**$test.html**

**Test Page**

This is just a test file to check that $ works ok in file names

**actiondefinition-definitions.html**

**Action Definition Detailed Descriptions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**ActionDefinition**

**actiondefinition-examples.html**

**Action Definition Examples**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

For ActionDefinition examples, see the examples in [DecisionSupportRule](file:///C:\temp\decisionsupportrule-examples.html) and [OrderSet](file:///C:\temp\orderset-examples.html).

**actiondefinition-mappings.html**

**Action Definition Mappings**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page provides mappings for Action Definition.

**actiondefinition.html**

**Action Definition**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The ActionDefinition structure is used to define an action to be taken as part of some context such as the result of evaluating a decision support rule, or completing an order set for a patient. The structure is definitional in that it does not define a concrete action, but rather a template for some action that will be filled out in context.

For example, as part of a Suicide Risk Assessment OrderSet, a provider may choose to refer the patient to a Mental Health specialist. The OrderSet includes a definition of the action as a ProcedureRequest, but without contextual information such as the specific patient involved. As part of placing the OrderSet, the context is used to provide the additional information and complete the ProcedureRequest resource.

Each action definition may be a single action, or a group of actions, each specified as a sub-action using the same ActionDefinition structure. This hierarchical structure allows the ActionDefinition to be used in describing simple recommendations, all the way up through complex multi-level order sets.

The behavior element allows intended behavior of the actions to be described. For each behavior, a type specifies the kind of behavior being described, such as grouping or selection behavior. Each behavior type is then specified using a different value set as defined by the table below:

| **Behavior Type** | **Type Code** | **Possible Behaviors** |
| --- | --- | --- |
| Grouping Behavior | grouping | [Action Grouping Behavior Value Set](file:///C:\temp\valueset-action-grouping-behavior.html) |
| Selection Behavior | selection | [Action Selection Behavior Value Set](file:///C:\temp\valueset-action-selection-behavior.html) |
| Required Behavior | required | [Action Required Behavior Value Set](file:///C:\temp\valueset-action-required-behavior.html) |
| Precheck Behavior | precheck | [Action Precheck Behavior Value Set](file:///C:\temp\valueset-action-precheck-behavior.html) |
| Cardinality Behavior | cardinality | [Action Cardinality Behavior Value Set](file:///C:\temp\valueset-action-cardinality-behavior.html) |

**Content**

**Constraints**

**administration-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Administration Module**

**Introduction**

The Administrative module covers the base data that is then linked into the other modules for clinical content, finance/billing, workflow, etc.  
This of course is built on the FHIR technology platform modules.

Before any clinical data can be recorded, the basic information on the patient must be recorded, and then often the basis of the interaction (such as an encounter)

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| * [Patient](file:///C:\temp\patient.html) * [RelatedPerson](file:///C:\temp\relatedperson.html) * [Group](file:///C:\temp\group.html) * [Practitioner](file:///C:\temp\practitioner.html) * [PractitionerRole](file:///C:\temp\practitionerrole.html) | * [Organization](file:///C:\temp\organization.html) * [Location](file:///C:\temp\location.html) * [HealthcareService](file:///C:\temp\healthcareservice.html) * [Endpoint](file:///C:\temp\endpoint.html) | * [EpisodeOfCare](file:///C:\temp\episodeofcare.html) * [Encounter](file:///C:\temp\encounter.html) * [Account](file:///C:\temp\account.html) * [Flag](file:///C:\temp\flag.html) | * [Device](file:///C:\temp\device.html) * [DeviceComponent](file:///C:\temp\devicecomponent.html) * [DeviceMetric](file:///C:\temp\devicemetric.html) * [Substance](file:///C:\temp\substance.html) |

**Patient Registers**

Track people involved in receiving healthcare, the basics nearly everything else references back to

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |

*Note:* [Patient linking](file:///C:\temp\patient.html#links) should also be considered when evaluating searches with references to other resources. e.g. Searching for conditions for a patient.  
At present the specification does not define if the links should be also followed to include conditions that reference the linked patients too. We are currently seeking feedback on this.

*Note:* The Person resource may be used as a centralized register of people that may eventually be involved in healthcare, and could be used as the central coe demographics register.  
However the fields/values in Person are duplicated in the other resources, and in many cases the Person resource will be hosted on external systems.

**Clinical Categorization Resources**

Most clinical actitivies occur grouped in some way. Long term care is typically covered by an EpisodeOfCare, where short term care by encounters. Account associates the tracking of transactions back to a Patient (or other resource). Flag is just used to highlight a warning or other notification about a patient (or other resource)

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |

*Note:* Resources shown with a dotted box are described in other sections of the specification: Coverage and Claim are from the [section on Finance](file:///C:\temp\financial-module.html).

**Service Provider Directory Resources**

Service Directory resources are usually stored in the administration section of applications, and may even be synchronized from external systems.

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |

**Devices and Substances**

Other assets are often registered in the administration system, and maintained as master files.

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |

**Security and Privacy**

Patient privacy is handled with security labels and tags in the Resource [Meta](file:///C:\temp\resource.html#Meta) property. This is the standard way that the FHIR specification provides this supporting information to a sub-system that implements it (which is not defined by FHIR).

One of the more common use cases is for marking a patient as being a [celebrity](file:///C:\temp\security-labels.html).

Note that privacy considerations apply to Person, Practitioner and RelatedPerson records in addition to Patient's.

While Organization, Location, Device and other non-person-identifying records are generally subject to less stringent security precautions, such data must still be protected to avoid safety issues (e.g. someone maliciously changing the ingredients associated with a drug to cause/fail to cause alerts)

Devices can be linked to Patients. If this occurs, they must be protected as any other patient-linked element

For more general considerations, see [the Security and Privacy module](file:///C:\temp\secpriv-module.html).

**Common use Cases**

Administration Resources are cornerstone resources that are used by clinical and other domains of the FHIR Standard.

* **Managing a Master Record of a Patient and a Person** (e.g. MPI)  
  A [Patient](file:///C:\temp\patient.html) resource is used to describe patient demographic and visit information and any updates to it. It can be used to communicate [Patient](file:///C:\temp\patient.html) information to other systems (e.g. other registries, clinical, ancillary and financial systems). Some systems distinguish the Patient Registry (or Client Registry) from the Person Registry. A [Person](file:///C:\temp\person.html) resource is a base for the Person Registry system. The Patient/Person Management use case includes creation, update, as well as merge/unmerge and link/unlink scenarios.
* **Managing a Master Record of a Provider and Service Catalogue** (e.g. Provider Registry, Service Directory)  
  A [Practitioner](file:///C:\temp\practitioner.html) resource is a base resource for enabling the registry of individuals, related to providing health care services. Other resources, such as [Organization](file:///C:\temp\organization.html), [Location](file:///C:\temp\location.html), [HealthcareService](file:///C:\temp\healthcareservice.html), are creating a complete picture of where, how and by whom the care services are enabled to a patient. The resources can be used for managing the master record or as a reference in clinical resources to inform about participants and places for various clinical resources.
* **Managing Other Administrative Records**  
  The Administration domain of the FHIR standard includes creation and update of [Device](file:///C:\temp\device.html) and [Substance](file:///C:\temp\substance.html) records. Resources can be used for managing a master record or communicating its information to other systems.
* **Enabling Patient Profiles, Clinical Reporting, Connecting clinical records**  
  Administration Resources are referred by almost all clinical resources. Querying systems, using the references to Administration Resources enables the creation of profiles and reports of various complexities.
* **Enabling Clinical Grouping and Financial Reporting**  
  Other use cases are included in the roadmap of resources, developed by the Patient Administration group. The roadmap section lists plans and updates of the current work.

**Developmental Roadmap**

The Patient Administration is currently working through resources that support:

* Encounters and Scheduling *(enhance maturity of encounters and further develop in/outpatient scheduling)*
* Service Provider Directory *(in co-ordination with the Argonaut Provider Directory group)*
* Financial Management interactions *(account/coverage, then charge item, which links administration to billing)*

Many of the administrative resources are part of the core resources that most systems use first and have formed the basis for most people's first experiences with FHIR.  
However this limited exposure has still to be proven in all contexts, such as veterinary, public health and clinical research.

**administration.html**

**Administrative Resources**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| *Note: The PA work-group would appreciate contributions to enhance the appointment resource(s) to cover additional use-cases if required for implementations* | | |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |

Additional Resources will be added in the future. A list of hypothesized resources can be found on the [HL7 wiki](http://wiki.hl7.org/index.php?title=FHIR_Resource_Types). Feel free to add any you think are missing or engage with one of the [HL7 Work Groups](http://www.hl7.org/Special/committees/index.cfm) to submit a [proposal](http://wiki.hl7.org/index.php?title=Category:FHIR_Resource_Proposal) to define a resource of particular interest.

**backboneelement-definitions.html**

**BackboneElement - Detailed Descriptions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**BackboneElement**

**backboneelement.html**

**BackboneElement**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The base definition for elements defined as part of a resource definition. [Data Type](file:///C:\temp\datatypes.html) elements do not use this type. For instance, [Patient.contact](file:///C:\temp\patient.html#resource) is an element that is defined as part of the patient resource, so it automatically has the type BackboneElement.

Note that the descendent types of BackboneElement are all declared implicitly as part of the definitions of the resources.

**Content**

**Specializations**

This table lists all the specializations of BackboneElement that are defined, and the path at which they are defined.

**ballot-intro.html**

**STU 3 Ballot Welcome**

Hello and welcome to the third FHIR STU ballot!

This specification represents a series of significant changes and enhancements from the second FHIR Draft Standard for Trial Use specification HL7 published in October, 2015. It also includes changes resulting from committee meetings, connectathons, 1000's of change proposals, and collaborations with other standards organizations. A [summary of changes](file:///C:\temp\history.html#v1.6.0) and a [complete list of changes to resources and data types](file:///C:\temp\diff.html) are available, along with [transforms](file:///C:\temp\r2maps.html) between R2 and R3 for many resources.

FHIR is presently a "Specification for Trial Use" (STU) ballot. FHIR is no longer a "Draft" specification. Indeed, there are many implementations in production all across the world, or in preparation for production adoption. There are still some parts of the specification in an early stage of development, while other parts are quite mature and stable. In fact, as of this version, most DSTU 2 patient resources are still valid STU 3 Patient resources, though this is not a requirement - yet. We anticipate that some content will become normative on the next ballot after this one. For full details concerning the maturity process in the FHIR specification, see the [FHIR maturity Model](file:///C:\temp\resource.html#maturity).

The FHIR specification is presented as a series of interlinked HTML pages. They can either be reviewed online or can be [downloaded](file:///C:\temp\fhir-spec.zip) for exploration on your own device. (175MB zip, ~1GB unzipped). The scope of the FHIR Ballot is any page where the URL starts with http://hl7.org/fhir/2016Sep.

A few notes to consider:

* Pay close attention to the FMM level of the artifacts (as described above)
* Not all of the existing outstanding issues will be resolved prior to the STU ballot passing. Some may be left open to allow feedback from the early adopter community.
* This specification is complete enough to be implementable in a variety of healthcare scenarios, including supporting the "Common Meaningful Use Data Set" and other content from the Consolidated CDA implementation guide. However, the set of resources is not complete. Resources may evolve and new ones will be introduced over time. Refer to [FHIR Timelines](file:///C:\temp\timelines.html) for additional guidance on expectations around the evolution of the FHIR specification, or the road maps on the module pages.

**Balloting Rules**

HL7 ballot rules require that participants sign up prior to opening of the ballot. If you did not sign up in advance, you can still submit comments using the [Propose a Change](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemAdd&tracker_id=677) link at the bottom of each page of the specification. Feedback from balloters will be given priority, but all suggestions will be considered as much as time allows. (And be sure to [sign up](http://wiki.hl7.org/index.php?title=FHIR_email_list_subscription_instructions) to the FHIR list-server and/or follow the [#FHIR](https://twitter.com/hashtag/fhir) hash-tag so you don't miss the chance to vote in the next ballot cycle.)

If you are signed up to ballot, you can download the balloting spreadsheet from the [Ballot Desktop](http://www.hl7.org/participate/onlineballoting.cfm). All ballot feedback must be provided using the spreadsheet template provided. (There's a help tab that explains the meaning of each of the columns.) For FHIR, you have the option of making your comments directly in the spreadsheet or submitting your comment using the FHIR [Change Tracker](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemBrowse) tool. If you take the latter approach, you **must** include a reference to each tracker item in your ballot spreadsheet along with a vote (negative, affirmative typo, etc.). All spreadsheets must be submitted along with an overall vote by end of day Eastern time on the designated ballot closure date for the comments to be considered as part of ballot disposition.

Note: By using the tracker, you reduce some of the administrative effort of managing the ballot process. As well, you will receive notifications when comments are made on your ballot comment as well as when dispositions are made. Comments not submitted to the tracker as part of the ballot process will be migrated to the tracker after all ballot submissions are received.

When submitting your ballot feedback, if you have a general comment on something that you see occurring multiple times, please include at least a couple of specific locations where you see the issue. As much as possible, capture each separate concern as a distinct row in the ballot sheet or separate [tracker item](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemBrowse). (If using tracker items for your submissions, you **MUST** still submit a ballot spreadsheet referencing the relevant tracker items.) It makes our job of reconciling much easier. Also, don't forget to fill in the section numbers (gray numbers to the left of each heading) and URLs.

If you have questions that are interfering with the ability to review the specification or submit ballot comments, please contact one of the co-chairs of the FHIR Management Group: [Lloyd McKenzie](mailto:lloyd@lmckenzie.com) or [David Hay](mailto:david.hay25@gmail.com).

Thanks for taking the time to review the FHIR specification. We appreciate any feedback you can provide.

**Balloting Roadmap**

You can review which ever parts of the specification that are of interest in any order you wish. However, if you are not familiar with the FHIR specification, you may wish to follow this roadmap, by clicking on the [Documentation](file:///C:\temp\documentation.html) link and read through at least the following:

For context:

* [1 page Summary (Glossy)](file:///C:\temp\summary.html)
* [Developer's Introduction](file:///C:\temp\overview-dev.html)
* [Clinical Introduction](file:///C:\temp\overview-clinical.html)
* [Architect's Introduction](file:///C:\temp\overview-arch.html)
* [Overview & Roadmap](file:///C:\temp\overview.html)
* [FHIR Timelines](file:///C:\temp\timelines.html)

To see scope:

* Click on the [Resources](file:///C:\temp\resourcelist.html) link at the top to get a sense of what resources exist and maybe drill into a couple of them to see what they look like before coming back to the Documentation tab for more context

To understand base approach:

* [Base Resource](file:///C:\temp\resource.html)
* [DomainResource](file:///C:\temp\domainresource.html)
* [Narrative](file:///C:\temp\narrative.html)
* [Formats: XML, JSON](file:///C:\temp\formats.html)
* [Extensibility](file:///C:\temp\extensibility.html)

To understand key infrastructure:

* [Data Types](file:///C:\temp\datatypes.html)
* [Using Codes in Resources](file:///C:\temp\terminologies.html)

For context for this ballot:

* [Outstanding Issues](file:///C:\temp\todo.html)

Feel free to drill down into any other sections of interest. If you want to ensure a complete review of the whole spec, the [Table of Contents](file:///C:\temp\toc.html) may be a useful guide.

**book.html**

**Fast Health Interoperability Resources (FHIR) v**

**Warning: This version of FHIR is the STU ballot, and the stable version for the September/October connectathons. Implementers are welcome to experiment with the content defined here, but should note that the contents are subject to change without prior notice.**

**Table of Contents**

**cda-intro.html**

**Clinical Document Architecture (CDA) on FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 0 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**What is CDA on FHIR?**

CDA on FHIR specifies how to implement CDA R2 with the FHIR [Composition](file:///C:\temp\composition.html) resource.  
The original HL7 Clinical Document Architecture (CDA) defined the structure and semantics of "clinical documents" for the purpose of exchange. A clinical document is a documentation of clinical observations and services, with the following characteristics:

* Persistence - A clinical document continues to exist in an unaltered state, for a time period defined by local and regulatory requirements (NOTE: There is a distinct scope of persistence for a clinical document, independent of the persistence of any XML-encoded CDA document instance).
* Stewardship - A clinical document is maintained by an organization entrusted with its care.
* Potential for authentication - A clinical document is an assemblage of information that is intended to be legally authenticated.
* Context - A clinical document establishes the context for its contents.
* Wholeness - Authentication of a clinical document applies to the whole and does not apply to portions of the document without the full context of the document.
* Human readability - A clinical document is human readable.

A CDA document on FHIR is a defined and complete information object that can include text, images, sounds, and other multimedia content.

**Scope of the CDA on FHIR**

The scope of CDA on FHIR is the standardization of clinical documents for exchange.  
  
The data format of clinical documents outside of the exchange context (e.g. the data format used to store clinical documents) is not addressed in this specification.  
  
CDA on FHIR does not specify the creation or management of documents, only their exchange markup. While it may be possible to directly use the CDA Schema in a document authoring environment, such use is not the primary purpose of the CDA specification.  
  
Document management is critically interdependent with the CDA specifications, but the specification of document management messages is outside the scope of the CDA.

**Goals and Design Principles**

The goals of the CDA on FHIR are:

* Give priority to delivery of patient care.
* Allow cost effective implementation across as wide a spectrum of systems as possible.
* Support exchange of human-readable documents between users, including those with different levels of technical sophistication.
* Promote longevity of all information encoded according to this architecture.
* Enable a wide range of post-exchange processing applications.
* Be compatible with a wide range of document creation applications.
* Promote exchange that is independent of the underlying transfer or storage mechanism.
* Prepare the design reasonably quickly.
* Enable policy-makers to control their own information requirements without extension to this specification.

A number of design principles follow from consideration of the above goals:

* This architecture must be compatible with XML and JSON.
* This architecture must be compatible with representations of clinical information arising from other HL7 committees.
* Technical barriers to use of the architecture should be minimized.
* The architecture specifies the representation of instances required for exchange.
* The architecture should impose minimal constraints or requirements on document structure and content required for exchange.
* The architecture must be scalable to accommodate fine-grained markup such as highly structured text and coded data.
* Document specifications based on this architecture should accommodate such constraints and requirements as supplied by appropriate professional, commercial, and regulatory agencies.
* Document specifications for document creation and processing, if intended for exchange, should map to this exchange architecture.
* CDA documents must be human readable using widely-available and commonly-deployed XML-aware browsers and print drivers and a generic CDA style sheet written in a standard style sheet language.
* Use open standards.

**General CDA on FHIR Concepts**

**Major Components of a CDA on FHIR Document**

This section serves as a high-level introduction to the major components of a CDA document, all of which are described again and in greater detail later on. The intent here is to familiarize the reader with the high-level concepts to facilitate an understanding of the sections that follow. [EDITORS: in CDA r2 there is a bunch of detail about how CDA is wrapped - and an example. Consider whether the discussion is relevant here: "A CDA document is wrapped by the <ClinicalDocument> element, and contains a header..."]

**Human Readability and Rendering CDA Documents**

The CDA requirement for human readability guarantees that a receiver of a CDA document can algorithmically display the clinical content of the note on a standard Web browser.

* There must be a deterministic way for a recipient of an arbitrary CDA document to render the attested content.
* Human readability shall not require a sender to transmit a special style sheet along with a CDA document. It must be possible to render all CDA documents with a single style sheet and general-market display tools.
* Human readability applies to the authenticated content. There may be additional information conveyed in the document that is there primarily for machine processing that is not authenticated and need not be rendered.
* When structured content is derived from narrative, there must be a mechanism to describe the process (e.g. by author, by human coder, by natural language processing algorithm, by specific software) by which machine-processable portions were derived from a block of narrative.
* When narrative is derived from structured content, there must be a mechanism to identify the process by which narrative was generated from structured data.

These principles and requirements have led to the current approach, where the material to be rendered is placed into the Section.content...[EDITORS: current design doesn't make it clear where to consistently find narrative]

**change.html**

**Appendix: The Role of Informatics in the Shift from Reactive to Proactive Healthcare**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

FHIR offers more than a standard solving existing problems in interoperability, it provides a platform for the future. Interoperability is "all about the people"; to get past the peak of inflated expectations to the plateau of productivity on [the hype cycle](https://en.wikipedia.org/wiki/Hype_cycle) it is imperative that there is a shared vision of what is trying to be achieved. Healthcare is in the midst of a transformation driven by the convergence of biological and information revolutions and by economic imperative and social change. FHIR's strength lies in it being grounded in the real world which is changing rapidly. This appendix offers a high-level view of the disruption underway and helps place FHIR into a wider context.

Just as John Snow used a branch of informatics and geospatial analysis to identify the source of cholera in London and so prove the germ theory and revolutionize medicine in 1854, so too will the application of informatics and associated technology be instrumental in the next big change in healthcare.

In countries with advanced economies, health systems are facing increased demand from aging populations and increased incidence of chronic disease. In many of these countries, there is also a shortage of skilled workers. Mounting evidence points to avoidable errors causing serious harm to patients. Indeed optimal care only occurs about half the time in even the best performing health systems (1). Small incremental improvements will not be enough to address the looming crisis in sustainable healthcare.

Convergence of trends in health care, biology, informatics and technology together with the associated social changes and economic imperative is driving a paradigm shift (2) that may be the answer. Informatics has a role in most aspects of this. The figure below provides a summary of the convergence of these trends and the likely results of disruption that will move us from reactive to proactive healthcare.

In terms of our understanding of physiological pathways, informatics is now the major tool of molecular biology. For example, the time spent computing exceeds chemical analysis for gene sequencing by 3 to 1. Informatics is being used to map neural networks and to build the models of systems biology (3) with ever increasing levels of precision and complexity that can't be done without the help of machines. Our understanding has changed so much in recent times that the American Academy of Science is now arguing that it is time for a new taxonomy (4).

Personalization is occurring both because of social change and increased biological knowledge and is being facilitated by cheap mobile computing, sensors and devices. These social forces and the enabling technologies are allowing greater participation by 'non-experts' in decision making, treatment, discovery and knowledge management.

Greater knowledge about how we think (5), advances in the information sciences and the increased availability of computing power means that our capacity to acquire knowledge and use it to predict the course of pathology has increased enormously, and that is fortuitous, because the explosion of information is impossible to deal with otherwise. These new capabilities can be used to provide better advice and to better prevent disease through discovery, monitoring and treatment.

The health system itself can also benefit from what looks like a second phase of utilization of information technology through on-line care provision, real integrated measurement of quality and integration of knowledge in work-flow. With openness and transparency there is also the prospect of co-opetition (cooperative competition) and with integrated outcome measures that pay-for-success contracting can be used as an economic lever.

Large scale change in the way healthcare is done is both essential and inevitable. It is likely that this will derive from the merging of the knowledge and machines of the biological and information revolutions facilitating a shift from reactive treatment to proactive personalized medicine. Only by such significant changes could the needed quantum improvement in the effectiveness and efficiency of healthcare be made.

The development of digital technology has disrupted other sectors, notably media, retail and manufacturing, and the health sector is unlikely to remain immune (6). Digitization of biology and health will allow machines to help, lead to a demystification of disease, the democratization of healthcare, and a move from the treatment of disease to the promotion and maintenance of wellness.

**References**

1. CareTrack: assessing the appropriateness of health care delivery in Australia. Runciman WB, Hunt TD, Hannaford NA, Hibbert PD, Westbrook JI, Coiera EW, Day RO, Hindmarsh DM, McGlynn EA, Braithwaite J. 2, 16 July 2012, Med J Aust. 2012 Jul 16;197(2):100-5. PubMed PMID: 227, Vol. 197(2), pp. 100-5
2. Kuhn, Thomas S. The Structure of Scientific Revolutions. Chicago : University of Chicago Press, 1962
3. Hood, Leroy. Institute of Systems Biology. [Online] [Cited: 25 May 2013.] <https://www.systemsbiology.org/leroy-hood>
4. Board on Life Sciences, Committee on A Framework for Developing a New Taxonomy of Disease. Toward Precision Medicine; Building a Knowledge Network for Biomedical Research and a New Taxonomy of Disease. Washington, DC : The National Academic Press, 2011
5. Kahneman, Daniel. Thinking, Fast and Slow. New York : Farrar, Straus and Giroux, 2011
6. Topol, Eric. The Creative Destruction of Medicine - How the Digital Revolution Will Create Better Health Care. New York : Basic Books, 2012.

This page maintained by [Michael Legg](mailto:michael.legg@mlanda.com.au)

**changelist.html**

**FHIR Change History**

**clinical.html**

**Clinical Resources**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |

Additional Resources will be added in the future. A list of hypothesized resources can be found on the [HL7 wiki](http://wiki.hl7.org/index.php?title=FHIR_Resource_Types). Feel free to add any you think are missing or engage with one of the [HL7 Work Groups](http://www.hl7.org/Special/committees/index.cfm) to submit a [proposal](http://wiki.hl7.org/index.php?title=Category:FHIR_Resource_Proposal) to define a resource of particular interest.

**clinicalreasoning-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Clinical Reasoning**

*Note to implementers: The user-facing external clinical decision support use case supported by the ServiceDefinition/$evaluate operation of the Clinical Reasoning module has significant overlap with the functionality provided by the*[*CDS Hooks*](http://cds-hooks.org)*specification. As part of FHIR STU4, it is the intention to unify the CDS Hooks specification with the Clinical Reasoning module, ensuring that implementers have a single consistent mechanism to support this use case that meets the requirements of both the Clinical Quality Framework and CDS Hooks communities. Although the functionality in both specifications is conceptually aligned, this unification will likely result in changes to both specifications. The CQF and CDS Hooks project teams are committed to this unification and will work to ensure that the resulting changes have as little impact as possible on current and ongoing implementation efforts, while meeting the needs of both communities. The project team is planning on a workable implementation tested at the connectathon in May of 2017, with a trial ballot following in September 2017, targeting publication as part of the STU4 FHIR specification.*

**Introduction**

The Clinical Reasoning module provides resources and operations to enable the representation, distribution, and evaluation of clinical knowledge artifacts such as clinical decision support rules, quality measures, order sets, and protocols. In addition, the module describes how expression languages can be used throughout the specification to provide dynamic capabilities.

