule. An ECA rule has a specific structure that is a restriction on the general structure of a CDS Knowledge Artifact. See the Implementation Guide for the constraints applicable to ECA rules. | | "Order Set" | - | Denotes an Order Set for use by physicians at the point of care. An Order Set has a specific structure that is a restriction on the general structure of a CDS Knowledge Artifact. See the Implementation Guide for the constraints applicable to Order Sets. | | "Documentation Template" | - | Denotes a Documentation Template for use by care providers. A Documentation Template has a specific structure that is a restriction on the general structure of a CDS Knowledge Artifact. See the Implementation Guide for the constraints applicable to Documentation Templates. | |

Known Direct Subtypes (1):

[ArtifactType](#b1149) []

All Direct / Indirect Based Attributes (1):

[artifactType](#b760)/@value

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [ArtifactTypeCore](#b1152) [] (restriction)  **ArtifactTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [ArtifactTypeCore](#b1152) |

XML Source

<xs:simpleType name="[**ArtifactTypeExt**](#b1155)">

<xs:restriction base="[**ArtifactTypeCore**](#b1152)"/>

</xs:simpleType>

simpleType "CalendarCycle"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("D" | "H" | "J" | "M" | "N" | "S" | "W" | "Y") | (xs:string | ("CD" | "CH" | "CM" | "CN" | "CS" | "CW" | "CY" | "DM" | "DW" | "DY" | "HD" | "MY" | "NH" | "SN" | "WY")) |

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [PIVL\_TS](#b886)/@[alignment](#b880) [], | [PeriodLiteral](#b862)/@[alignment](#b856) [] |

Annotation

vocSet: D10684 (C-0-D10684-cpt)

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([CalendarCycleOneLetter](#b1161) | [CalendarCycleTwoLetter](#b1164))  **CalendarCycle** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [CalendarCycleOneLetter](#b1161)
2. [CalendarCycleTwoLetter](#b1164)

XML Source (w/o annotations (1))

<xs:simpleType name="[**CalendarCycle**](#b1158)">

<xs:union memberTypes="[**CalendarCycleOneLetter**](#b1161) [**CalendarCycleTwoLetter**](#b1164)"/>

</xs:simpleType>

simpleType "CalendarCycleOneLetter"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "D", "H", "J", "M", "N", "S", "W", "Y" |

Known Direct Subtypes (1):

[CalendarCycle](#b1158) []

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [PIVL\_TS](#b886)/@[alignment](#b880) [], | [PeriodLiteral](#b862)/@[alignment](#b856) [] |

Annotation

abstDomain: V10701 (C-0-D10684-V10701-cpt)

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **CalendarCycleOneLetter** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "D", "H", "J", "M", "N", "S", "W", "Y" | |

XML Source (w/o annotations (1))

<xs:simpleType name="[**CalendarCycleOneLetter**](#b1161)">

<xs:restriction base="xs:string">

<xs:enumeration value="D"/>

<xs:enumeration value="H"/>

<xs:enumeration value="J"/>

<xs:enumeration value="M"/>

<xs:enumeration value="N"/>

<xs:enumeration value="S"/>

<xs:enumeration value="W"/>

<xs:enumeration value="Y"/>

</xs:restriction>

</xs:simpleType>

simpleType "CalendarCycleTwoLetter"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| xs:string | ("CD" | "CH" | "CM" | "CN" | "CS" | "CW" | "CY" | "DM" | "DW" | "DY" | "HD" | "MY" | "NH" | "SN" | "WY") |

Known Direct Subtypes (1):

[CalendarCycle](#b1158) []

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [PIVL\_TS](#b886)/@[alignment](#b880) [], | [PeriodLiteral](#b862)/@[alignment](#b856) [] |

Annotation

abstDomain: V10685 (C-0-D10684-V10685-cpt)

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([GregorianCalendarCycle](#b1221) | restriction of xs:string)  **CalendarCycleTwoLetter** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [GregorianCalendarCycle](#b1221)
2. anonymous simpleType:

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "CD", "CH", "CM", "CN", "CS", "CW", "CY", "DM", "DW", "DY", "HD", "MY", "NH", "SN", "WY" | |

XML Source (w/o annotations (1))

<xs:simpleType name="[**CalendarCycleTwoLetter**](#b1164)">

<xs:union memberTypes="[**GregorianCalendarCycle**](#b1221)">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="CD"/>

<xs:enumeration value="CH"/>

<xs:enumeration value="CM"/>

<xs:enumeration value="CN"/>

<xs:enumeration value="CS"/>

<xs:enumeration value="CW"/>

<xs:enumeration value="CY"/>

<xs:enumeration value="DM"/>

<xs:enumeration value="DW"/>

<xs:enumeration value="DY"/>

<xs:enumeration value="HD"/>

<xs:enumeration value="MY"/>

<xs:enumeration value="NH"/>

<xs:enumeration value="SN"/>

<xs:enumeration value="WY"/>

</xs:restriction>

</xs:simpleType>

</xs:union>

</xs:simpleType>

simpleType "Cardinality"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "Single", "Multiple" |

All Direct / Indirect Based Elements (1):

[responseCardinality](#b515) []

Annotation

Cardinality defines the expected cardinality of an element, single  
or multiple.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **Cardinality** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "Single", "Multiple" | |

XML Source (w/o annotations (1))

<xs:simpleType name="[**Cardinality**](#b1167)">

<xs:restriction base="xs:string">

<xs:enumeration value="Single"/>

<xs:enumeration value="Multiple"/>

</xs:restriction>

</xs:simpleType>

simpleType "CompressionAlgorithm"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "DF", "GZ", "Z", "ZL" |

Annotation

vocSet: D10620 (C-0-D10620-cpt)

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **CompressionAlgorithm** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "DF", "GZ", "Z", "ZL" | |

XML Source (w/o annotations (1))

<xs:simpleType name="[**CompressionAlgorithm**](#b1170)">

<xs:restriction base="xs:string">

<xs:enumeration value="DF"/>

<xs:enumeration value="GZ"/>

<xs:enumeration value="Z"/>

<xs:enumeration value="ZL"/>

</xs:restriction>

</xs:simpleType>

simpleType "ConditionRoleType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| "ApplicableScenario" | "ApplicableScenario" |

All Direct / Indirect Based Elements (1):

[conditionRole](#b413) []

Annotation

The roles that a condition plays in the execution  
of a component. Currently, only one role type is defined. Additional  
role types may be defined in the future (e.g., inclusion criteria,  
exclusion criteria)

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([ConditionRoleTypeCore](#b1176) | [ConditionRoleTypeExt](#b1179))  **ConditionRoleType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [ConditionRoleTypeCore](#b1176)
2. [ConditionRoleTypeExt](#b1179)

XML Source (w/o annotations (1))

<xs:simpleType name="[**ConditionRoleType**](#b1173)">

<xs:union memberTypes="[**ConditionRoleTypeCore**](#b1176) [**ConditionRoleTypeExt**](#b1179)"/>

</xs:simpleType>

simpleType "ConditionRoleTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "ApplicableScenario" | - | This role type specifies that a condition is used to determine whether or not a particular knowledge component should be executed. If the expression evaluates to true, then the component is executed. | |

Known Direct Subtypes (2):

[ConditionRoleType](#b1173) [], [ConditionRoleTypeExt](#b1179) []

All Direct / Indirect Based Elements (1):

[conditionRole](#b413) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **ConditionRoleTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  | | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "ApplicableScenario" | - | This role type specifies that a condition is used to determine whether or not a particular knowledge component should be executed. If the expression evaluates to true, then the component is executed. | | |

XML Source (w/o annotations (1))

<xs:simpleType name="[**ConditionRoleTypeCore**](#b1176)">

<xs:restriction base="xs:string">

<xs:enumeration value="ApplicableScenario"/>

<!-- <xs:enumeration value="Inclusion"></xs:enumeration>  
<xs:enumeration value="Exclusion"></xs:enumeration>  
<xs:enumeration value="Precondition"></xs:enumeration>  
<xs:enumeration value="Postcondition"></xs:enumeration> -->

</xs:restriction>

</xs:simpleType>

simpleType "ConditionRoleTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "ApplicableScenario" | - | This role type specifies that a condition is used to determine whether or not a particular knowledge component should be executed. If the expression evaluates to true, then the component is executed. | |

Known Direct Subtypes (1):

[ConditionRoleType](#b1173) []

All Direct / Indirect Based Elements (1):

[conditionRole](#b413) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [ConditionRoleTypeCore](#b1176) [] (restriction)  **ConditionRoleTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [ConditionRoleTypeCore](#b1176) |

XML Source

<xs:simpleType name="[**ConditionRoleTypeExt**](#b1179)">

<xs:restriction base="[**ConditionRoleTypeCore**](#b1176)"/>

</xs:simpleType>

simpleType "ContributorType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("Author" | "Editor" | "Endorser" | "Reviewer") | ("Author" | "Editor" | "Endorser" | "Reviewer") |

All Direct / Indirect Based Attributes (1):

[role](#b435)/@value

Annotation

Enumeration of roles that contribute to the development and maintenance of a knowledge artifact.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([ContributorTypeCore](#b1185) | [ContributorTypeExt](#b1188))  **ContributorType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [ContributorTypeCore](#b1185)
2. [ContributorTypeExt](#b1188)

XML Source (w/o annotations (1))

<xs:simpleType name="[**ContributorType**](#b1182)">

<xs:union memberTypes="[**ContributorTypeCore**](#b1185) [**ContributorTypeExt**](#b1188)"/>

</xs:simpleType>

simpleType "ContributorTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "Author", "Editor", "Endorser", "Reviewer" |

Known Direct Subtypes (2):

[ContributorType](#b1182) [], [ContributorTypeExt](#b1188) []

All Direct / Indirect Based Attributes (1):

[role](#b435)/@value

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **ContributorTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "Author", "Editor", "Endorser", "Reviewer" | |

XML Source

<xs:simpleType name="[**ContributorTypeCore**](#b1185)">

<xs:restriction base="xs:string">

<xs:enumeration value="Author"/>

<xs:enumeration value="Editor"/>

<xs:enumeration value="Endorser"/>

<xs:enumeration value="Reviewer"/>

</xs:restriction>

</xs:simpleType>

simpleType "ContributorTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "Author", "Editor", "Endorser", "Reviewer" |

Known Direct Subtypes (1):

[ContributorType](#b1182) []

All Direct / Indirect Based Attributes (1):

[role](#b435)/@value

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [ContributorTypeCore](#b1185) [] (restriction)  **ContributorTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [ContributorTypeCore](#b1185) |

XML Source

<xs:simpleType name="[**ContributorTypeExt**](#b1188)">

<xs:restriction base="[**ContributorTypeCore**](#b1185)"/>

</xs:simpleType>

simpleType "CoverageType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("PatientGender" | "PatientAgeGroup" | "ClinicalFocus" | "TargetUser" | "WorkflowSetting" | "WorkflowTask" | "ClinicalVenue") | ("PatientGender" | "PatientAgeGroup" | "ClinicalFocus" | "TargetUser" | "WorkflowSetting" | "WorkflowTask" | "ClinicalVenue") |

All Direct / Indirect Based Attributes (1):

[focus](#b447)/@value

Annotation

Specifies clinical metadata that can be used to retrieve, index and/or categorize the knowledge artifact. This metadata can either be specific to the applicable population (e.g., age category, DRG) or the specific context of care (e.g., venue, care setting, provider of care).

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([CoverageTypeCore](#b1194) | [CoverageTypeExt](#b1197))  **CoverageType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [CoverageTypeCore](#b1194)
2. [CoverageTypeExt](#b1197)

XML Source (w/o annotations (1))

<xs:simpleType name="[**CoverageType**](#b1191)">

<xs:union memberTypes="[**CoverageTypeCore**](#b1194) [**CoverageTypeExt**](#b1197)"/>

</xs:simpleType>

simpleType "CoverageTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "PatientGender" | - | The gender of the patient. For this item type, use HL7 administrative gender codes (OID: 2.16.840.1.113883.1.11.1) | | "PatientAgeGroup" | - | A patient demographic category for which this artifact is applicable. Allows specification of age groups using coded values originating from the MeSH Code system (OID: 2.16.840.1.113883.6.177). More specifically, only codes from the AgeGroupObservationValue value set are valid for this field [2.16.840.1.113883.11.75] | | "ClinicalFocus" | - | The clinical concept(s) addressed by the artifact. For example, disease, diagnostic test interpretation, medication ordering. Please refer to the implementation guide on which code system and codes to use. | | "TargetUser" | - | The user types to which an artifact is targeted. For example, PCP, Patient, Cardiologist, Behavioral Professional, Oral Health Professional, Prescriber, etc... taken from the NUCC Health Care provider taxonomyCode system (OID: 2.16.840.1.113883.6.101) | | "WorkflowSetting" | - | The settings in which the artifact is intended for use. For example, admission, pre-op, etc. | | "WorkflowTask" | - | The context for the clinical task(s) represented by this artifact. Can be any task context represented by the HL7 ActTaskCode value set (OID: 2.16.840.1.113883.1.11.19846). General categories include: order entry, patient documentation and patient information review | | "ClinicalVenue" | - | The venue in which an artifact could be used. For example, Outpatient, Inpatient, Home, Nursing home. The code value may originate from either the HL7 ActEncounter (OID: 2.16.840.1.113883.1.11.13955) or NUCC non-individual provider codes OID: 2.16.840.1.113883.1.11.19465 | |

Known Direct Subtypes (2):

[CoverageType](#b1191) [], [CoverageTypeExt](#b1197) []

All Direct / Indirect Based Attributes (1):

[focus](#b447)/@value

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **CoverageTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "PatientGender" | - | The gender of the patient. For this item type, use HL7 administrative gender codes (OID: 2.16.840.1.113883.1.11.1) | | "PatientAgeGroup" | - | A patient demographic category for which this artifact is applicable. Allows specification of age groups using coded values originating from the MeSH Code system (OID: 2.16.840.1.113883.6.177). More specifically, only codes from the AgeGroupObservationValue value set are valid for this field [2.16.840.1.113883.11.75] | | "ClinicalFocus" | - | The clinical concept(s) addressed by the artifact. For example, disease, diagnostic test interpretation, medication ordering. Please refer to the implementation guide on which code system and codes to use. | | "TargetUser" | - | The user types to which an artifact is targeted. For example, PCP, Patient, Cardiologist, Behavioral Professional, Oral Health Professional, Prescriber, etc... taken from the NUCC Health Care provider taxonomyCode system (OID: 2.16.840.1.113883.6.101) | | "WorkflowSetting" | - | The settings in which the artifact is intended for use. For example, admission, pre-op, etc. | | "WorkflowTask" | - | The context for the clinical task(s) represented by this artifact. Can be any task context represented by the HL7 ActTaskCode value set (OID: 2.16.840.1.113883.1.11.19846). General categories include: order entry, patient documentation and patient information review | | "ClinicalVenue" | - | The venue in which an artifact could be used. For example, Outpatient, Inpatient, Home, Nursing home. The code value may originate from either the HL7 ActEncounter (OID: 2.16.840.1.113883.1.11.13955) or NUCC non-individual provider codes OID: 2.16.840.1.113883.1.11.19465 | | |

XML Source (w/o annotations (7))

<xs:simpleType name="[**CoverageTypeCore**](#b1194)">

<xs:restriction base="xs:string">

<xs:enumeration value="PatientGender"/>

<xs:enumeration value="PatientAgeGroup"/>

<xs:enumeration value="ClinicalFocus"/>

<xs:enumeration value="TargetUser"/>

<xs:enumeration value="WorkflowSetting"/>

<xs:enumeration value="WorkflowTask"/>

<xs:enumeration value="ClinicalVenue"/>

</xs:restriction>

</xs:simpleType>

simpleType "CoverageTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "PatientGender" | - | The gender of the patient. For this item type, use HL7 administrative gender codes (OID: 2.16.840.1.113883.1.11.1) | | "PatientAgeGroup" | - | A patient demographic category for which this artifact is applicable. Allows specification of age groups using coded values originating from the MeSH Code system (OID: 2.16.840.1.113883.6.177). More specifically, only codes from the AgeGroupObservationValue value set are valid for this field [2.16.840.1.113883.11.75] | | "ClinicalFocus" | - | The clinical concept(s) addressed by the artifact. For example, disease, diagnostic test interpretation, medication ordering. Please refer to the implementation guide on which code system and codes to use. | | "TargetUser" | - | The user types to which an artifact is targeted. For example, PCP, Patient, Cardiologist, Behavioral Professional, Oral Health Professional, Prescriber, etc... taken from the NUCC Health Care provider taxonomyCode system (OID: 2.16.840.1.113883.6.101) | | "WorkflowSetting" | - | The settings in which the artifact is intended for use. For example, admission, pre-op, etc. | | "WorkflowTask" | - | The context for the clinical task(s) represented by this artifact. Can be any task context represented by the HL7 ActTaskCode value set (OID: 2.16.840.1.113883.1.11.19846). General categories include: order entry, patient documentation and patient information review | | "ClinicalVenue" | - | The venue in which an artifact could be used. For example, Outpatient, Inpatient, Home, Nursing home. The code value may originate from either the HL7 ActEncounter (OID: 2.16.840.1.113883.1.11.13955) or NUCC non-individual provider codes OID: 2.16.840.1.113883.1.11.19465 | |

Known Direct Subtypes (1):

[CoverageType](#b1191) []

All Direct / Indirect Based Attributes (1):

[focus](#b447)/@value

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [CoverageTypeCore](#b1194) [] (restriction)  **CoverageTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [CoverageTypeCore](#b1194) |

XML Source

<xs:simpleType name="[**CoverageTypeExt**](#b1197)">

<xs:restriction base="[**CoverageTypeCore**](#b1194)"/>

</xs:simpleType>

simpleType "DataEventType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("DataElementAdded" | "DataElementModified" | "DataElementRemoved" | "DataElementAccessed" | "DataElementAccessEnded") | ("DataElementAdded" | "DataElementModified" | "DataElementRemoved" | "DataElementAccessed" | "DataElementAccessEnded") |

All Direct / Indirect Based Attributes (1):

[*RequestBase*](#b967)/@[triggerType](#b962) []

Annotation

Enumeration of types of events related to access, creation, removal, or update of data.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([DataEventTypeCore](#b1203) | [DataEventTypeExt](#b1206))  **DataEventType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [DataEventTypeCore](#b1203)
2. [DataEventTypeExt](#b1206)

XML Source (w/o annotations (1))

<xs:simpleType name="[**DataEventType**](#b1200)">

<xs:union memberTypes="[**DataEventTypeCore**](#b1203) [**DataEventTypeExt**](#b1206)"/>

</xs:simpleType>

simpleType "DataEventTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "DataElementAdded", "DataElementModified", "DataElementRemoved", "DataElementAccessed", "DataElementAccessEnded" |

Known Direct Subtypes (2):

[DataEventType](#b1200) [], [DataEventTypeExt](#b1206) []

All Direct / Indirect Based Attributes (1):

[*RequestBase*](#b967)/@[triggerType](#b962) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **DataEventTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "DataElementAdded", "DataElementModified", "DataElementRemoved", "DataElementAccessed", "DataElementAccessEnded" | |