Clinical Reasoning involves the ability to represent and encode clinical knowledge in a very broad sense so that it can be integrated into clinical systems. This encoding may be as simple as controlling whether or not a particular section of an order set appears based on the conditions that a patient has, or it may be as complex as representing the care pathway for a patient with multiple conditions.

**Scope**

The Clinical Reasoning module focuses on enabling two primary use cases:

1. **Sharing** - The ability to represent clinical knowledge artifacts such as decision support rules, order sets, protocols, and quality measures, and to do so in a way that enables those artifacts to be shared across organizations.
2. **Evaluation** - The ability to evaluate clinical knowledge artifacts in the context of a specific patient or population, including the ability to request decision support guidance, impact clinical workflow, and retrospectively assess quality metrics.

To enable these use cases, the module defines several components that can each be used independently, or combined to enable more complex functionality. These components are:

* **Expression Logic** - the representation of logic using languages such as FHIRPath and Clinical Quality Language (CQL).
* **Definitional Resources** - resources that are not defined on any specific patient, but are used to define the actions to be performed as part of a clinical knowledge artifact such as an order set or decision support rule. These resources can be used directly, or with profiles to provide intended structure for specific types of resources.
* **Knowledge Artifacts** - representation of clinical knowledge such as decision support rules and clinical quality measures.

These basic components can then be used to enable a broad variety of clinical decision support and quality measurement use cases, including knowledge sharing, decision support services, and clinical quality assessment and reporting. The topics below provide more detailed discussion on each of these components and uses:

| **Topic** | **Description** |
| --- | --- |
| [Overview and Background](file:///C:\temp\main-pages.html#background) | If you are interested in the background and development of the FHIR Clinical Reasoning module, this topic covers where it came from and why it exists. See also the general FHIR introductions for [clinicians](file:///C:\temp\overview-clinical.html), [developers](file:///C:\temp\overview-dev.html) or [architects](file:///C:\temp\overview-arch.html) |
| [Using Expressions](file:///C:\temp\cqif\cqif-topics-using-expressions.html) | If you want to see how to add dynamic capabilities to FHIR resources using expressions, start here. |
| [Definitional Resources](file:///C:\temp\cqif\cqif-topics-definitional-resources.html) | If you want to see how to describe definitional resources using the ActivityDefinition resource, start here. |
| [Representing Knowledge Artifacts](file:///C:\temp\cqif\cqif-knowledge-artifact-representation.html) | If you want to represent knowledge artifacts such as Event-Condition-Action rules, Order Sets, or Protocols, start here. |
| [Sharing Knowledge Artifacts](file:///C:\temp\cqif\cqif-knowledge-artifact-distribution.html) | If you want to share and distribute knowledge artifacts, start here. |
| [Clinical Decision Support Service](file:///C:\temp\cqif\cqif-cds-on-fhir.html) | If you want to use the Clinical Reasoning module to provide or use Clinical Decision Support services, start here. |
| [Quality Reporting](file:///C:\temp\cqif\cqif-quality-reporting.html) | If you want to define or report clinical quality measures, start here. |

**Audience**

From the perspective of a Knowledge Author, this module describes an approach to representing knowledge artifacts within FHIR.

From the perspective of a Knowledge Content Provider, this module defines search functionality for using a FHIR server as a knowledge artifact repository.

From the perspective of a Knowledge Evaluation Service Provider, this module defines operations and profiles in support of evaluating quality measures, and defining a service for guidance request and response, consistent with the approach taken by the current [Decision Support Service](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=12) specification.

And finally, from the perspective of a Knowledge Evaluation Service Consumer, this module defines the expected available operations and behavior of a knowledge evaluation service.

**Index**

**Resources**

| **Resource** | **Description** |
| --- | --- |
| [ActivityDefinition](file:///C:\temp\activitydefinition.html) | A resource to represent definitional resources. |
| [DataRequirement](file:///C:\temp\metadatatypes.html#datarequirement.html) | A data type that represents a general data requirement for a knowledge asset such as a decision support rule or quality measure. |
| [PlanDefinition](file:///C:\temp\plandefinition.html) | Represents the description of a plan for accomplishing a particular goal. This resource is used to represent a broad variety of clinical knowledge artifacts including decision support rules, order sets, and protocols. |
| [Library](file:///C:\temp\library.html) | Provides a container for knowledge artifacts that includes logic libraries, model definitions, and asset collections. |
| [ServiceDefinition](file:///C:\temp\servicedefinition.html) | Describes a specific clinical decision support service and supports evaluation using the $evaluate operation. |
| [GuidanceResponse](file:///C:\temp\guidanceresponse.html) | Represents the response to a specific guidance request returned by the $evaluate operation. |
| [Measure](file:///C:\temp\measure.html) | Represents a clinical quality measure and provides evaluation through the $evaluate-measure operation. |
| [MeasureReport](file:///C:\temp\measurereport.html) | Represents the response to a specific measure evaluation request returned by the $evaluate-measure operation. |

**Extensions**

| **Extension** | **Description** |
| --- | --- |
| [cqlExpression](file:///C:\temp\cqif\extension-cqif-cqlexpression.html) | A general purpose extension that supports the use of Clinical Quality Language within FHIR |
| [fhirPathExpression](file:///C:\temp\cqif\extension-cqif-fhirpathexpression.html) | A general purpose extension that supports the use of FHIRPath within FHIR |
| [library](file:///C:\temp\cqif\extension-cqif-library.html) | A general purpose extension that supports the declaration of dependencies that can be accessed by expression logic |
| [qualityOfEvidence](file:///C:\temp\cqif\extension-cqif-qualityofevidence.html) | An extension that can be applied to Attachments to indicate the quality of evidence in support of a particular artifact or recommendation. |
| [strengthOfRecommendation](file:///C:\temp\cqif\extension-cqif-strengthofrecommendation.html) | An extension that can be applied to Attachments to indicate the strength of a recommendation. |
| [optionCode](file:///C:\temp\cqif\extension-cqif-optioncode.html) | An extension that can be applied to questionnaire response options to provide meaning to the given option. |
| [sourceValueSet](file:///C:\temp\cqif\extension-cqif-sourcevalueset.html) | An extension that can be applied to coding elements to indicate the intended or actual source value set for the coding. |

**Profiles**

| **Profile** | **Description** |
| --- | --- |
| [CQIF-Questionnaire](file:///C:\temp\cqif\cqif-questionnaire.html) | Defines extensions to the base Questionnaire that allow it to be used as a DocumentationTemplate with behavior specified via logic in CQL libraries. |

**Services**

| **Service** | **Description** |
| --- | --- |
| [Knowledge Repository](file:///C:\temp\capabilitystatement-knowledge-repository.html) | Defines minimum service capabilities for a knowledge repository |
| [Measure Processor](file:///C:\temp\capabilitystatement-measure-processor.html) | Defines minimum service capabilities for a measure processor |

**Security and Privacy**

Because Knowledge Artifacts are typically patient-independent, many of the resources in the clinical reasoning module have no patient security and privacy concerns beyond the normal sensitivity that should be paid in any electronic healthcare system environment. However, the evaluation use case, including decision support guidance request/response, as well as quality measure evaluation have significant patient security and privacy concerns.

For the clinical decision support evaluation use case, as with any patient-specific information, care should be taken to ensure that the request and response are properly secured both at rest and in-motion, and that all access to the patient's information is done via a properly authenticated and authorized mechanism. This is particularly true of decision support artifacts where the logic is ingested as part of the definition of the artifact. In this scenario, the evaluation engine must ensure that data access within the ingested logic is subject to the same authentication and authorization requirements as any other access.

For guidance services that receive patient information, ensure that logging and auditing trails do not inadvertently compromise patient privacy and security by logging potentially sensitive information in an unencrypted way. In addition, guidance and recommendations returned from the service must ensure that content that contains patient information is clearly indicated so that consuming clients can take the appropriate care in integrating and displaying the resulting guidance.

For quality measure evaluation, individual and patient-list reports have the potential to contain large amounts of patient information. As with the decision support guidance responses, care must be taken to ensure the patient information is only accessible to properly authenticated and authorized agents, and that inadvertent breaches are minimized by following appropriate logging and auditing protocols.

In particular, because expression languages, depending on their power and scope, can provide the ability to access large amounts of data, as well as the potential for infinite recursion or looping, care should be taken to ensure that implementations adequately safeguard against Denial-of-Service-style attacks that leverage these capabilities to compromise systems by overloading capacity.

For more general considerations, see [the Security and Privacy module](file:///C:\temp\secpriv-module.html).

**Common use Cases**

| **Use Case** | **Description** |
| --- | --- |
| [Providing a dynamic value for a resource element](file:///C:\temp\cqif\cqif-topics-using-expressions.html#dynamic-value-for-a-resource-element) | Using expressions to define the value for an element of a FHIR resource. |
| [Defining a CQL library](file:///C:\temp\cqif\cqif-knowledge-artifact-representation.html#defining-a-cql-library) | Using the Library resource to incorporate a Clinical Quality Language library for use in FHIR resources. |
| [Defining a Model Definition artifact](file:///C:\temp\library-quick-model-definition.xml.html) | Using the Library resource to incorporate the definition of an information model for use with expressions in FHIR. |
| [Defining an Event Condition Action rule](file:///C:\temp\cqif\cqif-knowledge-artifact-representation.html#event-condition-action-rule) | Using the PlanDefinition resource to represent an event-condition-action rule in FHIR. |
| [Defining a Referral Request activity](file:///C:\temp\cqif\cqif-topics-definitional-resources.html#referral-request-activity-definition) | Using the ActivityDefinition resource to define a referral request activity that can be used as part of a knowledge artifact. |
| [Defining a Medication Request activity](file:///C:\temp\cqif\cqif-topics-definitional-resources.html#referral-request-activity-definition) | Using the ActivityDefinition resource to define a medication request activity that can be used as part of a knowledge artifact. |
| [Defining an Order Set](file:///C:\temp\cqif\cqif-knowledge-artifact-representation.html#order-set) | Using the PlanDefinition resource to represent an order set. |
| [Defining a Protocol](file:///C:\temp\cqif\cqif-knowledge-artifact-representation.html#protocol) | Using the PlanDefinition resource to represent a protocol. |
| [Defining a Questionnaire with dynamic content](file:///C:\temp\cqif\cqif-knowledge-artifact-representation.html#documentation-template) | Using the Questionnaire resource and expression extensions to add dynamic functionality to a FHIR Questionnaire. |
| [Defining a Service Module](file:///C:\temp\cqif\cqif-cds-on-fhir.html#service-modules) | Using the ServiceDefinition resource to define a decision support service module. |
| [Obtaining guidance from a Decision Support Service](file:///C:\temp\cqif\cqif-cds-on-fhir.html#guidance-request) | Using the $evaluate operation to request and process guidance from a decision support service. |
| [Defining a Measure](file:///C:\temp\cqif\cqif-quality-reporting.html#representing-quality-measures) | Using the Measure resource to represent a clinical quality measure. |
| [Evaluating a Measure](file:///C:\temp\cqif\cqif-quality-reporting.html#invoking-measures) | Using the $evaluate-measure operation to request calculation of a clinical quality measure. |
| Applying an ActivityDefinition | Using the $apply operation to realize the intent resource defined by an ActivityDefinition. |
| Applying a PlanDefinition | Using the $apply operation to realize a plan definition for a specific context. |
| [Representing Quality of Evidence/Strength of Recommendation](file:///C:\temp\cqif\cqif-topics-supporting-documentation.html) | Using the qualityOfEvidence and strengthOfRecommendation extensions to indicate ratings associated with evidence for a particular artifact or recommendation. |

**Developmental Roadmap**

The resources defined for the Clinical Reasoning module are the result of the combined efforts of multiple communities working on the shared goal of harmonized standards and specifications for clinical decision support and quality measurement artifacts. The current state of the module reflects changes incorporated both from previous ballots on the FHIR-specific material, as well as content derived from several other balloted specifications in the CDS and CQM domains. The content at this point is capable of supporting the two primary use cases of sharing and evaluation in both domains and for a broad variety of artifacts.

In particular, the use of Clinical Quality Language (CQL) as a foundational mechanism for representing clinical quality logic enables decision support and quality measurement artifacts to share common definitions. For example, a Chlamydia Screening measure and related decision support artifacts focused on improving the measure can share a common library that describes the criteria for detecting when Chlamydia Screening is required in a patient. The decision support rule applies these criteria to determine when and how to impact a workflow, while the quality measure uses these same criteria to determine whether the screening goal has been met for a patient or population. In addition, the resources defined in this module use common patterns for describing the structure of artifacts and their associated metadata, enabling a consistent approach to the sharing and distribution of clinical knowledge artifacts.

The goals of the module over the next year are to provide a stable basis for implementation of the sharing and evaluation use cases. The Clinical Quality Framework Initiative will use these resources as the basis for implementation projects, targeting an FMM level of 3 or 4 for all module resources within a year. We are actively seeking comments on all areas of the module, with particular focus on supporting the following scenarios:

* The use of FHIRPath and CQL to provide dynamic functionality throughout FHIR
* FHIR-Based Knowledge Artifact Repository
* FHIR-Based Clinical Decision Support Service
* FHIR-Based Quality Measure Evaluation and Reporting
* FHIR-Based Order Set Catalog/Ordering Services
* The use of FHIR operations as the protocol for defining artifact evaluation requests and responses. In particular, the ability to bundle multiple requests into a single operation, as well as the use of a generic FHIR operation for any service module evaluation, versus defining an evaluation specific to a given knowledge artifact.
* The use of FHIR interactions in general as a mechanism for enabling knowledge artifact repository and distribution functionality.

**Background**

The FHIR Clinical Reasoning module is sponsored by the Clinical Decision Support (CDS) and Clinical Quality Information (CQI) HL7 Work Groups, with input and coordination from the FHIR Infrastructure and Service Oriented Architecture HL7 Work Groups.

The guidance in this module is prepared as a Universal Realm Specification with support from the [Clinical Quality Framework (CQF) initiative](http://cqframework.info), which is a public-private partnership sponsored by the Centers for Medicare & Medicaid Services (CMS) and the U.S. Office of the National Coordinator for Health Information Technology (ONC) to identify, develop, and harmonize standards for clinical decision support and electronic clinical quality measurement.

The Clinical Quality Framework initiative has focused on harmonizing the historically disjointed specifications used by the Clinical Quality Measurement and Clinical Decision Support communities. Specifically, the initiative has focused on the specifications used to represent knowledge artifacts within the two communities. The strategy employed has been to break the conceptual content of knowledge artifacts into three core components, to define common standards for these core components, and to re-use these common standards for both clinical decision support and clinical quality measurement:

* Metadata - Descriptive information about the artifact and its content
* Clinical Information - Information about a patient or population of concern within a given artifact
* Logic - The clinical reasoning involved in an artifact

The first component has resulted in the [Clinical Quality Common Metadata Conceptual Model](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=391), an informative document harmonizing metadata requirements between Quality Measurement and Decision Support artifacts.

The second component has resulted in the QUICK [Conceptual and Logical](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=378) Models, a harmonization of the [Virtual Medical Record (vMR)](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=338) used in Decision Support and the [Quality Data Model (QDM)](https://ecqi.healthit.gov/qdm) used in Quality Measurement, and with its core requirements realized in FHIR as the Quality Improvement Core (QICore) [profiles](http://hl7.org/fhir/qicore/qicore.html#contents). Ongoing work in this area is focusing on coordination with the Clinical Information Modeling Initiative (CIMI) and a methodology for producing FHIR profiles from CIMI models. Currently, the QICore FHIR profiles (which are in turn derived from the US-Core profiles) can be used to model clinical quality data, and to present a consistent model for use in authoring and evaluating clinical quality artifacts.

Finally, the third component has resulted in the [Clinical Quality Language specification](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=400), a harmonization of the expressive capabilities of the [Clinical Decision Support Knowledge Artifact Specification (CDS KAS)](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=337) (produced by the [Health eDecisions](http://healthedecisions.org) (HeD) Standards and Interoperability (S&I) initiative), and the [Health Quality Measures Format (HQMF)](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=97).

As part of the ongoing CQF initiative pilot efforts, these developing specifications are being used to support knowledge artifact sharing, as well as evaluation of knowledge artifacts as part of decision support request/response and measure evaluation.

This module continues the harmonization of quality domain specifications by defining an approach to using a FHIR server as a component of a knowledge system in both the Knowledge Repository and Knowledge Evaluation Service roles.

**Related Specifications**

The approach and representations within this guide are derived from and intended to be consistent with the following specifications:

* [HL7 Version 3 Standard: Clinical Decision Support Knowledge Artifact Specification, Release 1.3](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=337)
* [HL7 Version 3 Standard: Representation of the Health Quality Measure Format (eMeasure) DSTU, Release 2.1](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=97)
* [HL7 Version 3 Standard: Decision Support Service, Release 2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=12)
* [HL7 Version 3 Implementation Guide: Decision Support Service, Release 1](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=334)

**Copyright Information**

This material includes SNOMED Clinical Terms ® (SNOMED CT®), which are 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 Logical Observation Identifiers Names and Codes (LOINC®) (<http://loinc.org>). The LOINC table, LOINC codes, and LOINC panels and forms file are copyright © 1995-2016, Regenstrief Institute, Inc. and the LOINC Committee and available at no cost under the license at <http://loinc.org/terms-of-use>.

This material contains content from the Unified Code for Units of Measure (UCUM) (<http://unitsofmeasure.org>). The UCUM specification is copyright © 1999-2014, Regenstrief Institute, Inc. and available at no cost under the license at <http://unitsofmeasure.org/trac/wiki/TermsOfUse>.

This material contains quality measure content developed by the National Committee for Quality Assurance (NCQA). The measure content is copyright (c) 2008-2013 National Committee for Quality Assurance and used in accordance with the NCQA license terms for non-commercial use.

**Acknowledgements**

The guidance in this module is the work of a joint project between the HL7 Clinical Quality Information and Clinical Decision Support Work Groups with the co-sponsorhip of the FHIR Infrastructure, Implementable Technology Specifications, and Service Oriented Architecture Work Groups.

**clinicalsummary-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Clinical Summary Module**

**Introduction**

This Clinical Summary Module focuses on the FHIR Resources that represent core clinical information for a patient. The information contained in these Resources are those frequently documented, created or retrieved by healthcare providers during the course of clinical care. Resources generated during the course of diagnostic studies can be found in the [Diagnostics Module](file:///C:\temp\diagnostics-module.html), whereas the Resources related to medication ordering and administration process can be found in the [Medications Module](file:///C:\temp\medications-module.html).

As an introduction to FHIR APIs and Resources, please see the [DeveloperÃ¢â‚¬â„¢s Introduction](file:///C:\temp\overview-dev.html) or [Clinical Introduction](file:///C:\temp\overview-clinical.html) in the Overview section of the [Foundation Module](file:///C:\temp\foundation-module.html).

**Index**

The Clinical Summary Module covers the following resources:

|  |  |  |
| --- | --- | --- |
| * [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) * [AdverseEvent](file:///C:\temp\adverseevent.html) * [Condition](file:///C:\temp\condition.html) (Problem) * [Procedure](file:///C:\temp\procedure.html) * [FamilyMemberHistory](file:///C:\temp\familymemberhistory.html) * [ClinicalImpression](file:///C:\temp\clinicalimpression.html) | * [DetectedIssue](file:///C:\temp\detectedissue.html) * [CarePlan](file:///C:\temp\careplan.html) * [CareTeam](file:///C:\temp\careteam.html) * [Goal](file:///C:\temp\goal.html) * [ReferralRequest](file:///C:\temp\referralrequest.html) | * [ProcedureRequest](file:///C:\temp\procedurerequest.html) * [RiskAssessment](file:///C:\temp\riskassessment.html) * [VisionPrescription](file:///C:\temp\visionprescription.html) * [NutritionOrder](file:///C:\temp\nutritionorder.html) |

FHIR Resources have a low, moderate or high levels of complexity with respect to the number of primary and child elements as well as the number of referenced Resources, found in this module and others. To better understand the relationships between Resources, we recommend beginning with the lower complexity, core Resources such as [Patient](file:///C:\temp\patient.html), [Condition](file:///C:\temp\condition.html), and [FamilyMemberHistory](file:///C:\temp\familymemberhistory.html) before addressing a high complexity Resource such as [CarePlan](file:///C:\temp\careplan.html).

**Security and Privacy**

The diagnostic resources often represent patient-related data, and as such are susceptible to data breaching. Necessary privacy and security provision must be in place for searching and fetching this information. For more general considerations, see the [Security and Privacy module](file:///C:\temp\secpriv-module.html).

**Common use Cases**

* **Documenting a patientÃ¢â‚¬â„¢s condition** Ã¢â‚¬â€œ The [Condition](file:///C:\temp\condition.html) Resource is used extensively throughout FHIR Resources to associate information and activities with specific conditions. The [Condition](file:///C:\temp\condition.html) Resource is broadly defined to include problems, diagnoses and health concerns.
* **Retrieving a list of the patientÃ¢â‚¬â„¢s Problem List**
* **Documenting and retrieving the patientÃ¢â‚¬â„¢s allergies** - The [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) Resource is used to represent the patientÃ¢â‚¬â„¢s allergy or intolerance to a substance. There is vibrant debate within clinical community regarding what is appropriate to document as an allergy or intolerance. These terms are used both formally by the Allergy and Immunology community as well as informally by patients leading to confusion. Readers are referred to the Resource definition for the Scope and Usage of this resource. The AllergyIntolerance Resource also supports the documenting of the absence of an allergy.
* **Family History** Ã¢â‚¬â€œ The [FamilyMemberHistory](file:///C:\temp\familymemberhistory.html) Resource can be used to document know conditions of family members and support the creation of pedigrees.
* **Care Plans** Ã¢â‚¬â€œ The [CarePlan](file:///C:\temp\careplan.html) resource supports a problem based care plan with references to other Resources including [CareTeam](file:///C:\temp\careteam.html), [Condition](file:///C:\temp\condition.html), [Goal](file:///C:\temp\goal.html), and activities such as [ReferralRequest](file:///C:\temp\referralrequest.html)

**Developmental Roadmap**

Over the next 18 months, we will continue to advance the resources through the [Maturity Levels](file:///C:\temp\resource.html#maturity) through the process of development and testing of the Resources. We anticipate more widespread implementation of core Resources such as Condition. Complex Resources such as CarePlan are dependent on the maturation of its referred Resources and are expected to mature more gradually. The clinical community will need to develop use cases to test further mature the [ReferralRequest](file:///C:\temp\referralrequest.html) and [ProcedureRequest](file:///C:\temp\procedurerequest.html) Resources at opportunities such as the [Clinicians on FHIR sessions](http://wiki.hl7.org/index.php?title=Clinicians_on_FHIR) at the HL7 Working Group Meetings.

**comparison-cda.html**

**Clinical Document Architecture (CDA)**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

[CDA](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7) is HL7's most widely adopted [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) standard. It provides both a standardized header containing metadata about the document as well as the ability to convey a wide variety of clinical content organized into various sections. The document content can be un-encoded, such as a PDF through to a fully encoded HL7 v3 instance.

NOTE: While FHIR can be used to create [documents](file:///C:\temp\documents.html) using the [Composition Resource](file:///C:\temp\composition.html), FHIR can also be used to exchange traditional CDA R2 documents making use of the [DocumentReference](file:///C:\temp\documentreference.html) resource, and handling the CDA document itself as a binary attachment (as XDS does).

**CDA Similarities and Differences**

**Clinical document focus:** As its name implies, Clinical Document Architecture is limited to "clinical" use cases. The CDA model does not support exchange of content not deemed to have clinical relevance, such as financial information and is limited to documents that deal with patients. (In some cases, such as the HL7 Structured Product Labeling standard, non-patient-specific CDA-like specifications are created to get around this limitation.) FHIR documents have no limitation on their content and can have subjects other than patients.

**Human readability approach:** CDA and FHIR both require that content be human-readable and define specific rules for how the human readable text is presented.

**Clinical Statement vs. resources:** In CDA, the "content" of the document is expressed using a complex and extremely abstract model based on HL7's "Clinical Statement" project. Its purpose is to allow implementers to express pretty much any clinical concept in any degree of rigor and granularity. (In practice, there are limitations built into the CDA model that make the expression of certain clinical concepts challenging). This model provides significant power, but also presents challenges. The first is that RIM modeling expertise is required in order to express any particular piece of clinical information. It isn't obvious how to represent things like allergies or surgery or blood pressure "out of the box". Templates are required to support interoperability. The second is that common clinical concepts can be (and frequently are) modeled differently in different circumstances. With FHIR, all clinical (and non-clinical) content in a message is handled by referencing existing resource definitions. These resources make it clear how to represent common structures like allergies and blood pressure "out of the box" and ensure that there's only one way for core content to be represented. It does however create the limitation that an appropriate resource must have been defined in order to share content. In the early stages of FHIR development, it may be necessary to make use of the [Basic](file:///C:\temp\basic.html) resource if an appropriate standard resource has not yet been defined.

**Templates and Profiles:** As discussed above, CDA relies on the presence of templates in order to understand the meaning of instances. While the meaning can theoretically be determined by looking at RIM attributes and codes, the reality is that this is often not safe or sufficient. As such, pretty much every CDA instance includes *templateId* attribute values scattered throughout the instance to define meaning. With FHIR, meaning is defined by the resource. [Profiles](file:///C:\temp\profiling.html) can be used to define extensions, but they never refine the meaning of core elements. While the profiles used in constructing a particular instance can be declared within the instance via [tags](file:///C:\temp\resource.html#tag). However, such declaration is not required.