XML Source

<xs:simpleType name="[**DataEventTypeCore**](#b1203)">

<xs:restriction base="xs:string">

<xs:enumeration value="DataElementAdded"/>

<xs:enumeration value="DataElementModified"/>

<xs:enumeration value="DataElementRemoved"/>

<xs:enumeration value="DataElementAccessed"/>

<xs:enumeration value="DataElementAccessEnded"/>

</xs:restriction>

</xs:simpleType>

simpleType "DataEventTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "DataElementAdded", "DataElementModified", "DataElementRemoved", "DataElementAccessed", "DataElementAccessEnded" |

Known Direct Subtypes (1):

[DataEventType](#b1200) []

All Direct / Indirect Based Attributes (1):

[*RequestBase*](#b967)/@[triggerType](#b962) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [DataEventTypeCore](#b1203) [] (restriction)  **DataEventTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [DataEventTypeCore](#b1203) |

XML Source

<xs:simpleType name="[**DataEventTypeExt**](#b1206)">

<xs:restriction base="[**DataEventTypeCore**](#b1203)"/>

</xs:simpleType>

simpleType "DateGranularity"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "Year", "Month", "Week", "Day", "Hour", "Minute", "Second", "Millisecond" |

Annotation

DateGranularity specifies the granularities available for temporal operations  
such as DateAdd, DateDiff, and DatePart.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **DateGranularity** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "Year", "Month", "Week", "Day", "Hour", "Minute", "Second", "Millisecond" | |

XML Source (w/o annotations (1))

<xs:simpleType name="[**DateGranularity**](#b1209)">

<xs:restriction base="xs:string">

<xs:enumeration value="Year"/>

<xs:enumeration value="Month"/>

<xs:enumeration value="Week"/>

<xs:enumeration value="Day"/>

<xs:enumeration value="Hour"/>

<xs:enumeration value="Minute"/>

<xs:enumeration value="Second"/>

<xs:enumeration value="Millisecond"/>

</xs:restriction>

</xs:simpleType>

simpleType "EventType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("DataEvent" | "PeriodicEvent") | ("DataEvent" | "PeriodicEvent") |

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [eventType](#b568) (defined in [FireEventAction](#b571) complexType) [], | [eventType](#b1077) (in [trigger](#b1082)) [] |

Annotation

An enumeration of event types. Events occur external to the artifact that can be used as a trigger to the artifact.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([EventTypeCore](#b1215) | [EventTypeExt](#b1218))  **EventType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [EventTypeCore](#b1215)
2. [EventTypeExt](#b1218)

XML Source (w/o annotations (1))

<xs:simpleType name="[**EventType**](#b1212)">

<xs:union memberTypes="[**EventTypeCore**](#b1215) [**EventTypeExt**](#b1218)"/>

</xs:simpleType>

simpleType "EventTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "DataEvent" | - | An event in which a data item is created, removed, updated, or accessed. Expression is expected to be an ExpressionRef that references an ExpressionDef in ExternalData that contains a Request with a triggerType attribute specified. | | "PeriodicEvent" | - | A time-based event which occurs at the specified period. Expression is expected to be a Period literal expression specifying the period on which the event should be repeated | |

Known Direct Subtypes (2):

[EventType](#b1212) [], [EventTypeExt](#b1218) []

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [eventType](#b568) (defined in [FireEventAction](#b571) complexType) [], | [eventType](#b1077) (in [trigger](#b1082)) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **EventTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "DataEvent" | - | An event in which a data item is created, removed, updated, or accessed. Expression is expected to be an ExpressionRef that references an ExpressionDef in ExternalData that contains a Request with a triggerType attribute specified. | | "PeriodicEvent" | - | A time-based event which occurs at the specified period. Expression is expected to be a Period literal expression specifying the period on which the event should be repeated | | |

XML Source (w/o annotations (2))

<xs:simpleType name="[**EventTypeCore**](#b1215)">

<xs:restriction base="xs:string">

<xs:enumeration value="DataEvent"/>

<xs:enumeration value="PeriodicEvent"/>

</xs:restriction>

</xs:simpleType>

simpleType "EventTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "DataEvent" | - | An event in which a data item is created, removed, updated, or accessed. Expression is expected to be an ExpressionRef that references an ExpressionDef in ExternalData that contains a Request with a triggerType attribute specified. | | "PeriodicEvent" | - | A time-based event which occurs at the specified period. Expression is expected to be a Period literal expression specifying the period on which the event should be repeated | |

Known Direct Subtypes (1):

[EventType](#b1212) []

All Direct / Indirect Based Elements (2):

|  |  |
| --- | --- |
| [eventType](#b568) (defined in [FireEventAction](#b571) complexType) [], | [eventType](#b1077) (in [trigger](#b1082)) [] |

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [EventTypeCore](#b1215) [] (restriction)  **EventTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [EventTypeCore](#b1215) |

XML Source

<xs:simpleType name="[**EventTypeExt**](#b1218)">

<xs:restriction base="[**EventTypeCore**](#b1215)"/>

</xs:simpleType>

simpleType "GregorianCalendarCycle"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| xs:string |

Known Direct Subtypes (1):

[CalendarCycleTwoLetter](#b1164) []

Known Indirect Subtypes (1):

[CalendarCycle](#b1158) []

All Direct / Indirect Based Attributes (2):

|  |  |
| --- | --- |
| [PIVL\_TS](#b886)/@[alignment](#b880) [], | [PeriodLiteral](#b862)/@[alignment](#b856) [] |

Annotation

abstDomain: V10758 (C-0-D10684-V10685-V10758-cpt)

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **GregorianCalendarCycle** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |

XML Source (w/o annotations (1))

<xs:simpleType name="[**GregorianCalendarCycle**](#b1221)">

<xs:restriction base="xs:string"/>

</xs:simpleType>

simpleType "GroupOrganizationBehaviorType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("VisualGroup" | "LogicalGroup" | "SentenceGroup") | ("VisualGroup" | "LogicalGroup" | "SentenceGroup") |

All Direct / Indirect Based Elements (1):

[value](#b598) (defined in [GroupOrganizationBehavior](#b600) complexType) []

Annotation

Defines organization behavior of a group: gives the reason why the items are grouped together.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([GroupOrganizationBehaviorTypeCore](#b1227) | [GroupOrganizationBehaviorTypeExt](#b1230))  **GroupOrganizationBehaviorType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [GroupOrganizationBehaviorTypeCore](#b1227)
2. [GroupOrganizationBehaviorTypeExt](#b1230)

XML Source (w/o annotations (1))

<xs:simpleType name="[**GroupOrganizationBehaviorType**](#b1224)">

<xs:union memberTypes="[**GroupOrganizationBehaviorTypeCore**](#b1227) [**GroupOrganizationBehaviorTypeExt**](#b1230)"/>

</xs:simpleType>

simpleType "GroupOrganizationBehaviorTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "VisualGroup" | - | Any group marked with this behavior should be displayed as a visual group to the end user. | | "LogicalGroup" | - | A group with this behavior logically groups its sub-elements, and may be shown as a visual group to the end user, but it is not required to do so. | | "SentenceGroup" | - | A group of related alternative actions is a sentence group if the item referenced by the action is the same in all the actions, and each action simply constitutes a different variation on how to specify the details for that item. For example, two actions that could be in a SentenceGroup are "aspirin, 500 mg, 2 times per day" and "aspirin, 300 mg, 3 times per day". In both cases, aspirin is the item referenced by the action, and the two actions represent two different options for how aspirin might be ordered for the patient. Note that a SentenceGroup would almost always have an associated selection behavior of "AtMostOne", unless it's a required action, in which case, it would be "ExactlyOne". | |

Known Direct Subtypes (2):

[GroupOrganizationBehaviorType](#b1224) [], [GroupOrganizationBehaviorTypeExt](#b1230) []

All Direct / Indirect Based Elements (1):

[value](#b598) (defined in [GroupOrganizationBehavior](#b600) complexType) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **GroupOrganizationBehaviorTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "VisualGroup" | - | Any group marked with this behavior should be displayed as a visual group to the end user. | | "LogicalGroup" | - | A group with this behavior logically groups its sub-elements, and may be shown as a visual group to the end user, but it is not required to do so. | | "SentenceGroup" | - | A group of related alternative actions is a sentence group if the item referenced by the action is the same in all the actions, and each action simply constitutes a different variation on how to specify the details for that item. For example, two actions that could be in a SentenceGroup are "aspirin, 500 mg, 2 times per day" and "aspirin, 300 mg, 3 times per day". In both cases, aspirin is the item referenced by the action, and the two actions represent two different options for how aspirin might be ordered for the patient. Note that a SentenceGroup would almost always have an associated selection behavior of "AtMostOne", unless it's a required action, in which case, it would be "ExactlyOne". | | |

XML Source (w/o annotations (3))

<xs:simpleType name="[**GroupOrganizationBehaviorTypeCore**](#b1227)">

<xs:restriction base="xs:string">

<xs:enumeration value="VisualGroup"/>

<xs:enumeration value="LogicalGroup"/>

<xs:enumeration value="SentenceGroup"/>

</xs:restriction>

</xs:simpleType>

simpleType "GroupOrganizationBehaviorTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "VisualGroup" | - | Any group marked with this behavior should be displayed as a visual group to the end user. | | "LogicalGroup" | - | A group with this behavior logically groups its sub-elements, and may be shown as a visual group to the end user, but it is not required to do so. | | "SentenceGroup" | - | A group of related alternative actions is a sentence group if the item referenced by the action is the same in all the actions, and each action simply constitutes a different variation on how to specify the details for that item. For example, two actions that could be in a SentenceGroup are "aspirin, 500 mg, 2 times per day" and "aspirin, 300 mg, 3 times per day". In both cases, aspirin is the item referenced by the action, and the two actions represent two different options for how aspirin might be ordered for the patient. Note that a SentenceGroup would almost always have an associated selection behavior of "AtMostOne", unless it's a required action, in which case, it would be "ExactlyOne". | |

Known Direct Subtypes (1):

[GroupOrganizationBehaviorType](#b1224) []

All Direct / Indirect Based Elements (1):

[value](#b598) (defined in [GroupOrganizationBehavior](#b600) complexType) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [GroupOrganizationBehaviorTypeCore](#b1227) [] (restriction)  **GroupOrganizationBehaviorTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [GroupOrganizationBehaviorTypeCore](#b1227) |

XML Source

<xs:simpleType name="[**GroupOrganizationBehaviorTypeExt**](#b1230)">

<xs:restriction base="[**GroupOrganizationBehaviorTypeCore**](#b1227)"/>

</xs:simpleType>

simpleType "GroupSelectionBehaviorType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("Any" | "All" | "AllOrNone" | "ExactlyOne" | "AtMostOne" | "OneOrMore") | ("Any" | "All" | "AllOrNone" | "ExactlyOne" | "AtMostOne" | "OneOrMore") |

All Direct / Indirect Based Elements (1):

[value](#b603) (defined in [GroupSelectionBehavior](#b605) complexType) []

Annotation

Defines selection behavior of a group: specifies the number of selectable items in the group that may be selected by the end user when the items of the group are displayed.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([GroupSelectionBehaviorTypeCore](#b1236) | [GroupSelectionBehaviorTypeExt](#b1239))  **GroupSelectionBehaviorType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [GroupSelectionBehaviorTypeCore](#b1236)
2. [GroupSelectionBehaviorTypeExt](#b1239)

XML Source (w/o annotations (1))

<xs:simpleType name="[**GroupSelectionBehaviorType**](#b1233)">

<xs:union memberTypes="[**GroupSelectionBehaviorTypeCore**](#b1236) [**GroupSelectionBehaviorTypeExt**](#b1239)"/>

</xs:simpleType>

simpleType "GroupSelectionBehaviorTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Any" | - | Any number of the items in the group may be chosen, from zero to all. | | "All" | - | All the items in the group must be selected as a single unit. | | "AllOrNone" | - | All the items in the group are meant to be chosen as a single unit: either all must be selected by the end user, or none may be selected. | | "ExactlyOne" | - | The end user must choose one and only one of the selectable items in the group. The user may not choose none of the items in the group. | | "AtMostOne" | - | The end user may choose zero or at most one of the items in the group. | | "OneOrMore" | - | The end user must choose a minimum of one, and as many additional as desired. | |

Known Direct Subtypes (2):

[GroupSelectionBehaviorType](#b1233) [], [GroupSelectionBehaviorTypeExt](#b1239) []

All Direct / Indirect Based Elements (1):

[value](#b603) (defined in [GroupSelectionBehavior](#b605) complexType) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **GroupSelectionBehaviorTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "Any" | - | Any number of the items in the group may be chosen, from zero to all. | | "All" | - | All the items in the group must be selected as a single unit. | | "AllOrNone" | - | All the items in the group are meant to be chosen as a single unit: either all must be selected by the end user, or none may be selected. | | "ExactlyOne" | - | The end user must choose one and only one of the selectable items in the group. The user may not choose none of the items in the group. | | "AtMostOne" | - | The end user may choose zero or at most one of the items in the group. | | "OneOrMore" | - | The end user must choose a minimum of one, and as many additional as desired. | | |

XML Source (w/o annotations (6))

<xs:simpleType name="[**GroupSelectionBehaviorTypeCore**](#b1236)">

<xs:restriction base="xs:string">

<xs:enumeration value="Any"/>

<xs:enumeration value="All"/>

<xs:enumeration value="AllOrNone"/>

<xs:enumeration value="ExactlyOne"/>

<xs:enumeration value="AtMostOne"/>

<xs:enumeration value="OneOrMore"/>

</xs:restriction>

</xs:simpleType>

simpleType "GroupSelectionBehaviorTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Any" | - | Any number of the items in the group may be chosen, from zero to all. | | "All" | - | All the items in the group must be selected as a single unit. | | "AllOrNone" | - | All the items in the group are meant to be chosen as a single unit: either all must be selected by the end user, or none may be selected. | | "ExactlyOne" | - | The end user must choose one and only one of the selectable items in the group. The user may not choose none of the items in the group. | | "AtMostOne" | - | The end user may choose zero or at most one of the items in the group. | | "OneOrMore" | - | The end user must choose a minimum of one, and as many additional as desired. | |

Known Direct Subtypes (1):

[GroupSelectionBehaviorType](#b1233) []

All Direct / Indirect Based Elements (1):

[value](#b603) (defined in [GroupSelectionBehavior](#b605) complexType) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [GroupSelectionBehaviorTypeCore](#b1236) [] (restriction)  **GroupSelectionBehaviorTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [GroupSelectionBehaviorTypeCore](#b1236) |

XML Source

<xs:simpleType name="[**GroupSelectionBehaviorTypeExt**](#b1239)">

<xs:restriction base="[**GroupSelectionBehaviorTypeCore**](#b1236)"/>

</xs:simpleType>

simpleType "PrecheckBehaviorType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("Yes" | "No") | ("Yes" | "No") |

All Direct / Indirect Based Elements (1):

[value](#b898) (defined in [PrecheckBehavior](#b900) complexType) []

Annotation

Defines selection frequency behavior for an action or group; i.e., for most frequently selected items, the end-user system may provide convenience options in the UI (such as pre-selection) in order to (1) communicate to the end user what the most frequently selected item is, or should, be in a particular context, and (2) save the end user time.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([PrecheckBehaviorTypeCore](#b1245) | [PrecheckBehaviorTypeExt](#b1248))  **PrecheckBehaviorType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [PrecheckBehaviorTypeCore](#b1245)
2. [PrecheckBehaviorTypeExt](#b1248)

XML Source (w/o annotations (1))

<xs:simpleType name="[**PrecheckBehaviorType**](#b1242)">

<xs:union memberTypes="[**PrecheckBehaviorTypeCore**](#b1245) [**PrecheckBehaviorTypeExt**](#b1248)"/>

</xs:simpleType>

simpleType "PrecheckBehaviorTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Yes" | - | An action with this behavior is one of the most frequent actions that is, or should be, included by an end user, for the particular context in which the action occurs. The system displaying the action to the end user should consider "pre-checking" such an action as a convenience for the user. | | "No" | - | An action with this behavior is one of the less frequent actions included by the end user, for the particular context in which the action occurs. The system displaying the actions to the end user would typically not "pre-check" such an action. | |

Known Direct Subtypes (2):

[PrecheckBehaviorType](#b1242) [], [PrecheckBehaviorTypeExt](#b1248) []

All Direct / Indirect Based Elements (1):

[value](#b898) (defined in [PrecheckBehavior](#b900) complexType) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **PrecheckBehaviorTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "Yes" | - | An action with this behavior is one of the most frequent actions that is, or should be, included by an end user, for the particular context in which the action occurs. The system displaying the action to the end user should consider "pre-checking" such an action as a convenience for the user. | | "No" | - | An action with this behavior is one of the less frequent actions included by the end user, for the particular context in which the action occurs. The system displaying the actions to the end user would typically not "pre-check" such an action. | | |

XML Source (w/o annotations (2))

<xs:simpleType name="[**PrecheckBehaviorTypeCore**](#b1245)">

<xs:restriction base="xs:string">

<xs:enumeration value="Yes"/>

<xs:enumeration value="No"/>

</xs:restriction>

</xs:simpleType>

simpleType "PrecheckBehaviorTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Yes" | - | An action with this behavior is one of the most frequent actions that is, or should be, included by an end user, for the particular context in which the action occurs. The system displaying the action to the end user should consider "pre-checking" such an action as a convenience for the user. | | "No" | - | An action with this behavior is one of the less frequent actions included by the end user, for the particular context in which the action occurs. The system displaying the actions to the end user would typically not "pre-check" such an action. | |

Known Direct Subtypes (1):

[PrecheckBehaviorType](#b1242) []

All Direct / Indirect Based Elements (1):

[value](#b898) (defined in [PrecheckBehavior](#b900) complexType) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [PrecheckBehaviorTypeCore](#b1245) [] (restriction)  **PrecheckBehaviorTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [PrecheckBehaviorTypeCore](#b1245) |

XML Source

<xs:simpleType name="[**PrecheckBehaviorTypeExt**](#b1248)">

<xs:restriction base="[**PrecheckBehaviorTypeCore**](#b1245)"/>

</xs:simpleType>

simpleType "RangeConstraintType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("Minimum" | "Maximum" | "List" | "Component") | ("Minimum" | "Maximum" | "List" | "Component") |

All Direct / Indirect Based Elements (1):

[constraintType](#b925) []

Annotation

The enumeration of different types of range constraints on values.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([RangeConstraintTypeCore](#b1254) | [RangeConstraintTypeExt](#b1257))  **RangeConstraintType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [RangeConstraintTypeCore](#b1254)
2. [RangeConstraintTypeExt](#b1257)

XML Source (w/o annotations (1))

<xs:simpleType name="[**RangeConstraintType**](#b1251)">

<xs:union memberTypes="[**RangeConstraintTypeCore**](#b1254) [**RangeConstraintTypeExt**](#b1257)"/>

</xs:simpleType>

simpleType "RangeConstraintTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Minimum" | - | The constraint specifies the lower bound or minimum value of a documentation item value. This constraint type applies to quantitative value types only. | | "Maximum" | - | The constraint specifies the upper bound or minimum value of a documentation item value. This constraint type applies to quantitative value types only. | | "List" | - | The constraint restricts the value to items from a list. | | "Component" | | | |

Known Direct Subtypes (2):

[RangeConstraintType](#b1251) [], [RangeConstraintTypeExt](#b1257) []

All Direct / Indirect Based Elements (1):

[constraintType](#b925) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **RangeConstraintTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "Minimum" | - | The constraint specifies the lower bound or minimum value of a documentation item value. This constraint type applies to quantitative value types only. | | "Maximum" | - | The constraint specifies the upper bound or minimum value of a documentation item value. This constraint type applies to quantitative value types only. | | "List" | - | The constraint restricts the value to items from a list. | | "Component" | | | | |

XML Source (w/o annotations (3))

<xs:simpleType name="[**RangeConstraintTypeCore**](#b1254)">

<xs:restriction base="xs:string">

<xs:enumeration value="Minimum"/>

<xs:enumeration value="Maximum"/>

<xs:enumeration value="List"/>

<xs:enumeration value="Component"/>

</xs:restriction>

</xs:simpleType>

simpleType "RangeConstraintTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Minimum" | - | The constraint specifies the lower bound or minimum value of a documentation item value. This constraint type applies to quantitative value types only. | | "Maximum" | - | The constraint specifies the upper bound or minimum value of a documentation item value. This constraint type applies to quantitative value types only. | | "List" | - | The constraint restricts the value to items from a list. | | "Component" | | | |

Known Direct Subtypes (1):

[RangeConstraintType](#b1251) []

All Direct / Indirect Based Elements (1):

[constraintType](#b925) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [RangeConstraintTypeCore](#b1254) [] (restriction)  **RangeConstraintTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [RangeConstraintTypeCore](#b1254) |

XML Source

<xs:simpleType name="[**RangeConstraintTypeExt**](#b1257)">

<xs:restriction base="[**RangeConstraintTypeCore**](#b1254)"/>

</xs:simpleType>

simpleType "RequestCardinality"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "Single", "Multiple" |

All Direct / Indirect Based Attributes (1):

[*RequestBase*](#b967)/@[cardinality](#b958) []

Annotation

RequestCardinality defines the expected cardinality of the request, single  
or multiple.  
  