**Mark-up language:** CDA defines its own XML syntax for narrative content, loosely based on HTML. FHIR makes use of a constrained set of XHTML which is somewhat more expressive than the CDA markup. Conversions from FHIR to CDA will need to take these constraints into account (or alternatively provide a fully rendered version of the document).

**CDA Interoperability Considerations**

CDA is a type of [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) specification. Therefore, all [considerations](file:///C:\temp\comparison-v3.html#V3-interoperability) that apply to v3 messaging also apply to CDA. In addition, the following topics are specific to CDA implementations.

**What to map:** The right-hand side (clinical content) portion of the CDA model qualifies as an abstract model [as discussed above](file:///C:\temp\comparison-v3.html#V3-abstractModels). While the CDA header can reasonably be mapped to the HL7 [Composition](file:///C:\temp\composition.html) resource and related resources, mappings between FHIR and CDA should be done at the template level rather than the CDA specification itself.

**Human readable granularity:** With FHIR, narrative only exists for the resources at the root of each section. With CDA, narrative exists for each section. Usually this means the narrative in CDA and FHIR will correspond. However, in some cases, a section will contain other sub-sections. In CDA, these "container" sections can have narrative. In FHIR, they cannot. Applications will need to have some way of managing this if converting.

**Discrete to human-readable linkages:** To ensure semantic traceability, both FHIR and CDA allow establishing linkages between text in the narrative and specific discrete elements in the encoded part of a document. If converting between FHIR and CDA, these linkages need to be converted as well. However, this is complicated by the fact that the granularity at which linkages can occur is different between the two specifications. In CDA, linkages can only occur at the level of a section or one of a couple of the entry types. With FHIR, linkages can occur at any level at all, including individual data type components or even portions of extensions. Converting from CDA to FHIR will be straight-forward, however there will be information loss when converting the other way.

**comparison-other.html**

**Other HL7 Standards**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

HL7 has produced a number of other standards that don't overlap with FHIR as closely as those listed above, primarily because they aren't focused solely on information exchange. However, they deserve a brief mention:

**EHR Functional Model (EHR-FM)**

This specification defines a number of functional behaviors for Electronic Health Record systems. FHIR is a healthcare information exchange standard that can be used to satisfy some of these functional behaviors. Details on how FHIR fits into the EHR-FM can be found [here](file:///C:\temp\ehr-fm.html)

**Context Management Specifications (CCOW)**

CCOW is a standard for allowing independent systems to synchronize context on a single workstation, providing a seamless interface for the user of that workstation (e.g. ensuring consistent user authentication, display of the same patient, display of the same order, etc.) In theory, FHIR resources could be used as an alternative CCOW implementation technology, however the business case for doing this is not clear. CCOW profiles include [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) mappings. These mappings can be used to help identify the equivalent FHIR data elements when establishing CCOW linkages in FHIR-based systems.

**Arden Syntax**

Arden Syntax is a language for defining decision support rules. These rules make reference to data elements that are used as part of the decision making process. However, the specification does not define how these data elements are identified. FHIR element and extension identifiers would provide one mechanism for identifying the relevant data elements.

**Virtual Medical Record**

The Virtual Medical Record is a draft specification under development by HL7 that also serves the decision support space. It defines a logical medical record that decision support rules can be constructed against. At present, this model is a custom model created specifically for VMR. However, the Decision Support work group is evaluating the possibility of using FHIR as a structure for future versions of the specification.

**comparison-v2.html**

**HL7 Version 2**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

[HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) was HL7's first information exchange standard and is one of its most widely adopted, being prominent in in-patient settings throughout the world, though also used in a variety of other contexts as well. HL7 v2 uses messages composed of re-useable segments to communicate healthcare-related information between a sending and receiving system as well as to invoke particular behavior (patient transfers, lab orders, etc.) It also supports one-way communication through notifications, provides support for queries and other workflow.

**HL7 v2 Similarities and Differences**

**Event-based:** FHIR supports an event-based messaging paradigm similar to the HL7 v2 messaging structure (though unlike HL7 v2, FHIR supports other paradigms as well including documents, REST and other service models). Refer to the [Message Header](file:///C:\temp\messageheader.html) resource.

**Granularity:** HL7 v2's "Segment" structure provides re-useable chunks of data that roughly correspond to FHIR's idea of [resources](file:///C:\temp\resource.html). However, HL7 v2 segments can't be independently manipulated. Additionally, not all segments have the characteristics of independent identity held by FHIR resources. Due to differences in scope and approach to extensibility, HL7 v2 segments and data types are frequently cluttered with data elements that are not used by (or even understood by) the majority of implementations.

Segments can be composed into repeating and/or optional collections called "groups" to represent full healthcare business objects. For example, the "Order" component of an OMP (Pharmacy/Treatment Order Message) includes:

* an ORC segment dealing with workflow aspects of the order
* an RXO segment dealing with pharmacy-specific order aspects
* optional TQ1 and TQ2 segments describing the timing of the order
* optional NTE segments dealing with supplemental notes or rendering of the order
* optional RXR segments describing route information
* etc.

The HL7 v2 approach to granularity emphasizes re-use of "patterns" of information. For example, timing and route information are not useful on their own, but they are useful in many circumstances. Due to the 3-level nesting limit, separate segments are also required for data structures that would otherwise nest too deeply. FHIR takes a different approach to reusability, focusing on objects that can be maintained independently. The [MedicationRequest](file:///C:\temp\medicationrequest.html) resource encompasses all of the aspects of the above segments, with the exception of some of the workflow aspects of ORC which is handled by the [Task](file:///C:\temp\task.html) resource. The [MedicationRequest](file:///C:\temp\medicationrequest.html) resource is itself complex, having nested structures for dosage instructions, dispensing instructions, etc. that are not simple data types.

**Extensibility:** HL7 v2 provides an extensibility mechanism through the use of "Z-segments". The meaning of these extensions is opaque without prior manual explanation by the sender. Extensions are supposed to be restricted to data elements that do not affect the meaning of the "standard" segments. FHIR [Extensions](file:///C:\temp\extensibility.html), on the other hand, can appear at any level (including within data types). [ModifierExtensions](file:///C:\temp\extensibility.html#modifierExtension) may be used in circumstances where an extension can change the meaning of other elements (e.g. the introduction of a negation indicator on a record). Finally, the meaning of FHIR extensions is discoverable by resolving the URI that defines the extension. The URI approach also ensures that extensions created by independent systems won't collide. (This can be an issue with Z-segments.)

**Inter-version compatibility:** HL7 version 2 has strict processes for maintaining forward and backward compatibility. Content can only be added to the end of existing fields, components, etc. Applications are expected to ignore unexpected content or repetitions. FHIR promises similar compatibility rules. The path to an element within a FHIR instance will remain unchanged in future versions. Specific rules on handling "new" elements (ignoring, checking for "must understand" indicators, etc. will be developed during the STU period.

**Human readability:** In general, HL7 v2 instances do not provide for human readable versions of the content exchanged. While some systems may make use of NTE segments to provide a human-readable rendering of all or part of a message payload, the rules for when or if this occurs is site-specific. FHIR requires human readable content to be provided for each resource.

**Update behavior:** HL7 v2 data is typically exchanged in "snapshot" mode - updates are communicated by sending a complete copy of the instance with the new data filled in. However, some segments and messages in HL7 v2 support more sophisticated exchanges where only changed data is sent and codes or special values indicate what sort of change is to occur (e.g. add this address, remove this name). Out-of-the-box, FHIR only functions using snapshot mode. While the use of [ModifierExtensions](file:///C:\temp\extensibility.html#modifierExtension) to introduce equivalent behavior to HL7 v2 is possible, doing so would create interoperability issues and would make use of the resources difficult outside the messaging paradigm.

**Optionality & Profiles:** Both HL7 v2 and FHIR provide a similar degree of flexibility at the international standard level. Most data elements are optional. However, there are two differences. FHIR resources are much more limited in terms of what elements are included in the core specification - only those elements that the vast majority of systems will support. HL7 v2 tends to include many elements that are used in only very limited circumstances. FHIR uses extensions for those circumstances. HL7 v2 and FHIR both provide formal mechanisms for defining profiles to give guidance on the use of the specification. However, the HL7 v2 mechanism has not been widely used. FHIR [Profiles](file:///C:\temp\profiling.html) form an essential component of the methodology and are built into tooling, increasing the likelihood of their use.

**HL7 v2 Interoperability Considerations**

**Mapping:** One of the biggest challenges with HL7 v2 interoperability is the variation of implementation. Even when identical scenarios are being handled in similar business environments, the data elements supported can vary and even the place where a given data element is placed in an instance can vary. As a result, defining consistent mapping rules between HL7 v2 and FHIR at an international or even regional level is not terribly realistic. The FHIR mappings provided give a starting point for consideration, but mappings will generally need to be done on an implementation by implementation basis.

**Extensions:** While some HL7 v2 elements will map to FHIR core, a large percentage will not. Where a HL7 v2 element is not supported by core, an extension will be needed to share the information. Where there is interest, HL7 may choose to publish and maintain extensions for HL7 v2 elements that are not supported as part of the core FHIR specification. The FHIR extension registry should be searched prior to defining local extensions. If time permits, the relevant HL7 WG should be contacted with a request to define additional HL7 v2 extensions if needed ones are not present. If time does not permit, applications can define their own extensions, but should have a migration plan for if/when HL7 defines it later. For Z-segments, URIs should be defined to be specific to the system/environment that defined the Z-segment (e.g. http://acme.org/fhir/extensions/consent), not based on the name of the Z-segment itself (given that Z-segments with the same name but different meaning may exist) (e.g. http://hl7.org/ZAC).

**Resource identification:** HL7 v2 messages will often reference objects that have already been referred to in previous messages. When converting the messages to FHIR, these references will need to point to the same resource URI. Given that not all HL7 v2 message objects have identifiers in the message, this can be somewhat problematic. An approach to handling this issue exists for FHIR [transactions](file:///C:\temp\http.html#transaction). However, the ramifications of using this approach in a messaging environment have not yet been resolved. Implementers will need to explore their own strategies as part of early adoption.

**Merging references and resources:** HL7 v2 message instances may well reference the same "object" numerous times. For example, a message containing a patient's medication history is likely to include references to the same clinicians and clinics/hospitals many times. While in some cases, the data captured for a given object might be identical in all uses, in other cases the information might vary. For example, the sending system might convey historical phone numbers for old records and current phone numbers for newer records. Alternatively, the message design might allow expression of different amounts of detail in different portions of the message or the sending application might simply be designed to convey different amounts of detail in different portions of the message (e.g. conveying phone number for an ordering clinician, but not for a data-entry clinician). When converting to FHIR, all references to the same "object" will generally have a single resource identifier and be referenced only once in the instance - with the complete set of information needed/available. This creates two challenges:

1. *How does the conversion software recognize when two portions of a message are referencing the same object?* While some references may have unique identifiers or names that are sufficient to confirm "same object", others may not - though some other combination of attributes may be sufficient. The specific rules will need to be determined by the implementer performing the conversion
2. *If multiple versions of data are present, what set of data should be used - or should multiple versions be sent with distinct history ids? (And if the latter, what is the 'order' of the versions?* If the order of the versions can be determined by data in the message (e.g. assuming older order dates have "older" demographics), dates can be specified on the entry *updated* element to indicate relative ordering. If the ordering can't be determined, it will be difficult to merge the data into a single resource or represent it using multiple resources.

**Identified vs. Contained resources:** Each HL7 v2 message will be mapped to multiple resource instances - often 10s or even 100s of resource instances. To maintain consistency with the HL7 v2 messaging paradigm, all resource data will typically be sent over the wire as part of the FHIR message rather than being sent by reference as would be typical in a RESTful implementation. However, FHIR provides two different ways of communicating the resources as part of the message [bundle](file:///C:\temp\compartmentdefinition.html): they can either be sent as "fully identified" resources (direct entries in the bundle with their own identity, and able to be the subject of independent transactions), or they can be sent as [contained](file:///C:\temp\references.html#contained) resources, meaning they are only identified relative to another resources and cannot be retrieved or otherwise manipulated on their own. A HL7 v2 to FHIR conversion process will need to make the determination of what data elements are or must be present, for a resource to be fully identified. In some cases, the determination will be done at the time of mapping. In other cases, it may depend on the content of a particular instance. As an example, an XCN containing just a name (|^Smith^John|) doesn't contain enough information to identify the physician from any other John Smith, so will need to be contained resource, whereas an XCN of |12345^Smith^John| generally does, though the conversion process will need to be aware of the scope and management processes around the identifier.

**Generating human-readable content:** FHIR requires that every resource have a human readable [narrative](file:///C:\temp\narrative.html) that contains all information relevant to human decision-making. When converting from HL7 v2, developers (likely with guidance from clinicians) will need to determine what information from the message should be rendered and how to generate this content.

**Nulls and update modes:** In HL7 v2, "action" codes can determine whether particular segments represent information to be added, updated or deleted. Fields can be populated with "null" (two consecutive double-quotes with no other characters) to note a field is to be deleted. An omitted element or repetition is generally interpreted as "retain existing data unchanged". This contrasts with the FHIR approach of requiring all data to be present as a snapshot. Systems will either need to build in logic to generate a full snapshot of each resource or consider using the [Patch Operation](file:///C:\temp\http.html#patch)instead.

**comparison-v3.html**

**HL7 v3 (and ISO 21090)**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

[HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) was intended to be the next generation of HL7's messaging standards. It introduced a common Reference Information Model (RIM), data type model and set of vocabulary as well as a formal standards development methodology. In addition, it introduced the use of "documents" as an alternative architecture to messaging for sharing healthcare information (see [the CDA comparison](file:///C:\temp\comparison-cda.html)). While nominally covering both, the term "v3" is typically used to refer to "v3 messaging". The data types used as a basis for v3 have also been adopted by ISO as ISO 21090. The HL7 RIM has also been adopted as an ISO standard.

v3 messaging has been adopted by a number of large projects, particularly in the electronic health record area, though it has not achieved the market penetration of [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185). The HL7 RIM and the ISO 21090 data types have also been used by other SDOs and projects that have not leveraged the full HL7 v3 methodology. Most of the comments and guidance provided here will apply to those solutions as well.

**v3 Similarities and Differences**

**Reference model:** The use of the HL7 RIM is a core aspect of the HL7 v3 methodology and it is front and center in the specification and the wire format. All data elements in HL7 v3 instances are derived from either the RIM or the ISO data types. In FHIR, this is true of most resources and data type elements, but not all. Some resources ([StructureDefinition](file:///C:\temp\structuredefinition.html), [CapabilityStatement](file:///C:\temp\capabilitystatement.html), [ValueSet](file:///C:\temp\valueset.html), etc.) deal with content that is outside the RIM's scope. And in a few circumstances, adjustments have been made in the FHIR data types that are not yet supported in the HL7 v3 data types model. The expectation is that these changes will be supported in the next version of the v3 data types model. The main difference is that the serialization format of FHIR is not driven by the RIM mappings. This results in considerably more concise and intuitive instances. It is possible to implement FHIR with absolutely no knowledge of the HL7 RIM.

**Codes:** v3 places considerable reliance on coded attributes to convey the meaning of instances. Examples include *classCode*, *moodCode*, *determinerCode*, etc. The allowed codes for these attributes are strictly controlled by HL7. FHIR also has attributes that are limited to codes defined in the FHIR specification - those using the [code](file:///C:\temp\datatypes.html#code) data type. However, these are generally limited to attributes with business meaning - status, contact types, etc.

Both FHIR and v3 make use of value sets to define the sets of codes that can be used for attributes in particular contexts. However, in FHIR, a [ValueSet](file:///C:\temp\valueset.html) is just another type of resource, meaning it can be sent as part of an instance just like any other piece of data. (The same is true of [StructureDefinition](file:///C:\temp\structuredefinition.html), [CapabilityStatement](file:///C:\temp\capabilitystatement.html) and other meta-level resources.)

**Granularity & referencing:** HL7 v3 models are broken into 3 main types - wrappers, payloads and Common Message Element Types (CMETs). These are combined into interactions to define the set of content that can be sent over the wire at one time. In some cases, the granularity of each of these models will exactly align with the granularity of FHIR resources, but not always. v3 models are divided based on the expectation of re-use. FHIR models are divided based on whether the objects they represent are considered to be able to "stand alone". In HL7 v3, numerous models can exist to represent the same essential underlying healthcare information construct. For example, at the HL7 International level, there are 10 different CMETs for the concept of "Patient". In addition, some payload models represent patient directly without using CMETs. Further variation exists in the v3 models created by HL7 affiliates and other v3 implementers. Each of these different CMETs has their own schema and may use different element names, different levels of nesting and different constraints. With FHIR, there is only one [Patient](file:///C:\temp\patient.html) resource. Many profiles can be created on that resource, but all of them will use the same schema and support the same serialization format.

**Design by constraint:** The design methodology in v3 is one of "design by constraint". The idea is that all data needed for any sort of healthcare communication is represented in the HL7 RIM. All other data models simply constrain the RIM to reflect the needs of particular domain spaces. This starts at the international level with further refinement happening in individual countries, projects and finally specific implementations. As models become closer to the implementer, they become less abstract. The result is a tendency for v3 models to be extremely broad in their coverage and capability and also somewhat abstract. They need to be this way in order to ensure that all possible implementations in the space covered by that model can be properly constrained. As well, each model produces its own schema and, in most cases, constrained schemas are not strictly wire-compatible with the schemas of the model being constrained.

FHIR takes a different approach. FHIR resources do not attempt to represent all data elements that could possibly be used in a space. Instead, only those data elements that are expected to be used by "most" implementations within the scope of the resource are considered part of the core resource definition. (This is sometimes referred to as "The 80% rule" - if approximately 80% of systems maintaining the resource will support the element, then it is part of core). All other data elements are expected to be handled using extensions. [Profiles](file:///C:\temp\profiling.html) are used both to constrain resources and to define extensions appropriate to narrower implementation spaces. Serialization format interoperability is retained across all profiles on a given resource.

**Context conduction:** When conveying healthcare information between humans, much data can be inferred from context. For example, if a report has an "author" noted on a cover page, it is generally inferred that each statement within the report is authored by that same person. This inference grows more challenging when data needs to be analyzed by computers, whether for query, decision support or other analysis. Thus far, the HL7 v3 methodology has provided three distinct mechanisms to allow data models to define how "context" should propagate through models, making explicit for computers what humans would normally understand intuitively. FHIR has chosen a different path. In FHIR, no context is conducted - everything is explicit. If a report about a patient contains 100 observations all about that same patient, each observation will include a reference to the patient. However, this is relatively painless because it's only a reference - an id and possibly a short display value. One of the benefits of this approach is that each resource can be safely consumed and examined without concern for the context in which that resource was communicated. The meaning of each resource instance is fully self-contained.

**Null flavors:** In healthcare, it's quite common for data to be unknown, unavailable, have an exceptional value or otherwise fall outside the bounds of a "normal" value. To deal with this, v3 introduced the concept of "null flavor" on almost every attribute and data type property in its models. These coded null flavors could be sent in place of or in addition to the data that would typically be sent for the attribute, association or data type property. Examples include the ideas of "Unknown", "Not asked", "Positive infinity", "Trace amount", "Masked", "Other", etc. Unless an element was explicitly marked as "mandatory" - meaning no null flavors were permitted, these null flavors could appear absolutely anywhere.

FHIR approaches the problem differently. Null flavors are only introduced in the core specification in those circumstances where it is expected that most systems will need them. Where needed, the flavors are constrained to those relevant to that element.

**v3 Interoperability Considerations**

**Using RIM mappings:** Most resource elements and data type properties include mappings to the RIM. These mappings serve two purposes. They help to define FHIR semantics in terms of HL7's reference models, helping to ensure that the Work Groups defining the data elements have a good and consistent understanding of the meaning of every element. They also provide guidance for implementers of v3 specifications that may be looking to migrate to or map between v3 and FHIR. However, for the latter use it's important to understand some limitations on the RIM mappings. The RIM is a language which allows the same "idea" to be conveyed in a number of different ways with varying granularity and expressiveness. Thus it's entirely possible for a RIM element to map to a core FHIR element even though its RIM representation is somewhat different than described in the mapping. In addition, not all v3 models adhere to good modeling practices, so some data elements that would appear to map to a FHIR element might not map if the information has not been well represented. Therefore RIM mappings should be taken as a guide, not an absolute, and mappings must be done in the context of the v3 specification being mapped. (Also see [Abstract models](file:///C:\temp\main-pages.html#v3-abstractModels) below.)

**v3 extensions:** While the core of the v3 methodology is "design by constraint", it still makes provision for the use of extensions - either in a foreign namespace or denoted by a special attribute. When converting between v3 and FHIR, the use of such extensions will need to be taken into account. As a rule, most v3 extensions will map to FHIR extensions, as the v3 design-by-constraint principle suggests that anything that would qualify as "core" in FHIR would already have been part of the base v3 specification.

**Abstract models:** As previously noted, many of the v3 models created at the HL7 International level are quite abstract. As a result, the models can be used to say a wide variety of things, often in a wide variety of different ways. This makes defining a mapping between those specifications and FHIR (or any other specification) quite tricky. For practical v3 <-> FHIR interoperability, mappings will need to be created at the level of message specifications, implementation guides and/or templates that are more concrete and closer to the implementation level. For example, mapping all of CDA to FHIR would be impossible given the expressive capability of the right-hand-side of the CDA model. However, mapping the Consolidated CDA (CCDA) templates to FHIR is quite possible.

**Context conduction:** As discussed above, HL7 v3 models rely on context conduction - either implicitly or explicitly controlled. When converting to FHIR, the context will need to be propagated into each resource.

**Update mode:** In HL7 v3 instances, updates are generally handled in snapshot mode, similar to the FHIR approach - if any information changes, the entire record is sent, including the modified data elements. However, the v3 methodology does support the introduction of an "updateMode" property to allow only the changes to be sent for all or part of an instance. Each element repetition is flagged with an updateMode to indicate whether the element is to be added, removed, updated, etc. Additional updateModes allow further control over updates. As with the [V2 discussion](file:///C:\temp\comparison-v2.html#V2-updateMode)above, implementers will need to generate a full snapshot of  each resource or or consider using the [Patch Operation](file:///C:\temp\http.html#patch) instead.

**Additional considerations:** Most of the implementation considerations for interoperating between FHIR and [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) also hold with v3. Specifically: [Extensions](file:///C:\temp\comparison-v2.html#V2-extensions), [Independent vs. Contained resources](file:///C:\temp\comparison-v2.html#V2-contained), [Resource Identification](file:///C:\temp\comparison-v2.html#V2-identification), [Merging references and resources](file:///C:\temp\comparison-v2.html#V2-merging) and [Generating human-readable content](file:///C:\temp\comparison-v2.html#V2-humanReadable).

**comparison.html**

**Appendix: The Relationship between FHIR and other HL7 Standards**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

[Health Level Seven (HL7)](http://www.hl7.org) has been developing healthcare information exchange and related standards since 1987. In that time, the organization has produced a number of standards families - many used throughout the world to automate healthcare data sharing and improve patient care. FHIR has been written to be implementable without any knowledge of these other specifications. However, FHIR does leverage this prior experience, both in terms of applying best practices learned from experience and attempting to avoid some of the pitfalls of earlier work.

This appendix describes the relationship of FHIR to some of HL7's other standard families. It may be of interest to those coming to FHIR with previous experience with other HL7 standards as well as those who may need to support interoperability between FHIR solutions and implementations of other HL7 standards.

In this appendix:

* [HL7 v2](file:///C:\temp\comparison-v2.html)
* [v3 (RIM / messaging)](file:///C:\temp\comparison-v3.html)
* [CDA & CCDA etc.](file:///C:\temp\comparison-cda.html)
* [Other HL7 standards](file:///C:\temp\comparison-other.html)

**Notes:**

* In addition to the major standards families identified below, HL7 produces numerous implementation guides, some of which have earned as much prominence as a standard family themselves. The general guidance given below for each standard family should hold for all implementation guides based on that standard. For example, the guidance listed for [CDA](file:///C:\temp\comparison-cda.html) would apply to Consolidated CDA (CCDA) and other CDA Implementation Guides.
* While this appendix focuses on the relationship between FHIR and other HL7 standards, relationships also exist to non-HL7 standards. Some resources provide direct implementation of functionality from other standards including [DICOM](http://medical.nema.org) (see the [ImagingStudy](file:///C:\temp\imagingstudy.html) resource) and [IHE](http://www.ihe.net/) (e.g. the [AuditEvent](file:///C:\temp\auditevent.html) and [DocumentReference](file:///C:\temp\documentreference.html) resources).
* Many FHIR resources draw requirements from or provide mappings to other standards. Some resources also provide additional guidance on how to use them with external specifications as part of their implementation notes. As well, a wiki page to capture additional guidance on the use of FHIR with other specifications can be found [here](http://wiki.hl7.org/index.php?title=FHIR_and_Other_Specifications)
* FHIR can satisfy the needs covered by all of the previous primary HL7 interoperability standards ([HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185), [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) and CDA). In many cases, it also provides additional benefits in terms of ease of interoperability. Therefore, the possibility exists that FHIR could gradually replace some or all of these standards. However it is unclear how rapidly (or even if) the market will make such a migration. It is likely that most of these standards will exist in parallel for quite some time. HL7 has committed to ongoing maintenance of existing standards for as long as the HL7 membership requires.

**compatibility.html**

**Inter-version Compatibility**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The following rules will apply to resources, profiles and other content within the specification once those portions of the specification reach [full normative status](file:///C:\temp\timelines.html#normative). These rules ensure that implementations may exercise FHIR interfaces and process the content of FHIR resources safely while exchanging data between applications using different versions of FHIR.