If a request results in a cardinality higher than expected, a run-time  
exception should be thrown.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **RequestCardinality** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "Single", "Multiple" | |

XML Source (w/o annotations (1))

<xs:simpleType name="[**RequestCardinality**](#b1260)">

<xs:restriction base="xs:string">

<xs:enumeration value="Single"/>

<xs:enumeration value="Multiple"/>

</xs:restriction>

</xs:simpleType>

simpleType "RequiredBehaviorType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("Must" | "Could" | "MustUnlessDocumented") | ("Must" | "Could" | "MustUnlessDocumented") |

All Direct / Indirect Based Elements (1):

[value](#b970) (defined in [RequiredBehavior](#b972) complexType) []

Annotation

Defines requiredness behavior for selecting an action or an action group; i.e., whether the action or action group is required or optional.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([RequiredBehaviorTypeCore](#b1266) | [RequiredBehaviorTypeExt](#b1269))  **RequiredBehaviorType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [RequiredBehaviorTypeCore](#b1266)
2. [RequiredBehaviorTypeExt](#b1269)

XML Source (w/o annotations (1))

<xs:simpleType name="[**RequiredBehaviorType**](#b1263)">

<xs:union memberTypes="[**RequiredBehaviorTypeCore**](#b1266) [**RequiredBehaviorTypeExt**](#b1269)"/>

</xs:simpleType>

simpleType "RequiredBehaviorTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Must" | - | An action with this behavior must be included in the actions processed by the end user; the end user may not choose not to include this action. | | "Could" | - | An action with this behavior may be included in the set of actions processed by the end user. | | "MustUnlessDocumented" | - | An action with this behavior must be included in the set of actions processed by the end user, unless the end user provides documentation as to why the action was not included. | |

Known Direct Subtypes (2):

[RequiredBehaviorType](#b1263) [], [RequiredBehaviorTypeExt](#b1269) []

All Direct / Indirect Based Elements (1):

[value](#b970) (defined in [RequiredBehavior](#b972) complexType) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **RequiredBehaviorTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | enumeration: | |  |  |  | | --- | --- | --- | | "Must" | - | An action with this behavior must be included in the actions processed by the end user; the end user may not choose not to include this action. | | "Could" | - | An action with this behavior may be included in the set of actions processed by the end user. | | "MustUnlessDocumented" | - | An action with this behavior must be included in the set of actions processed by the end user, unless the end user provides documentation as to why the action was not included. | | |

XML Source (w/o annotations (3))

<xs:simpleType name="[**RequiredBehaviorTypeCore**](#b1266)">

<xs:restriction base="xs:string">

<xs:enumeration value="Must"/>

<xs:enumeration value="Could"/>

<xs:enumeration value="MustUnlessDocumented"/>

</xs:restriction>

</xs:simpleType>

simpleType "RequiredBehaviorTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enumeration: | |  |  |  | | --- | --- | --- | | "Must" | - | An action with this behavior must be included in the actions processed by the end user; the end user may not choose not to include this action. | | "Could" | - | An action with this behavior may be included in the set of actions processed by the end user. | | "MustUnlessDocumented" | - | An action with this behavior must be included in the set of actions processed by the end user, unless the end user provides documentation as to why the action was not included. | |

Known Direct Subtypes (1):

[RequiredBehaviorType](#b1263) []

All Direct / Indirect Based Elements (1):

[value](#b970) (defined in [RequiredBehavior](#b972) complexType) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [RequiredBehaviorTypeCore](#b1266) [] (restriction)  **RequiredBehaviorTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [RequiredBehaviorTypeCore](#b1266) |

XML Source

<xs:simpleType name="[**RequiredBehaviorTypeExt**](#b1269)">

<xs:restriction base="[**RequiredBehaviorTypeCore**](#b1266)"/>

</xs:simpleType>

simpleType "ResourceRelationshipType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("AdaptedFrom" | "AssociatedResource" | "DependsOn" | "DerivedFrom" | "SimilarTo" | "VersionOf") | ("AdaptedFrom" | "AssociatedResource" | "DependsOn" | "DerivedFrom" | "SimilarTo" | "VersionOf") |

All Direct / Indirect Based Attributes (1):

[relationship](#b974)/@value

Annotation

A specific status is associated with each version of an artifact.  
  
See the Implementation Guide for a state-transition diagram showing the legal transitions from each state; each state is equivalent to particular status.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([ResourceRelationshipTypeCore](#b1275) | [ResourceRelationshipTypeExt](#b1278))  **ResourceRelationshipType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [ResourceRelationshipTypeCore](#b1275)
2. [ResourceRelationshipTypeExt](#b1278)

XML Source (w/o annotations (1))

<xs:simpleType name="[**ResourceRelationshipType**](#b1272)">

<xs:union memberTypes="[**ResourceRelationshipTypeCore**](#b1275) [**ResourceRelationshipTypeExt**](#b1278)"/>

</xs:simpleType>

simpleType "ResourceRelationshipTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "AdaptedFrom", "AssociatedResource", "DependsOn", "DerivedFrom", "SimilarTo", "VersionOf" |

Known Direct Subtypes (2):

[ResourceRelationshipType](#b1272) [], [ResourceRelationshipTypeExt](#b1278) []

All Direct / Indirect Based Attributes (1):

[relationship](#b974)/@value

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **ResourceRelationshipTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "AdaptedFrom", "AssociatedResource", "DependsOn", "DerivedFrom", "SimilarTo", "VersionOf" | |

XML Source

<xs:simpleType name="[**ResourceRelationshipTypeCore**](#b1275)">

<xs:restriction base="xs:string">

<xs:enumeration value="AdaptedFrom"/>

<xs:enumeration value="AssociatedResource"/>

<xs:enumeration value="DependsOn"/>

<xs:enumeration value="DerivedFrom"/>

<xs:enumeration value="SimilarTo"/>

<xs:enumeration value="VersionOf"/>

</xs:restriction>

</xs:simpleType>

simpleType "ResourceRelationshipTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "AdaptedFrom", "AssociatedResource", "DependsOn", "DerivedFrom", "SimilarTo", "VersionOf" |

Known Direct Subtypes (1):

[ResourceRelationshipType](#b1272) []

All Direct / Indirect Based Attributes (1):

[relationship](#b974)/@value

Annotation

To add new items to the enumeration ArtifactStatus, comment out or delete the restriction on ArtifactStatusTypeCore.(Optionally, also remove  
the directive above to include the artifactstatustypecore.xsd).  
Now, add a new restriction such as on string and define new enumerated types. See the commented out example below. !!! BKM get example from ArtifactLifeCycleEventTypeExt

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [ResourceRelationshipTypeCore](#b1275) [] (restriction)  **ResourceRelationshipTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [ResourceRelationshipTypeCore](#b1275) |

XML Source (w/o annotations (1))

<xs:simpleType name="[**ResourceRelationshipTypeExt**](#b1278)">

<xs:restriction base="[**ResourceRelationshipTypeCore**](#b1275)"/>

</xs:simpleType>

simpleType "SetOperator"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "I", "U", "D" |

All Direct / Indirect Based Attributes (1):

[SXCM\_TS](#b1058)/@[operator](#b1056) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **SetOperator** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "I", "U", "D" | |

XML Source

<xs:simpleType name="[**SetOperator**](#b1281)">

<xs:restriction base="xs:string">

<xs:enumeration value="I"/>

<xs:enumeration value="U"/>

<xs:enumeration value="D"/>

</xs:restriction>

</xs:simpleType>

simpleType "ValueType"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| ("Boolean" | "Code" | "CodedOrdinal" | "EntityName" | "Identifier" | "Integer" | "IntegerInterval" | "Period" | "PhysicalQuantity" | "PhysicalQuantityInterval" | "QuantityInterval" | "Ratio" | "RatioInterval" | "Real" | "RealInterval" | "SimpleCode" | "String" | "Timestamp" | "TimestampInterval" | "URL") | ("Boolean" | "Code" | "CodedOrdinal" | "EntityName" | "Identifier" | "Integer" | "IntegerInterval" | "Period" | "PhysicalQuantity" | "PhysicalQuantityInterval" | "QuantityInterval" | "Ratio" | "RatioInterval" | "Real" | "RealInterval" | "SimpleCode" | "String" | "Timestamp" | "TimestampInterval" | "URL") |

All Direct / Indirect Based Elements (1):

[responseDataType](#b514) []

Annotation

A specification of a constraint on the range of values for an item.

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  union of ([ValueTypeCore](#b1287) | [ValueTypeExt](#b1290))  **ValueType** |

|  |  |
| --- | --- |
| Derivation: | by union |

Member Types

1. [ValueTypeCore](#b1287)
2. [ValueTypeExt](#b1290)

XML Source (w/o annotations (1))

<xs:simpleType name="[**ValueType**](#b1284)">

<xs:union memberTypes="[**ValueTypeCore**](#b1287) [**ValueTypeExt**](#b1290)"/>

</xs:simpleType>

simpleType "ValueTypeCore"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "Boolean", "Code", "CodedOrdinal", "EntityName", "Identifier", "Integer", "IntegerInterval", "Period", "PhysicalQuantity", "PhysicalQuantityInterval", "QuantityInterval", "Ratio", "RatioInterval", "Real", "RealInterval", "SimpleCode", "String", "Timestamp", "TimestampInterval", "URL" |

Known Direct Subtypes (2):

[ValueType](#b1284) [], [ValueTypeExt](#b1290) []

All Direct / Indirect Based Elements (1):

[responseDataType](#b514) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  **ValueTypeCore** |

|  |  |
| --- | --- |
| Derivation: | restriction of xs:string |
| Facets: | |  |  | | --- | --- | | enumeration: | "Boolean", "Code", "CodedOrdinal", "EntityName", "Identifier", "Integer", "IntegerInterval", "Period", "PhysicalQuantity", "PhysicalQuantityInterval", "QuantityInterval", "Ratio", "RatioInterval", "Real", "RealInterval", "SimpleCode", "String", "Timestamp", "TimestampInterval", "URL" | |

XML Source

<xs:simpleType name="[**ValueTypeCore**](#b1287)">

<xs:restriction base="xs:string">

<xs:enumeration value="Boolean"/>

<xs:enumeration value="Code"/>

<xs:enumeration value="CodedOrdinal"/>

<xs:enumeration value="EntityName"/>

<xs:enumeration value="Identifier"/>

<xs:enumeration value="Integer"/>

<xs:enumeration value="IntegerInterval"/>

<xs:enumeration value="Period"/>

<xs:enumeration value="PhysicalQuantity"/>

<xs:enumeration value="PhysicalQuantityInterval"/>

<xs:enumeration value="QuantityInterval"/>

<xs:enumeration value="Ratio"/>

<xs:enumeration value="RatioInterval"/>

<xs:enumeration value="Real"/>

<xs:enumeration value="RealInterval"/>

<xs:enumeration value="SimpleCode"/>

<xs:enumeration value="String"/>

<xs:enumeration value="Timestamp"/>

<xs:enumeration value="TimestampInterval"/>

<xs:enumeration value="URL"/>

</xs:restriction>

</xs:simpleType>

simpleType "ValueTypeExt"

|  |  |
| --- | --- |
| Namespace: | [urn:hl7-org:v3/hed](#b249) |

|  |
| --- |
| Simple Content Model |
| *enumeration of* xs:string |

Simple Content Restrictions:

|  |  |
| --- | --- |
| Enumeration: | "Boolean", "Code", "CodedOrdinal", "EntityName", "Identifier", "Integer", "IntegerInterval", "Period", "PhysicalQuantity", "PhysicalQuantityInterval", "QuantityInterval", "Ratio", "RatioInterval", "Real", "RealInterval", "SimpleCode", "String", "Timestamp", "TimestampInterval", "URL" |

Known Direct Subtypes (1):

[ValueType](#b1284) []

All Direct / Indirect Based Elements (1):

[responseDataType](#b514) []

Type Definition Detail

|  |
| --- |
| Type Derivation Tree  xs:string (restriction)  [ValueTypeCore](#b1287) [] (restriction)  **ValueTypeExt** |

|  |  |
| --- | --- |
| Derivation: | restriction of [ValueTypeCore](#b1287) |

XML Source

<xs:simpleType name="[**ValueTypeExt**](#b1290)">

<xs:restriction base="[**ValueTypeCore**](#b1287)"/>

</xs:simpleType>

## Schema Datatypes.xsd

Schema "datatypes.xsd"

Target Namespace:

[urn:hl7-org:v3/cdsdt](#b5)

Defined Components:

elements (5 local), complexTypes (25), simpleTypes (17), attribute groups (9)

Default Namespace-Qualified Form:

Local Elements: qualified; Local Attributes: unqualified

Schema Location:

C:\Users\Bryn\Documents\Src\HeD\Source\src\main\schema\complete\datatypes.xsd

Imported by Schemas (1):

[knowledgedocument.xsd](#b1292)

Annotation

Introduction  
------------  
  
This XML document was originally developed in the course of development of  
the ISO/HL7 21090 standard (Healthcare Datatypes). This XML document contains  
either a part or all of the Healthcare Datatypes schema implementation for  
Healthcare Datatypes examples conformant to the Healthcare Datatypes schemas.  
  
ISO/HL7 gives users of Healthcare Datatypes free license to this XML document  
or modifications thereof for use in hardware or software products claiming  
conformance to Healthcare Datatypes.  
  
Those intending to use this XML document in hardware or software products are  
advised that its use may infringe existing patents. The original developers  
of this XML document and his/her company, the subsequent editors and their  
companies, and ISO/HL7 have no liability for use of this XML document or  
modifications thereof in an implementation.  
  
Copyright is not released for non Healthcare Datatypes conforming products.  
The organizations who contributed to this XML document retain the full  
right to use the code for their own purpose, assign or donate their  
contribution to a third party and inhibit third parties from using  
their contribution for non Healthcare Datatypes conforming products.  
  
Copyright (c) 2005-2009 ISO/HL7.  
  
This XML document is provided for informative purposes only. If any parts  
of this XML document contradict the normative part of the corresponding  
standard document then the normative part should be used as the definitive  
specification.  
  
This notice must be included in all copies or derivative works.  
  
Basic Profile for vMR  
---------------------  
  
vMR goals for "Basic" profile:  
Adequately support "Basic vMR" uses including  
enough demographics to enable the structuring of patient-focused output  
support for basic medical concepts in a wide variety of vocabularies  
support for the expression of all common laboratory results  
support for all common coded observations about a patient  
support for common substance administrations  
Remove elements whose use is primarily for human readability  
this includes simplifying ST datatypes to just the value  
Remove elements whose use is primarily for highly-specialized edge cases  
this includes simplifying CD datatype to omit translations, and changing  
its displayName element to be an attribute of xs:string  
  
Extended vMR Profiles are planned for templated extensions to the basic vMR,  
as required. Following those general constraints, we have come up with the  
following list of elements as essential to "Basic" vMR use, expecting them to  
have broad applicability, even in highly specialized uses of CDS:  
• ANY abstract extends HXIT  
• XP abstract extends ANY  
• EntityNamePartQualifier  
• set\_EntityNamePartQualifier  
• EntityNamePartType  
• EntityNameUse  
• set\_EntityNameUse  
• ENXP extends XP  
• EN extends ANY  
• AddressPartType  
• PostalAddressUse  
• set\_PostalAddressUse  
• AD extends ANY  
• ADXP extends XP  
• BL extends ANY  
• CD extends ANY  
• Code  
• CS extends ANY  
• Decimal  
• HXIT abstract  
• II extends ANY  
• INT extends QTY  
• IVL\_INT extends ANY  
• IVL\_PQ extends ANY  
• IVL\_QTY extends ANY  
• IVL\_REAL extends ANY  
• IVL\_TS extends ANY  
• PQ extends QTY  
• QTY abstract extends ANY  
• REAL extends QTY  
• RTO extends QTY  
• ST extends ANY  
• TelecommunicationCapability  
• set\_TelecommunicationCapability  
• TelecommunicationAddressUse  
• set\_TelecommunicationAddressUse  
• TEL extends ANY  
• TS extends QTY  
• Uid extends String  
• UncertaintyType  
• Uri extends String  
  
Profiling this schema  
---------------------  
  
As described in ISO 21090, conformance profiles are able to  
describe subsets of ISO 21090. If a subset schema is derived  
from this schema (by, for instance, deleting types, elements or  
attributes, or by tightening schema or schematron constraints),  
this heading documentation (copyright statement, this paragraph,  
and the version history) must be reproduced as is, along with a  
a section above this one explaining who prepared the revised  
schema, what conformance profile it conforms to, and where the  
conformance profile may be found.  
  
Version History  
---------------  
Version 0.03 20110113 Technical Correction (Lloyd McKenzie)  
Changes from 0.02:  
Corrected schematron to reflect new names of controlInformationRoot and controlInformationExtension  
  
Version 0.02 20090526 Final Ballot Candidate (Grahame Grieve)  
Changes from 0.01:  
Add documentation for each element  
Remove flavors as schema types  
HXIT: Renamed controlActRoot and controlActExtension to controlInformationRoot and controlInformationExtension  
ANY: made flavorId a set instead of a singleton  
ENXP/ADXP: Introduce XP type and move attributes there  
EN: Reorganise EN.use, ENXP.partType and ENXP.qualifier enumerations  
TEL: add capabilities attribute  
SD.TEXT / SD.TITLE: new types defined  
CD: change codingRationale contents  
II: correct one enumeration value on reliability  
REAL/MO: remove precision attribute  
PQ: remove PQV and redefine PQ and PQR accordingly  
QSC: change code from enumerated attribute to CD  
  
Version 0.01 20080317 First Release (Grahame Grieve)

## Schema KnowledgeDocument.xsd

Schema "knowledgedocument.xsd"

Target Namespace:

[urn:hl7-org:v3/hed](#b249)

Defined Components:

elements (2 global + 212 local), complexTypes (168), simpleTypes (54)

Default Namespace-Qualified Form:

Local Elements: qualified; Local Attributes: unqualified

Schema Location:

C:\Users\Bryn\Documents\Src\HeD\Source\src\main\schema\complete\knowledgedocument.xsd

Imports Schemas (1):

[datatypes.xsd](#b1291)

Annotation

This file allows organizations to extend the enumeration ArtifactLifeCycleEventType.

## Namespace Bindings

Namespace Bindings

|  |  |
| --- | --- |
| **Prefix** | **Namespace URI / Binding Location** |
| - | [urn:hl7-org:v3/cdsdt](#b5)   |  |  | | --- | --- | | File: | [datatypes.xsd](#b1291) | | Element: | <xs:schema ... > | |
| - | [urn:hl7-org:v3/hed](#b249)   |  |  | | --- | --- | | File: | [knowledgedocument.xsd](#b1292) | | Element: | <xs:schema ... > | |
| altova | http://www.altova.com/xml-schema-extensions   |  |  | | --- | --- | | File: | [datatypes.xsd](#b1291) | | Element: | <xs:schema ... > | |
| dt | [urn:hl7-org:v3/cdsdt](#b5)   |  |  | | --- | --- | | File: | [knowledgedocument.xsd](#b1292) | | Element: | <xs:schema ... > | |
| xs | http://www.w3.org/2001/XMLSchema   |  |  | | --- | --- | | File: | [datatypes.xsd](#b1291) | | Element: | <xs:schema ... > | |
| xs | http://www.w3.org/2001/XMLSchema   |  |  | | --- | --- | | File: | [knowledgedocument.xsd](#b1292) | | Element: | <xs:schema ... > | |

# Appendix A – Referenced documents

The reference documents specific to the CDS Knowledge Artifact are presented in the table below. This implementation guide is built based on the premise of reusing as much existing material as possible, and as such, the list of references is fairly extensive, and may be expanded upon as the implementation guide is developed and balloted.

| Reference Name | Location | How this reference was used in development of the CDS Knowledge Artifact Implementation Guide |
| --- | --- | --- |
| S&I Framework Health eDecisions - CDS Artifact Sharing Use Case | <http://wiki.siframework.org/file/view/SIFramework_HeD_UC1_CDSArtifactSharing_v1.0.docx> | The use case serves as the baseline for all functional and system requirements associated with the CDS knowledge artifact, and provides scoping for the implementation guide |
| The Arden Syntax for Medical Logic Systems Version 2.7 | <http://www.hl7.org/documentcenter/private/standards/Arden/v27/Arden%20Syntax%202.7_PDF.zip> | The Arden Syntax Version 2.8 is used as the primary reference source for the Expression section of this document. It also provides additional reference material on datatypes associated with the CDS Knowledge Artifact. |
| HL7 Version 3 Domain Analysis Model: Virtual Medical Record for Clinical Decision Support (vMR-CDS), Release 1 | <http://www.hl7.org/documentcenter/private/standards/v3/V3DAM_CDS_VMR_R1_INFORM_2012APR.zip> | The Domain Analysis Model (DAM) for the Virtual Medical Record (vMR) serves as the primary resource for the structure of the CDS Knowledge Artifact. |
| HL7 Version 3 Implementation Guide: Virtual Medical Record for Clinical Decision Support (vMR-CDS) for GELLO, Release 1 Draft Standard for Trial Use | <http://www.hl7.org/documentcenter/public/standards/dstu/V3IG_CDS_VMR_GELLO_DSTU_R1_2012APR.pdf> | GELLO is included in this list of references due to the structure of the vMR/GELLO implementation guide that was balloted by HL7. The HeD initiative wishes to emulate several elements of that structure to assist implementers of the CDS Knowledge Artifact |
| Guidelines Element Model (GEM) | <http://gem.med.yale.edu/default.htm> | The Guidelines Element Model (GEM) is included within this implementation guide as one of the schemas that has been harmonized within the new CDS Knowledge Artifact schema. |
| Digital Infrastructure for the Learning Health System: The Foundation for Continuous Improvement in Health and Health Care: Workshop Series Summary. Institute of Medicine. | This PDF is available from The National Academies Press at:  <http://www.nap.edu/catalog.php?record_id=12912> www,nap.edu October 2011. ISBN 978-0-309-15416-1" | Cited as reference for learning health system in Executive Summary |
| AHRQ eRecs – Structuring Care Recommendations for Clinical Decision Support | <http://images.ahrq.gov/publishedimages/communities/a_e/ahrq_funded_projects/projects/calendaryearupdateshtmlpages/2011_2900900022i2_osheroff_pdf_3.pdf> | The AHRQ eRecs specification will be harmonized to facilitate representation of data elements and logical expressions in a structured, codified format, enabling further local processing into CDS rules. |
| Clinical Decision Support Consortium Level 3 XML examples | <http://cdsportal.partners.org/CDSCSearch.aspx> | As part of promoting greater CDS adoption with a wider community of CDS vendors, the Harmonized HeD schema also maps to the CDSC L3 schema. While this schema is not considered a formally balloted standard through a Standards Development Organization (SDO), it nevertheless contains valuable best practices and implementation experience from the field. L3 serves as the "container" for the harmonized schema. |
| Health eDecisions Value Sets and Terminology Implementation Guide (HeD VST IG) | [https://docs.google.com/a/esacinc.com/document/d/1JdjRxR2rQRioBXXxxiw9jpl6ZP5c2pDEbdW8xq5xjEc/edit#](https://docs.google.com/a/esacinc.com/document/d/1JdjRxR2rQRioBXXxxiw9jpl6ZP5c2pDEbdW8xq5xjEc/edit) | During the Harmonization phase of HeD, the Value Sets and Terminoloy sub-work group (VST SWG) was launched. This sub-work group’s objective was to identify value sets or coding systems (terminologies) and map these to data elements used by HeD CDS artifacts. This implementation guide provides guiding principles for aligning existing value sets or coding systems to data elements in CDS artifacts. |

Table 168 – Appendix A - List of CDS References for Implementers

# appendix b - acronyms

The following acronyms are referenced in this implementation guide:

| Acronym | Definition/Description |
| --- | --- |
| CDS | Clinical Decision Support |
| CDSC L3 | Clinical Decision Support Consortium Level 3 |
| CREF | Allscripts Common Rule Engine Format (CREF) specification |
| DAM | Domain Analysis Model |
| EHR | Electronic Health Record |
| EMR | Electronic Medical Record |
| eRecs | AHRQ Electronic Recommendations |
| GEM | Guidelines Element Model |
| HIE | Health Information Exchange |
| HIT | Health Information Technology |
| HITECH Act | Health Information Technology for Economic and Clinical Health Act |
| HIPAA | Health Insurance Portability and Accountability Act |
| HeD | Health eDecisions |
| HITSP | Health Information Technology Standards Panel |
| HL7 | Health Level 7 |
| MU | Meaningful Use |
| ONC | Office of the National Coordinator for Health IT |
| S&I Framework | Standards & Interoperability Framework |
| vMR | HL7 Virtual Medical Record |
| XML | Extensible Markup Language |
| XSD | XML Schema |

Table 169 – Appendix B - List of Acronyms used in this Guide

# Appendix C – Definitions

Because the CDS Knowledge Artifact introduces new concepts and terms, an additional appendix of definitions is provided to support implementer understanding of terms that may be used in various sections of this document, specifically documentation and examples in Sections 4-6. This appendix DOES NOT include terms and definitions already provided in the S&I Framework HeD CDS Artifact Sharing Use Case

| Term | Definition/Description |
| --- | --- |
| Abstract | An abstract element or complex type cannot used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used. |
| Complex Type | A complex type is an XML element that contains other elements and/or attributes. |
| Enumeration | Used to limit an element to a set of constrained values – enumerations are used in the CDS Knowledge Artifact to define “value sets” |
| Simple Type | A simple type contains constraints and information about the values of attributes or text-only elements. |
| Value Set | A uniquely identifiable set of valid concept identifiers, where any concept identifier in a coded element can be tested to determine whether it is a member of the Value Set. |

Table 170 - Appendix C - List of Definitions

# Appendix D – Language Implementation

This appendix contains more detailed information relating to the intended semantics of the HeD Schema expression language. These topics are specifically relevant for readers interested in building translation, semantic validation, or evaluation applications for the HeD Schema expression language.

## Clinical Data Retrieval in HeDS Artifacts

This section discusses the problem of clinical data retrieval in the clinical decision support space in general, and how the problem is resolved in the HeDS specification.

### Defining Clinical Data

The problem of determining what data needs to be involved in the evaluation of any given artifact if that artifact contains arbitrary queries against the data model, is equivalent to the problem of query containment from database theory. This problem is known to be undecidable for arbitrary queries of the relational algebra, but is also shown to be both decidable and equivalent to the problem of query evaluation for the restricted class of conjunctive queries (Foundations of Databases, Abiteboul, Hull, Vianu).

In the Clinical Decision Support space, this problem is further complicated by the problem of terminology mapping. The meaning of a particular clinical statement within a patient’s data is represented with a vocabulary consisting of codes which determine the kind of statement being represented. For example, a diagnosis clinical statement may be classified using the ICD-9 vocabulary, identifying the specific diagnosis represented.

In order for Clinical Decision Support artifacts to operate correctly, the meaning of each clinical statement, as identified by the vocabularies involved, must be preserved. However, this meaning is often represented in different vocabularies in different systems. A mapping between the vocabularies is therefore required in order to facilitate expression and evaluation of the artifact.

In addition, patient data is represented in differing schemas across various patient data sources, and must therefore be mapped structurally into the patient data model used by an artifact.

These problems collectively constitute what is referred to as the “curly braces problem” in the Arden space. This problem arises because of the difficulty in defining the structural and semantic aspects of the data involved.

The solution to this problem proposed by the HeDS specification is to create a well-defined and relatively simple interface between the clinical data provided by patient data sources, and the usage of that data within the artifact.

First, all clinical data within a HeDS knowledge artifact is represented using the HL7 Virtual Medical Record (VMR). This allows content to be authored without regard to the specific data models used by various patient data sources.

Second, all references to clinical data within a HeDS knowledge artifact are represented using a specific type of expression that only allows a well-defined set of clinically relevant criteria to be used to reference the data. The purpose of this restriction is two-fold: First, it allows the data required for evaluation to be determined solely by inspection of the artifact. And second, it allows for easy and reliable implementation of the interface between the evaluation engine and the patient data source, because the criteria used to request information from the patient data source are simple and well-defined.

Third, by using standard terminologies within this data interface, the HeDS specification can guarantee that any given clinical statement referenced in an artifact has the same meaning as the data that is provided to the artifact from the patient data source. At a high level, this is the terminology problem; ensuring that the vocabularies used within the artifact are accurately mapped to the vocabularies used by the patient data source.

These three motivating factors inform the design of the Request expression used within the HeDS specification.

### Conformance Levels

Although HeDS uses the VMR, there are many possibilities for variance in the way that VMR data is provided. This problem leads to the potential for artifacts to reference properties within the model that may or not be provided within a given specific instance of patient data expressed in the VMR schema. To address this potential problem, the request expressions within a HeDS artifact specify not only the type of the data (meaning the specific model type being requested), but optionally a template identifier that further constrains the data that is expected in a given request. If a template identifier is provided, then the request expression is expected to return only data that matches the constraints in the given template.

To help communicate validity of an artifact for a specific use, the HeDS specification defines two conformance levels related to this use of templates:

#### Strict Conformance

A HeDS artifact can be said to be strictly conforming if all references to clinical statement model properties (elements and attributes of VMR model types) within the artifact are explicitly constrained by the templates used in the requests.

#### Loose Conformance

A HeDS artifact can be said to be loosely conforming if the artifact references properties that are not explicitly constrained by the templates used in the requests. This is not to say that the artifact is necessarily invalid, just that the structure of the clinical data provided to the request may or may not contain the elements referenced by properties within the artifact.

### Artifact Data Requirements

Because of the way data access is modeled within HeDS, the data requirements of a particular artifact can be clearly and accurately defined by inspecting only the ClinicalRequest expressions defined within the artifact. The following table broadly describes the data defined by each Request:

|  |  |
| --- | --- |
| Item | Description |
| Clinical Data Type | The type of clinical data to be retrieved. |
| Codes | The set of codes defining the clinical data. Only clinical data matching codes in the set will be retrieved. If no codes are specified, clinical data with any code will be retrieved. |
| Date Range | The date range for clinical data. Only data within the specified date range will be retrieved. If no date range is specified, clinical data of any date will be retrieved. |
| Timing | The IsInitial attribute specifies whether or not the request is part of the initial data requirements for the artifact. |
| Triggering | If specified, this indicates whether the request is defining the triggering criteria for the artifact. |

These criteria are designed to allow the implementation environment to communicate the data requirements for an artifact, or group of artifacts, to a consumer to allow the consumer to gather all and only the relevant clinical information for transport to the evaluation environment. This supports the near-real-time clinical decision support scenario where the evaluation environment is potentially separate from the medical records system environment.

To support further reducing the overall size of data required to be transported, the following steps can be taken to combine request descriptors that deal with the same type of clinical data.

First, create a request context for each unique type of request using the request data type and the triggering context for each initial request.

Next, for each request, add the codes to the matching request context (by data type and triggering context), recording the associated date range, if any, for each code. Note that the empty set of codes should be represented as the single code “ALL” for the purposes of this method. As date ranges are recorded, they must be merged so that for each code in each request context, no two date range intervals overlap or meet.

Once the date ranges for each code within each unique request context are determined, the unique set of date ranges for all codes is calculated, accumulating the set of associated codes. Each unique date range for the context then results in a final descriptor.

This process produces a set of clinical data descriptors with the following structure:

|  |  |
| --- | --- |
| Property | Description |
| Clinical Data Type | The type of clinical data required. |
| Triggering Context | The triggering context, if any. |
| Codes | The set of applicable codes, possibly empty (meaning all codes). |
| Date Range | The applicable date range, possibly empty (meaning all dates). |

Collectively, these descriptors then represent the minimum initial data requirements for the artifact, with any overlapping requests for the same type of data collapsed into a single request descriptor.

In addition to being used to describe the initial data requirements, this same process can be used to collapse additional data requests that are evaluated as part of further evaluation of the artifact.

## Expression Language Conceptual Model

In order to completely specify the semantics of the expression logic defined within HeDS, the intended execution model for expressions must be clearly defined. The following sections discuss the conceptual components of the expression language, and how these components are defined to operate.

### Data Model

The data model for the HeDS expression language provides the overall structure and definition for the types of operations and capabilities that can be represented within the language. Note that the schema itself is layered into a core expression schema, and a more specific, clinical expression schema. The expression schema deals with defining the core operations that are available without respect to any specific model. The clinical expression schema then extends those operations to include references to clinical data.

Note that although the expression language deals with various categories of types, these are only conceptually defined within the expression language schema. There is no expectation within the core expression language that any particular data model be used, only that whatever concrete data model is actually used can be concretely mapped to the type categories defined within HeDS. Because these type categories are extremely broad, this allows the HeDS expression language component to be used with a large class of concrete data models without modifying the underlying specification.

#### Values

A *value* within the HeDS expression language represents some piece of data. All values are of some *type*, which designates what operations can be performed on the value. There are four categories of types within the HeDS expression language:

1. Scalar types – Types representing simple values such as strings, integers, dates, and decimals.
2. Structured types – Types representing composite values consisting of sets of named properties, each of which has a declared type, and may or may not have a current value of that type.
3. Collection types – Types representing lists of values of some declared type.
4. Interval types – Types representing an interval of some declared type, called the *point* type.

#### Scalar Types

Scalar types allow for the representation of simple, atomic types, such as integers and strings. For example, the value **5** is a value of type *Integer*, meaning that it can be used in operations that require integer-valued input such as addition or comparison.

The expression language itself does not define any scalar types. The clinical expression layer introduces expressions for dealing with the subset of ISO 21090 data types defined as part of the HeDS specification. These are the same types used by the VMR, so literal expressions are defined at the clinical expression layer to facilitate expressing values of these data types.

#### Structured Types

Structured types allow for the representation of composite types. Typically, these types correspond to the model types defined in the clinical data model used for the artifact. Structured types are defined as containing a set of named properties, each of which are of some type, and may have a value of that type.

As with scalar types, the core expression layer does not define any structured types, it only provides facilities for constructing values of structured types and for operating on structured values.

#### Collection Types

Collection types allow for the representation of lists and sets of values of any type. All the values within a collection are expected to be of the same type.

Collections may be empty, and are defined to be 1-based for indexing purposes.

#### Interval Types

Interval types allow for the representation of intervals of some type. For example, an interval of integers allows the expression of the interval 1 to 5. Intervals can be open or closed at the beginning and/or end of the interval, and the beginning or end of the interval can be unspecified.

The core expression layer does not define any interval types, it only provides facilities for constructing values of interval types, and for operating on intervals.

### Language Elements

The expression language specified as part of HeDS is defined as an Abstract Syntax Tree. Whereas a traditional language would have syntax and require lexical analysis and parsing, the HeDS specification allows expressions to be represented directly as trees. This removes potential ambiguities such as operator order precedence, and makes analysis and processing of the expressions in the language much easier.

Concretely, this is accomplished by defining the language elements as types in an XML schema. Each language element is represented by a type in the XML schema. For example, the following element represents an integer literal expression:

<expression xsi:type="Literal" valueType="xs:int" value="6"/>

Arguments to operations are represented naturally using the hierarchical structure of the XML document. For example, the following fragment represents an expression for adding the integer values 2 and 2:

<expression xsi:type="Add">

<operand xsi:type="IntegerLiteral" value="2"/>

<operand xsi:type="IntegerLiteral" value="2"/>

</expression>

This structure allows expressions of arbitrary complexity to be built up using the language elements defined in the schema. Essentially, the language consists of only two kinds of elements: 1) Expressions, and 2) Expression Definitions.

Each expression returns a value of some type, and an expression definition allows a given expression to be defined with an identifier so that it can be referenced in other expressions.

These expressions and expression definitions are then used throughout the HeDS specification wherever logic needs to be applied within an artifact.

### Semantic Validation

Semantic Validation of an expression within the HeDS expression language is the process of verifying that the meaning of the expression is valid. This involves determining the type of each expression, and verifying that the arguments to each operation have the correct type.

This process proceeds as follows:

The graph of the expression being validated is traversed and the type of each node is determined. If the node has children (operands) the type of each child is determined in order to determine the type of the node. The following table defines the categories of nodes and the process for determining the type of each category:

|  |  |
| --- | --- |
| Node Category | Type Determination |
| Literal | The type of the node is the type of the literal being represented. |
| Property | The type of the node is the declared type of the property being referenced. |
| ParameterRef | The type of the node is the parameterType of the parameter being referenced. |
| ExpressionRef | The type of the node is the type of the expression being referenced. |
| ClinicalRequest | The type of the node is a list of the type of the data being requested. |
| ValueSet | The type of the node is a list of codes. |
| Operator | Generally, the type of the node is determined by resolving the type of each operand, and then using that signature to determine the resulting type of the operator. |

During validation, the implementation must maintain a stack of symbols that track the type of the object currently in scope. This allows the type of context-sensitive operators such as Current and Property to be determined. Refer to the Execution Model section for a description of the evaluation-time stack.

Details for the specifics of type determination for each operator are provided with the documentation for those operators.

### Execution Model

All logic in the HeDS expression language is represented as *expressions*. The language is pure functional, meaning no operations are allowed to have side effects of any kind. An expression may consist of any number of other expressions and operations, so long as they are all combined according to the semantic rules for each operation as described in the Semantic Validation section.