During the period of trial use of the specification (and once normative status is reached for elements that remain at draft or trial use status), changes may occur based on issues identified during early implementation of the specification. These changes do not need to adhere to the rules listed below.

**Version identification**

There is no explicit version marker in the resource content. FHIR adheres to the DICOM approach to versioning where content can safely be processed by instances independent of version. When dealing with STU-level content, applications may wish to use [resource tags](file:///C:\temp\resource.html#tags) to help manage this during the period of trial use.

The conformance layer ([CapabilityStatement](file:///C:\temp\capabilitystatement.html) and [StructureDefinition](file:///C:\temp\structuredefinition.html)) has mandatory properties declaring the FHIR specification version, and these may be used to determine which version of FHIR an implementation is using to aid in validation.

**Change frequency**

New versions of the FHIR specification will be produced regularly in accordance with the [FHIR publication timeline](file:///C:\temp\timelines.html). New versions of the specification will include additional draft and trial use content as well as promotion of previous trial use content to normative. Once content reaches normative, changes are expected to be infrequent. This is for two reasons:

1. The core specification focuses on those capabilities expected to be supported by most systems. For new capabilities to be introduced, it would need to be reflective of an overall change in the world-wide healthcare implementation environment.
2. If the implementation community has already consolidated around a standard approach to solving a FHIR implementation issue (e.g. using a particular extension), FHIR will not introduce confusion into the implementation community by defining a conflicting mechanism for solving that problem in the core specification.

**Forward compatible behavior**

In a typical scenario, mixed versions may need to exist, so applications SHOULD ignore elements that they do not recognize unless they are modifierExtensions. However, in a healthcare context, many application vendors are unwilling to consider this approach because of concerns about clinical risk or technical limitations in their software (e.g. schema based processing). Applications are not required to ignore unknown elements, but SHALL declare whether they will do so in their [Capability Statements](file:///C:\temp\capabilitystatement.html).

Unrecognized search criteria SHALL always be ignored. (Search criteria supported in a query are echoed as part of the search response so there is no risk in ignoring unexpected search criteria.)

Attempts to perform HTTP operations on unexpected URLs SHOULD be responded to with an appropriate error code.

**Backward compatible behavior**

Until the FHIR specification reaches normative status, systems may not make any assumptions about backward compability. Backward compatibility rules do not kick in until resources are normative and ignoring unexpected elements is unsafe until then. Implementers are encouraged to use a strategy of using distinct endpoints for different versions until all relevant resources and data types are normative. Once normative, the specification will follow the Permitted changes for normative content.

**Permitted changes for normative content**

|  |  |
| --- | --- |
| **Category** | **Allowed changes** |
| Elements | Once normative, subsequent versions of this specification may introduce new elements and/or content (e.g. XML attributes, etc.) at any location in the bundle, resource and/or data type structures. However, the names, path and meaning of previously existing data elements will not be changed. This includes no change to resource names and no changes to names assigned to slices and other elements within profiles. |
| Cardinality | Minimum element cardinalities will not be changed. Upper cardinality may change from 1 to \* only in circumstances where all elements except for the first repetition can be safely ignored. (This may mean that an order is assigned to the repeating items or that there is no preference as to which element is retained.) Systems should follow the rules above for unexpected elements. |
| Descriptions | Descriptive information about a resource - short labels, definitions, usage notes, aliases, examples, rationale, mappings, etc. may be updated or revised to provide additional clarity or guidance, but not in such a manner as to invalidate a reasonable interpretation of the previously documented use of an element. (This does not preclude fixing obvious errors.) |
| Value Sets | The definition of any value set that is marked as "immutable" will never change. The expansions for immutable value sets may still change if no "stable date" is declared and the value set does not restrict code system and/or value set references to specific versions if the referenced code system(s) or value set(s) change.  For non-immutable value sets:   * Value sets with an enumerated list of codes and having a 'fixed' binding may have additional codes introduced but will never have codes removed, though they may be deprecated. * Value sets making use of filters may have filters loosened or tightened to accommodate changes to underlying code systems. StableDates and referenced code system and value set versions may be adjusted to point to newer versions. * Definitions and display values for codes may change, but only in a manner that would not change the reasonable interpretation of data captured using the previous definitions or names. * Abstract codes may be made concrete. Concrete codes will not be made abstract.   For both immutable and non-immutable value sets, additional designations may be declared. |
| Terminology Bindings | Fixed bindings will remain fixed and will continue to point to the same value set. If the reference is version-specific, it will not change. Example bindings and preferred bindings may change to point to distinct value sets. Example bindings may be replaced with preferred bindings. |
| Data Types | Data types will not be removed or changed. New data types may be introduced. Types declared on existing elements will not be removed or changed. Additional data types may be added to elements which are already expressed as a choice of data types only if those elements are optional (minimum cardinality = 0). |
| Value Constraints | The allowed list of Data types will not be added, removed or changed. Invariants, regular expressions, fixed values and patterns will not be added, removed or changed. |
| Flags | The *Is Modifier* and *Is Summary* flags will not be changed. The *Must Support* flag may be changed to true, but will not be removed. |
| Slicing | Slicing rules and aggregation characteristics will not be changed. |
| Search Criteria | Search criteria may be added but not removed or renamed. Existing criteria will not have their type or path changed or have their description altered in any way that would invalidate the reasonable behavior of existing systems (with the exception of correcting obvious errors). |
| Operations | New operations may be defined but operations may not be removed or renamed. Existing parameters will not be removed or renamed, nor may their type or lower cardinality be changed. Upper cardinality may be changed from 1 to \*. (Systems should ignore unexpected repetitions.) Additional optional parameters may be introduced; e.g. Operation signatures cannot change; instead, additional operation variants will be defined. |
| Restful interface | Existing endpoints will not be renamed or removed, nor have their expected behavior changed in a manner that would cause reasonable systems designed against prior versions to be non-interoperable. Additional endpoints may be introduced. |
| Profiles and extension definitions | Profile structure, extension definitions and search criteria definitions will not be removed or have their URIs changed. New profile structures, extension definitions and search criteria definitions may be introduced. Profiles may have their statuses changed to "retired". Profiles referenced by data elements for structures or data types may be replaced with a reference to a distinct profile that is "compatible" with the previously referenced profile according to these forward and backward compatibility rules. |

Additional discussion on inter-versioning issues can be found here: <http://wiki.hl7.org/index.php?title=FHIR_interversion_compatibility>.

**conformance-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Conformance Module**

The Conformance Module represents metadata about the datatypes, resources and API features of the FHIR specification and can be used to create derived specifications.

|  |  |  |
| --- | --- | --- |
| * [Capability statement](file:///C:\temp\capabilitystatement.html) * [StructureDefinition](file:///C:\temp\structuredefinition.html) * [MessageDefinition](file:///C:\temp\messagedefinition.html) * [OperationDefinition](file:///C:\temp\operationdefinition.html) * [SearchParameter](file:///C:\temp\searchparameter.html) | * [CompartmentDefinition](file:///C:\temp\compartmentdefinition.html) * [DataElement](file:///C:\temp\dataelement.html) * [ImplementationGuide](file:///C:\temp\implementationguide.html) * [ElementDefinition (datatype)](file:///C:\temp\elementdefinition.html) | * [Full profiling details](file:///C:\temp\profiling.html) * [Detailed conformance rules](file:///C:\temp\conformance-rules.html) |

**Introduction**

The core FHIR specification describes a set of resources, frameworks and APIs that are used in many different contexts in healthcare. However, there is wide variability between jurisdictions and across the healthcare ecosystem around practices, requirements, regulations,education and what actions are feasible and/or beneficial.

For this reason, the FHIR specification is a "platform specification" - it creates a common platform or foundation on which a variety of different solutions are implemented. As a consequence, this specification usually requires further adaptation to particular contexts of use.

Typically, these adaptations specify:

* Rules about which resource elements are or are not used, and what additional elements are added that are not part of the base specification
* Rules about which of FHIR's RESTful API, messaging and document features are used, and how
* Rules about which terminologies are used in particular elements
* Descriptions of how the Resource elements and API features map to local requirements and/or implementations

Note that because of the nature of the healthcare ecosystem, there may be multiple overlapping sets of adaptations - by healthcare domain,by country, by institution, and/or by vendor/implementation.

FHIR provides a set of resources that can be used to represent and share the adaptations listed above in a computable fashion. These resources are collectively called the *conformance resources*. Although these conformance resources can be used in isolation they are typically used in the context of an *Implementation Guide* or a *Capability Statement*:

* *Implementation Guides* are documents published by a domain, institution or vendor that describe how FHIR is adapted to support a certain usecase (or set of usecases). An implementation guide combines a set of conformance resources and supporting narrative into a document for use by implementers.
* A *capability statement* which uses the conformance resources to document how a client or server has implemented FHIR, i.e. which aspects of the specification and API are implemented and how.

**Contents**

The content of an Implementation Guide is described using the [ImplementationGuide](file:///C:\temp\implementationguide.html) resource, while the capability statement is represented by the [CapabilityStatement](file:///C:\temp\capabilitystatement.html) resource. These two resources make use of the complete set of conformance resources to fully capture the set of adaptations they represent. Note that the CapabilityStatement resource is one of the *conformance resources*, the first just describing the capabilities of a system, while the latter is the set of all conformance resources, including:

* The [StructureDefinition](file:///C:\temp\structuredefinition.html) defines how a particular structure (Resource, Extension or Data Type) is used to:
  + Describe how existing elements in resources and/or data types are used and identify existing elements that are not used. Exact rules on how to do this can be found in the section on [conformance rules](file:///C:\temp\conformance-rules.html)
  + Define [extensions](file:///C:\temp\extensibility.html) that can be used in resources or data types.
  + Reference the [Value Sets](file:///C:\temp\valueset.html) that specify the content of coded elements, see the [terminology module](file:///C:\temp\terminology-module.html) for more details.
* The [MessageDefinition](file:///C:\temp\messagedefinition.html) describes messages that can be sent and received, including the driving event, contents to be exchanged and responsibilities on receipt.
* The [OperationDefinition](file:///C:\temp\operationdefinition.html) describes additional operations in addition to [those in the base specification](file:///C:\temp\operations.html).
* The [SearchParameter](file:///C:\temp\searchparameter.html) describes additional search capabilities in addition to those in the base specification
* The [CompartmentDefinition](file:///C:\temp\compartmentdefinition.html) describes a logical grouping for resources, used in access control or search
* The [DataElement](file:///C:\temp\dataelement.html) resource is a formal description of a single piece of information that can be gathered and reported.

Conformance resources may be used independently, not just within the context of an ImplementationGuide resource or capability statement. See the section [Common use cases](file:///C:\temp\main-pages.html#uses) for examples of such uses.

The conformance resources and their relationships are shown below:

Resources shown with a dotted box are described in other sections of the specification: ValueSet, ConceptMap and StructureMap are from the [section on terminology](file:///C:\temp\terminology-module.html), TestScript is part of the section on [Implementer Support](file:///C:\temp\implsupport-module.html).

**Security and Privacy**

The conformance resources do not represent patient-related data, and as such are less susceptible to data breaching. Some caution is required however:

* StructureDefinitions may contain invariants formulated as structured expressions that are evaluated by external engines (i.e. xpath), which -if unproperly sandboxed- could provide low-level access to the system
* A server may be implemented in such a way that it will automatically support additional search parameters or compartments definitions uploaded to that system. This may result in searches that use large amounts of processing power, expose sensitive patient data, or both.
* If a server acts as a repository of conformance resources, it should tightly control modifications to its data: updates to conformance resources are usually managed by a review process to ensure backwards compatibility and stability. Worst case, changes to these resources may result in dependent systems running the risk of being misconfigured.

**Common use Cases**

Conformance resources are commonly used as part of an [Implementation Guide](file:///C:\temp\implementationguide.html) or [CapabilityStatement](file:///C:\temp\capabilitystatement.html) resource. There are many ways to use the resources independently however, including:

* An instance of a resource may reference one or more StructureDefinitions to claim conformance to the rules layed out in those StructureDefinitions
* A server may serve CapabilityStatement resources, effectively functioning as a discovery endpoint for services within an organization
* A server may hold a set of DataElement resources to become a Data Dictionary of reuseable datamodels as per ISO 11179.
* A code-generation tool may use StructureDefinitions and OperationDefinitions to generate code that represents the structures as classes and operations as remotely callable functions to provide an easier programming model to a software developer.
* A server may act as a repository of NameSpace resources, so vendors and implementers can quickly look up the urls or oids for a given terminology or identifier system.

**Developmental Roadmap**

A subset of the conformance resources have been tested and used in production tooling and as such have reached a maturity level where changes become less likely. These are CapabilityStatement, StructureDefinition, ValueSet, OperationDefinition and SearchParameter.

Other resources are still under development:

* ImplementationGuide: used in the HL7 production tooling, but has not received much use outside of these tools yet.
* CompartmentDefinition: new in STU3, and as such has not undergone much production use
* DataElement: the exact role of "Data Element Definitions" is still being studied

Over the past two years, these resources have been mainly used in the tools used to build the FHIR publication, early-adopter implementation guides and the FHIR Foundation conformance resource registry. In the 2016-2018 timeframe we expect to see more widespread use of these resources in validation tooling, code-generators and more extensive model-based guide authoring tools.

**conformance-rules.html**

**Conformance**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The FHIR specification describes a set of [resources](file:///C:\temp\resource.html), and several different frameworks for exchanging resources between different systems. Because of its general nature and wide applicability, the rules made in this specification are generally fairly loose. As a consequence, this specification allows that different applications may not be able to be interoperate because of how they use optional features. As a consequence, applications claiming conformance to this specification make the claim in respect of a specific exchange framework, and in regard to a specific details about their usage of those frameworks and resource contents.

Applications claim conformance to one (or more) of the following exchange frameworks:

* "RESTful FHIR": the [RESTful API](file:///C:\temp\http.html)
* "FHIR messaging": [Message based exchange](file:///C:\temp\messaging.html)
* "FHIR documents": [Document based exchange](file:///C:\temp\documents.html)

To provide details about specific usage of the frameworks and resource contents, FHIR provides a [conformance layer](file:///C:\temp\profiling.html) that implementers and national/regional programs can use to provide a computable statement about how the resources and their exchange paradigms are used to solve particular use cases. The conformance layer itself is implemented using the following key resources:

|  |  |
| --- | --- |
| [Value Set](file:///C:\temp\valueset.html) | Defines a set of coded values (see "[Using Codes](file:///C:\temp\terminologies.html)" for more details) |
| [StructureDefinition](file:///C:\temp\structuredefinition.html) | Makes rules about how a resource (or type) and its data elements are used in a particular context. A structure definition references value sets for the coded elements in a resource |
| [CapabilityStatement](file:///C:\temp\capabilitystatement.html) | A statement of the kinds of resources and operations provided and/or consumed by an application. The Capability Statement references profiles to describe specific use of resources by the application |
| [Implementation Guide](file:///C:\temp\implementationguide.html) | A single coherent collection of capability statements, profiles, value set, and documentation describing a set of interoperable applications |

The specification also [provides a number of tools that can assist with enforcing technical conformance](file:///C:\temp\validation.html) to this base specification and profiles on it.

Conformance with this specification does not provide any guarantee of patient or data safety. However, choosing to not conform to this specification carries additional risk in two ways:

* FHIR has been subject to a level of review and vetting unlikely to be received by any non-conformant variation; variations may result in introduction of undetected risks
* FHIR-like solutions (based on FHIR, but not conformant) may set expectations by trading partners which are not met due to the non-conformance of the system and these un-met expectations may also result in risk

Any assertion of conformance to FHIR that does not reference a [CapabilityStatement](file:///C:\temp\capabilitystatement.html) instance has limited utility. System capabilities that cannot be described by a CapabilityStatement cannot be considered FHIR conformant.

**Base Conformance Rules**

The contents of a resource and the formats used to represent resources SHALL conform to the rules described in this specification, as defined in the narrative of the specification, and as controlled by the conformance properties defined below.

Note: This specification uses the conformance verbs SHALL, SHOULD, and MAY as defined in [RFC 2119](http://www.ietf.org/rfc/rfc2119.txt). Unlike RFC 2119, however, this specification allows that different applications may not be able to interoperate because of how they use optional features.

Data elements defined in resources and data types have 3 properties that are directly related to conformance: Cardinality, Is-Modifier, and Must-Support. These interact to place conformance requirements on implementations.

**Cardinality**

All attributes defined in FHIR have cardinality as part of their definition - a minimum number of required appearances and a maximum number. These numbers specify the number of times the attribute may appear in any instance of the resource type. This specification only defines the following cardinalities: 0..1, 0..\*, 1..1, and 1..\*. Profiles that describe specific use cases may use other values for cardinality within the limits of the cardinality defined by the base resource.

Note that when present, elements cannot be empty - they SHALL have a value attribute, child elements, or extensions. This means that setting an element to a minimum cardinality of 1 does not ensure that valid data will be present; specific XPath constraints are required to ensure that the required data will be present.

In this specification, very few elements have a minimum cardinality of 1. Resources are used in many contexts, often quite removed from their primary use case, and sometimes even basic information is quite incomplete. For this reason, the only elements that have a minimum cardinality of 1 are those where they are necessary to any understanding of the resource or element that contains them. The minimum cardinalities should not be taken as a guide to what elements are expected to be present in any particular use of the resource, including their normal/primary usage purpose. In some cases, this specification publishes additional profiles that define which elements are required in particular situations. Similar profiles are published by jurisdictions, vendors, or projects.

For elements that have cardinality > 1, the order in which they appear may have meaning. Unless the element definition (either in this specification or the extension) defines a meaning to the order explicitly (using [ElementDefinition.orderMeaning](file:///C:\temp\elementdefinition.html), the meaning of the order is not defined, and implementations are allowed to reorder the elements. Note that it is not possible to define a meaning for the order of the elements in a [profile](file:///C:\temp\profiling.html) using a [StructureDefinition](file:///C:\temp\structuredefinition.html). When there is no definition of the meaning of the order, implementations that need to choose a single element from a list of elements for some use SHALL do so based on the semantics of the content of the elements that repeats. Profiles and Implementation guides may often make rules about this selection process.

Clients should not depend on servers maintaining ordering of elements, unless the retrieved resource conforms to a profile which mandates maintenance of ordering. If a server cannot maintain ordering, it must must strip off known profile tags which require maintenance of ordering, and strip off unknown profiles (since they might require maintenance of ordering).

**Is-modifier**

Is-Modifier is a boolean property that is assigned when an element is defined, either as part of the base resource contents in this specification, or when [extensions are defined](file:///C:\temp\structuredefinition.html). An element is labeled "Is-Modifier = true" if the value it contains may change the interpretation of the element that contains it (including if the element is the resource as a whole). Typical examples of elements that are labeled "Is-Modifier" are elements such as "status", "active", "refuted", or "certainty". Whether an element is a modifier cannot be changed when element usage is described in a profile (i.e. a constraining [Structure Definition](file:///C:\temp\structuredefinition.html)). When an element is labeled as Is-Modifier, the documentation must be clear about why it is a modifier.

A typical example of a modifier element is one that negates the element that contains it. For instance, in the following fragment of a resource definition:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [**Name**](file:///C:\temp\formats.html#table) | [**Flags**](file:///C:\temp\formats.html#table) | [**Card.**](file:///C:\temp\formats.html#table) | [**Type**](file:///C:\temp\formats.html#table) | [**Description & Constraints**](file:///C:\temp\formats.html#table) |
| [AllergyIntolerance](file:///C:\temp\allergyintolerance-definitions.html#AllergyIntolerance) |  |  | [DomainResource](file:///C:\temp\domainresource.html) | Allergy or Intolerance (generally: Risk Of Adverse reaction to a substance) |
| [onset](file:///C:\temp\allergyintolerance-definitions.html#AllergyIntolerance.onset_x_) | ÃŽÂ£ | 0..1 | [dateTime](file:///C:\temp\datatypes.html#dateTime) | Date(/time) when manifestations showed |
| [patient](file:///C:\temp\allergyintolerance-definitions.html#AllergyIntolerance.patient) | ÃŽÂ£ | 1..1 | [Reference](file:///C:\temp\references.html)([Patient](file:///C:\temp\patient.html)) | Who the sensitivity is for |
| [verificationStatus](file:///C:\temp\allergyintolerance-definitions.html#AllergyIntolerance.verificationStatus) | ?! ÃŽÂ£ | 0..1 | [code](file:///C:\temp\datatypes.html#code) | unconfirmed | confirmed | refuted | entered-in-error [AllergyIntoleranceVerificationStatus](file:///C:\temp\valueset-allergy-verification-status.html) ([Required](file:///C:\temp\terminologies.html#required)) |
| [criticality](file:///C:\temp\allergyintolerance-definitions.html#AllergyIntolerance.criticality) | ÃŽÂ£ | 0..1 | [code](file:///C:\temp\datatypes.html#code) | CRITL | CRITH | CRITU [AllergyIntoleranceCriticality](file:///C:\temp\valueset-allergy-intolerance-criticality.html) ([Required](file:///C:\temp\terminologies.html#required)) |

The value of the 'verificationStatus' element affects the entire meaning of the resource - if it is set to refuted, the entire resource must be understood differently, and so it is not safe for implementations to ignore it. As a consequence, it is labelled as 'is modifier = true'. In this tabular representation of the resource, this shows as the flag '?!'. The [JSON](file:///C:\temp\json.html) and [XML](file:///C:\temp\xml.html) representations of a resource definition have their own representation of 'is modifier = true' status, and it is defined directly in a [ElementDefinition](file:///C:\temp\elementdefinition-definitions.html#ElementDefinition.isModifier).

Is-Modifier elements SHALL be represented in the narrative summary of the resource.

If the value of a modifier element is not explicit in the instance, or known by the context, the resource may not be able to be safely understood. Wherever possible, elements labeled "Is-Modifier = true" also have a minimum cardinality of 1, or a default value, in order to introduce certainty in their handling. However sometimes this is not possible - much legacy data is not well described. Implementations producing resources SHOULD ensure that appropriate values for isModifier elements are provided at all times.

Implementations processing the data in resources SHALL understand the impact of the element when using the data. Implementations are not required to "support" the element in any meaningful way - they may achieve this understanding by rejecting instances that contain values outside those they support (for instance, an application may refuse to accept observations with a reliability other than "ok"). Alternatively, implementations may be able to be sure that, due to their implementation environment, such values will never occur. However applications SHOULD always check the value irrespective of this.

Note that processing the data of a resource typically means copying or filtering data out of a resource for use in another context (display to a human, decision support, exchange in another format where not all information is included, or storing it for this kind of use). Servers and background processes that simply move whole resources around unchanged are not "processing the data of the resource", and therefore these applications are not required to check Is-Modifier elements.

Every element in the base resource has a value of "true" or "false" for the Is-Modifier flag. The value of the flag cannot be changed by profiles on the resource, in either direction. When a profile defines an extension, it labels the extension with the Is-Modifier flag, and this cannot be changed in other profiles. Note that extensions that have is-Modifier = true are represented differently in resource instances ("modifierExtension" instead of "extension"), and there are [additional rules about how they are handled](file:///C:\temp\extensibility.html#modifierExtension).

Here is a list of all the modifier elements defined in the specification:

**Must-Support**

Labeling an element Must-Support means that implementations that produce or consume resources SHALL provide "support" for the element in some meaningful way. Because the base FHIR specification is intended to be independent of any particular implementation context, no elements are flagged as "must-support" as part of the base specification. This flag is intended for use in profiles that have a defined implementation context.

For this reason, the specification itself never labels any elements as must-support. This is done in [Resource Profiles](file:///C:\temp\profiling.html#mustsupport), where the profile labels an element as mustSupport=true. When a profile does this, it SHALL also make clear exactly what kind of "support" is required, as this can mean many things.

Note that an element that has the property IsModifier is not necessarily a "key" element (e.g. one of the important elements to make use of the resource), nor is it automatically mustSupport - however both of these things are more likely to be true for IsModifier elements than for other elements.

**Constraints**

All elements may have constraints attached to them (also known as 'invariants'). Constraints defined on an element have the following properties:

|  |  |
| --- | --- |
| **Key** | Identifies the constraint uniquely amongst all the constraints in the context - typically, this is used to refer to the constraint in an error message |
| **Requirements** | An explanation of why the constraint has been applied - what harmful conditions are being avoided |
| **Severity** | Whether the constraint is an error, or a warning. The exact difference in meaning of these depends on context, but an error is associated with "SHALL" and systems rejecting content, whereas a warning might not be |
| **Human Description** | A human description of the rule intended to be show as the explanation for a message when the constraint is not met |
| **Expression** | An [FHIRPath](http://hl7.org/fluentpath) expression that must evaluate to true when run on the element |
| **XPath** | An XPath expression that must evaluate to true when run on the element in the XML representation |

Many constraints are defined in the base specification. In addition, additional constraints may be defined in [profiles](file:///C:\temp\profiling.html) that apply to resources. Systems are not required to evaluate the constraints, just as they are not required to check for conformance, or schema validity. However, systems SHOULD always ensure that all resources are valid against all applicable constraints.

Elements can also be explicitly associated with constraints defined elsewhere. This is a notification to implementers that the element is affected by the constraint. It has no meaning when the constraints are evaluated.

[Profiles](file:///C:\temp\profiling.html) may define additional constraints that apply to an element, but they cannot alter or remove constraints that are already applied.

**Other Metadata**

In addition to the conformance metadata, each element has other metadata properties:

* **isSummary** - How the element behaves in [Summary Searches](file:///C:\temp\search.html#summary)
* **defaultValue / meaningWhenMissing** - What it [means when an element is missing](file:///C:\temp\elementdefinition.html#missing)
* **maxLength** - How long the representation of the primitive value is allowed to be (not including XML or JSON escaping)

**Examples and Reference Implementations**

This specification includes many examples. While every effort has been made to ensure that the examples are fully conformant to the specification, if the examples disagree with the specification, the specification is considered correct and normative, not the examples. This same rule applies to the reference implementations.