Because the language is pure functional, every expression and operator is defined to return the same value on every evaluation within the same artifact evaluation. In particular this means:

1. All clinical data returned by request expressions within the artifact must return the same set on every evaluation. An implementation would likely use a snapshot of the required clinical data in order to achieve this behavior.
2. Invocations of non-deterministic operations such as Now() and Today() are defined to return the timestamp associated with the evaluation request, rather than the clock of the engine performing the evaluation.

Once an expression has been semantically validated, its return type is known. This means that the expression is guaranteed to return either a value of that type, or a *null*, indicating the evaluation did not result in a value.

In general, operations are defined to result in null if any of their arguments are null. For example, the result of evaluating 2 + null is null. In this way, missing information results in an unknown result. There are exceptions to this rule, notably the logical operators, and the null-handling operators. The behavior for these operators (and others that do not follow this rule) are described in detail in the documentation for each operator.

Evaluation takes place within an execution model that provides access to the data and parameters provided to the evaluation. Data is provided to the evaluation as a set of lists of structured values representing a patient’s clinical information. In order to be represented in this data set, a given structured value must be a *cacheable* item. A cacheable item must have the following:

|  |  |
| --- | --- |
| Property | Description |
| Identifier | A property, or set of properties, that uniquely identify the item. |
| Codes | A code, or list of codes that identify the associated clinical codes for the item. |
| Date | A date time defining the clinically relevant date and/or time of the item. |

Evaluation consists of two phases, a *pre-processing* phase, and an *evaluation* phase. The pre-processing phase is used to determine the initial data requirements for a rule. During this phase any request expressions in the rule are analyzed to determine what data must be provided to the evaluation in order to successfully complete a rule evaluation. All requests with IsInitial set to true, as well as any trigger requests, are considered and a set of data descriptors is produced using the method described in the Artifact Data Requirements section. This means in particular that the Codes and DateRange expressions in each of the initial and triggering requests must be compile-time evaluable. This means that these expressions may not reference any clinical information, though they are allowed to reference parameter values.

During the evaluation phase, the result of the expression is determined. Conceptually, evaluation proceeds as follows:

The graph of the expression being evaluated is traversed and the result of each node is calculated. If the node has children (operands), the result of each child is evaluated before the result of the node can be determined. The following table describes the general categories of nodes and the process of evaluation for each:

|  |  |
| --- | --- |
| Node Category | Evaluation |
| Literal | The result of the node is the value of the literal represented. |
| Operation | The result of the node is the result of the operation described by the node given the results of the operand nodes of the expression. |
| ClinicalRequest | The result of the node is the result of retrieving the data represented by the request, i.e. a list of structured values of the type defined in the request representing the patient information being retrieved. |
| ExpressionRef | The result of the node is the result of evaluating the referenced expression. |
| ParameterRef | The result of the node is the value of the referenced parameter. |

During evaluation, the implementation must maintain a stack that is used to represent the value that is currently in context. Certain operations within the expression language are defined with a scope, and these operations use the stack to represent this scope. The following table details these operations:

|  |  |
| --- | --- |
| Operation | Stack Effect |
| ObjectRedefine | The *source* operand is pushed on to the stack prior to evaluating the *property* expressions. The stack is popped before the result is returned. |
| Filter | For each item in the *source* operand, the item is pushed on to the stack, the *condition* expression is evaluated, and the item is popped off of the stack. |
| ForEach | For each item in the *source* operand, the item is pushed on to the stack, the *element* expression is evaluated, and the item is popped off of the stack. |

The *scope* attribute of these operators provides an optional name for the item being pushed on to the stack. This name can be used within the **Current** and **Property** expressions to determine which element on the stack is being accessed. If no scope is provided, the top of the stack is assumed.

Details for the evaluation behavior of each specific operator are provided as part of the documentation for each operator.

# ****Appendix E – Examples****

## FLACC Example

1. <?xml version="1.0" encoding="UTF-8"?>
2. <?schematron-schema href="../main/schematron/knowledgeartifact.sch"?>
3. <?schematron-schema href="../main/schematron/documentationtemplates.sch"?>
4. <knowledgeDocument xmlns="urn:hl7-org:v3/hed"
5. xmlns:vmr="org.opencds.vmr.v1\_0.schema.vmr"
6. xmlns:dt="urn:hl7-org:v3/cdsdt" xmlns:p1="http://www.w3.org/1999/xhtml"
7. xmlns:xml="http://www.w3.org/XML/1998/namespace" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8. xsi:schemaLocation="urn:hl7-org:v3/hed ../main/schema/knowledgedocument.xsd ">
9. <!-- FLACC Pain Scale Documentation Template This example describes the FLACC pain scale as a documentation template. The example was chosen to illustrate the use of a documentation template to describe a typical practitioner administered patient evaluation instrument. -->
10. <metadata>
11. <identifiers>
12. <identifier root="FLACCExampleForHeD" version="2" />
13. </identifiers>
14. <artifactType value="Documentation Template" />
15. <schemaIdentifier root="urn:hl7-org:v3/hed" version="1.0" />
16. <dataModels>
17. <modelReference>
18. <description value="Virtual Medical Record model" />
19. <referencedModel value="org.opencds.vmr.v1\_0.schema.vmr"/>
20. </modelReference>
21. </dataModels>
22. <title value="FLACC Pain Scale" />
23. <relatedResources>
24. <relatedResource>
25. <relationship value="DerivedFrom" />
26. <resources>
27. <resource>
28. <title value="FLACC Scale" />
29. <location
30. value="http://painconsortium.nih.gov/pain\_scales/FLACCScale.pdf" />
31. <description value="NIH Pain Intnesity Instruments" />
32. </resource>
33. </resources>
34. </relatedResource>
35. </relatedResources>
36. <supportingEvidence>
37. <evidence>
38. <resources>
39. <resource>
40. <citation
41. value="Merkel, SI, Voepel-Lewis, T., Shayevitz, JR, &amp; Malviya, S. (1997). The FLACC: a behavioral
42. scale for scoring postoperative pain in young children. Pediatric Nursing, 23(3): 293-297" />
43. </resource>
44. </resources>
45. </evidence>
46. </supportingEvidence>
47. <applicability>
48. <coverage>
49. <focus value="PatientAgeGroup" />
50. <description value="Population between the age of 2 months to 7 years" />
51. <value code="D007223" codeSystem="2.16.840.1.113883.11.75"
52. codeSystemName="MeSH - AgeGroupObservationValue" displayName="Infant; 1 to 23 months"/>
53. </coverage>
54. <coverage>
55. <focus value="PatientAgeGroup" />
56. <description value="Population between the age of 2 months to 7 years" />
57. <value code="D002675" codeSystem="2.16.840.1.113883.11.75"
58. codeSystemName="MeSH - AgeGroupObservationValue" displayName="child, preschool; 2 to 5 years" />
59. </coverage>
60. <coverage>
61. <focus value="PatientAgeGroup" />
62. <description value="Population between the age of 2 months to 7 years" />
63. <value code="D002648" codeSystem="2.16.840.1.113883.11.75"
64. codeSystemName="MeSH - AgeGroupObservationValue" displayName="child; 6 to 12 years"/>
65. </coverage>
66. </applicability>
67. <status value="Draft" />
68. <contributions>
69. <contribution>
70. <contributor xsi:type="Person">
71. <contacts>
72. <contact value="mailto:aziz.boxwala@meliorix.com" />
73. </contacts>
74. <name use="C">
75. <dt:part value="Aziz" type="GIV" />
76. <dt:part value="Boxwala" type="FAM" />
77. </name>
78. <affiliation>
79. <name value="Meliorix Inc." />
80. </affiliation>
81. </contributor>
82. <role value="Author" />
83. </contribution>
84. </contributions>
85. </metadata>
86. <externalData>
87. <def name="Patient">
88. <expression xsi:type="ClinicalRequest" cardinality="Single"
89. dataType="vmr:EvaluatedPerson" isInitial="true" />
90. </def>
91. </externalData>
92. <expressions>
93. <def name="PatientAge">
94. <expression xsi:type="Property" path="demographics.age">
95. <source xsi:type="ExpressionRef" name="Patient" />
96. </expression>
97. </def>
98. </expressions>
99. <conditions>
100. <condition>
101. <logic xsi:type="And">
102. <description>Patient is between 2 months and 7 years of age</description>
103. <operand xsi:type="GreaterOrEqual">
104. <description>Patient is more than 2 months old</description>
105. <operand xsi:type="ExpressionRef" name="PatientAge"></operand>
106. <!-- NOTE: A physical quantity with units of months is used here. The
107. intended semantics are that the implementation engine will perform any conversions
108. required to perform the comparison. -->
109. <operand xsi:type="PhysicalQuantityLiteral" value="2" unit="mo" />
110. </operand>
111. <operand xsi:type="LessOrEqual">
112. <description>Patient is less than 7 years old</description>
113. <operand xsi:type="ExpressionRef" name="PatientAge"></operand>
114. <operand xsi:type="PhysicalQuantityLiteral" value="7" unit="a" />
115. </operand>
116. </logic>
117. <conditionRole>ApplicableScenario</conditionRole>
118. </condition>
119. </conditions>
120. <actionGroup>
121. <title value="FLACC Scale" />
122. <representedConcepts>
123. <concept code="38213-5" displayName="FLACC pain assessment panel:-:Pt:^Patient:-"
124. codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" />
125. </representedConcepts>
126. <subElements>
127. <!-- Define the container to store responses to the documentation items. -->
128. <simpleAction xsi:type="DeclareResponseAction" />
129. <simpleAction xsi:type="CollectInformationAction">
130. <documentationConcept>
131. <displayText value="Face" />
132. <itemCodes>
133. <itemCode code="38216-8" codeSystem="2.16.840.1.113883.6.1"
134. codeSystemName="LOINC" />
135. </itemCodes>
136. <responseDataType>Integer</responseDataType>
137. <responseCardinality>Single</responseCardinality>
138. <responseRange xsi:type="ListConstraint"
139. strictSelection="true">
140. <constraintType>List</constraintType>
141. <item>
142. <value xsi:type="IntegerLiteral" value="0" />
143. <displayText value="No particular expression or smile" />
144. </item>
145. <item>
146. <value xsi:type="IntegerLiteral" value="1" />
147. <displayText
148. value="Occasional grimace or frown, withdrawn, disinterested" />
149. </item>
150. <item>
151. <value xsi:type="IntegerLiteral" value="2" />
152. <displayText value="Frequent to constant quivering chin, clenched jaw" />
153. </item>
154. </responseRange>
155. </documentationConcept>
156. <responseBinding property="Face" />
157. </simpleAction>
158. <simpleAction xsi:type="CollectInformationAction">
159. <documentationConcept>
160. <displayText value="Legs" />
161. <itemCodes>
162. <itemCode code="38217-6" codeSystem="2.16.840.1.113883.6.1"
163. codeSystemName="LOINC" />
164. </itemCodes>
165. <responseDataType>Integer</responseDataType>
166. <responseCardinality>Single</responseCardinality>
167. <responseRange xsi:type="ListConstraint"
168. strictSelection="true">
169. <constraintType>List</constraintType>
170. <item>
171. <value xsi:type="IntegerLiteral" value="0" />
172. <displayText value="Normal position or relaxed" />
173. </item>
174. <item>
175. <value xsi:type="IntegerLiteral" value="1" />
176. <displayText value="Uneasy, restless, tense" />
177. </item>
178. <item>
179. <value xsi:type="IntegerLiteral" value="2" />
180. <displayText value="Kicking, or legs drawn up" />
181. </item>
182. </responseRange>
183. </documentationConcept>
184. <responseBinding property="Legs" />
185. </simpleAction>
186. <simpleAction xsi:type="CollectInformationAction">
187. <documentationConcept>
188. <displayText value="Activity" />
189. <itemCodes>
190. <itemCode code="38218-4" codeSystem="2.16.840.1.113883.6.1"
191. codeSystemName="LOINC" />
192. </itemCodes>
193. <responseDataType>Integer</responseDataType>
194. <responseCardinality>Single</responseCardinality>
195. <responseRange xsi:type="ListConstraint"
196. strictSelection="true">
197. <constraintType>List</constraintType>
198. <item>
199. <value xsi:type="IntegerLiteral" value="0" />
200. <displayText value="Lying quietly, normal position, moves easily" />
201. </item>
202. <item>
203. <value xsi:type="IntegerLiteral" value="1" />
204. <displayText value="Squirming, shifting back and forth, tense" />
205. </item>
206. <item>
207. <value xsi:type="IntegerLiteral" value="2" />
208. <displayText value="Arched, rigid or jerking" />
209. </item>
210. </responseRange>
211. </documentationConcept>
212. <responseBinding property="Activity" />
213. </simpleAction>
214. <simpleAction xsi:type="CollectInformationAction">
215. <documentationConcept>
216. <displayText value="Cry" />
217. <itemCodes>
218. <itemCode code="38219-2" codeSystem="2.16.840.1.113883.6.1"
219. codeSystemName="LOINC" />
220. </itemCodes>
221. <responseDataType>Integer</responseDataType>
222. <responseCardinality>Single</responseCardinality>
223. <responseRange xsi:type="ListConstraint"
224. strictSelection="true">
225. <constraintType>List</constraintType>
226. <item>
227. <value xsi:type="IntegerLiteral" value="0" />
228. <displayText value="No cry (awake or asleep)" />
229. </item>
230. <item>
231. <value xsi:type="IntegerLiteral" value="1" />
232. <displayText value="Moans or whimpers; occasional complaint" />
233. </item>
234. <item>
235. <value xsi:type="IntegerLiteral" value="2" />
236. <displayText
237. value="Crying steadily, screams or sobs, frequent complaints" />
238. </item>
239. </responseRange>
240. </documentationConcept>
241. <responseBinding property="Cry" />
242. </simpleAction>
243. <simpleAction xsi:type="CollectInformationAction">
244. <documentationConcept>
245. <displayText value="Consolability" />
246. <itemCodes>
247. <itemCode code="38220-0" codeSystem="2.16.840.1.113883.6.1"
248. codeSystemName="LOINC" />
249. </itemCodes>
250. <responseDataType>Integer</responseDataType>
251. <responseCardinality>Single</responseCardinality>
252. <responseRange xsi:type="ListConstraint"
253. strictSelection="true">
254. <constraintType>List</constraintType>
255. <item>
256. <value xsi:type="IntegerLiteral" value="0" />
257. <displayText value="Content, relaxed" />
258. </item>
259. <item>
260. <value xsi:type="IntegerLiteral" value="1" />
261. <displayText
262. value="Reassured by occasional touching, hugging or being talked to, distractible" />
263. </item>
264. <item>
265. <value xsi:type="IntegerLiteral" value="2" />
266. <displayText value="Difficult to console or comfort" />
267. </item>
268. </responseRange>
269. </documentationConcept>
270. <responseBinding property="Consolability" />
271. </simpleAction>
272. <simpleAction xsi:type="CollectInformationAction">
273. <documentationConcept>
274. <displayText value="Total Score" />
275. <description
276. value="Each of the five (5) categories is scored from 0-2, which results in a total score between 0
277. and 10." />
278. <itemCodes>
279. <itemCode code="38215-0" codeSystem="2.16.840.1.113883.6.1"
280. codeSystemName="LOINC" />
281. </itemCodes>
282. <responseDataType>Integer</responseDataType>
283. <responseCardinality>Single</responseCardinality>
284. <responseRange xsi:type="ExpressionConstraint">
285. <constraintType>Minimum</constraintType>
286. <constraint xsi:type="IntegerLiteral" value="0" />
287. </responseRange>
288. <responseRange xsi:type="ExpressionConstraint">
289. <constraintType>Maximum</constraintType>
290. <constraint xsi:type="IntegerLiteral" value="10" />
291. </responseRange>
292. </documentationConcept>
293. <initialValue xsi:type="Sum">
294. <source xsi:type="List">
295. <element xsi:type="Property" path="Face">
296. <source xsi:type="ParameterRef" name="Responses" />
297. </element>
298. <element xsi:type="Property" path="Legs">
299. <source xsi:type="ParameterRef" name="Responses" />
300. </element>
301. <element xsi:type="Property" path="Activity">
302. <source xsi:type="ParameterRef" name="Responses" />
303. </element>
304. <element xsi:type="Property" path="Cry">
305. <source xsi:type="ParameterRef" name="Responses" />
306. </element>
307. <element xsi:type="Property" path="Consolability">
308. <source xsi:type="ParameterRef" name="Responses" />
309. </element>
310. </source>
311. </initialValue>
312. <responseBinding property="TotalScore" />
313. </simpleAction>
314. </subElements>
315. </actionGroup>
316. </knowledgeDocument>