The examples reflected in this specification do \*not\* represent actual people. Any resemblance to real people - alive or dead - is entirely coincidental. In some cases, examples may be drawn from real clinical data. However, if this has occurred, the content has been scrubbed to remove any identifying information.

**contributor-mappings.html**

**Contributor Mappings**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page provides mappings for Contributor

**cpt.html**

**Using CPT with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Summary**

|  |  |
| --- | --- |
| Source | CPT is made managed by the [American Medical Association](http://www.ama-assn.org/ama) |
| System | The URI <http://www.ama-assn.org/go/cpt> identifies the CPT code system |
| Version | Where a version is appropriate, use the year of release e.g. 2014 |
| Code | "Code" in the CPT tables (a 5 character code) |
| Display | "Description" in the CPT tables |
| Inactive | Todo: Describe how it is determined which concepts are inactive |
| Subsumption | No Subsumption relationships are defined by CPT |
| Filter Properties | None defined |

**Version Issues**

CPT is released each October. CPT versions are identified simply by the year of their release.

**Example Usage**

<coding>

<system value="http://www.ama-assn.org/go/cpt"/>

<version value="2014"/>

<code value="31502"/>

<display value="Tracheotomy tube change prior to establishment of fistula tract"/>

</coding>

**Copyright/License Issues**

CPT is a registered trademark of the American Medical Association. The AMA holds the copyright for the CPT coding system.

When Value Sets include CPT codes, the copyright element should include the text "CPT copyright 2014 American Medical Association. All rights reserved.".

**CPT Filter Properties**

No filter properties have been defined at this time.

**Implicit Value Sets**

The value set URL http://hl7.org/fhir/ValueSet/cpt-all is a value set that includes all CPT codes.

**credits.html**

**Community and Credits**

|  |  |
| --- | --- |
| [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Community**

The FHIR community meets inside the wider [HL7 community](http://hl7.org) and draws on its extensive human resources, institutional memory, previous standards and corporate support. HL7 itself owns FHIR and makes it freely available and the community relies on HL7 provided infrastructure.

The primary resources used by the FHIR community are the [HL7 wiki](http://wiki.hl7.org/index.php?title=FHIR), the [FHIR email list](http://wiki.hl7.org/index.php?title=FHIR_email_list_subscription_instructions) and the [gForge](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemBrowse&tracker_id=677) (for subversion and tracker capabilities). In addition, the community holds regular face to face meetings as part of the [HL7 Working Group meetings](http://www.hl7.org/events/workgroupmeetings.cfm?ref=nav). The formal governance arrangements that manage FHIR development are documented on HL7's [wiki](http://wiki.hl7.org/index.php?title=FHIR_Governance_Process)

**Credits**

FHIR is a specification produced by the HL7 Community. Many individuals contribute to the FHIR specification. Of particular note:

* The Editorial team is James Agnew (University Health Network), Grahame Grieve (Health Intersections), Eric Haas (Health eData), Rob Hausam (Hausam Consuling), David Hay (Orion), Ewout Kramer (Furore), Lloyd McKenzie (Gevity), Josh Mandel (Harvard), Viet Nguyen (Systems Made Simple), Brian Postlethaite (HealthConnex), Bryn Rhodes (Database Consulting Group)
* The [Governance board](http://wiki.hl7.org/index.php?title=FHIR_Governance_Board) is Calvin Beebe, Woody Beeler, Lorraine Constable, Grahame Grieve, David Hay, Ewout Kramer, Wayne Kubrik, Cecil Lynch, Lloyd McKenzie, John Quinn and Dave Shaver
* The [Management board](http://wiki.hl7.org/index.php?title=FHIR_Management_Group) is Hans Buitendijk, Grahame Grieve, David Hay, Paul Knapp, Josh Mandel, Lloyd McKenzie, John Moehrke, Brian Pech, Brian Postlethwaite and John Quinn. (Former members: Woody Beeler, Lorraine Constable, Jean Duteau, Hugh Glover and Ron Parker)
* The basic HTML design done by [studiojoyo.com](http://www.studiojoyo.com) - thanks
* The following organizations have helped by attending Connectathons: [Accenture](http://www.accenture.com), [AEGIS.net Inc](http://www.aegis.net), [Agfa Healthcare](http://www.agfahealthcare.com), [ahdis gmbh / HL7 Switzerland](http://www.ahdis.ch), [Akana, Inc.](http://www.akana.com), [Allscripts](http://www.allscripts.com), [Almerys](http://www.almerys.com), L'Atelier du Soft (HL7 France), [American Immunization Registry Association](http://www.immregistries.org), [American Society of Clinical Oncology](http://www.asco.org), [Analysists International (AIC)](http://aictalent.acsicorp.com), [Anthem, Inc.](https://www.antheminc.com), [Apelon, Inc.](http://apelon.com), Apertura, [Blue Cross Blue Shield of Alabama](http://www.bcbsal.org), [Blue Cross Blue Shield of Louisiana](http://www.bcbsla.com), [Blue Wave Informatics](http://www.bluewaveinformatics.co.uk), [Boston Children's Hospital](http://www.childrenshospital.org), [BRIT Systems](http://www.brit.com), [Cambia Health Solutions](http://www.cambiahealth.com), [Canada Health Infoway (CHI)](http://infoway-inforoute.ca), [Center for Medical Interoperability](http://medicalinteroperability.org), [Centers for Disease Control and Prevention](http://www.cdc.gov), [Cerner Corporation](http://www.cerner.com), [Choice Hospital Systems](http://www.choise-hs.com), [CIGNA](http://www.cigna.com), [CIOX Health](http://www.cioxhealth.com), [Cleveland Clinic Health System](http://my.clevelandclinic.org), [Cognitive medical Systems](http://www.cognitivemedicalsystems.com), [College of American Pathologists](http://www.cap.org), [Corepoint Health](http://www.corepointhealth.com), [CSIRO ICT](http://www.ict.csiro.au),[Database Consulting Group](http://www.databaseconsultinggroup.com), [Datuit, LLC](http://www.datuit.com), [DCA Health Solutions](http://ehealth.data.com.au), [Dell](http://www.dell.com), [Deontik Pty Ltd](http://deontik.com), [DIPS](http://www.dips.no), Drajer, [Dynamic Health It](http://www.dynamichealthit.com), [ecGroup Inc](http://ecgroupinc.com), Ediden Group Inc., [Edifecs, Inc.](http://www.edifecs.com), [Edmond Scientific Company](http://www.edmondsci.com), [Elsevier](http://www.elsevier.ca), [Epic](http://www.epic.com), [ESAC Inc](http://esacinc.com), [Eversolve, LLC](http://www.eversolve.com), [Fujifilm Australia](http://www.fujifilm.com.au), [Furore](http://furore.com), [GE Healthcare](http://ge.com),[Gea-Interactive](http://2013.gea-interactive.com.au), [Global Village Consulting](http://global-village.net), [Gevity](http://www.gevityinc.com), Haas Consulting, Hausam Consulting LLC, [Health Care Service Corportation](http://www.hcsc.com), [HEALTH CONNEX](https://healthconnex.com.au), [Health Intersections](http://www.healthintersections.com.au), [Health IQ](http://www.healthiq.com.au), [Healthcentrix](http://www.healthcentrix.com), HealthFire, [Health Spring](https://www.healthspring.in), [Healthwise](https://www.healthwise.org), [Hefei University of Technology](http://www.hfut.edu.cn), [Helse Vest IKT AS](http://www.helse-vest-ikt.no), [HL7 Argentina](http://www.hl7argentina.org.ar), [HL7 Australia](http://www.hl7.org.au), HL7 Canada, [HL7 Germany](http://www.hl7.de), [HL7 Korea](http://www.hl7korea.or.kr), [HL7 New Zealand](http://www.hl7.org.nz), [HL7 Netherlands](http://www.hl7.nl), [HL7 New Zealand](http://www.hl7.org.nz), [HL7 Norway](http://www.hl7.no), [HL7 UK](http://www.hl7.org.uk), [IBM](http://www.ibm.com), [Icahn School of Medicine at Mount Sinai](http://icahn.mssm.edu), IHIS Research Center - Kyungpook National University, [Inovalon](http://www.inovalon.com),[Intelligent Medical Objects (IMO)](https://www.e-imo.com), [Interfaceware](http://www.interfaceware.com), [InterSystems](http://www.intersystems.com), [Inovalon](http://www.inovalon.com), [Intermountain Healthcare](http://www.intermountainhealthcare.org), [iSalus Solutions](http://www.isalussolutions.com), [JBS International](http://www.jbsinternational.com), JKM Software LLC, [J P Systems](http://www.jpsys.com), [Kestral Computing](http://www.kestral.com.au), Knapp Consulting Inc., [Kyungpook National University](http://en.knu.ac.kr), [Lamprey Networks, Inc.](http://www.lampreynetworks.com), [Lantana Consulting Group](http://lantanagroup.com), [The Lazy Company](http://thelazycompany.com), [Leidos, Inc.](https://www.leidos.com), [Mater Pathology](http://pathology.mater.org.au), [Mayo Clinic](http://mayoclinic.org), [McKesson](http://www.mckesson.com), [Medical Objects](https://www.medical-objects.com.au), [Michigan Health Information Network](http://mihin.org), [MITRE Corporation](http://www.mitre.org), [Mohawk College](http://www.mohawkcollege.ca), [MSIA Australia](http://www.msia.com.au), [National E-Health Transition Authority (NEHTA)](http://www.nehta.gov.au), [National Institute of Standards and Technology (NIST)](http://www.nist.gov), [National Marrow Donor Program](http://bethematch.org), [NaviNet](https://www.navinet.net), [NCQA](http://www.ncqa.org), [Nortal AS](http://www.nortal.com), [NProgram](http://www.nprogram.co.uk), [Office of the National Coordinator for Health Intormation Technology](http://www.healthit.gov), [Open Mapping Software](http://www.openmapsw.com), [Optum](https://www.optum.com), [Oridashi](http://oridashi.com.au), [Orion Healthcare](http://www.orionhealth.com), [Pareto Intelligence, LLC](https://www.paretointelligence.com), [Partners HealthCare System, Inc.](http://www.partners.org), [PenRad](http://www.penrad.com), [Philips Healthcare](http://www.usa.philips.com/healthcare), [Premier Healthcare Alliance](https://www.premierinc.com), [Qvera](http://www.qvera.com), [Queensland Health](http://www.health.qld.gov.au), [Regenstrief Institute](http://www.regenstrief.org), [RelayHealth](http://www.relayhealth.com), [Ringholm](http://www.ringholm.comÃ¢â‚¬Å½), [River Rock Associates](http://www.riverrockassociates.com), [Roche Diagnostics International Ltd.](http://www.roche-diagnostics.comÃ¢â‚¬Å½), [Rouge Valley Health Systems](http://www.rougevalley.ca), [RxREVU](http://rxrevu.com), [SAMHSA](http://www.samhsa.gov), [Smart Health Solutions](http://www.smarthealth.com.au), [Smart Platforms](https://smartplatforms.org), [Surescripts](http://surescripts.com), [Synapse, Inc.](http://www.synapsegroupinc.com), [Sysmex AU](http://www.sysmex.com.au), [Sysmex NZ Ltd](http://www.sysmex.co.nz), [Systems Made Simple](http://www.systemsmadesimple.com), [Texas Health Services Authority](http://www.hietexas.org), [Thrasys](http://www.thrasys.com), [Transcend Insights](https://www.transcendinsights.com), [U.S. Dept. of Veteran Affairs](http://www.va.gov), Ucaresoft, [University Health Network](http://www.uhn.ca), [University of Applied Sciences Upper Austria](http://www-en.fh-ooe.at), [University Of Washington Medical Center](http://www.uwmedicine.org/uw-medical-center), [Vanderbilt University Medical Center](http://www.mc.vanderbilt.edu), Vermonster, Vernetzt LLC, [Videntity, Inc.](http://www.videntity.com), [Web MD Health Services](https://www.webmdhealthservices.com), [West Health](http://www.westhealth.org), [Wolters Kluwer Health](http://wolterskluwer.com), [yConsult](http://yconsult.com.au), [YouCentric](http://www.youcentric.com), and [Zynx Health](http://www.zynxhealth.com)
* The FHIR specification includes content developed by or reviewed by the following [HL7 Work Groups](http://www.hl7.org/Special/committees/index.cfm): Clinical Genomics (CG), Clinical Decision Support (CDS) Clinical Quality Information (CQI) Conformance & Guidance for Implementation/Testing (CGIT), Community Based Collaborative Care (CBCC), Electronic Health Record (EHR), Financial Management (FM), FHIR Infrastructure (FHIR-I), Health Care Device (Devices), Health Standards Integration (HSI), Imaging Integration (II), Implementable Technology Specifications (ITS), Infrastructure & Messaging (InM), Modeling & Methodology (MnM), Orders & Observations (OO), Patient Administration (PA), Patient Care (PC), Pharmacy, Public Health & Emergency Response (PHER), Security, Structured Documents (SD), Templates, US Realm Steering Committee, and Vocabulary
* In addition, some resources were developed in coordination with the following organizations: [DICOM](http://medical.nema.org/standard.html) and [IHE](http://http:/www.ihe.net)
* The following individuals were development leads or primary content contributors for resource development: James Agnew, Woody Beeler, David Booth, Lorraine Constable, Bo Dagnall, Jean-Henri Duteau, Richard Ettema, Hugh Glover, Grahame Grieve, Eric Haas, Rob Hausam, David Hay, Alexander Henket, Paul Knapp, Ewout Kramer, Patrick Loyd, Joginder Madra, Joshua Mandel, Brett Marquard, Lloyd McKenzie, John Moehrke, Claude Nanjo, Chris Nickerson, Brian Postlethwaite, Brynn Rhodes, and Rik Smithies
* Invaluable support through review of ballots and/or participation in FHIR formal QA processes was provided by: Tomasz Adamusiak, Marla Albitz, Al Amyot, Larry Babb, Yongjian Bao, Nagesh Bashyam, Randy Bates, Calvin Beebe, Woody Beeler, Ruth Berge, Peter Bernhardt, Bernd Blobel, Andy Bond, Keith Boone, Aziz Boxwala , Laura Bright, Helen Broberg, Hans Buitendijk, Carl Burnett, Dave Carlson, Victor Chai, Dennis Cheung, George Cole, Kathleen Connor, Teresa Conway, Todd Cooper, Carmela Couderc, Guilherme Del Fiol, Savithri Devaraj, Gary Dickinson, Robert Dieterle, Gay Dolin, Bob Dollin, Jean Duteau, Michael Ekaireb, Sergio Enriquez, David Fallas, Rick Geimer, Zabrina Gonzaga, William Goosen, Matthew Graham, Ollie Gray, Andy Gregorowicz, Grahame Grieve, Wei Guo, Eric Haas, Russel Hamm, Yegor Hanov, Smita Hastak, John Hatem, Rob Hausam, David Hay, Mike Henderson, Yan Heras, Scott Hollington, Rob Horn, Wendy Huang, Lenel James, Hozefa Jerapurwala, Emma Jones, Peter Jordan, Anthony Julian, Richard Kavanagh, Kensaku Kawamoto, Thomson Khun, Ted Klein, Paul Knapp, Helmut Koenig, Ewout Kramer, Mark Kramer, James Kretz, Eric Larson, Khalid Latif, Michael Lawley, Russ Leftwich, Ben Levy, Daniel Loewenstein, Stefano Lotti, Patrick Loyd, Thomas Lukasik, Jay Lyle, Josh Mandel, Perry Mar, Brett Marquard, Rute Martins, Jason Mathews , Vince McCauley, Robert McClure, Clem McDonald, Lloyd McKenzie, Tim McNeil, Riki Merrick, Bob Milius, Michelle Miller, Zoran Milosevic, John Moehrke, Rick Moore, Sean Moore, Saurabh Moudgil, Galen Mulrooney, Stephen Munro, Claude Nanjo, Dale Nelson, Shamil Nizamov, Frank Oemig, Tom Oniki, Nancy Orvis, David Parker, Brian Pech, Melva Peters, Vasil Peytchev, Fran Pivonka, Pu, Dave Pyke, Gila Pyke, Stan Rankins, Gordon Raup, Dianne Reeves, Greg Rehwoldt, Bryn Rhodes , Mitra Rocca, Michael Rossman, Iryna Roy, Daniel Rutz, Rob Savage, Brian Scheller, Amnon Shabo, Kevin Shekleton, Ioana Singureanu, Anne Smith, Rik Smithies, Harry Solomon, Corey Spears, Rene Spronk, Andy Stechishin, Howard Strasberg, Jenni Syed, David Tao, Sylvia Thun, Leslie Tompkins, Chris Toomey, Andrew Torres, Mollie Ullman-Coliere, Cary Ussery, Michael van der Zel, Bas van Poppel, Serafina Versaggi, Mead Walker, Jason Walonoski, Kathy Walsh, Yunwei Wang, Ian Williams and Timothy Williams
* Innumerable others have assisted through discussions on the Implementer's Skype chat, http://chat.fhir.org, the FHIR list server, participation at HL7 Working Group Meetings and through submission of change requests
* THANK YOU - without the support of this community, FHIR would not exist.

**cvx.html**

**Using CVX (Vaccine Administered) with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Summary**

|  |  |
| --- | --- |
| Source | Available at <http://www2a.cdc.gov/vaccines/iis/iisstandards/vaccines.asp?rpt=cvx>. Developed by The CDC's National Center of Immunization and Respiratory Diseases ([NCIRD](http://www.cdc.gov/ncird/)) |
| System | The URI to identify CVX codes is http://hl7.org/fhir/sid/cvx |
| Version | If it is desired to exchange the version, use the date of the last updated code in the format YYYYMMDD |
| Code | Use the value of "CVX Code" column in the web page |
| Display | Use the value of the "Short Description" column in the web page |
| Inactive | Todo: Describe how it is determined which concepts are inactive |
| Subsumption | No Subsumption relationships are defined by CVX |
| Filter Properties | None are described yet |

**Version Issues**

If it is desired to exchange the version, use the date of the last updated code in the format YYYYMMDD.

**Copyright/License Issues**

CVX has no copyright acknowledgement needed, nor are there any license terms to adhere to.

**CVX Filter Properties**

No need for filters identified yet.

**Implicit Value Sets**

No need for implicit value sets identified yet.

**datatypes-definitions.html**

**Data Type Detailed Descriptions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page provides the detailed descriptions for the data types

**Table of Contents**

|  |  |
| --- | --- |
| **Primitive Types** | **Complex Types** |

**Primitive Types**

See also [Base Definition](file:///C:\temp\datatypes.html#primitive), [Examples](file:///C:\temp\datatypes-examples.html#primitive), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#primitive), [Mappings](file:///C:\temp\datatypes-mappings.html#primitive) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#primitive).

*todo*

**Attachment**

See also [Base Definition](file:///C:\temp\datatypes.html#Attachment), [Examples](file:///C:\temp\datatypes-examples.html#Attachment), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Attachment), [Mappings](file:///C:\temp\datatypes-mappings.html#Attachment) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Attachment).

**Identifier**

See also [Base Definition](file:///C:\temp\datatypes.html#identifier), [Examples](file:///C:\temp\datatypes-examples.html#identifier), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#identifier), [Mappings](file:///C:\temp\datatypes-mappings.html#identifier) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#identifier).

**Coding**

See also [Base Definition](file:///C:\temp\datatypes.html#Coding), [Examples](file:///C:\temp\datatypes-examples.html#Coding), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Coding), [Mappings](file:///C:\temp\datatypes-mappings.html#Coding) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Coding).

**CodeableConcept**

See also [Base Definition](file:///C:\temp\datatypes.html#CodeableConcept), [Examples](file:///C:\temp\datatypes-examples.html#CodeableConcept), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#CodeableConcept), [Mappings](file:///C:\temp\datatypes-mappings.html#CodeableConcept) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#CodeableConcept).

**Quantity**

See also [Base Definition](file:///C:\temp\datatypes.html#Quantity), [Examples](file:///C:\temp\datatypes-examples.html#Quantity), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Quantity), [Mappings](file:///C:\temp\datatypes-mappings.html#Quantity) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Quantity).

**Range**

See also [Base Definition](file:///C:\temp\datatypes.html#Range), [Examples](file:///C:\temp\datatypes-examples.html#Range), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Range), [Mappings](file:///C:\temp\datatypes-mappings.html#Range) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Range).

**Ratio**

See also [Base Definition](file:///C:\temp\datatypes.html#Ratio), [Examples](file:///C:\temp\datatypes-examples.html#Ratio), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Ratio), [Mappings](file:///C:\temp\datatypes-mappings.html#Ratio) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Ratio).

**Period**

See also [Base Definition](file:///C:\temp\datatypes.html#Period), [Examples](file:///C:\temp\datatypes-examples.html#Period), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Period), [Mappings](file:///C:\temp\datatypes-mappings.html#Period) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Period).

**SampledData**

See also [Base Definition](file:///C:\temp\datatypes.html#SampledData), [Examples](file:///C:\temp\datatypes-examples.html#SampledData), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#SampledData), [Mappings](file:///C:\temp\datatypes-mappings.html#SampledData) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#SampledData).

**HumanName**

See also [Base Definition](file:///C:\temp\datatypes.html#HumanName), [Examples](file:///C:\temp\datatypes-examples.html#HumanName), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#HumanName), [Mappings](file:///C:\temp\datatypes-mappings.html#HumanName) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#HumanName).

**Address**

See also [Base Definition](file:///C:\temp\datatypes.html#Address), [Examples](file:///C:\temp\datatypes-examples.html#Address), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Address), [Mappings](file:///C:\temp\datatypes-mappings.html#Address) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Address).

**ContactPoint**

See also [Base Definition](file:///C:\temp\datatypes.html#ContactPoint), [Examples](file:///C:\temp\datatypes-examples.html#ContactPoint), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#ContactPoint), [Mappings](file:///C:\temp\datatypes-mappings.html#ContactPoint) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#ContactPoint).

**Timing**

See also [Base Definition](file:///C:\temp\datatypes.html#Timing), [Examples](file:///C:\temp\datatypes-examples.html#Timing), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Timing), [Mappings](file:///C:\temp\datatypes-mappings.html#Timing) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Timing).

**Signature**

See also [Base Definition](file:///C:\temp\datatypes.html#Signature), [Examples](file:///C:\temp\datatypes-examples.html#Signature), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Signature), [Mappings](file:///C:\temp\datatypes-mappings.html#Signature) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Signature).

**Annotation**

See also [Base Definition](file:///C:\temp\datatypes.html#Annotation), [Examples](file:///C:\temp\datatypes-examples.html#Annotation), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Annotation), [Mappings](file:///C:\temp\datatypes-mappings.html#Annotation) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Annotation).

**datatypes-examples.html**

**Data Type Examples**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page includes additional examples of the data types, based on common usages and questions

**Table of Contents**

**Primitive Types**

**Complex Types**

**Primitive Types**

A boolean true value:

<active value="true" />

A negative integer value:

<score value="-14" />

A high-precision decimal value:

<pi value="3.14159265358979323846264338327950288419716939937510" />

A stream of bytes, base64 encoded:

<data value="/9j/4...KAP//Z" /> <!-- covers many lines -->

A unicode string:

<caption value="Noodles are called ?? in Chinese" />

A URI that points to a website:

<reference value="http://hl7.org/fhir" />

A URI that is a urn:

<id value="urn:isbn:0451450523" />

A date of birth:

<date value="1951-06-04" />

An approximate date of birth:

<date value="1951-06" />

The instant a document was created, including time zone:

<instant value="2013-06-08T10:57:34+01:00" />

The instant a document was created, expressed in UTC, with milliseconds:

<instant value="2013-06-08T09:57:34.2112Z" />

2:35pm in the afternoon:

<time value="14:35" />

**String Patterns**

A URI that is the root oid of HL7:

<root value="urn:oid:2.16.840.1.113883" />

A URI that is a uuid:

<id value="urn:uuid:a5afddf4-e880-459b-876e-e4591b0acc11" />

A code:

<code value="acq4+acq5" />

A code with single internal space:

<code value="Question 4b" />

A numeric id:

<id value="314" />

An alphanumeric id:

<id value="alpha-gamma-14" />

**Attachment**

See also [Base Definition](file:///C:\temp\datatypes.html#Attachment), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Attachment), [Mappings](file:///C:\temp\datatypes-mappings.html#Attachment), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Attachment) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Attachment).

A PDF document:

<document>

<contentType value="application/pdf" />

<language value="en" />

<data value="/9j/4...KAP//Z" /> <!-- covers many lines -->

<title value="Definition of Procedure" />

</document>

document : {

contentType : { value : "application/pdf" },

language : { value : "en" },

data : { value : "/9j/4...KAP//Z"},

title : { value : "Definition of Procedure" }

}

Since the JSON examples have the same structure as the XML, only XML is shown for the rest of the examples.

A reference to a DICOM image via WADO:

<image>

<contentType value="application/dicom" />

<url value="http://10.1.2.3:1000/wado?requestType=WADO&amp;wado\_details..." />

<hash value="EQH/..AgME" />

</image>

**Identifier**

See also [Base Definition](file:///C:\temp\datatypes.html#Identifier), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Identifier), [Mappings](file:///C:\temp\datatypes-mappings.html#Identifier), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Identifier) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Identifier).

**Examples**

A primary key from an application table (an OID in the space allocated by HL7 to some organization to further sub-allocate):

<identifier>

<use value="official" />

<system value="urn:oid:2.16.840.1.113883.16.4.3.2.5" />

<value value="123" />

</identifier>

A patient identifier defined by a hospital:

<identifier>

<use value="official" />

<system value="http://www.acmehosp.com/patients" />

<value value="44552" />

<period>

<start value="2003-05-03" />

</period>

</identifier>

In this case, the period is used to track when the identifier was first assigned to the patient.