## RespiratoryOrder Example

1. <?xml version="1.0" encoding="UTF-8"?>
2. <?schematron-schema href="../main/schematron/knowledgeartifact.sch"?>
3. <?schematron-schema href="../main/schematron/ordersets.sch"?>
4. <knowledgeDocument xmlns="urn:hl7-org:v3/hed"
5. xmlns:vmr="org.opencds.vmr.v1\_0.schema.vmr" xmlns:dt="urn:hl7-org:v3/cdsdt"
6. xmlns:p1="http://www.w3.org/1999/xhtml" xmlns:xml="http://www.w3.org/XML/1998/namespace"
7. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8. xsi:schemaLocation="urn:hl7-org:v3/hed ../main/schema/knowledgedocument.xsd ">
9. <!--
10. Respiratory Order
11. This example defines a typical respiratory order.
12. This example was chosen to illustrate a modular order set that could be used alone or in conjunction with another order set, and to illustrate the handling of Boolean selection logic within an order set.
13. -->
14. <metadata>
15. <identifiers>
16. <identifier root="www.zynx.com/cds/orderset/RespiratoryProtocol" extension="1234" version="2.7"/>
17. </identifiers>
18. <artifactType value="Order Set"/>
19. <schemaIdentifier root="urn:hl7-org:v3/hed" version="1.0" />
20. <dataModels>
21. <modelReference>
22. <description value="Virtual Medical Record model" />
23. <referencedModel value="org.opencds.vmr.v1\_0.schema.vmr" />
24. </modelReference>
25. </dataModels>
26. <title value=" Respiratory Order Linkable Order Set"/>
27. <description value="Order Set Illustrating Respiratory Orders. This order set is a linkable order set. That is, it is intended to be modular and included as part of a larger order set."/>
28. <documentation>
29. <description value="Explanation"/>
30. <content>
31. <xhtml:div xmlns:xhtml="http://www.w3.org/1999/xhtml">Order Set Illustrating Respiratory Orders. This order set is a linkable order set. That is, it is intended to be modular and included as part of a larger order set. It is used here to illustrate boolean relationships between orderables in an order set.</xhtml:div>
32. </content>
33. </documentation>
34. <applicability>
35. <!-- Can we assume that anything subsumed under respiratory therapy is applicable or does one need to enumerate each option? -->
36. <coverage>
37. <focus value="ClinicalFocus"/>
38. <description value="Respiratory Therapy Order - Parent Orderable"/>
39. <value code="53950000" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Respiratory therapy (procedure)" />
40. </coverage>
41. <coverage>
42. <focus value="ClinicalFocus"/>
43. <description value="Ventilator settings"/>
44. <value code="410210009" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Ventilator care management (procedure)"/>
45. </coverage>
46. <coverage>
47. <focus value="ClinicalFocus"/>
48. <description value="Pulse oximetry"/>
49. <value code="252465000" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Pulse oximetry (procedure)"/>
50. </coverage>
51. <coverage>
52. <focus value="ClinicalFocus"/>
53. <description value="Blood gas, arterial"/>
54. <value code="32564009" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Arterial specimen collection for laboratory test (procedure)"/>
55. </coverage>
56. <coverage>
57. <focus value="ClinicalFocus"/>
58. <description value="Oxygen via nasal cannula"/>
59. <value code="371907003" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Oxygen administration by nasal cannula (procedure)"/>
60. </coverage>
61. <coverage>
62. <focus value="ClinicalFocus"/>
63. <description value="Oxygen via nonrebreather face mask"/> <!-- No exact match in SNOMED -->
64. <value code="371908008" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Oxygen administration by mask (procedure)" />
65. </coverage>
66. <coverage>
67. <focus value="ClinicalFocus"/>
68. <description value="Oxygen via simple face mask"/> <!-- No exact match in SNOMED -->
69. <value code="371908008" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Oxygen administration by mask (procedure)"/>
70. </coverage>
71. <coverage>
72. <focus value="ClinicalFocus"/>
73. <description value="Oxygen via Venturi mask"/>
74. <value code="429253002" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Oxygen administration by Venturi mask (procedure)"/>
75. </coverage>
76. </applicability>
77. <status value="Draft"/>
78. <eventHistory>
79. <!-- How do we handle change tracking -->
80. <artifactLifeCycleEvent>
81. <eventType value="Created"/>
82. <eventDateTime value="20121130"/>
83. </artifactLifeCycleEvent>
84. </eventHistory>
85. <contributions>
86. <contribution>
87. <contributor xsi:type="Organization">
88. <addresses>
89. <address>
90. <dt:part type="SAL" value="10880 Wilshire Boulevard"/>
91. <dt:part type="CTY" value="Los Angeles"/>
92. <dt:part type="ZIP" value="90024"/>
93. <dt:part type="STA" value="CA"/>
94. <dt:part type="CNT" value="USA"/>
95. </address>
96. </addresses>
97. <contacts>
98. <contact value="310-825-3333" use="WP"/>
99. </contacts>
100. <name value="Zynx Health"/>
101. </contributor>
102. <role/>
103. </contribution>
104. </contributions>
105. <publishers>
106. <publisher xsi:type="Organization">
107. <addresses>
108. <address>
109. <dt:part type="SAL" value="10880 Wilshire Boulevard"/>
110. <dt:part type="CTY" value="Los Angeles"/>
111. <dt:part type="ZIP" value="90024"/>
112. <dt:part type="STA" value="CA"/>
113. <dt:part type="CNT" value="USA"/>
114. </address>
115. </addresses>
116. <contacts>
117. <contact value="310-825-3333" use="WP"/>
118. </contacts>
119. <name value="Zynx Health"/>
120. </publisher>
121. </publishers>
122. </metadata>
123. <expressions/>
124. <actionGroup> <!-- Respiratory Order -->
125. <behaviors>
126. <behavior xsi:type="GroupSelectionBehavior">
127. <value>AtMostOne</value>
128. </behavior>
129. </behaviors>
130. <representedConcepts>
131. <concept code="53950000" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Respiratory therapy (procedure)"/>
132. </representedConcepts>
133. <subElements>
134. <actionGroup> <!-- Ventilator group -->
135. <behaviors>
136. <behavior xsi:type="GroupSelectionBehavior">
137. <value>All</value>
138. </behavior>
139. </behaviors>
140. <subElements>
141. <simpleAction xsi:type="CreateAction"> <!-- Ventilator Settings -->
142. <textEquivalent value="Ventilator Settings"/>
143. <actionSentence xsi:type="ObjectExpression" objectType="vmr:ProcedureProposal">
144. <property name="code">
145. <value xsi:type="CodeLiteral" code="410210009" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Ventilator care management (procedure)"/>
146. </property>
147. </actionSentence>
148. </simpleAction>
149. <actionGroup> <!-- Oxygenation Assessment group -->
150. <behaviors>
151. <behavior xsi:type="GroupSelectionBehavior">
152. <value>Any</value>
153. </behavior>
154. </behaviors>
155. <subElements>
156. <simpleAction xsi:type="CreateAction"> <!-- Pulse Oxymetry -->
157. <textEquivalent value="Pulse Oxymetry"/>
158. <actionSentence xsi:type="ObjectExpression" objectType="vmr:ProcedureProposal">
159. <property name="code">
160. <value xsi:type="CodeLiteral" code="252465000" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Pulse oximetry (procedure)"/>
161. </property>
162. </actionSentence>
163. </simpleAction>
164. <simpleAction xsi:type="CreateAction"> <!-- Blood gas, arterial -->
165. <textEquivalent value="Blood gas, arterial"/>
166. <actionSentence xsi:type="ObjectExpression" objectType="vmr:ProcedureProposal">
167. <property name="code">
168. <value xsi:type="CodeLiteral" code="32564009" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Arterial specimen collection for laboratory test (procedure)"/>
169. </property>
170. </actionSentence>
171. </simpleAction>
172. </subElements>
173. </actionGroup>
174. </subElements>
175. </actionGroup>
176. <actionGroup> <!-- Supplemental oxygen group -->
177. <behaviors>
178. <behavior xsi:type="GroupSelectionBehavior">
179. <value>All</value>
180. </behavior>
181. </behaviors>
182. <subElements>
183. <actionGroup> <!-- Supplemental oxygen -->
184. <behaviors>
185. <behavior xsi:type="GroupSelectionBehavior">
186. <value>ExactlyOne</value>
187. </behavior>
188. </behaviors>
189. <subElements>
190. <simpleAction xsi:type="CreateAction"> <!-- Oxygen via nasal canula -->
191. <textEquivalent value="Oxygen via nasal canula"/>
192. <actionSentence xsi:type="ObjectExpression" objectType="vmr:ProcedureProposal">
193. <property name="code">
194. <value xsi:type="CodeLiteral" code="371907003" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Oxygen administration by nasal cannula (procedure)"/>
195. </property>
196. </actionSentence>
197. </simpleAction>
198. <simpleAction xsi:type="CreateAction"> <!-- Oxygen via nonrebreather face mask. Note that this is not an exact match with SNOMED CT. -->
199. <textEquivalent value="Oxygen via nonrebreather face mask"/>
200. <actionSentence xsi:type="ObjectExpression" objectType="vmr:ProcedureProposal">
201. <property name="code">
202. <value xsi:type="CodeLiteral" code="371908008" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Oxygen administration by mask (procedure)"/>
203. </property>
204. </actionSentence>
205. </simpleAction>
206. <simpleAction xsi:type="CreateAction"> <!-- Oxygen via simple face mask. Note that this is not an exact match with SNOMED CT. -->
207. <textEquivalent value="Oxygen via simple face mask"/>
208. <actionSentence xsi:type="ObjectExpression" objectType="vmr:ProcedureProposal">
209. <property name="code">
210. <value xsi:type="CodeLiteral" code="371908008" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Oxygen administration by mask (procedure)"/>
211. </property>
212. </actionSentence>
213. </simpleAction>
214. <simpleAction xsi:type="CreateAction"> <!-- Oxygen via venturi mask -->
215. <textEquivalent value="Oxygen via venturi mask"/>
216. <actionSentence xsi:type="ObjectExpression" objectType="vmr:ProcedureProposal">
217. <property name="code">
218. <value xsi:type="CodeLiteral" code="429253002" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Oxygen administration by Venturi mask (procedure)"/>
219. </property>
220. </actionSentence>
221. </simpleAction>
222. </subElements>
223. </actionGroup>
224. <actionGroup> <!-- Oxygenation assessment group -->
225. <behaviors>
226. <behavior xsi:type="GroupSelectionBehavior">
227. <value>OneOrMore</value>
228. </behavior>
229. </behaviors>
230. <subElements>
231. <simpleAction xsi:type="CreateAction"> <!-- Pulse Oxymetry -->
232. <textEquivalent value="Pulse Oxymetry"/>
233. <actionSentence xsi:type="ObjectExpression" objectType="vmr:ProcedureProposal">
234. <property name="code">
235. <value xsi:type="CodeLiteral" code="252465000" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Pulse oximetry (procedure)"/>
236. </property>
237. </actionSentence>
238. </simpleAction>
239. <simpleAction xsi:type="CreateAction"> <!-- Blood gas, arterial -->
240. <textEquivalent value="Blood gas, arterial"/>
241. <actionSentence xsi:type="ObjectExpression" objectType="vmr:ProcedureProposal">
242. <property name="code">
243. <value xsi:type="CodeLiteral" code="32564009" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT" displayName="Arterial specimen collection for laboratory test (procedure)"/>
244. </property>
245. </actionSentence>
246. </simpleAction>
247. </subElements>
248. </actionGroup>
249. </subElements>
250. </actionGroup>
251. </subElements>
252. </actionGroup>
253. </knowledgeDocument>

## DopamineComplexIVOrderWithComplexLiteral Example

1. <?xml version="1.0" encoding="UTF-8"?>
2. <?schematron-schema href="../main/schematron/knowledgeartifact.sch"?>
3. <?schematron-schema href="../main/schematron/ordersets.sch"?>
4. <knowledgeDocument xmlns="urn:hl7-org:v3/hed"
5. xmlns:vmr="org.opencds.vmr.v1\_0.schema.vmr" xmlns:dt="urn:hl7-org:v3/cdsdt"
6. xmlns:p1="http://www.w3.org/1999/xhtml" xmlns:xml="http://www.w3.org/XML/1998/namespace"
7. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8. xsi:schemaLocation="urn:hl7-org:v3/hed ../main/schema/knowledgedocument.xsd
9. org.opencds.vmr.v1\_0.schema.vmr extschema/vmr.xsd">
10. <!-- Dopamine Complex IV Order This example describes a complex IV order.
11. This example was chosen to illustrate a modular order set that could be used alone or in conjunction with another order set, to illustrate a composite order that contains both a diluent (base solution) and an additive, and to illustrate the proposed vMR extension mechanism. -->
12. <!-- Please note that all terminology bindings are for illustrative purposes only. At this time, the HeD community has not defined the terminologies and value sets that will be bound to specific clinical attribute types -->
13. <metadata>
14. <identifiers>
15. <identifier root="www.zynx.com/cds/orderset/ivsets"
16. extension="42364" version="0.9" />
17. </identifiers>
18. <artifactType value="Order Set" />
19. <schemaIdentifier root="urn:hl7-org:v3/hed" version="1.0" />
20. <templateIds>
21. <!-- An IV Set would require a template ID since it makes use of the vMR extension scheme -->
22. <templateId root="http://www.zynx.com/cds/template/"
23. extension="1237" />
24. <templateId
25. root="http://www.exampleURI.com/HeD/templates/attributeExtension" />
26. <templateId root="http://www.exampleURI.com/HeD/templates/ComplexIVOrders" />
27. </templateIds>
28. <dataModels>
29. <modelReference>
30. <description value="Virtual Medical Record model" />
31. <referencedModel value="org.opencds.vmr.v1\_0.schema.vmr" />
32. </modelReference>
33. </dataModels>
34. <title value="IV Set Order" />
35. <description value="Order set illustrating IV Sets" />
36. <documentation>
37. <description value="Explanation" />
38. <content>
39. <xhtml:div xmlns:xhtml="http://www.w3.org/1999/xhtml">IV Sets are examples of complex
40. medications commonly ordered in inpatient order sets.
41. </xhtml:div>
42. </content>
43. </documentation>
44. <status value="Draft" />
45. <eventHistory>
46. <artifactLifeCycleEvent>
47. <eventType value="Created" />
48. <eventDateTime value="20121130" />
49. </artifactLifeCycleEvent>
50. </eventHistory>
51. <contributions>
52. <contribution>
53. <contributor xsi:type="Organization">
54. <addresses>
55. <address>
56. <dt:part type="SAL" value="10880 Wilshire Boulevard" />
57. <dt:part type="CTY" value="Los Angeles" />
58. <dt:part type="ZIP" value="90024" />
59. <dt:part type="STA" value="CA" />
60. <dt:part type="CNT" value="USA" />
61. </address>
62. </addresses>
63. <contacts>
64. <contact value="310-825-3333" use="WP" />
65. </contacts>
66. <name value="Zynx Health" />
67. </contributor>
68. <role />
69. </contribution>
70. </contributions>
71. <publishers>
72. <publisher xsi:type="Organization">
73. <addresses>
74. <address>
75. <dt:part type="SAL" value="10880 Wilshire Boulevard" />
76. <dt:part type="CTY" value="Los Angeles" />
77. <dt:part type="ZIP" value="90024" />
78. <dt:part type="STA" value="CA" />
79. <dt:part type="CNT" value="USA" />
80. </address>
81. </addresses>
82. <contacts>
83. <contact value="310-825-3333" use="WP" />
84. </contacts>
85. <name value="Zynx Health" />
86. </publisher>
87. </publishers>
88. </metadata>
89. <actionGroup>
90. <behaviors>
91. <behavior xsi:type="GroupSelectionBehavior">
92. <value>All</value>
93. </behavior>
94. </behaviors>
95. <subElements>
96. <!-- Here we build the IV Set container order. All attributes here pertain to the IV Set as a whole. -->
97. <simpleAction xsi:type="CreateAction">
98. <textEquivalent value="DOPamine drip 800mg/500mL D5W" />
99. <actionSentence xsi:type="ObjectRedefine">
100. <description>
101. DOPamine drip 800mg/500mL D5W
102. Starting Dose: 2 mcg/kg/min,
103. Titrate Increment: 0.5 mcg/kg/min every 5 minutes,
104. Keep MAP Greater Than: 60,
105. Hold if HR Greater than: 120,
106. Priority: Routine,
107. Start Time Offset: now,
108. Special Inst: maximum dose of 20mcg/kg/min, titrate,
109. Comments: Ordered as: DOPamine drip 800mg/500mL D5W,
110. Diluent: D5W Titratable Base 500 mL, IV,
111. Additive: DOPamine (for infusion) 800 mg
112. </description>
113. <source xsi:type="ComplexLiteral">
114. <value xsi:type="vmr:SubstanceAdministrationProposal">
115. <id xmlns="" root="12345" /> <!-- TODO: id should not be required -->
116. <substance xmlns="">
117. <id xmlns="" root="12345" />
118. <substanceCode code="1160755" codeSystem="2.16.840.1.113883.6.88"
119. codeSystemName="RxNorm" displayName="Dopamine Injectable Product" />
120. <!-- GAP WITH WORKAROUND: Unfortunately, there is no good way to model the concentration well using RTO. What we need is a ratio of physical quantities. For now, we place this in a non-interoperable string field. -->
121. <strength numerator="800" denominator="500" /> <!-- This is supposed to be 800mg/500ml -->
122. </substance>
123. <!-- Dose Restriction: "maximum dose of 20mcg/kg/min, titrate" -->
124. <doseRestriction xmlns="">
125. <maxDoseForInterval value="20" unit="mcg/kg" /> <!-- <originalText>maximum dose of 20mcg/kg/min, titrate</originalText> -->
126. <timeInterval value="1" unit="min" />
127. </doseRestriction>
128. <!-- Starting Dose: 2 mcg/kg/min -->
129. <relatedClinicalStatement xmlns=""> <!-- TODO: Why is this xmlns required -->
130. <targetRelationshipToSource code="???"
131. codeSystem="???" codeSystemName="???" displayName="extendClassWithAttribute" />
132. <observationResult>
133. <id root="12345" />
134. <observationFocus code="???" codeSystem="???"
135. codeSystemName="???" displayName="Starting Dose" />
136. <observationValue>
137. <physicalQuantity value="2" unit="mcg/kg/min" />
138. </observationValue>
139. </observationResult>
140. </relatedClinicalStatement>
141. <!-- Titrate Increment: 0.5 mcg/kg/min -->
142. <relatedClinicalStatement xmlns="">
143. <targetRelationshipToSource code="???"
144. codeSystem="???" codeSystemName="???" displayName="extendClassWithAttribute" />
145. <observationResult>
146. <id root="12345" />
147. <observationFocus code="???" codeSystem="???"
148. codeSystemName="???" displayName="Titrate Increment" />
149. <observationValue>
150. <physicalQuantity value="0.5" unit="mcg/kg/min" />
151. </observationValue>
152. </observationResult>
153. </relatedClinicalStatement>
154. <!-- Titrate Frequency: q5min -->
155. <relatedClinicalStatement xmlns="">
156. <targetRelationshipToSource code="???"
157. codeSystem="???" codeSystemName="???" displayName="extendClassWithAttribute" />
158. <observationResult>
159. <id root="12345" />
160. <observationFocus code="???" codeSystem="???"
161. codeSystemName="???" displayName="Titrate Frequency" />
162. <observationValue>
163. <concept code="???" codeSystem="???" codeSystemName="???"
164. displayName="q5min" />
165. </observationValue>
166. </observationResult>
167. </relatedClinicalStatement>
168. <!-- Nursing Instruction: "Hold if HR Greater than: 120" -->
169. <relatedClinicalStatement xmlns="">
170. <targetRelationshipToSource code="???"
171. codeSystem="???" codeSystemName="???" displayName="extendClassWithAttribute" />
172. <observationResult>
173. <id root="12345" />
174. <observationFocus code="???" codeSystem="???"
175. codeSystemName="???" displayName="Nursing Instructions" />
176. <observationValue>
177. <text value="Hold if HR Greater than: 120" />
178. </observationValue>
179. </observationResult>
180. </relatedClinicalStatement>
181. <!-- Nursing Instruction: "Keep MAP Greater Than: 60" -->
182. <relatedClinicalStatement xmlns="">
183. <targetRelationshipToSource code="???"
184. codeSystem="???" codeSystemName="???" displayName="extendClassWithAttribute" />
185. <observationResult>
186. <id root="12345" />
187. <observationFocus code="???" codeSystem="???"
188. codeSystemName="???" displayName="Nursing Instructions" />
189. <observationValue>
190. <text value="Keep MAP Greater Than: 60" />
191. </observationValue>
192. </observationResult>
193. </relatedClinicalStatement>
194. <!-- Priority: Routine" -->
195. <!-- NOTE: Not using Criticality due to semantic overload of Priority and Criticality -->
196. <relatedClinicalStatement xmlns="">
197. <targetRelationshipToSource code="extendClassWithAttribute"
198. codeSystem="HeDVMRExtensionAttributes" codeSystemName="HeDVMRExtensionAttributes"
199. displayName="extendClassWithAttribute" />
200. <observationResult>
201. <id root="12345" />
202. <observationFocus code="???" codeSystem="???"
203. codeSystemName="???" displayName="Priority" />
204. <observationValue>
205. <concept code="???" codeSystem="???" codeSystemName="???"
206. displayName="Routine" />
207. </observationValue>
208. </observationResult>
209. </relatedClinicalStatement>
210. <!-- Specify the drug diluent for this Complex IV -->
211. <relatedClinicalStatement xmlns="">
212. <targetRelationshipToSource code="74626007"
213. codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT"
214. displayName="Drug diluent" />
215. <substanceAdministrationProposal>
216. <id root="12345" />
217. <substance>
218. <id root="12345" />
219. <substanceCode code="400420008" codeSystem="2.16.840.1.113883.6.96"
220. codeSystemName="SNOMED-CT"
221. displayName="Dextrose 5g/100mL (5%) injection solution 500mL vial" />
222. </substance>
223. </substanceAdministrationProposal>
224. </relatedClinicalStatement>
225. <!-- Specify the drug additive for this Complex IV -->
226. <relatedClinicalStatement xmlns="">
227. <targetRelationshipToSource code="81388006"
228. codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT"
229. displayName="Drug additive" />
230. <substanceAdministrationProposal>
231. <id root="12345" />
232. <substance>
233. <id root="12345" />
234. <substanceCode code="1160755" codeSystem="2.16.840.1.113883.6.88"
235. codeSystemName="RxNorm" displayName="Dopamine Injectable Product" />
236. </substance>
237. <doseQuantity lowValue="800" lowUnit="mg"
238. highValue="800" highUnit="mg" /> <!-- TODO: highValue and highUnit are required? -->
239. </substanceAdministrationProposal>
240. </relatedClinicalStatement>
241. </value>
242. </source>
243. <!-- Start Time Offset: Now -->
244. <property name="proposedAdministrationTimeInterval"> <!-- IVL\_TS -->
245. <value xsi:type="Interval">
246. <begin xsi:type="Now" />
247. </value>
248. </property>
249. </actionSentence>
250. </simpleAction>
251. </subElements>
252. </actionGroup>
253. </knowledgeDocument>

## HeartFailureAdmissionToMedSurgOrderSet Example

1. <?xml version="1.0" encoding="UTF-8"?>
2. <?schematron-schema href="../main/schematron/knowledgeartifact.sch"?>
3. <?schematron-schema href="../main/schematron/ordersets.sch"?>
4. <!-- Rationale: -->
5. <knowledgeDocument xmlns="urn:hl7-org:v3/hed"
6. xmlns:vmr="org.opencds.vmr.v1\_0.schema.vmr"
7. xmlns:dt="urn:hl7-org:v3/cdsdt" xmlns:p1="http://www.w3.org/1999/xhtml"
8. xmlns:xml="http://www.w3.org/XML/1998/namespace" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
9. xsi:schemaLocation="urn:hl7-org:v3/hed ../main/schema/knowledgedocument.xsd ">
10. <!--
11. Heart Failure Admission to Med/Surg Unit
12. This is a partial order set for Heart Failure, Admission to the Med/Surg unit of a hospital.
13. This example was chosen to illustrate a routine order set and how to represent sections, reminders, orders, order details, performance measures, evidence links, selection (and other) types of behaviors, and links to modular order sets. -->
14. <!-- Note that all coding systems and codes in this example are for illustrative purposes only. The Health eDecisions Terminologies and Value Sets sub-working group is working to finalize a set of terminologies and value sets for all the coded values in a Health eDecisions artifact. Once that set of terminologies and value sets is finalized, this example will be updated to include the correct references. For example, LOINC is used in this example as the terminology for laboratory orders, but this is subject to change, depending on the results of the sub-working group. -->
15. <metadata>
16. <!-- This section contains all the metadata for the artifact which can be used to support searches -->
17. <identifiers>
18. <identifier root="www.zynx.com/cds/orderset/HeartFailureAdmissionToMedSurg"
19. extension="1234" version="3.0"/>
20. </identifiers>
21. <artifactType value="Order Set"/>
22. <schemaIdentifier
23. root="urn:hl7-org:v3/hed" version="1.0" />
24. <templateIds>
25. <templateId root="http://www.zynx.com/cds/template/" extension="1234" version="3.0"/>
26. </templateIds>
27. <dataModels>
28. <modelReference>
29. <description value="Virtual Medical Record model" />
30. <referencedModel value="org.opencds.vmr.v1\_0.schema.vmr"/>
31. </modelReference>
32. </dataModels>
33. <title value="Heart Failure Admission to Med/Surg"/>
34. <description value="The Heart Failure module addresses the medical inpatient management of adult patients with new-onset or acute exacerbations of heart failure. The emphasis of this module is on medical management. Surgical and other nonmedical interventions are not covered in depth. This module addresses acute cardiogenic pulmonary edema due to heart failure with reduced left ventricular ejection fraction. This module does not fully address management of acute myocardial infarction or unstable angina."/>
35. <documentation>
36. <title value="Zynx Heart Failure Module"/>
37. <location value="https://www.zynx.com/Reference/Content.aspx?ItemID=216945"/>
38. <description value="Addresses the medical inpatient management of adult patients with new-onset or acute exacerbations of heart failure"/>
39. </documentation>
40. <applicability>
41. <!-- These coverage codes are to help users who are searching for an applicable Order Set -->
42. <coverage>
43. <focus value="ClinicalFocus"/>
44. <description value="Heart Failure - SNOMED CT"/>
45. <value codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT" code="84114007"
46. displayName="Heart Failure (disorder)"/>
47. </coverage>
48. <coverage>
49. <focus value="ClinicalFocus"/>
50. <description value="Heart Failure - ICD-9"/>
51. <value codeSystem="2.16.840.1.113883.3.7.1.6.1" codeSystemName="ICD-9" code="428.9"
52. displayName="Heart failure, unspecified"/>
53. </coverage>
54. <coverage>
55. <focus value="PatientAgeGroup"/>
56. <description value="Adult"/>
57. <value code="133936004" codeSystem="2.16.840.1.113883.6.96"
58. codeSystemName="SNOMED-CT" displayName="Adult"/>
59. </coverage>
60. <coverage>
61. <focus value="ClinicalVenue"/>
62. <description value="Inpatient"/>
63. <value codeSystem="2.16.840.1.113883.12.4" codeSystemName="Patient class (HL7)"
64. code="I" displayName="Inpatient"/>
65. </coverage>
66. </applicability>
67. <status value="Active"/>
68. <eventHistory>
69. <artifactLifeCycleEvent>
70. <eventType value="Created"/>
71. <eventDateTime value="20110125"/>
72. </artifactLifeCycleEvent>
73. <artifactLifeCycleEvent>
74. <eventType value="Pre-published"/>
75. <eventDateTime value="20110911"/>
76. </artifactLifeCycleEvent>
77. <artifactLifeCycleEvent>
78. <eventType value="Published"/>
79. <eventDateTime value="20120125"/>
80. </artifactLifeCycleEvent>
81. </eventHistory>
82. <publishers>
83. <publisher xsi:type="Organization">
84. <addresses>
85. <address>
86. <dt:part type="SAL" value="10880 Wilshire Boulevard"/>
87. <dt:part type="CTY" value="Los Angeles"/>
88. <dt:part type="ZIP" value="90024"/>
89. <dt:part type="STA" value="CA"/>
90. <dt:part type="CNT" value="USA"/>
91. </address>
92. </addresses>
93. <contacts>
94. <contact value="310-825-3333" use="WP"/>
95. </contacts>
96. <name value="Zynx Health"/>
97. </publisher>
98. </publishers>
99. </metadata>
100. <externalData>
101. <def name="Patient">
102. <expression xsi:type="ClinicalRequest" cardinality="Single" dataType="vmr:EvaluatedPerson"
103. isInitial="true"/>
104. </def>
105. <def name="AdverseReactionToACEInhibitors">
106. <expression xsi:type="ClinicalRequest" cardinality="Multiple"
107. dataType="vmr:AdverseEvent" isInitial="true">
108. <codes xsi:type="List">
109. <element xsi:type="CodeLiteral" code="293500009"
110. codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT"
111. displayName="Angiotensin-converting-enzyme inhibitor adverse reaction (disorder)"/>
112. <element xsi:type="CodeLiteral" code="295036000"
113. codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT"
114. displayName="Angiotensin-converting-enzyme inhibitor allergy (disorder)"/>
115. <element xsi:type="CodeLiteral" code="407579007"
116. codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT"
117. displayName="History of - angiotensin II receptor antagonist allergy (situation)"/>
118. <element xsi:type="CodeLiteral" code="407590002"
119. codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED-CT"
120. displayName="Angiotensin II receptor antagonist adverse reaction "/>
121. </codes>
122. </expression>
123. </def>
124. <def name="AdministeredACEInhibitors">
125. <expression xsi:type="ClinicalRequest" cardinality="Multiple"
126. dataType="vmr:SubstanceAdministrationEvent"
127. codeProperty="substanceAdministrationGeneralPurpose.code"
128. dateProperty="administrationTimeInterval.low" useValueSets="true">
129. <description>ACE inhibitor administered to patient</description>
130. <codes xsi:type="CodeLiteral" code="N0000000181"
131. codeSystem="2.16.840.1.113883.3.26.1.5" codeSystemName="NDF-RT"
132. displayName="Angiotensin-Converting Enzyme Inhibitors"/>
133. </expression>
134. </def>
135. <def name="PrescribedACEInhibitors">
136. <expression xsi:type="ClinicalRequest" cardinality="Multiple"
137. dataType="vmr:SubstanceAdministrationProposal"
138. codeProperty="substanceAdministrationGeneralPurpose.code"
139. dateProperty="administrationTimeInterval.low" useValueSets="true">
140. <description>Patient prescribed ACE inhibitor medication</description>
141. <codes xsi:type="CodeLiteral" code="N0000000181"
142. codeSystem="2.16.840.1.113883.3.26.1.5" codeSystemName="NDF-RT"
143. displayName="Angiotensin-Converting Enzyme Inhibitors"/>
144. </expression>
145. </def>
146. </externalData>
147. <expressions>
148. <def name="PatientAge">
149. <expression xsi:type="Property" path="demographics.age">
150. <source xsi:type="ExpressionRef" name="Patient"/>
151. </expression>
152. </def>
153. </expressions>
154. <actionGroup>
155. <subElements>
156. <actionGroup>
157. <!-- General -->
158. <title value="General"/>
159. <subElements>
160. <!-- Note that both of these "reminder" elements are modelled as MessageActions, but we need to revisit the semantics of MessageAction so we can distinguish message actions that might be ordered by a provider (e.g., "send email to patient about xxx") vs. messages that are simply included in-line with a list of orders in an Order Set. Note that group selection behaviours related to selection should not apply to the latter type of message, but should apply to the former type. -->
161. <simpleAction xsi:type="MessageAction">
162. <supportingEvidence>
163. <evidence>
164. <resources>
165. <!-- shows both a content-provider evidence link (Zynx) as well as a link to a third-party source -->
166. <resource>
167. <title value="Zynx Evidence"/>
168. <location
169. value="https://www.zynx.com/Reference/Content.aspx?ItemID=216984&amp;ver=1"
170. />
171. </resource>
172. <resource>
173. <title value="European Journal of Heart Failure 2009"/>
174. <location
175. value="http://eurjhf.oxfordjournals.org/content/11/12/1208.full.pdf"
176. />
177. </resource>
178. </resources>
179. </evidence>
180. </supportingEvidence>
181. <message xsi:type="StringLiteral" value="ADHERE Risk Model"/>
182. </simpleAction>
183. <simpleAction xsi:type="MessageAction">
184. <supportingEvidence>
185. <evidence>
186. <resources>
187. <resource>
188. <title value="Zynx Evidence"/>
189. <location
190. value="https://www.zynx.com/Reference/Content.aspx?ItemID=216985&amp;ver=1"
191. />
192. </resource>
193. </resources>
194. </evidence>
195. </supportingEvidence>
196. <message xsi:type="StringLiteral" value="Heart Failure Survival Score"/>
197. </simpleAction>
198. </subElements>
199. </actionGroup>
200. <actionGroup>
201. <!-- Activity -->
202. <behaviors>
203. <behavior xsi:type="GroupSelectionBehavior">
204. <!-- Indicate that the physician must choose exactly one of the items in this group as they are mutually exclusive. -->
205. <value>ExactlyOne</value>
206. </behavior>
207. </behaviors>
208. <title value="Activity"/>
209. <subElements>
210. <!-- Create some proposed procedures to be presented to the physician at CPOE time -->
211. <simpleAction xsi:type="CreateAction">
212. <!-- Create a proposed procedure to be presented to the physician at CPOE time -->
213. <textEquivalent value="Ambulate"/>
214. <actionSentence xsi:type="ObjectExpression"
215. objectType="vmr:ProcedureProposal">
216. <property name="code">
217. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.96" code="62013009" codeSystemName="SNOMED-CT" displayName="Ambulating Patient"/>
218. </property>
219. </actionSentence>
220. </simpleAction>
221. <simpleAction xsi:type="CreateAction">
222. <textEquivalent value="Bed rest"/>
223. <actionSentence xsi:type="ObjectExpression"
224. objectType="vmr:ProcedureProposal">
225. <property name="code">
226. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.96" code="183074009" codeSystemName="SNOMED-CT" displayName="Recommendation to rest in bed"/>
227. <!-- Terminology note: this is supposed to be an order, not a recommendation, so SNOMED-CT term is not a good fit for this -->
228. </property>
229. </actionSentence>
230. </simpleAction>
231. </subElements>
232. </actionGroup>
233. <actionGroup>
234. <!-- Nursing Orders -->
235. <title value="Nursing Orders"/>
236. <subElements>
237. <actionGroup>
238. <behaviors>
239. <!-- Indicate that the physician may choose any items in the group. "Any" includes choosing none. -->
240. <behavior xsi:type="GroupSelectionBehavior">
241. <value>Any</value>
242. </behavior>
243. </behaviors>
244. <title value="Assessments"/>
245. <subElements>
246. <simpleAction xsi:type="CreateAction">
247. <!-- Cardiac monitor -->
248. <behaviors>
249. <behavior xsi:type="PrecheckBehavior">
250. <value>Yes</value>
251. </behavior>
252. </behaviors>
253. <textEquivalent value="Cardiac Monitor"/>
254. <actionSentence xsi:type="ObjectExpression"
255. objectType="vmr:ProcedureProposal">
256. <property name="code">
257. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.96" code="88140007" codeSystemName="SNOMED-CT" displayName="Cardiac monitor surveillance"/>
258. </property>
259. </actionSentence>
260. </simpleAction>
261. <simpleAction xsi:type="CreateAction">
262. <!-- Measure blood pressure, orthostatic -->
263. <textEquivalent value="Measure blood pressure, orthostatic"/>
264. <actionSentence xsi:type="ObjectExpression"
265. objectType="vmr:ProcedureProposal">
266. <property name="code">
267. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.96" code="425058005" codeSystemName="SNOMED-CT" displayName="Taking orthostatic vital signs"/>
268. <!-- Terminology comment: SNOMED Term is an indirect match for the desired order -->
269. </property>
270. </actionSentence>
271. </simpleAction>
272. </subElements>
273. </actionGroup>
274. <actionGroup>
275. <behaviors>
276. <behavior xsi:type="GroupSelectionBehavior">
277. <value>Any</value>
278. </behavior>
279. </behaviors>
280. <title value="Interventions"/>
281. <subElements>
282. <simpleAction xsi:type="CreateAction">
283. <textEquivalent value="Elevate head of bed"/>
284. <actionSentence xsi:type="ObjectExpression"
285. objectType="vmr:ProcedureProposal">
286. <property name="code">
287. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.96" code="423171007" codeSystemName="SNOMED-CT" displayName="Elevation of head of bed"/>
288. </property>
289. </actionSentence>
290. </simpleAction>
291. <simpleAction xsi:type="CreateAction">
292. <textEquivalent value="Urinary catheter initiation/management"/>
293. <actionSentence xsi:type="ObjectExpression"
294. objectType="vmr:ProcedureProposal">
295. <property name="code">
296. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.96" code="429723008" codeSystemName="SNOMED-CT" displayName="Procedure involving urinary catheter"/>
297. </property>
298. </actionSentence>
299. </simpleAction>
300. </subElements>
301. </actionGroup>
302. </subElements>
303. </actionGroup>
304. <actionGroup>
305. <!-- Medications -->
306. <behaviors>
307. <behavior xsi:type="GroupOrganizationBehavior">
308. <!-- Indicate that this group should be visually separated from other groups in the order set, perhaps as
309. indented section with the group title as the title, etc. -->
310. <value>VisualGroup</value>
311. </behavior>
312. </behaviors>
313. <title value="Medications"/>
314. <subElements>
315. <actionGroup>
316. <subElements>
317. <simpleAction xsi:type="MessageAction">
318. <!-- if patient meets certain criteria, then remind the physician to consider ordering an ACE inhibitor. -->
319. <conditions>
320. <condition>
321. <logic xsi:type="And">
322. <operand xsi:type="GreaterOrEqual">
323. <operand xsi:type="ExpressionRef" name="Age"/>
324. <operand xsi:type="IntegerLiteral" value="18"/>
325. </operand>
326. <operand xsi:type="IsEmpty">
327. <operand xsi:type="ExpressionRef"
328. name="AdverseReactionToACEInhibitors"/>
329. </operand>
330. </logic>
331. <conditionRole>ApplicableScenario</conditionRole>
332. </condition>
333. </conditions>
334. <textEquivalent
335. value="If the patient is over 18 years old and not allergic to an ACE inhibitor and not receiving an ACE inhibitor and LVEF is less than 40%, then consider prescribing an ACE inhibitor."/>
336. <message xsi:type="StringLiteral"
337. value="If the patient is over 18 years old and not allergic to an ACE inhibitor and not receiving an ACE inhibitor and LVEF is less than 40%, then consider prescribing an ACE inhibitor."
338. />
339. </simpleAction>
340. </subElements>
341. </actionGroup>
342. <actionGroup>
343. <!-- ACE inhibitors -->
344. <supportingResources>
345. <!-- Quality Measures -->
346. <resource>
347. <!-- Note that no identifier is given for this quality measure, but one could be if desired. -->
348. <title
349. value="ARRA EHR Stage 2 Meaningful Use Quality Measure by the Centers for Medicare and Medicaid Services (2012)"/>
350. <location value="TBD"/>
351. <description value="ARRA EHR Stage 2 Meaningful Use Quality Measure by the Centers for Medicare and Medicaid Services (2012)"/>
352. </resource>
353. <resource>
354. <title
355. value="Physician Consortium for Performance Improvement Performance Measure by the American Medical Association (2012)"/>
356. <location value="TBD"/>
357. <description value="Physician Consortium for Performance Improvement Performance Measure by the American Medical Association (2012)"/>
358. </resource>
359. </supportingResources>
360. <behaviors>
361. <!-- Note interaction between group selection behavior and required behavior. This says that the physician is required to choose
362. exactly one of these medications unless documentation is provided explaining why it was not chosen. -->
363. <behavior xsi:type="GroupSelectionBehavior">
364. <value>ExactlyOne</value>
365. </behavior>
366. <behavior xsi:type="RequiredBehavior">
367. <value>MustUnlessDocumented</value>
368. </behavior>
369. </behaviors>
370. <title value="Angiotensin-Converting Enzyme Inhibitors"/>
371. <description value="For patients with diastolic heart failure who are intolerant to an ACE inhibitor, consider the use of an ARB. For patients with diastolic heart failure, consider the use of an ACE inhibitor. For patients with systolic heart failure who are hemodynamically stable and are intolerant to an ACE inhibitor due to cough, use an ARB"/>
372. <representedConcepts>
373. <!-- Coded this as an ACE inhibitors group -->
374. <concept code="N0000000181" codeSystem="2.16.840.1.113883.3.26.1.5"
375. codeSystemName="NDF-RT"
376. displayName="Angiotensin-Converting Enzyme Inhibitors"/>
377. </representedConcepts>
378. <subElements>
379. <actionGroup>
380. <!-- This is a group of orders in which captopril is the orderable substance for each. We use the representedConcept to specify this explicitly, and we use
381. SentenceGroup behavior to indicate that each item in the group references the same orderable. -->
382. <behaviors>
383. <behavior xsi:type="GroupOrganizationBehavior">
384. <value>SentenceGroup</value>
385. </behavior>
386. </behaviors>
387. <representedConcepts>
388. <concept codeSystem="2.16.840.1.113883.6.88" code="1998"
389. codeSystemName="RxNorm" displayName="captopril"/>
390. </representedConcepts>
391. <subElements>
392. <simpleAction xsi:type="CreateAction">
393. <textEquivalent
394. value="captopril 6.25 milligram orally 3 times a day"/>
395. <actionSentence xsi:type="ObjectExpression"
396. objectType="vmr:SubstanceAdministrationProposal">
397. <property name="substance">
398. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.88" code="1998" codeSystemName="RxNorm" displayName="captopril"/>
399. </property>
400. <property name="doseQuantity">
401. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="6.25" lowUnit="mg" highValue="6.25" highUnit="mg" lowIsInclusive="true" highIsInclusive="true"/>
402. </property>
403. <property name="deliveryRoute">
404. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.3.26.1.1.1" code="001" codeSystemName="FDA Routes of Administration" displayName="ORAL"/>
405. </property>