An identifier that refers to a patient FHIR resource on a particular system:

<identifier>

<system value="urn:ietf:rfc:3986" />

<value value="http://pas-server/xxx/Patient/443556" />

</identifier>

This is not a resource reference - it's a logical reference by the patient identifier.

A UUID:

<identifier>

<use value="temp" />

<system value="urn:ietf:rfc:3986" />

<value value="urn:uuid:a76d9bbf-f293-4fb7-ad4c-2851cac77162" />

</identifier>

UUIDs are often used for temporary identifiers, though this is not necessary.

A US SSN:

<identifier>

<use value="usual" />

<type>

<coding>

<system value="http://hl7.org/fhir/v2/0203"/>

<code value="SSN"/>

</coding>

</type>

<system value="http://hl7.org/fhir/sid/us-ssn" />

<value value="000111111" />

</identifier>

Notes:

* US SSNs are often presented like this: 000-11-1111, the dashes are for presentation and should be removed, as specified in the [definition of ssn-us](file:///C:\temp\terminologies-systems.html#identifiersystems)
* The use of "usual" means that this institution prefers to use SSN when identifying the patient

A medical record number assigned on 5-July 2009:

<identifier>

<use value="usual" />

<type>

<coding>

<system value="http://hl7.org/fhir/v2/0203"/>

<code value="MRN"/>

</coding>

</type>

<system value="urn:oid:0.1.2.3.4.5.6.7" />

<value value="2356" />

<period>

<start value="2009-07-05" />

</period>

</identifier>

**Coding**

See also [Base Definition](file:///C:\temp\datatypes.html#Coding), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Coding), [Mappings](file:///C:\temp\datatypes-mappings.html#Coding), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Coding) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Coding).

**Examples**

A simple code for headache, in ICD-10:

<code>

<system value="http://hl7.org/fhir/sid/icd-10" />

<code value="G44.1" />

</code>

A SNOMED CT expression:

<problem>

<system value="http://snomed.info/sct" />

<code value="128045006:{363698007=56459004}" />

</problem>

**CodeableConcept**

See also [Base Definition](file:///C:\temp\datatypes.html#CodeableConcept), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#CodeableConcept), [Mappings](file:///C:\temp\datatypes-mappings.html#CodeableConcept), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#CodeableConcept) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#CodeableConcept).

**Examples**

A simple code for headache initially coded in SNOMED CT (by picking the SNOMED CT code from a pick-list), and then translated to ICD-10:

<concept>

<coding>

<system value="http://hl7.org/fhir/sid/icd-10" />

<code value="R51" />

</coding>

<coding>

<system value="http://snomed.info/sct" />

<code value="25064002" />

<display value="Headache" />

<userSelected value="true" />

</coding>

<text value="general headache" />

</concept>

A concept represented in an institution's local coding systems for unit for which no UCUM equivalent exists:

<unit>

<coding>

<system value="urn:oid:2.16.840.1.113883.19.5.2" />

<code value="tab" />

<display value="Tablet" />

</coding>

<coding>

<system value="http://unitsofmeasure.org" />

</coding>

</unit>

A SNOMED CT expression:

<diagnosis>

<coding>

<system value="http://snomed.info/sct" />

<code value="128045006:{363698007=56459004}" />

</coding>

<text value="Cellulitis of the foot" />

</diagnosis>

In this case, there is no display element, because no display is defined for SNOMED CT expressions.

Using the value set:

The results on a urinalysis strip:

<valueCoding>

<system value="http://example.org/codes/simple-grades" />

<code value="+" />

</valueCoding>

And where the value set would be something like this:

<ValueSet xmlns="http://hl7.org/fhir">

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml">

<p>Possible Clinistix codes: neg, trace, +, ++, and +++</p>

</div>

</text>

<url value="http://hl7.org/fhir/ValueSet/clinistix"/>

<name value="Codes for Clinistix"/>

<publisher value="HL7"/>

<contact>

<name value="FHIR project team"/>

<telecom>

<system value="url"/>

<value value="http://hl7.org/fhir"/>

</telecom>

</contact>

<description value="Clinistix Codes"/>

<status value="draft"/>

<experimental value="true"/>

<date value="2013-10-01"/>

<compose>

<include>

<system value="http://example.org/codes/simple-grades"/>

<concept>

<code value="neg"/>

</concept>

<concept>

<code value="trace"/>

</concept>

<concept>

<code value="+"/>

</concept>

<concept>

<code value="+"/>

</concept>

<concept>

<code value="++"/>

</concept>

<concept>

<code value="+++"/>

</concept>

</include>

</compose>

</ValueSet>

**Quantity**

See also [Base Definition](file:///C:\temp\datatypes.html#Quantity), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Quantity), [Mappings](file:///C:\temp\datatypes-mappings.html#Quantity), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Quantity) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Quantity).

**Examples**

A duration:

<time>

<value value="25" />

<unit value="sec" />

<system value="http://unitsofmeasure.org" />

<code value="s" />

</time>

A concentration where the value was out of range:

<result>

<value value="40000" />

<comparator value="&gt;" />

<unit value="mcg/L" />

<system value="http://unitsofmeasure.org" />

<code value="ug" />

</result>

An amount of prescribed medication:

<dose>

<value value="3" />

<unit value="capsules" />

<system value="http://snomed.info/sct" />

<code value="385049006" />

</dose>

A price (coded using currency codes defined in ISO 4217):

<cost>

<value value="25.45" />

<unit value="US$" />

<system value="urn:iso:std:iso:4217" />

<code value="USD" />

</cost>

**Range**

See also [Base Definition](file:///C:\temp\datatypes.html#Range), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Range), [Mappings](file:///C:\temp\datatypes-mappings.html#Range), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Range) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Range).

**Examples**

Range of Quantity (distance):

<estimate>

<low>

<value value="1.6" />

<unit value="m" />

</low>

<high>

<value value="1.9" />

<unit value="m" />

</high>

</estimate>

**Ratio**

See also [Base Definition](file:///C:\temp\datatypes.html#Ratio), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Ratio), [Mappings](file:///C:\temp\datatypes-mappings.html#Ratio), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Ratio) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Ratio).

**Examples**

Titer (Ratio of integer:integer)

<result>

<numerator>

<value value="1" />

</numerator>

<denominator>

<value value="128" />

</denominator>

</result>

Unit cost (Ratio of Money:Quantity):

<charge>

<numerator>

<value value="103.50" />

<unit value="US$" />

<code value="USD" />

<system value="urn:iso:std:iso:4217" />

</numerator>

<denominator>

<value value="1" />

<unit value="day" />

<code value="day" />

<system value="http://unitsofmeasure.org" />

</denominator>

</charge>

**Period**

See also [Base Definition](file:///C:\temp\datatypes.html#Period), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Period), [Mappings](file:///C:\temp\datatypes-mappings.html#Period), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Period) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Period).

**Examples**

23rd May 2011 to 27th May, including 27th May:

<coverage>

<start value="2011-05-23" />

<end value="2011-05-27" />

</coverage>

**SampledData**

See also [Base Definition](file:///C:\temp\datatypes.html#SampledData), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#SampledData), [Mappings](file:///C:\temp\datatypes-mappings.html#SampledData), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#SampledData) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#SampledData).

**Example**

The output from an EKG device:

<sampledData>

<origin>

<value value="0"/>

<unit value="μV"/>

<system value="http://unitsofmeasure.org"/>

<code value="uV"/>

</origin>

<period value="2"/>

<factor value="2.5"/>

<dimensions value="1"/>

<data value="-4 -13 -18 -18 -18 -17 -16 -16 -16 -16 -16 -17 -18 -18 -18 ...."/>

</sampledData>

**HumanName**

See also [Base Definition](file:///C:\temp\datatypes.html#HumanName), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#HumanName), [Mappings](file:///C:\temp\datatypes-mappings.html#HumanName), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#HumanName) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#HumanName).

A Simple example

<name>

<family value="Everyman" />

<given value="Adam" />

<given value="A." />

</name>

Composite names

<name>

<family value="Contrata" />

<given value="Mary Jane" />

</name>

These cases can be quite ambiguous - is "Mary Jane" one name, or two? Different systems, and data enterers may treat this differently, and the person themselves may not know. Parts are allowed to contain spaces, but systems should consider how to treat these cases. Composite names separated by "-" should be treated as a single name part.

A common pattern: a person is called by a name other than that expected from their official name (first given name in most cultures).

<name>

<use value="official" />

<family value="Chalmers" />

<given value="Peter" />

<given value="James" />

</name>

<name>

<use value="usual" />

<given value="Jim" />

</name>

This same pattern is often encountered with immigrants, who retain their real name for official use, but adopt a localized name for everyday use:

<name>

<use value="official" />

<family value="Sczypinski" />

<given value="Piotr" />

<given value="Andre" />

</name>

<name>

<use value="usual" />

<family value="Skipper" />

<given value="Jim" />

</name>

Karen van Hentenryck is of Dutch origin, and the "van" is a voorvoegsel.

<name>

<use value="official" />

<family value="van Hentenryck" >

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-own-prefix" >

<valueString value="van" />

</extension>

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-own-name">

<valueString value="Hentenryck" />

</extension>

</family>

<given value="Karen" />

</name>

See [the Extensibility Example for more information](file:///C:\temp\extensibility-examples.html) about the use of extensions.

Complex example from Germany: Dr.phil. Regina Johanna Maria von Hochheim-Weilenfels, NCFSA. This example shows extensive use of multiple given names, prefixes, suffixes, for academic degrees, nobility titles, and professional designations.

<name>

<use value="official" />

<family value="von Hochheim-Weilenfels">

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-own-prefix" >

<valueString value="von" />

</extension>

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-own-name">

<valueString value="Hochheim-Weilenfels" />

</extension>

</family>

<given value="Regina" />

<given value="Johanna" />

<given value="Maria" />

<prefix value="Dr. phil.">

<extension url="http://hl7.org/fhir/StructureDefinition/iso21090-EN-qualifier" >

<valueCode value="AC" />

</extension>

</prefix>

<suffix value="NCFSA" />

</name>

<name>

<use value="maiden" />

<family value="Hochheim" />

</name>

This example makes use of the ISO 21090 extensions to carry the rarely used ISO 21090 qualifier attribute "AC".

Japanese example in the three forms: ideographic (Kanji), syllabic (Hiragana) and alphabetic (Romaji).

<name>

<extension url="http://hl7.org/fhir/StructureDefinition/iso21090-EN-representation">

<valueCode value="IDE" />

</extension>

<family value="Ã¦Å“Â¨Ã¦Ââ€˜" />

<given value="Ã©â‚¬Å¡Ã§â€Â·" />

</name>

<name>

<extension url="http://hl7.org/fhir/StructureDefinition/iso21090-EN-representation">

<valueCode value="SYL" />

</extension>

<family value="Ã£ÂÂÃ£â€šâ‚¬Ã£â€šâ€°" />

<given value="Ã£ÂÂ¿Ã£ÂÂ¡Ã£ÂÅ " />

</name>

<name>

<extension url="http://hl7.org/fhir/StructureDefinition/iso21090-EN-representation">

<valueCode value="ABC" />

</extension>

<family value="KIMURA" />

<given value="MICHIO" />

</name>

The three forms may be differentiated by the character subset each contains, but some systems require the differentiation to be made manually, which can be done using the ED representation extension.

Russian example in the two forms: cyrillic, and latin:

<name>

<family value="Ãâ€¢ÃÅ“Ãâ€¢Ãâ€ºÃËœÃÂ" />

<given value="ÃËœÃâ€™ÃÂÃÂ" />

<given value="Ãâ€™Ãâ€ºÃÂÃâ€ÃËœÃÅ“ÃËœÃÂ ÃÅ¾Ãâ€™ÃËœÃÂ§" />

</name>

<name>

<family value="EMELIN" />

<given value="IVAN" />

<given value="VLADIMIROVICH" />

</name>

In Russian usage, these names are known as the domestic and foreign names respectively. The two forms are differentiated by the character subset each contains.

Scandinavian example: Erikson is the family name. Jan Erik are the given names, and Ãƒâ€“stlund the family name of the mother, which is taken as a Mellannamn.

<name>

<use value="official" />

<family value="Erikson" />

<given value="Jan" />

<given value="Erik" />

<given value="Ãƒâ€“stlund">

<extension url="http://hl7.org/fhir/StructureDefinitioniso-20190#name-qualifier" >

<valueCoding>

<code value="MID" />

<system value="http://hl7.org/fhir/v3/EntityNamePartQualifier2" />

</valueCoding>

</extension>

</given>

</name>

This example makes use of the ISO 21090 extension to carry the culture specific ISO 21090 qualifier attribute "MID" for the Mellannamn.

Then Jan Erikson has a daughter, Karin, with his wife Margrete Hansen. The first communications of the new born name is "Margrete Jente" (Margrete's Girl) and the mother's family name, not the given name (Karin). The father's Family name is not used at all. This is a known temporary name assigned directly after the birth of the child.

<name>

<use value="temp" />

<!-- use could be OR+OLD, depends how record keeping is done -->

<family value="Hansen" />

<given value="Margrete Jente" />

</name>

The baby's name is subsequently changed to the fathers' family name, and to use the mother's name as mellomnamn.

<name>

<use value="official" />

<family value="Erikson" />

<given value="Karin" />

<given value="Hansen">

<extension url="http://hl7.org/fhir/StructureDefinitioniso-20190#name-qualifier" >

<valueCoding>

<code value="MID" />

<system value="http://hl7.org/fhir/v3/EntityNamePartQualifier2" />

</valueCoding>

</extension>

</given>

</name>

Later, Karin gets married to Per Berg, and decides to adopt Berg as her family name, and also decides to use Erikson as the mellom navn. (Note: Karin could have chosen to use another mellom navn, e.g. the family name of her mother, her father or other family names as specified by naming laws of the country in question).

<name>

<use value="old" />

<family value="Erikson" />

<given value="Karin" />

<given value="Hansen">

<extension url="http://hl7.org/fhir/StructureDefinitioniso-20190#name-qualifier" >

<valueCoding>

<code value="MID" />

<system value="http://hl7.org/fhir/v3/EntityNamePartQualifier2" />

</valueCoding>

</extension>

</given>

</name>

<name>

<use value="official" />

<family value="Berg" />

<given value="Karin" />

<given value="Erikson">

<extension url="http://hl7.org/fhir/StructureDefinitioniso-20190#name-qualifier" >

<valueCoding>

<code value="MID" />

<system value="http://hl7.org/fhir/v3/EntityNamePartQualifier2" />

</valueCoding>

</extension>

</given>

</name>

<name>

<use value="usual" />

<family value="Berg" />

<given value="Karin" />

</name>

**W3C International Examples**

These examples are taken from the [W3C International Examples](http://www.w3.org/International/questions/qa-personal-names), which should be consulted for further information.

A patronymic is "The part of a name that links to the genealogy":

<name>

<text value="BjÃƒÂ¶rk GuÃƒÂ°mundsdÃƒÂ³ttir"/>

<family value="GuÃƒÂ°mundsdÃƒÂ³ttir"/>

<given value="BjÃƒÂ¶rk"/>

</name>

A patronymic with a "son/daughter of" appellation:

<name>

<text value="Isa bin Osman"/>

<family value="bin Osman"/>

<given value="Isa"/>

</name>

A Chinese name with a generational name:

<name>

<text value="Ã¦Â¯â€ºÃ¦Â³Â½Ã¤Â¸Å“"/> <!-- left to right -->

<family value="Ã¦Â¯â€º"/>

<given value="Ã¦Â³Â½Ã¤Â¸Å“"/>

</name>

<name>

<text value="Mao Ze Dong"/> <!-- left to right -->

<family value="Mao"/>

<given value="Ze Dong"/>

</name>

Note that many systems in China do not store family and given names separately, and just use text.

Additional Western name (see also example above):

<name>

<use value="official" />

<family value="Yao" />

<given value="Ming" />

</name>

<name>

<use value="usual" />

<given value="Fred" />

</name>

Composite Family name:

<name>

<family value="CarreÃƒÂ±o QuiÃƒÂ±ones" >

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-fathers">

<valueString value="CarreÃƒÂ±o" />

</extension>

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-mothers">

<valueString value="QuiÃƒÂ±ones" />

</extension>

</family>

<given value="MarÃƒÂ­a-Jose" />

</name>

Note that it is optional whether to break down the family name to mother's and father's parts, and not always of value, as in this Brazilian Example:

<name>

<family value="Santos Tavares Melo Silva" />

<given value="JosÃƒÂ©" />

<given value="Eduardo" />

</name>

Note that this naming pattern can become quite extreme ("Pedro de AlcÃƒÂ¢ntara JoÃƒÂ£o Carlos Leopoldo Salvador Bibiano Francisco Xavier de Paula LeocÃƒÂ¡dio Miguel Gabriel Rafael Gonzaga"). A more practical example is

<name>

<family value="Costa Teixeira" >

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-fathers">

<valueString value="Costa" />

</extension>

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-mothers">

<valueString value="Teixeira" />

</extension>

</family>

<given value="Manuel" />

</name>

After marriage, this may change to

<name>

<family value="Costa Teixeira Sanches" >

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-fathers">

<valueString value="Costa" />

</extension>

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-mothers">

<valueString value="Teixeira" />

</extension>

<extension url="http://hl7.org/fhir/StructureDefinition/humanname-partners-name">

<valueString value="Sanches" />

</extension>

</family>

<given value="Manuel" />

</name>

Russian Examples (using Cyrillic):

<name>

<family value="Ãâ€¢ÃÂ»Ã‘Å’Ã‘â€ ÃÂ¸ÃÂ½" />

<given value="Ãâ€˜ÃÂ¾Ã‘â‚¬ÃÂ¸Ã‘Â" />

<given value="ÃÂÃÂ¸ÃÂºÃÂ¾ÃÂ»ÃÂ°ÃÂµÃÂ²ÃÂ¸Ã‘â€¡" />

</name>

<name>

<family value="Ãâ€¢ÃÂ»Ã‘Å’Ã‘â€ ÃÂ¸ÃÂ½ÃÂ°" />

<given value="ÃÂÃÂ°ÃÂ¸ÃÂ½ÃÂ°" />

<given value="ÃËœÃÂ¾Ã‘ÂÃÂ¸Ã‘â€žÃÂ¾ÃÂ²ÃÂ½ÃÂ°" />

</name>

Example with Initial:

<name>

<family value="Public" />

<given value="John" />

<given value="Q." />

</name>

Other Examples:

<name>

<text value="Velikkakathu Sankaran Achuthanandan"/>

<family value="Velikkakathu" />

<given value="Sankaran" />

<given value="Achuthanandan" />

</name>

<name>

<text value="Kogaddu Birappa Timappa Nair"/>

<family value="Nair" />

<given value="Birappa" />

<given value="Timappa" />

<prefix value="Kogaddu" />

</name>

<name>

<text value="Aditya Pratap Singh Chauhan"/>

<family value="Singh" />

<given value="Aditya" />

<given value="Pratap" />

<suffix value="Chauhan" />

</name>

<name>

<text value="Madurai Mani Iyer"/>

<given value="Mani" />

<prefix value="Madurai" />

<suffix value="Iyer" />

</name>

<name>

<text value="Abu Karim Muhammad al-Jamil ibn Nidal ibn Abdulaziz al-Filistini"/>

<family value="ibn Nidal ibn Abdulaziz" />

<given value="Muhammad" />

<given value="al-Jamil" />

<prefix value="Abu Karim" />

<suffix value="al-Filistini" />

</name>

*Todo: need to discuss this with Indian / Arabic implementers*. Note that collecting and storing the *text* element makes the primary purpose of the structured parts for index/searching, and fidelity of the name parts is not critical.

**Address**

See also [Base Definition](file:///C:\temp\datatypes.html#Address), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Address), [Mappings](file:///C:\temp\datatypes-mappings.html#Address), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Address) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Address).

Note about these examples: the most important element in an address is the 'text' element - this defines what is printed on the envelope, the actual postal address. The other data elements are provided to support either data analysis based on a patient's address, or for the many systems that exchange fully structured addresses specific to a particular culture. For this reason, these examples focus on the structured data elements of the address.

**Example**

HL7 office's address.

<address>

<use value="work" />

<text value="1050 W Wishard Blvd

RG

5th floor

Indianapolis, IN 46240" />

<line value="1050 W Wishard Blvd" />

<line value="RG 5th floor" />

<city value="Indianapolis" />

<state value="IN" />

<postalCode value="46240" />

</address>

A UK example address, with the county 'HUDDERSFIELD'.

<address>

<text value="1 Back Lane&#13;&#10;Holmfirth&#13;&#10;HUDDERSFIELD&#13;&#10;HD7 1HQ"/>

<line value="1 Back Lane"/>

<city value="Holmfirth"/>

<district value="HUDDERSFIELD"/>

<postalCode value="HD7 1HQ"/>

</address>

A temporary postal address - i.e. an address that it doesn't make sense to try and visit.

<address>

<use value="temp"/>

<type value="postal"/>

<line value="PO Box 31445"/>

<city value="Erewhon"/>

<postalCode value="0001"/>

</address>

The next set of examples are taken from the official [international postal union](http://www.upu.int) examples.

|  |  |
| --- | --- |
| Rue LougoraÃƒÂ¯a 12, app. 10 | thoroughfare type, name and number, apartment |
| Korolevo | locality |
| 223016 NOVY DVOR | postcode + post office name |
| Minsk | District name of district |
| Minsk | Region name of region |
| BELARUS | Country |

<address>

<line value="Rue LougoraÃƒÂ¯a 12, app. 10"/>

<city value="Korolevo"/>

<district value="Minsk"/>

<state value="Minsk" />

<country value="BELARUS" />

</address>

|  |  |
| --- | --- |
| Protea Apt 12 | apartment number |
| 22 Ally Hassan Mwinyi | premise + street |
| 14111 MSASANI | postcode + locality |
| DAR ES SALAM | province |
| TANZANIA | country |

<address>

<line value="Protea Apt 12"/>

<line value="22 Ally Hassan Mwinyi"/>

<city value="MSASANI"/>

<state value="DAR ES SALAM" />

<postalCode value="14111"/>

<country value="TANZANIA" />

</address>

|  |  |
| --- | --- |
| 15 Shiri | premise + sub-locality 2 |
| Kimashuku Village | sub-locality |
| 25204 MACHAME KUSINI | postcode + locality |
| HAI | sub-province |
| KILIMANJARO | province |
| TANZANIA | country |

<address>

<extension url="http://hl7.org/fhir/StructureDefinition/iso21090-ADXP-precinct">

<valueCode value="Kimashuku Village"/>

</extension>

<line value="15 Shiri"/>

<city value="MACHAME KUSINI"/>

<district value="HAI"/>

<state value="KILIMANJARO" />

<postalCode value="25204"/>

<country value="TANZANIA" />

</address>

|  |  |
| --- | --- |
| 705-1104 | building number + door |
| 56, Dalgubeol-daero 323beon-gil, Suseong-gu | street |
| Daegu | city |
| Rep. OF KOREA | country |
| 706-907 | postcode |

<address>

<line value="705-1104"/>

<line value="56, Dalgubeol-daero 323beon-gil, Suseong-gu"/>

<city value="Daegu"/>

<country value="Rep. OF KOREA" />

<postalCode value="706-907"/>

</address>

|  |  |
| --- | --- |
| 3F | building floor |
| 42, Toegye-ro 77beon-gil | street |
| Chuncheon-si | city |
| Gangwon-do | province |
| Rep. OF KOREA | country |
| 200-066 | postcode |

<address>

<line value="3F"/>

<line value="42, Toegye-ro 77beon-gil"/>

<city value="Chuncheon-si"/>

<state value="Gangwon-do" />

<country value="Rep. OF KOREA" />

<postalCode value="200-066"/>

</address>

|  |  |
| --- | --- |
| Ga-B101 | building number + door |
| 136, Sesil-ro | street |
| Busan | city |
| Rep. OF KOREA | country |
| 612-837 | postcode |

<address>

<line value="Ga-B101"/>

<line value="136, Sesil-ro"/>

<city value="Busan"/>

<country value="Rep. OF KOREA" />

<postalCode value="612-837"/>

</address>

**ContactPoint**

See also [Base Definition](file:///C:\temp\datatypes.html#ContactPoint), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#ContactPoint), [Mappings](file:///C:\temp\datatypes-mappings.html#ContactPoint), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#ContactPoint) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#ContactPoint).

**Example**

Home phone number:

<telecom>

<system value="phone" />

<value value="(555) 675 5745" />

<use value="home" />

</telecom>

In ISO 21090, [CDA](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7) and other [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) specifications, the TEL data type requires that the telephone number be represented as a formal URI in [RFC 3166](http://www.ietf.org/rfc/rfc3966.txt) syntax. Many CDA documents violate this rule, since the telephone numbers are not collected from the user in conformance with the underlying rules (particularly around extensions, notes about use, etc.), so the base FHIR type does not require the phone number to be conformant. A ISO 21090 extension can be used to convey this if desired:

**Example**

Formally correct URI:

<telecom>

<extension url="http://hl7.org/fhir/StructureDefinition/iso21090-TEL-address" >

<valueUri value="tel:+15556755745" />

</extension>

<system value="phone" />

<value value="(555) 675 5745" />

<use value="home" />

</telecom>

**Timing**

See also [Base Definition](file:///C:\temp\datatypes.html#Timing), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Timing), [Mappings](file:///C:\temp\datatypes-mappings.html#Timing), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Timing) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Timing).

**Example**

A series of appointments for radiotherapy:

<schedule>

<event value="2012-01-07T09:00:00+10:00" />

<event value="2012-01-14T09:00:00+10:00" />

<event value="2012-01-22T11:00:00+10:00" />

</schedule>

BID (twice a day) (no start or end specified):

<schedule>

<repeat>

<frequency value="2" />

<period value="1" />

<periodUnit value="d" />

</repeat>

</schedule>

1/2 an hour before breakfast for 10 days from 23-Dec 2011:

<schedule>

<repeat>

<boundsPeriod>

<start value="2011-12-23" />

<end value="2012-01-02" />

</boundsPeriod>

<when value="ACM" />

<offset value="30" />

</repeat>

</schedule>

Note that the end date is inclusive like the end date of a Period.

TID, for 14 days:

<schedule>

<repeat>

<boundsDuration>

<value value="14" />

<unit value="d" />

<system value="http://unitsofmeasure.org" />

<code value="d" />

</boundsDuration>

<frequency value="3" />

<period value="1" />

<periodUnit value="d" />

</repeat>

</schedule>

BID, start on 7/1/2015 at 1:00 PM:

<schedule>

<repeat>

<boundsPeriod>

<start value="2015-07-01T13:00:00" />

</boundsPeriod>

<frequency value="2" />

<period value="1" />

<periodUnit value="d" />

</repeat>

</schedule>

Take just once, with no specified time

<schedule>

<repeat>

<count value="1" />

</repeat>

</schedule>

**Signature**

See also [Base Definition](file:///C:\temp\datatypes.html#Signature), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Signature), [Mappings](file:///C:\temp\datatypes-mappings.html#Signature), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Signature) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Signature).

**Example**

todo

<signature>

<!-- todo -->

</signature>

**Annotation**

See also [Base Definition](file:///C:\temp\datatypes.html#Annotation), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Annotation), [Mappings](file:///C:\temp\datatypes-mappings.html#Annotation), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Annotation) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Annotation).

**Example**

todo

<Annotation>

<!-- todo -->

</Annotation>

**datatypes-extras.html**

**Data Type Profiles and Extensions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page lists profiles and extensions defined in this specification itself for the data types. For background information, see [Profiling Resources](file:///C:\temp\profiling.html#resources) and [Extensibility](file:///C:\temp\extensibility.html). Additional profiles and resources may be found in published [Implementation Guides](http://fhir.org/guides/registry), or in the [Conformance resource registry](http://registry.fhir.org).

**Table of Contents**

|  |  |
| --- | --- |
| **Primitive Types** | **Complex Types** |

**Primitive Types**

See also [Base Definition](file:///C:\temp\datatypes.html#primitive), [Examples](file:///C:\temp\datatypes-examples.html#primitive), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#primitive), [Mappings](file:///C:\temp\datatypes-mappings.html#primitive) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#primitive).

|  |
| --- |
|  |

**Attachment**

See also [Base Definition](file:///C:\temp\datatypes.html#Attachment), [Examples](file:///C:\temp\datatypes-examples.html#Attachment), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Attachment), [Mappings](file:///C:\temp\datatypes-mappings.html#Attachment) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Attachment).

|  |
| --- |
|  |

**Identifier**

See also [Base Definition](file:///C:\temp\datatypes.html#identifier), [Examples](file:///C:\temp\datatypes-examples.html#identifier), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#identifier), [Mappings](file:///C:\temp\datatypes-mappings.html#identifier) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#identifier).

|  |
| --- |
|  |

**Coding**

See also [Base Definition](file:///C:\temp\datatypes.html#Coding), [Examples](file:///C:\temp\datatypes-examples.html#Coding), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Coding), [Mappings](file:///C:\temp\datatypes-mappings.html#Coding) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Coding).

|  |
| --- |
|  |

**CodeableConcept**

See also [Base Definition](file:///C:\temp\datatypes.html#CodeableConcept), [Examples](file:///C:\temp\datatypes-examples.html#CodeableConcept), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#CodeableConcept), [Mappings](file:///C:\temp\datatypes-mappings.html#CodeableConcept) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#CodeableConcept).

|  |
| --- |
|  |

**Quantity**

See also [Base Definition](file:///C:\temp\datatypes.html#Quantity), [Examples](file:///C:\temp\datatypes-examples.html#Quantity), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Quantity), [Mappings](file:///C:\temp\datatypes-mappings.html#Quantity) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Quantity).

|  |
| --- |
|  |

**Range**

See also [Base Definition](file:///C:\temp\datatypes.html#Range), [Examples](file:///C:\temp\datatypes-examples.html#Range), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Range), [Mappings](file:///C:\temp\datatypes-mappings.html#Range) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Range).

|  |
| --- |
|  |

**Ratio**

See also [Base Definition](file:///C:\temp\datatypes.html#Ratio), [Examples](file:///C:\temp\datatypes-examples.html#Ratio), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Ratio), [Mappings](file:///C:\temp\datatypes-mappings.html#Ratio) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Ratio).

|  |
| --- |
|  |

**Period**

See also [Base Definition](file:///C:\temp\datatypes.html#Period), [Examples](file:///C:\temp\datatypes-examples.html#Period), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Period), [Mappings](file:///C:\temp\datatypes-mappings.html#Period) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Period).

|  |
| --- |
|  |

**SampledData**

See also [Base Definition](file:///C:\temp\datatypes.html#SampledData), [Examples](file:///C:\temp\datatypes-examples.html#SampledData), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#SampledData), [Mappings](file:///C:\temp\datatypes-mappings.html#SampledData) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#SampledData).

|  |
| --- |
|  |

**HumanName**

See also [Base Definition](file:///C:\temp\datatypes.html#HumanName), [Examples](file:///C:\temp\datatypes-examples.html#HumanName), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#HumanName), [Mappings](file:///C:\temp\datatypes-mappings.html#HumanName) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#HumanName).

|  |
| --- |
|  |

**Address**

See also [Base Definition](file:///C:\temp\datatypes.html#Address), [Examples](file:///C:\temp\datatypes-examples.html#Address), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Address), [Mappings](file:///C:\temp\datatypes-mappings.html#Address) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Address).

|  |
| --- |
|  |

**ContactPoint**

See also [Base Definition](file:///C:\temp\datatypes.html#ContactPoint), [Examples](file:///C:\temp\datatypes-examples.html#ContactPoint), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#ContactPoint), [Mappings](file:///C:\temp\datatypes-mappings.html#ContactPoint) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#ContactPoint).

|  |
| --- |
|  |

**Timing**

See also [Base Definition](file:///C:\temp\datatypes.html#Timing), [Examples](file:///C:\temp\datatypes-examples.html#Timing), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Timing), [Mappings](file:///C:\temp\datatypes-mappings.html#Timing) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Timing).

|  |
| --- |
|  |

**Signature**

See also [Base Definition](file:///C:\temp\datatypes.html#Signature), [Examples](file:///C:\temp\datatypes-examples.html#Signature), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Signature), [Mappings](file:///C:\temp\datatypes-mappings.html#Signature) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Signature).

|  |
| --- |
|  |

**Annotation**

See also [Base Definition](file:///C:\temp\datatypes.html#Annotation), [Examples](file:///C:\temp\datatypes-examples.html#Annotation), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Annotation), [Mappings](file:///C:\temp\datatypes-mappings.html#Annotation) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Annotation).

|  |
| --- |
|  |

**datatypes-mappings.html**

**Data Type Mappings**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page provides mappings for the data types. There are mappings to [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185), [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186), and (where appropriate) vCard.

**Table of Contents**

|  |  |
| --- | --- |
| **Primitive Types** | **Complex Types** |

**Primitive Types**

See also [Base Definition](file:///C:\temp\datatypes.html#primitive), [Examples](file:///C:\temp\datatypes-examples.html#primitive), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#primitive), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#primitive) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#primitive).

**Attachment**

See also [Base Definition](file:///C:\temp\datatypes.html#Attachment), [Examples](file:///C:\temp\datatypes-examples.html#Attachment), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Attachment), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Attachment) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Attachment).

**Identifier**

See also [Base Definition](file:///C:\temp\datatypes.html#identifier), [Examples](file:///C:\temp\datatypes-examples.html#identifier), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#identifier), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#identifier) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#identifier).

**Coding**

See also [Base Definition](file:///C:\temp\datatypes.html#Coding), [Examples](file:///C:\temp\datatypes-examples.html#Coding), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Coding), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Coding) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Coding).

**CodeableConcept**

See also [Base Definition](file:///C:\temp\datatypes.html#CodeableConcept), [Examples](file:///C:\temp\datatypes-examples.html#CodeableConcept), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#CodeableConcept), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#CodeableConcept) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#CodeableConcept).

**Quantity**

See also [Base Definition](file:///C:\temp\datatypes.html#Quantity), [Examples](file:///C:\temp\datatypes-examples.html#Quantity), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Quantity), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Quantity) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Quantity).

**Range**

See also [Base Definition](file:///C:\temp\datatypes.html#Range), [Examples](file:///C:\temp\datatypes-examples.html#Range), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Range), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Range) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Range).

**Ratio**

See also [Base Definition](file:///C:\temp\datatypes.html#Ratio), [Examples](file:///C:\temp\datatypes-examples.html#Ratio), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Ratio), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Ratio) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Ratio).

**Period**

See also [Base Definition](file:///C:\temp\datatypes.html#Period), [Examples](file:///C:\temp\datatypes-examples.html#Period), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Period), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Period) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Period).

**SampledData**

See also [Base Definition](file:///C:\temp\datatypes.html#SampledData), [Examples](file:///C:\temp\datatypes-examples.html#SampledData), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#SampledData), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#SampledData) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#SampledData).

**HumanName**

See also [Base Definition](file:///C:\temp\datatypes.html#HumanName), [Examples](file:///C:\temp\datatypes-examples.html#HumanName), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#HumanName), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#HumanName) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#HumanName).

[**vCard**](http://tools.ietf.org/html/rfc6350)**Mappings**

* HumanName.text = vCard "FN" field
* HumanName.use = use of the vCard "TYPE" parameter
* HumanName.family, .given, .prefix, .suffix = parts of vCard "N" field. Note that there is no FHIR equivalent for the poorly defined "additional" name field. In FHIR, given names go in "middle" names
* The vCard nickname corresponds to a name with the use "nickname"

**Address**

See also [Base Definition](file:///C:\temp\datatypes.html#Address), [Examples](file:///C:\temp\datatypes-examples.html#Address), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Address), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Address) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Address).

**ContactPoint**

See also [Base Definition](file:///C:\temp\datatypes.html#ContactPoint), [Examples](file:///C:\temp\datatypes-examples.html#ContactPoint), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#ContactPoint), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#ContactPoint) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#ContactPoint).

**Timing**

See also [Base Definition](file:///C:\temp\datatypes.html#Timing), [Examples](file:///C:\temp\datatypes-examples.html#Timing), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Timing), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Timing) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Timing).

**Signature**

See also [Base Definition](file:///C:\temp\datatypes.html#Signature), [Examples](file:///C:\temp\datatypes-examples.html#Signature), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Signature), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Signature) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Signature).

**Annotation**

See also [Base Definition](file:///C:\temp\datatypes.html#Annotation), [Examples](file:///C:\temp\datatypes-examples.html#Annotation), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Annotation), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Annotation) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Annotation).

**datatypes-version-maps.html**

**Data Type Detailed Descriptions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page provides the detailed descriptions for the data types

**Table of Contents**

|  |  |
| --- | --- |
| **Primitive Types** | **Complex Types** |

**Primitive Types**

See also [Base Definition](file:///C:\temp\datatypes.html#primitive), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#primitive), [Examples](file:///C:\temp\datatypes-examples.html#primitive), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#primitive) and [Mappings](file:///C:\temp\datatypes-mappings.html#primitive).

**Attachment**

See also [Base Definition](file:///C:\temp\datatypes.html#Attachment), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Attachment), [Examples](file:///C:\temp\datatypes-examples.html#Attachment), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Attachment) and [Mappings](file:///C:\temp\datatypes-mappings.html#Attachment).

**Identifier**

See also [Base Definition](file:///C:\temp\datatypes.html#identifier), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#identifier), [Examples](file:///C:\temp\datatypes-examples.html#identifier), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#identifier) and [Mappings](file:///C:\temp\datatypes-mappings.html#identifier).

**Coding**

See also [Base Definition](file:///C:\temp\datatypes.html#Coding), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Coding), [Examples](file:///C:\temp\datatypes-examples.html#Coding), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Coding) and [Mappings](file:///C:\temp\datatypes-mappings.html#Coding).

**CodeableConcept**

See also [Base Definition](file:///C:\temp\datatypes.html#CodeableConcept), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#CodeableConcept), [Examples](file:///C:\temp\datatypes-examples.html#CodeableConcept), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#CodeableConcept) and [Mappings](file:///C:\temp\datatypes-mappings.html#CodeableConcept).

**Quantity**

See also [Base Definition](file:///C:\temp\datatypes.html#Quantity), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Quantity), [Examples](file:///C:\temp\datatypes-examples.html#Quantity), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Quantity) and [Mappings](file:///C:\temp\datatypes-mappings.html#Quantity).

**Range**

See also [Base Definition](file:///C:\temp\datatypes.html#Range), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Range), [Examples](file:///C:\temp\datatypes-examples.html#Range), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Range) and [Mappings](file:///C:\temp\datatypes-mappings.html#Range).

**Ratio**

See also [Base Definition](file:///C:\temp\datatypes.html#Ratio), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Ratio), [Examples](file:///C:\temp\datatypes-examples.html#Ratio), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Ratio) and [Mappings](file:///C:\temp\datatypes-mappings.html#Ratio).

**Period**

See also [Base Definition](file:///C:\temp\datatypes.html#Period), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Period), [Examples](file:///C:\temp\datatypes-examples.html#Period), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Period) and [Mappings](file:///C:\temp\datatypes-mappings.html#Period).

**SampledData**

See also [Base Definition](file:///C:\temp\datatypes.html#SampledData), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#SampledData), [Examples](file:///C:\temp\datatypes-examples.html#SampledData), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#SampledData) and [Mappings](file:///C:\temp\datatypes-mappings.html#SampledData).

**HumanName**

See also [Base Definition](file:///C:\temp\datatypes.html#HumanName), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#HumanName), [Examples](file:///C:\temp\datatypes-examples.html#HumanName), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#HumanName) and [Mappings](file:///C:\temp\datatypes-mappings.html#HumanName).

**Address**

See also [Base Definition](file:///C:\temp\datatypes.html#Address), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Address), [Examples](file:///C:\temp\datatypes-examples.html#Address), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Address) and [Mappings](file:///C:\temp\datatypes-mappings.html#Address).

**ContactPoint**

See also [Base Definition](file:///C:\temp\datatypes.html#ContactPoint), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#ContactPoint), [Examples](file:///C:\temp\datatypes-examples.html#ContactPoint), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#ContactPoint) and [Mappings](file:///C:\temp\datatypes-mappings.html#ContactPoint).

**Timing**

See also [Base Definition](file:///C:\temp\datatypes.html#Timing), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Timing), [Examples](file:///C:\temp\datatypes-examples.html#Timing), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Timing) and [Mappings](file:///C:\temp\datatypes-mappings.html#Timing).

**Signature**

See also [Base Definition](file:///C:\temp\datatypes.html#Signature), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Signature), [Examples](file:///C:\temp\datatypes-examples.html#Signature), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Signature) and [Mappings](file:///C:\temp\datatypes-mappings.html#Signature).

**Annotation**

See also [Base Definition](file:///C:\temp\datatypes.html#Annotation), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Annotation), [Examples](file:///C:\temp\datatypes-examples.html#Annotation), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Annotation) and [Mappings](file:///C:\temp\datatypes-mappings.html#Annotation).

**datatypes.html**

**Data Types**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 3 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The FHIR specification defines a set of data types that are used for the resource elements. There are four categories of data types:

1. Simple / primitive types, which are single elements with a primitive value ([below](file:///C:\temp\main-pages.html#primitive))
2. General purpose complex types, which are re-usable clusters of elements ([below](file:///C:\temp\main-pages.html#xomplex))
3. [Complex data types for metadata](file:///C:\temp\metadatatypes.html)
4. Special purpose data types: [Reference](file:///C:\temp\references.html#Reference), [Narrative](file:///C:\temp\narrative.html#Narrative), [Extension](file:///C:\temp\extensibility.html#Extension), [Meta](file:///C:\temp\resource.html#Meta), and [Dosage](file:///C:\temp\dosage.html#Dosage)

This page describes the general purpose data types.

**Table of Contents**

**Primitive Types**

**Complex Types**

Legend: White = abstract type. Light blue = concrete types. Light Pink = Profile on Type.

**Primitive Types**

The following table describes the primitive types that are used in this specification. Primitive types are those with only a value, and no additional elements as children (though, like all types, they have [extensions](file:///C:\temp\extensibility.html)). See also the [Examples](file:///C:\temp\datatypes-examples.html#primitives).

|  |  |  |  |
| --- | --- | --- | --- |
| **Primitive Types** | | | |
| **FHIR Name** | **Value Domain** | **XML Representation** | **JSON representation** |
| boolean | true | false | xs:boolean, except that **0 and 1 are not valid values** | JSON boolean (true or false) |
| integer | A signed 32-bit integer (for larger values, use decimal) | xs:int, except that **leading 0 digits are not allowed** | JSON number (with no decimal point) |
|  | Regex: [0]|[-+]?[1-9][0-9]\* | | |
| string | A sequence of Unicode characters | xs:string | JSON String |
|  | Note that strings SHALL NOT exceed 1MB in size. String should not contain Unicode character points below 32, except for u0009 (horizontal tab), u0010 (carriage return) and u0013 (line feed) | | |
|  | *Todo: is there any reason not to make this a SHALL?* | | |
| decimal | Rational numbers that have a decimal representation. See below about the precision of the number | xs:decimal, except that **decimals may not use exponents, and leading 0 digits are not allowed** | A JSON number, except that **exponents are not allowed** |
| uri | A Uniform Resource Identifier Reference ([RFC 3986](http://tools.ietf.org/html/rfc3986)). Note: URIs are case sensitive. For UUID (urn:uuid:53fefa32-fcbb-4ff8-8a92-55ee120877b7) use all lowercase | xs:anyURI | A JSON string - a URI |
|  | URIs can be absolute or relative, and may have an optional fragment identifier | | |
| base64Binary | A stream of bytes, base64 encoded ([RFC 4648](http://tools.ietf.org/html/rfc4648)) | xs:base64Binary | A JSON string - base64 content |
|  | *Todo: is it possible to impose an upper absolute limit on a base64Binary (for denial of service reasons, like on string)?* | | |
| instant | An instant in time - **known at least to the second and always includes a time zone**. Note: This is intended for precisely observed times (typically system logs etc.), and not human-reported times - for them, use date and dateTime. instant is a more constrained dateTime | xs:dateTime | A JSON string - an xs:dateTime |
|  | Note: This type is for system times, not human times (see date and dateTime below). | | |
| date | A date, or partial date (e.g. just year or year + month) as used in human communication. **There is no time zone**. Dates SHALL be valid dates | union of xs:date, xs:gYearMonth, xs:gYear | A JSON string - a union of xs:date, xs:gYearMonth, xs:gYear |
|  | Regex: -?[0-9]{4}(-(0[1-9]|1[0-2])(-(0[0-9]|[1-2][0-9]|3[0-1]))?)? | | |
| dateTime | A date, date-time or partial date (e.g. just year or year + month) as used in human communication. If hours and minutes are specified, a time zone SHALL be populated. Seconds must be provided due to schema type constraints but may be zero-filled and may be ignored. Dates SHALL be valid dates. **The time "24:00" is not allowed** | union of xs:dateTime, xs:date, xs:gYearMonth, xs:gYear | A JSON string - a union of xs:dateTime, xs:date, xs:gYearMonth, xs:gYear |
|  | Regex: -?[0-9]{4}(-(0[1-9]|1[0-2])(-(0[0-9]|[1-2][0-9]|3[0-1])(T([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9](\.[0-9]+)?(Z|(\+|-)((0[0-9]|1[0-3]):[0-5][0-9]|14:00)))?)?)? | | |
| time | A time during the day, with no date specified (can be converted to a [Duration](file:///C:\temp\main-pages.html#Duration) since midnight). Seconds must be provided due to schema type constraints but may be zero-filled and may be ignored. **The time "24:00" is not allowed, and neither is a time zone** | xs:time | A JSON string - an xs:time |
|  | Regex: ([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9](\.[0-9]+)? | | |
| code | Indicates that the value is taken from a set of controlled strings defined elsewhere (see [Using codes](file:///C:\temp\terminologies.html) for further discussion). Technically, a code is restricted to a string which has at least one character and no leading or trailing whitespace, and where there is no whitespace other than single spaces in the contents | xs:token | JSON string |
|  | Regex: [^\s]+([\s]?[^\s]+)\* | | |
| oid | An OID represented as a URI ([RFC 3001](http://www.ietf.org/rfc/rfc3001.txt)); e.g. urn:oid:1.2.3.4.5 | xs:anyURI | JSON string - uri |
|  | Regex: urn:oid:[0-2](\.[1-9]\d\*)+ | | |
| id | Any combination of upper or lower case ASCII letters ('A'..'Z', and 'a'..'z', numerals ('0'..'9'), '-' and '.', with a length limit of 64 characters. (This might be an integer, an un-prefixed OID, UUID or any other identifier pattern that meets these constraints.) | xs:string | JSON string |
|  | Regex: | | |
| markdown | A string that may contain markdown syntax for optional processing by a markdown presentation engine | xs:string | JSON string |
| unsignedInt | Any non-negative integer (e.g. >= 0) | xs:nonNegativeInteger | JSON number |
|  | Regex: [0]|([1-9][0-9]\*) | | |
| positiveInt | Any positive integer (e.g. >= 1) | xs:positiveInteger | JSON number |
|  | Regex: +?[1-9][0-9]\* | | |

Notes:

* For all the types, the XML, JSON and Turtle representations of the primitive values are the same except for different escaping in XML and JSON
* Boolean values can also be represented using coded values (such as [HL7 v2 Table 0136](file:///C:\temp\v2\0136\index.html)). See [Observation](file:///C:\temp\observation.html#valuex) for one such use
* The regexes may allow a broader set of values than are actually valid (e.g. leap years) so additional validation is needed
* The regexes should be qualified with start of string and end of string anchors based on the regex implementation used (e.g. caret '^' and dollar-sign '$' for JavaScript, POSIX, XML and XPath; '\A' and '\Z' for .NET, Java, Python and others; please verify these definitions with the regex implementation used).
* The data types code, string, and uri can be [bound to a value set](file:///C:\temp\terminologies.html#strings)
* The precision of the decimal value has signficance:
  + e.g. 0.010 is regarded as different to 0.01, and the original precision should be preserved
  + Implementations SHALL handle decimal values in ways that preserve and respect the precision of the value as represented for presentation purposes
  + Implementations are not required to perform calculations with these numbers differently, though they may choose to do so (i.e. preserve significance)
  + See implementation comments for [XML](file:///C:\temp\xml.html#schema-gen), [JSON](file:///C:\temp\json.html#decimal) and [RDF](file:///C:\temp\rdf.html#decimal)
  + In object code, implementations that might meet this constraint are GMP implementations or equivalents to Java BigDecimal that implement arbitrary precision, or a combination of a (64 bit) floating point value with a precision field
  + Note that there is no absolute limit to the magnitude of the value, though large and/or highly precise values are extremely rare in medicine. One element where highly precise decimals may be encountered is the [Location](file:///C:\temp\location.html) coordinates
* About the id datatype:
  + Ids are case sensitive. UUIDs SHALL be sent using lowercase letters
  + The ID type includes identifiers consistent with [ISO 18232](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=38610), but also includes other identifier formats as well, and is not case insensitive like ISO 18232.
  + In a typical FHIR URL, like http://example.com/fhir/Patient/1234, the last part "1234" (highlighted in red) is the part that is an id datatype
  + A full UUID is a uri, not an id. UUIDs in URIs SHALL also be represented in lowercase (urn:uuid:59bf0ef4-e89c-4628-9b51-12ae3fdbe22b)
* About the markdown datatype:
  + Systems are not required to have markdown support, and there is considerable variation in markdown syntax, so the content of a string should be readable without markdown processing, per markdown philosophy
  + The preferred markdown syntax is described at <http://daringfireball.net/projects/markdown/syntax> (see tests here: <http://daringfireball.net/projects/downloads/MarkdownTest_1.0.zip>)
  + Markdown content should not contain Unicode character points below 32, except for u0009 (horizontal tab), u0010 (carriage return) and u0013 (line feed). *Todo: is there any reason not to make this a SHALL?*

**STU Note:** [CommonMark](http://commonmark.org/) is a new arrival but may gain sufficient impetus to be preferred in a future version.

Feedback is welcome [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**Representations in XML, JSON, and Turtle**

All elements using these primitive types may have one or more of a value as described above, an internal identity (e.g. xml:id), and extensions. For an example, take an element of name "count" and type "integer".

**XML**

The value is represented in XML as an attribute named "value":

<count value="2"/>

The full representation, with id, extensions and value:

<count id="a1" value="2">

<extension url="...">

<valueXX.../>

</extension>

</count>

**JSON**

In JSON, for convenience, the value is represented as the property itself:

"count" : 2

The full representation, with id, extensions and value, showing the id and extensions in the sibling property:

"count" : 2

"count\_" : {

"id" : "a1",

"extension" : [{

"url" : "...",

"valueXXX" : "...."

}]

}

**RDF**

The value is represented in RDF as an relationship with the URI "http://h;7.org/fhir/value". Using the normal prefix, this becomes:

fhir:Type.count [ fhir:value "2"^^xsd:integer ]

For the types date and DateTime, the type must be specified explicitly. For all other types, it is optional. The full representation, with id, extensions and value:

fhir:Type.count [

Element.id "a1";

fhir:value "2"^^xsd:integer;

Element.extension [

fhir:Extension.url "..";

fhir:Extension.valueXX...