406. <!-- vMR does not have a way to say "3 times per day". Instead, a dosing period must be specified. Since 3 times per day would be every 8 hours for a fixed
407. dosing period, we set 8 hours as the dosing period. The value of dosingPeriodIntervalIsImportant is used to determine whether the interval itself is important
408. or not. If not, then a dosing period of "every 8 hours" is interpreted as "3 times per day". -->
409. <!-- Since dosing frequency is typically a coded value in EMR's, perhaps we should consider adding a coded frequency field to vMR? Otherwise, this may be very difficult
410. for current EMR's to process. -->
411. <property name="dosingPeriod">
412. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="8" lowUnit="h" highValue="8" highUnit="h" lowIsInclusive="true" highIsInclusive="true"/>
413. </property>
414. <property name="dosingPeriodIntervalIsImportant">
415. <value xsi:type="BooleanLiteral" value="false"/>
416. </property>
417. </actionSentence>
418. </simpleAction>
419. <simpleAction xsi:type="CreateAction">
420. <textEquivalent
421. value="captopril 12.5 milligram orally 3 times a day"/>
422. <actionSentence xsi:type="ObjectExpression"
423. objectType="vmr:SubstanceAdministrationProposal">
424. <property name="substance">
425. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.88" code="1998" codeSystemName="RxNorm" displayName="captopril"/>
426. </property>
427. <property name="doseQuantity">
428. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="12.5" lowUnit="mg" highValue="12.5" highUnit="mg" lowIsInclusive="true" highIsInclusive="true"/>
429. </property>
430. <property name="deliveryRoute">
431. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.3.26.1.1.1" code="001" codeSystemName="FDA Routes of Administration" displayName="ORAL"/>
432. </property>
433. <property name="dosingPeriod">
434. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="8" lowUnit="h" highValue="8" highUnit="h" lowIsInclusive="true" highIsInclusive="true"/>
435. </property>
436. <property name="dosingPeriodIntervalIsImportant">
437. <value xsi:type="BooleanLiteral" value="false"/>
438. </property>
439. </actionSentence>
440. </simpleAction>
441. <simpleAction xsi:type="CreateAction">
442. <textEquivalent
443. value="captopril 25 milligram orally 3 times a day"/>
444. <actionSentence xsi:type="ObjectExpression"
445. objectType="vmr:SubstanceAdministrationProposal">
446. <property name="substance">
447. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.88" code="1998" codeSystemName="RxNorm" displayName="captopril"/>
448. </property>
449. <property name="doseQuantity">
450. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="25" lowUnit="mg" highValue="25" highUnit="mg" lowIsInclusive="true" highIsInclusive="true"/>
451. </property>
452. <property name="deliveryRoute">
453. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.3.26.1.1.1" code="001" codeSystemName="FDA Routes of Administration" displayName="ORAL"/>
454. </property>
455. <property name="dosingPeriod">
456. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="8" lowUnit="h" highValue="8" highUnit="h" lowIsInclusive="true" highIsInclusive="true"/>
457. </property>
458. <property name="dosingPeriodIntervalIsImportant">
459. <value xsi:type="BooleanLiteral" value="false"/>
460. </property>
461. </actionSentence>
462. </simpleAction>
463. </subElements>
464. </actionGroup>
465. <simpleAction xsi:type="CreateAction">
466. <textEquivalent value="lisinopril 2.5 milligram orally once a day"/>
467. <actionSentence xsi:type="ObjectExpression"
468. objectType="vmr:SubstanceAdministrationProposal">
469. <property name="substance">
470. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.88" code="29046" codeSystemName="RxNorm" displayName="Lisinopril"/>
471. </property>
472. <property name="doseQuantity">
473. <!-- units should be coded value, but ISO datatypes do not support a coded value for the units of meausre. We will work with the vMR group to address this. In the meantime, we are using coded values for all UOM from UCUM (https://phinvads.cdc.gov/vads/ViewCodeSystem.action?id=2.16.840.1.113883.6.8) until this issue is addressed ...-->
474. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="2.5" lowUnit="mg" highValue="2.5" highUnit="mg" lowIsInclusive="true" highIsInclusive="true"/>
475. </property>
476. <property name="deliveryRoute">
477. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.3.26.1.1.1" code="001" codeSystemName="FDA Routes of Administration" displayName="ORAL"/>
478. </property>
479. <property name="dosingPeriod">
480. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="24" lowUnit="h" highValue="24" highUnit="h" lowIsInclusive="true" highIsInclusive="true"/>
481. </property>
482. <property name="dosingPeriodIntervalIsImportant">
483. <value xsi:type="BooleanLiteral" value="false"/>
484. </property>
485. </actionSentence>
486. </simpleAction>
487. </subElements>
488. </actionGroup>
489. <actionGroup>
490. <!-- Beta Blockers -->
491. <supportingResources>
492. <!-- Quality Measures, none with IDs -->
493. <resource>
494. <title
495. value="ARRA EHR Stage 2 Meaningful Use Quality Measure by the Centers for Medicare and Medicaid Services (2012)"/>
496. <location
497. value="https://www.zynx.com/Reference/Content.aspx?ItemID=216994"/>
498. <description value="ARRA EHR Stage 2 Meaningful Use Quality Measure by the Centers for Medicare and Medicaid Services (2012)"/>
499. </resource>
500. <resource>
501. <title
502. value="Physician Consortium for Performance Improvement Performance Measure by the American Medical Association (2012)"/>
503. <location
504. value="https://www.zynx.com/Reference/Content.aspx?ItemID=216994"/>
505. <description value="Physician Consortium for Performance Improvement Performance Measure by the American Medical Association (2012)"/>
506. </resource>
507. <resource>
508. <title
509. value="Get With The Guidelines Achievement Measure by the American Heart Association/American Stroke Association (2012)"/>
510. <location
511. value="https://www.zynx.com/Reference/Content.aspx?ItemID=216994"/>
512. <description value="Get With The Guidelines Achievement Measure by the American Heart Association/American Stroke Association (2012)"/>
513. </resource>
514. <resource>
515. <title
516. value="Target Measure by the American Heart Association/American Stroke Association (2012)"/>
517. <location
518. value="https://www.zynx.com/Reference/Content.aspx?ItemID=216994"/>
519. <description value="Target Measure by the American Heart Association/American Stroke Association (2012)"/>
520. </resource>
521. </supportingResources>
522. <behaviors>
523. <!-- Note interaction between group selection behavior and required behavior. -->
524. <behavior xsi:type="GroupSelectionBehavior">
525. <value>ExactlyOne</value>
526. </behavior>
527. <behavior xsi:type="RequiredBehavior">
528. <value>MustUnlessDocumented</value>
529. </behavior>
530. </behaviors>
531. <title value="Beta-Blockers"/>
532. <description value="For patients with diastolic heart failure and a previous MI, use a beta-blocker. For patients with diastolic heart failure, consider the use of a beta-blocker. For patients with systolic heart failure who are hemodynamically stable, use beta-blocker therapy (eg, bisoprolol, carvedilol, metoprolol extended release)."/>
533. <representedConcepts>
534. <!-- Coded this as a beta blockers group -->
535. <concept code="N0000000161" codeSystem="2.16.840.1.113883.3.26.1.5"
536. codeSystemName="NDF-RT" displayName="Adrenergic beta-Antagonists"/>
537. </representedConcepts>
538. <subElements>
539. <simpleAction xsi:type="CreateAction">
540. <textEquivalent
541. value="carvedilol 3.125 milligram orally 2 times a day"/>
542. <actionSentence xsi:type="ObjectExpression"
543. objectType="vmr:SubstanceAdministrationProposal">
544. <property name="substance">
545. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.88" code="20352" codeSystemName="RxNorm" displayName="carvedilol"/>
546. </property>
547. <property name="doseQuantity">
548. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="3.125" lowUnit="mg" highValue="3.125" highUnit="mg" lowIsInclusive="true" highIsInclusive="true"/>
549. </property>
550. <property name="deliveryRoute">
551. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.3.26.1.1.1" code="001" codeSystemName="FDA Routes of Administration" displayName="ORAL"/>
552. </property>
553. <!-- vMR does not have a way to say "2 times per day". Instead, a dosing period must be specified. Since 2 times per day would be every 12 hours for a fixed
554. dosing period, we set 12 hours as the dosing period. The value of dosingPeriodIntervalIsImportant is used to determine whether the interval itself is important
555. or not. If not, then a dosing period of "every 12 hours" is interpreted as "2 times per day". -->
556. <!-- Since dosing frequency is typically a coded value in EMR's, perhaps we should consider adding a coded frequency field to vMR? Otherwise, this may be very difficult
557. for current EMR's to process. -->
558. <property name="dosingPeriod">
559. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="12" lowUnit="h" highValue="12" highUnit="h" lowIsInclusive="true" highIsInclusive="true"/>
560. </property>
561. <property name="dosingPeriodIntervalIsImportant">
562. <value xsi:type="BooleanLiteral" value="false"/>
563. </property>
564. </actionSentence>
565. </simpleAction>
566. <actionGroup> <!-- Note that this sentence group has the same orderable item in both cases - "metoprolol succinate SR 25 mg 24 hr tab", but at two different dose levels.
567. The representedConcept for the group captures this common orderable, and each CreateAction in the sub-elements creates each of the two dosing levels. -->
568. <behaviors>
569. <behavior xsi:type="GroupOrganizationBehavior">
570. <value>SentenceGroup</value>
571. </behavior>
572. </behaviors>
573. <representedConcepts>
574. <concept codeSystem="2.16.840.1.113883.6.88"
575. code="TBD - now missing from RxNorm" codeSystemName="RxNorm"
576. displayName="24 HR Metoprolol Succinate 25 MG Extended Release Tablet"
577. />
578. </representedConcepts>
579. <subElements>
580. <simpleAction xsi:type="CreateAction">
581. <textEquivalent
582. value="metoprolol succinate SR 25 mg 24 hr tab 0.5 tablet orally once a day"/>
583. <actionSentence xsi:type="ObjectExpression"
584. objectType="vmr:SubstanceAdministrationProposal">
585. <property name="substance">
586. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.88" code="TBD - now missing from RxNorm" codeSystemName="RxNorm" displayName="24 HR Metoprolol Succinate 25 MG Extended Release Tablet"/>
587. </property>
588. <property name="doseQuantity">
589. <!-- units should be coded value, but ISO datatypes do not support a coded value for the units of measure. We will work with the vMR group to address this. In the meantime, we are using coded values for all UOM from UCUM (https://phinvads.cdc.gov/vads/ViewCodeSystem.action?id=2.16.840.1.113883.6.8) until this issue is addressed ...-->
590. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="0.5" lowUnit="tbl" highValue="0.5" highUnit="tbl" lowIsInclusive="true" highIsInclusive="true"/>
591. </property>
592. <property name="deliveryRoute">
593. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.3.26.1.1.1" code="001" codeSystemName="FDA Routes of Administration" displayName="ORAL"/>
594. </property>
595. <property name="dosingPeriod">
596. <!-- This means once per day -->
597. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="24" lowUnit="h" highValue="24" highUnit="h" lowIsInclusive="true" highIsInclusive="true"/>
598. </property>
599. <property name="dosingPeriodIntervalIsImportant">
600. <value xsi:type="BooleanLiteral" value="false"/>
601. </property>
602. </actionSentence>
603. </simpleAction>
604. <simpleAction xsi:type="CreateAction">
605. <textEquivalent
606. value="metoprolol succinate SR 25 mg 24 hr tab 1 tablet orally once a day"/>
607. <actionSentence xsi:type="ObjectExpression"
608. objectType="vmr:SubstanceAdministrationProposal">
609. <property name="substance">
610. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.88" code="TBD - now missing from RxNorm" codeSystemName="RxNorm" displayName="24 HR Metoprolol Succinate 25 MG Extended Release Tablet"/>
611. </property>
612. <property name="doseQuantity">
613. <!-- units should be coded value, but ISO datatypes do not support a coded value for the units of measure. We will work with the vMR group to address this. In the meantime, we are using coded values for all UOM from UCUM (https://phinvads.cdc.gov/vads/ViewCodeSystem.action?id=2.16.840.1.113883.6.8) until this issue is addressed ...-->
614. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="1" lowUnit="tbl" highValue="1" highUnit="tbl" lowIsInclusive="true" highIsInclusive="true"/>
615. </property>
616. <property name="deliveryRoute">
617. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.3.26.1.1.1" code="001" codeSystemName="FDA Routes of Administration" displayName="ORAL"/>
618. </property>
619. <property name="dosingPeriod">
620. <!-- This means once per day -->
621. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="24" lowUnit="h" highValue="24" highUnit="h" lowIsInclusive="true" highIsInclusive="true"/>
622. </property>
623. <property name="dosingPeriodIntervalIsImportant">
624. <value xsi:type="BooleanLiteral" value="false"/>
625. </property>
626. </actionSentence>
627. </simpleAction>
628. </subElements>
629. </actionGroup>
630. <simpleAction xsi:type="CreateAction">
631. <textEquivalent
632. value="metoprolol succinate SR 50 mg 24 hr tab 1 tablet orally once a day"/>
633. <actionSentence xsi:type="ObjectExpression"
634. objectType="vmr:SubstanceAdministrationProposal">
635. <property name="substance">
636. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.88" code="TBD - now missing from RxNorm" codeSystemName="RxNorm" displayName="24 HR Metoprolol Succinate 50 MG Extended Release Tablet"/>
637. </property>
638. <property name="doseQuantity">
639. <!-- units should be coded value, but ISO datatypes do not support a coded value for the units of measure. We will work with the vMR group to address this. In the meantime, we are using coded values for all UOM from UCUM (https://phinvads.cdc.gov/vads/ViewCodeSystem.action?id=2.16.840.1.113883.6.8) until this issue is addressed ...-->
640. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="1" lowUnit="tbl" highValue="1" highUnit="tbl" lowIsInclusive="true" highIsInclusive="true"/>
641. </property>
642. <property name="deliveryRoute">
643. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.3.26.1.1.1" code="001" codeSystemName="FDA Routes of Administration" displayName="ORAL"/>
644. </property>
645. <property name="dosingPeriod">
646. <!-- This means once per day -->
647. <value xsi:type="PhysicalQuantityIntervalLiteral" lowValue="24" lowUnit="h" highValue="24" highUnit="h" lowIsInclusive="true" highIsInclusive="true"/>
648. </property>
649. <property name="dosingPeriodIntervalIsImportant">
650. <value xsi:type="BooleanLiteral" value="false"/>
651. </property>
652. </actionSentence>
653. </simpleAction>
654. </subElements>
655. </actionGroup>
656. </subElements>
657. </actionGroup>
658. <actionGroup>
659. <!-- Laboratory -->
660. <title value="Laboratory"/>
661. <representedConcepts>
662. <!-- Coded this as a laboratory tests group -->
663. <concept code="15220000" codeSystem="2.16.840.1.113883.6.96"
664. codeSystemName="SNOMED-CT" displayName="Laboratory Test (procedure)"/>
665. </representedConcepts>
666. <subElements>
667. <simpleAction xsi:type="CreateAction">
668. <textEquivalent value="B-type natriuretc peptide (BNP)"/>
669. <actionSentence xsi:type="ObjectExpression"
670. objectType="vmr:ProcedureProposal">
671. <property name="code">
672. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.1" code="42637-9" codeSystemName="LOINC" displayName="Natriuretic peptide B [Mass/​volume] in Blood"/>
673. </property>
674. </actionSentence>
675. </simpleAction>
676. <simpleAction xsi:type="CreateAction">
677. <textEquivalent
678. value="Complete blood cell count with automated white blood cell differential"/>
679. <actionSentence xsi:type="ObjectExpression"
680. objectType="vmr:ProcedureProposal">
681. <property name="code">
682. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.1" code="57021-8" codeSystemName="LOINC" displayName="CBC W Auto Differential panel in Blood"/>
683. </property>
684. </actionSentence>
685. </simpleAction>
686. </subElements>
687. </actionGroup>
688. <actionGroup>
689. <!-- Diagnostic Tests -->
690. <title value="Diagnostic Tests"/>
691. <representedConcepts>
692. <!-- Coded this as a diagnostic tests group -->
693. <concept code="103693007" codeSystem="2.16.840.1.113883.6.96"
694. codeSystemName="SNOMED-CT" displayName="Diagnostic procedure (procedure)"/>
695. </representedConcepts>
696. <subElements>
697. <simpleAction xsi:type="CreateAction">
698. <textEquivalent value="Radiograph, chest, 2 views"/>
699. <actionSentence xsi:type="ObjectExpression"
700. objectType="vmr:ProcedureProposal">
701. <property name="code">
702. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.96" code="399208008" codeSystemName="SNOMED-CT" displayName="Plain chest X-ray (procedure)"/>
703. </property>
704. </actionSentence>
705. </simpleAction>
706. <simpleAction xsi:type="CreateAction">
707. <textEquivalent value="12-lead ECG "/>
708. <actionSentence xsi:type="ObjectExpression"
709. objectType="vmr:ProcedureProposal">
710. <property name="code">
711. <value xsi:type="CodeLiteral" codeSystem="2.16.840.1.113883.6.96" code="268400002" codeSystemName="SNOMED-CT" displayName="12 lead ECG (procedure)"/>
712. </property>
713. </actionSentence>
714. </simpleAction>
715. </subElements>
716. </actionGroup>
717. <actionGroup>
718. <!-- Additional Orders -->
719. <title value="Additional Orders"/>
720. <subElements>
721. <!-- There are currently no elements or attibutes defined to help display these references to other order sets within this parent order set.
722. More input is needed from the community to determine whether such are needed. Such attributes should only be included here if their
723. values are specific to this order set; otherwise, the referenced order sets should be retrieved for any specific values needed to display
724. these references in the target system. In addition, note that these references should be resolved at the time that this artifact is imported
725. into the target system, with the referenced knowledge artifact being included "in-line" with the rest of this parent order set, or
726. represented with an internal system link, depending on the capabilities of the target system. -->
727. <actionGroupReference root="www.zynx.com/cds/orderset/RespiratoryProtocol"
728. extension="1234" version="2.7"/>
729. <actionGroupReference root="www.zynx.com/cds/orderset/RegularInsulinSlidingScale"
730. extension="43064" version="1.5"/>
731. </subElements>
732. </actionGroup>
733. </subElements>
734. </actionGroup>
735. </knowledgeDocument>