]

]

For additional details, see the [XML](file:///C:\temp\xml.html), [JSON](file:///C:\temp\json.html) and [Turtle](file:///C:\temp\rdf.html) format definitions. When the value is missing, and there are no extensions, the element is not represented at all. This means that in xml, attributes are never present with a length of 0 (value=""), and properties are never a 0 length string or null in JSON ("name" : "" is not valid). (note: there is one specific [use of the null](file:///C:\temp\json.html#null) in the JSON representation).

According to XML schema, leading and trailing whitespace in the value attribute is ignored for the types boolean, integer, decimal, base64Binary, instant, uri, date, dateTime, oid, and uri. Note that this means that the schema aware XML libraries give different attribute values to non schema aware libraries when reading the XML instances. For this reason, the value attribute for these types SHOULD not have leading and trailing spaces. String values should only have leading and trailing spaces if they are part of the content of the value. In JSON and Turtle whitespace in string values is always significant. Primitive types other than string SHALL NOT have leading or trailing whitespace.

**Complex Types**

In XML, these types are represented as XML Elements with child elements with the name of the defined elements of the type. The name of the element is defined where the type is used. In JSON, the data type is represented by an object with properties named the same as the XML elements. Since the JSON representation is almost exactly the same, only the first [example](file:///C:\temp\datatypes-examples.html#Attachment) has an additional explicit JSON representation.

Complex data types may be "profiled". A [Structure Definition](file:///C:\temp\structuredefinition.html) or type "constraint" makes a set of rules about which elements SHALL have values and what the possible values are.

**UML Diagrams of the Data types**

**Attachment**

See also [Examples](file:///C:\temp\datatypes-examples.html#Attachment), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Attachment), [Mappings](file:///C:\temp\datatypes-mappings.html#Attachment), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Attachment) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Attachment).

This type is for containing or referencing attachments - additional data content defined in other formats. The most common use of this type is to include images or reports in some report format such as PDF. However it can be used for any data that has a MIME type.

The actual content of an Attachment can be conveyed directly using the data element or a URL reference can be provided. If both are provided, the reference SHALL point to the same content as found in the data. The reference can never be reused to point to some different data (i.e. the reference is version specific). The URL reference SHALL point to a location that resolves to actual data; some URIs such as cid: meet this requirement. If the URL is a relative reference, it is interpreted in the same way as a [resource reference](file:///C:\temp\references.html#references).

The contentType element SHALL always be populated when an Attachment contains data, and MAY be populated when there is a url. It can include charset information and other mime type extensions as appropriate. If there is no character set in the contentType then the correct course of action is undefined, though some media types may define a default character set and/or the correct character set may be able to be determined by inspection of the content.

The hash is included so that applications can verify that the content returned by the URL has not changed. The hash and size relate to the data before it is represented in base64 form.

In many cases where Attachment is used, the cardinality is >1. A valid use of repeats is to convey the same content in different mime types and languages. Guidance on the meaning of repeating elements SHALL be provided in the definition of the repeating resource element or extension that references this type. The language element describes the language of the attachment using the [codes defined in BCP 47](http://tools.ietf.org/html/bcp47).

**Constraints**

If neither data nor a URL is provided, the value should be understood as an assertion that no content for the specified mimeType and/or language is available for the combination of language and contentType.

The context of use may frequently make rules about the kind of attachment (and therefore, the kind of mime types) that can be used.

Attachment is used in the following places:

**Coding**

See also [Examples](file:///C:\temp\datatypes-examples.html#Coding), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Coding), [Mappings](file:///C:\temp\datatypes-mappings.html#Coding), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Coding) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Coding).

A Coding is a representation of a defined concept using a symbol from a defined "code system" - see [Using Codes in resources](file:///C:\temp\terminologies.html) for more details.

The meaning of the Coding is defined by the code. The system provides the source of the definition of the code, along with an optional version reference. The display is a human display for the text defined by the system - it is not intended for computation.

The system is a URI that references the code system that defines the code. Choosing the correct system is important; for more information about the code system URI, read [Managing Terminology System URIs](file:///C:\temp\terminologies.html#system). The system URI SHALL NOT contain a reference to a value set (e.g. ValueSet.url). If the code is taken from a CodeSystem resource, CodeSystem.url is the correct value for the system uri. Resolvable URLs are generally preferred by implementers over non-resolvable URNs, particularly opaque URNs such as OIDs (urn:oid:) or UUIDs (urn:uuid:).

A code system version may also be supplied. If the meaning of codes within the code system is consistent across releases, this is not required. The version SHOULD be exchanged when the system does not maintain consistent definitions across versions. Note that the following systems SHOULD always have a version specified:

* National releases of SNOMED CT (consistency of definitions varies amongst jurisdictions, and some jurisdictions may make their own rules on this)
* Various versions of ICD (note: the major releases are labeled as different code systems altogether, but there is variation within versions)

More generally, any classification (e.g. a code system that includes concepts with relative definitions such as "not otherwise coded" will require a version. See the [discussion of code system versions in the Code System resource](file:///C:\temp\codesystem.html#versioning) for further discussion on versioning.

If present, the code SHALL be a syntactically correct symbol as defined by the system. In some code systems such as SNOMED CT, the symbol may be an expression composed of other predefined symbol (e.g. post-coordination). Note that codes are case sensitive unless specified otherwise by the code system. The display is a text representation of the code defined by the system and is used to display the meaning of the code by an application that is not aware of the system.

Where the code system defines multiple possible display strings, one of these SHALL be used in display. If one is labeled as preferred, it SHOULD be used. If the code system does not define a text representation (e.g. SNOMED CT Expressions) then display cannot be populated, and the meaning of the code won't be accessible to systems that don't understand the code expression.

In some cases, the system may not be known - only the code is known. In this case, no useful processing of the code may be performed unless the system can be safely inferred by the context. This practice should be avoided where possible, as information sharing in a wider context is very likely to arise eventually, and codes cannot be used in the absence of a known system.

If the system is present, and there is no code, then this is understood to mean that there is no suitable code in the system in which to represent the code.

If two codings have the same system, version and code then they have the same meaning. If the version information is missing, or the system, version or the code elements differ, then how the codes are related can only be determined by consulting the definitions of the system(s) and any [mappings](file:///C:\temp\conceptmap.html) available.

A coding may be marked as a "userSelected" if a user selected the particular coded value in a user interface (e.g. the user selects an item in a pick-list). If a user selected coding exists, it is the preferred choice for performing translations etc.

**Constraints**

The context of use (as defined in the resource or applicable profile) usually makes rules about what codes and systems are allowed or required in a particular context by [binding](file:///C:\temp\terminologies.html) the element to a value set.

Coding is used in the following places:

**Design Note**: This specification defines two types for representing coded values:

* **Coding**: a simple direct reference to a code defined by a code system
* **CodeableConcept**: a text description and/or a list of Codings (i.e. a list of references to codes defined by code systems)

The Coding data type corresponds to the simple case of selecting a single code from a code list. However this type is rarely used in the FHIR specifications; long experience with exchanging coded values in HL7 shows that in the general case, systems need to able to exchange multiple translation codes, and/or an original text.

The Coding data type is used directly when there is certainty that the value must be selected directly from one of the available codes, and the list of possible codes is agreed to by all participants. This is not usually the case in the context of FHIR - general interoperability - so Coding is mostly used in extensions, which are usually intended to be defined for a well-controlled context of use.

**CodeableConcept**

See also [Examples](file:///C:\temp\datatypes-examples.html#CodeableConcept), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#CodeableConcept), [Mappings](file:///C:\temp\datatypes-mappings.html#CodeableConcept), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#CodeableConcept) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#CodeableConcept).

A CodeableConcept represents a value that is usually supplied by providing a reference to one or more terminologies or ontologies, but may also be defined by the provision of text. This is a common pattern in healthcare data.

Each coding is a representation of the concept as described above. The concept may be coded multiple times in different code systems (or even multiple times in the same code systems, where multiple forms are possible, such as with SNOMED CT). The different codings may have slightly different granularity due to the differences in the definitions of the underlying codes. There is no meaning associated with the ordering of coding within a CodeableConcept. A typical use of CodeableConcept is to send the local code that the concept was coded with, and also one or more translations to publicly defined code systems such as LOINC or SNOMED CT. Sending local codes is useful and important for the purposes of debugging and integrity auditing.

Whether or not coding elements are present, the text is the representation of the concept as entered or chosen by the user, and which most closely represents the intended meaning of the user or concept. Very often the text is the same as a display of one of the codings. One of the codings may be flagged as the userSelected - the code or concept that the user actually selected directly. When none of the coding elements is marked as userSelected, the text (if present) is the preferred source of meaning.

**Constraints**

The context of use usually makes rules about what codes and systems are allowed or required in a particular context by [binding](file:///C:\temp\terminologies.html) the element to a value set.

CodeableConcept is used in the following places:

**Quantity**

See also [Examples](file:///C:\temp\datatypes-examples.html#Quantity), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Quantity) and [Mappings](file:///C:\temp\datatypes-mappings.html#Quantity), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Quantity) and and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Quantity).

A measured amount (or an amount that can potentially be measured).

The value contains the numerical value of the quantity, including an implicit precision. If no comparator is specified, the value is a point value (i.e. '='). The comparator element can never be ignored.

The unit element contains a displayable unit that defines what is measured. The unit may additionally be coded in some formal way using the code and the system (see [Coding](file:///C:\temp\main-pages.html#Coding) for further information about how to use the system element).

If the unit is able to be coded in UCUM and a code is provided, it SHOULD be a UCUM code. If a UCUM unit is provided in the code then a canonical value can be generated for purposes of comparison between quantities. Note that the unit element will often contain text that is actually a valid UCUM unit, but it cannot be assumed that the unit actually contains a valid UCUM unit.

**Constraints**

The context of use may frequently define what kind of measured quantity this is and therefore what kind of unit can be used. The context of use may additionally require a code from a particular system. The context of use may also restrict the values for the value or range.

Quantity is used in the following places:

**Defined Variations on Quantity**

There are several additional data types that are specializations of Quantity that only introduce new restrictions on the existing elements defined as part of the Quantity data type:

|  |  |  |
| --- | --- | --- |
| **Type Name** | **Rules** | **Formal Definitions** |
| Age |  | [XML](file:///C:\temp\age.profile.xml.html), [JSON](file:///C:\temp\age.profile.json.html) |
|  | Usage: | |
| Count |  | [XML](file:///C:\temp\count.profile.xml.html), [JSON](file:///C:\temp\count.profile.json.html) |
|  | Usage: | |
| Distance |  | [XML](file:///C:\temp\distance.profile.xml.html), [JSON](file:///C:\temp\distance.profile.json.html) |
|  | Usage: | |
| Duration |  | [XML](file:///C:\temp\duration.profile.xml.html), [JSON](file:///C:\temp\duration.profile.json.html) |
|  | Usage: | |
| Money |  | [XML](file:///C:\temp\money.profile.xml.html), [JSON](file:///C:\temp\money.profile.json.html) |
|  | Usage: | |
| In addition to the specializations, there is one profile on Quantity used in several resources: | | |
| **Profile Name** | **Rules** | **Formal Definitions** |
| Simple Quantity |  | [XML](file:///C:\temp\simplequantity.profile.xml.html), [JSON](file:///C:\temp\simplequantity.profile.json.html) |
|  | Usage: | |

Note that the profile is different from the other specializations because it is not a type, just rules applied where the Quantity type is used.

**Range**

See also [Examples](file:///C:\temp\datatypes-examples.html#Range), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Range), [Mappings](file:///C:\temp\datatypes-mappings.html#Range), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Range) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Range).

A set of ordered Quantity values defined by a low and high limit.

A Range specifies a set of possible values; usually, one value from the range applies (e.g. "give the patient between 2 and 4 tablets"). Ranges are typically used in instructions.

The unit and code/system elements of the low or high elements SHALL match. If the low or high elements are missing, the meaning is that the low or high boundaries are not known and therefore neither is the complete range.

The comparator flag on the low or high elements cannot be present. Note that the Range type should not be used to represent out of range measurements: A quantity type with the comparator element should be used instead.

The low and the high values are inclusive, and are assumed to have arbitrarily high precision; e.g. the range 1.5 to 2.5 includes 1.50, and 2.50 but not 1.49 or 2.51.

**Constraints**

Range is used in the following places:

**Ratio**

See also [Examples](file:///C:\temp\datatypes-examples.html#Ratio), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Ratio), [Mappings](file:///C:\temp\datatypes-mappings.html#Ratio), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Ratio) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Ratio).

A relationship between two Quantity values expressed as a numerator and a denominator.

Common factors in the numerator and denominator are not automatically cancelled out. The Ratio data type is used for titers (e.g. "1:128") and other quantities produced by laboratories that truly represent ratios. Ratios are not simply "structured numbers" - for example blood pressure measurements (e.g. "120/60") are not ratios. In addition, ratios are used where common factors in the numerator and denominator do not cancel out. The most common example of this is where the ratio represents a unit cost, and the numerator is a currency (e.g. 50/$10).

A proper ratio has both a numerator and a denominator; however these are not mandatory in order to allow an invalid ratio with an extension with further information.

**Constraints**

The context of use may require particular types of Quantity for the numerator or denominator.

Ratio is used in the following places:

**Period**

See also [Examples](file:///C:\temp\datatypes-examples.html#Period), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Period), [Mappings](file:///C:\temp\datatypes-mappings.html#Period), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Period) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Period).

A time period defined by a start and end date/time.

A period specifies a range of times. The context of use will specify whether the entire range applies (e.g. "the patient was an inpatient of the hospital for this time range") or one value from the period applies (e.g. "give to the patient between 2 and 4 pm on 24-Jun 2013").

If the start element is missing, the start of the period is not known. If the end element is missing, it means that the period is ongoing, or the start may be in the past, and the end date in the future, which means that period is expected/planned to end at the specified time

The end value includes any matching date/time. For example, the period 2011-05-23 to 2011-05-27 includes all the times from the start of the 23rd May through to the end of the 27th of May.

Period is used in the following places:

**SampledData**

See also [Examples](file:///C:\temp\datatypes-examples.html#SampledData), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#SampledData), [Mappings](file:///C:\temp\datatypes-mappings.html#SampledData), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#SampledData) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#SampledData).

Data that comes from a series of measurements taken by a device, with upper and lower limits. There may be more than one dimension in the data.

A SampledData provides a concise way to handle the data produced by devices that sample a physical particular state at a high frequency. A typical use for this is for the output of an ECG or EKG device. It has a series of raw decimal values (which are mostly simple integers), along with adjustments for scale and factor. These are interpreted such that

original measured value[i] = SampledData.data[i] \* SampledData.scaleFactor + SampledData.origin.value

The digits are a set of decimal values separated by a single space (Unicode character u20). In addition to decimal values, the special values "E" (error), "L" (below detection limit) and "U" (above detection limit) can also be used. If there is more than one dimension, the different dimensions are interlaced - all the data points for a particular time are represented together. The default value for factor is 1.

SampledData is used in the following places:

**Identifier**

See also [Examples](file:///C:\temp\datatypes-examples.html#Identifier), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Identifier), [Mappings](file:///C:\temp\datatypes-mappings.html#Identifier), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Identifier) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Identifier).

A numeric or alphanumeric string that is associated with a single object or entity within a given system. Typically, identifiers are used to connect content in resources to external content available in other frameworks or protocols. Identifiers are associated with objects, and may be changed or retired due to human or system process and errors.

The value SHALL be unique within the defined system and have a consistent meaning wherever it appears. Both system and value are always case sensitive.

The system is a URI that defines a set of identifiers (i.e. how the value is made unique). It might be a specific application or a recognized standard/specification for a set or identifiers or a way of making identifiers unique. FHIR defines [some useful or important system URIs directly](file:///C:\temp\identifier-registry.html). Here are some example identifier namespaces:

* http://hl7.org/fhir/sid/us-ssn identifiers US SSN values
* http://ns.electronichealth.net.au/id/hi/ihi/1.0 for Australian IHI numbers
* urn:ietf:rfc:3986 for when the value of the identifier is itself a globally unique URI

If the system is a URL, it SHOULD resolve. Resolution might be to a web page that describes the identifier system and/or supports look-up of identifiers. Alternatively it could be to a NamingSystem resource instance. Resolvable URLs are generally preferred by implementers over non-resolvable URNs, particularly opaque URNs such as OIDs (urn:oid:) or UUIDs (urn:uuid:). If used, OIDs and UUIDs may be registered in the [HL7 OID registry](http://hl7.org/oid) and SHOULD be registered if the content is shared or exchanged across institutional boundaries.

It is up to the implementer organization to determine an appropriate URL or URN structure that will avoid collisions and to manage that space (and the resolvability of URLs) over time.

Note that the scope of a given identifier system may extend beyond identifiers that might be captured by a single resource. For example, some systems might draw all "order" identifiers from a single namespace, though some might be used on [MedicationRequest](file:///C:\temp\medicationrequest.html) while others would appear on [ProcedureRequest](file:///C:\temp\procedurerequest.html).

If the identifier value itself is naturally a globally unique URI (e.g. an OID, a UUID, or a URI with no trailing local part), then the system SHALL be "urn:ietf:rfc:3986", and the URI is in the value (OIDs and UUIDs using urn:oid: and urn:uuid: - see [examples](file:///C:\temp\datatypes-examples.html#Identifier)).

In some cases, the system may not be known - only the value is known (e.g. a simple device that scans a barcode), or the system is known implicitly (simple exchange in a limited context, often driven by barcode readers). In this case, no useful matching may be performed using the value unless the system can be safely inferred by the context. Applications should provide a system whereever possible, as information sharing in a wider context is very likely to arise eventually, and values without a system are inherently limited in use.

In addition to the system (which provides a uniqueness scope) and the value, identifiers may also have a type, which may be useful when a system encounters identifiers with unknown system values. Note, however, that the type of an identifier is not a well controlled vocabulary with wide variations in practice. The type deals only with general categories of identifiers and SHOULD not be used for codes that correspond 1..1 with the Identifier.system. Some identifiers may fall into multiple categories due to variations in common usage.

The assigner is used to indicate what registry/state/facility/etc. assigned the identifier. As a [Reference](file:///C:\temp\references.html), the assigner can include just a text description in the display.

Identifier is used in the following places:

**HumanName**

See also [Examples](file:///C:\temp\datatypes-examples.html#HumanName), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#HumanName), [Mappings](file:///C:\temp\datatypes-mappings.html#HumanName), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#HumanName) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#HumanName).

A name of a human with text, parts and usage information.

Names may be changed or repudiated. People may have different names in different contexts. Names may be divided into parts of different type that have variable significance depending on context, though the division into parts is not always significant. With personal names, the different parts may or may not be imbued with some implicit meaning; various cultures associate different importance with the name parts and the degree to which systems SHALL care about name parts around the world varies widely.

This table summarizes where common parts of a person's name are found.

|  |  |  |
| --- | --- | --- |
| **Name** | **Example** | **Destination / Comments** |
| Surname | Smith | Family Name |
| First name | John | Given Name |
| Title | Mr | Prefix |
| Middle Name | Samuel | Subsequent Given Names |
| Patronymic | bin Osman | Family Name |
| Multiple family names | CarreÃƒÂ±o QuiÃƒÂ±ones | Family Name. See note below about decomposition of family name |
| Initials | Q. | Given Name as initial ("." recommended) |
| Nick Name | Jock | Given name, with Use = common |
| Qualifications | PhD | Suffix |
| Honorifics | Senior | Suffix |
| Voorvoegsel / Nobility | van Beethoven | Family Name. See note below about decomposition of family name |

For further information, including all [W3C International Examples](http://www.w3.org/International/questions/qa-personal-names), consult the [examples](file:///C:\temp\datatypes-examples.html#HumanName). **Note: Implementers should read the name examples for a full understanding of how name works.**

The multiple given parts and family name combine to form a single name. Where a person has alternate names that may be used in place of each other (e.g. Nicknames, Aliases), these are different instances of HumanName.

The text element specifies the entire name as it should be represented. This may be provided instead of or as well as specific parts, and can be built from the parts, though the correct order of assembly is culture dependent: the order of the parts within a given part type has significance and SHALL be observed. The appropriate order between family name and given names depends on culture and context of use. Note that there is an [extension](file:///C:\temp\extension-humanname-assembly-order.html) for the few times name assembly order is not fixed by the culture.

The given name parts may contain whitespace, though generally don't. Initials may be used in place of the full name if that is all that is recorded. Systems that operate across cultures should generally rely on the text form for presentation, and use the parts for index/search functionality. For this reasons, applications SHOULD populate the text element for future robustness.

In some cultures (e.g. German, Dutch, Spanish, Portuguese), family names are complex and composed of various parts that may need to be managed separately, e.g. they have differing significance for searching. In these cases, the full family name is populated in family, and a decomposition of the name can be provided using the family extensions [own-name](file:///C:\temp\extension-humanname-own-name.html), [own-prefix](file:///C:\temp\extension-humanname-own-prefix.html), [partner-name](file:///C:\temp\extension-humanname-partner-name.html), [partner-prefix](file:///C:\temp\extension-humanname-partner-prefix.html), [fathers-family](file:///C:\temp\extension-humanname-fathers-family.html) and [mothers-family](file:///C:\temp\extension-humanname-mothers-family.html).

For robust search, servers should search the parts of a family name independently. E.g. Searching either Carreno or Quinones should match a family name of "CarreÃƒÂ±o QuiÃƒÂ±ones". HL7 affiliates may make more specific recommendations about how search should work in their specific culture.

HumanName is used in the following places:

**Address**

See also [Examples](file:///C:\temp\datatypes-examples.html#Address), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Address), [Mappings](file:///C:\temp\datatypes-mappings.html#Address), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Address) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Address).

An address expressed using postal conventions (as opposed to GPS or other location definition formats). This data type may be used to convey addresses for use in delivering mail as well as for visiting locations and which might not be valid for mail delivery. There are a variety of postal address formats defined around the world.

The text element specifies the entire address as it should be represented. This may be provided instead of or as well as the specific parts. Applications updating an address SHALL ensure either that the text and the parts are in agreement, or that only one of the two is present.

**Constraints**

Address is used in the following places:

**ContactPoint**

See also [Examples](file:///C:\temp\datatypes-examples.html#ContactPoint), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#ContactPoint), [Mappings](file:///C:\temp\datatypes-mappings.html#ContactPoint), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#ContactPoint) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#ContactPoint).

Details for all kinds of technology-mediated contact points for a person or organization, including telephone, email, etc.

If capturing a phone, fax or similar contact point, the value should be a properly formatted telephone number according to [ITU-T E.123](http://www.itu.int/rec/T-REC-E.123-200102-I/e). However, this is frequently not possible due to legacy data and/or clerical practices when recording contact details. For this reason, phone, fax, page and email addresses are not handled as formal URLs. For other kinds of contacts, the system is "other" and the value SHOULD be a URL so that its use can be determined automatically. Typical URL schemes used in the value are http(s): for web addresses, and URL schemes for various kinds of messaging systems. If the value is not a URL, then human interpretation will be required.

The rank element can be used to specify a preference for the order in which a set of contacts is used. Contacts are ranked with lower values coming before higher values. Note that rank does not necessarily follow the order in which the contacts are represented in the instance.

**Constraints**

ContactPoint is used in the following places:

**Timing**

See also [Examples](file:///C:\temp\datatypes-examples.html#Timing), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Timing), [Mappings](file:///C:\temp\datatypes-mappings.html#Timing), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Timing) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Timing).

Specifies an event that may occur multiple times. Timing schedules are used for specifying when events are expected or requested to occur, and may also be used to represent a schedule that was followed for past events. A Timing schedule can be a list of events and/or criteria for when the event happens, which can be expressed in a structured form and/or as a code. When both event and a repeating specification are provided, the list of events should be understood as an interpretation of the information in the repeat structure.

If the timing schedule has repeating criteria, the repeat can occur a given number of times per the specified duration or in relation to some repeating real world event. If no end condition is specified, the schedule will terminate on some criteria that are expressed elsewhere.

This table summarizes some common uses of the Timing Data Type criteria.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **description** | **duration** | **durationUnit** | **frequency** | **frequencyMax** | **period** | **periodUnit** | **periodMax** | **Day of Week** | **Time Of Day** | **when** | **offset** | **bounds[x]** | **count** |
| Every 8 hours |  |  | 1 |  | 8 | h |  |  |  |  |  |  |  |
| Every 7 days |  |  | 1 |  | 7 | d |  |  |  |  |  |  |  |
| 3 times a day |  |  | 3 |  | 1 | d |  |  |  |  |  |  |  |
| 3-4 times a day |  |  | 3 | 4 | 1 | d |  |  |  |  |  |  |  |
| Every 4-6 hours |  |  | 1 |  | 4 | h | 6 |  |  |  |  |  |  |
| Every 21 days for 1 hour | 1 | hr | 1 |  | 21 | d |  |  |  |  |  |  |  |
| Three times a week for Ã‚Â½ hour | 0.5 | hr | 3 |  | 1 | wk |  |  |  |  |  |  |  |
| With breakfast |  |  |  |  |  |  |  |  |  | CM |  |  |  |
| For 5 minutes, 10 minutes before meals | 5 | min |  |  |  |  |  |  |  | AC | 10 |  |  |
| 1 tablet 3 times daily, 30 minutes before meals |  |  | 3 |  | 1 | d |  |  |  | AC | 30 |  |  |
| BID, 30 mins before meal, for next 10 days |  |  | 2 |  | 1 | d |  |  |  | AC | 30 | Duration = 10 days |  |
| TID, for 14 days |  |  | 3 |  | 1 | d |  |  |  |  |  | Duration = 14 days |  |
| BID, start on 7/1/2015 at 1:00 PM |  |  | 2 |  | 1 | d |  |  |  |  |  | Period.start = 2015-07-01T13:00:00 |  |
| Mon, Wed, Fri Morning |  |  | 1 |  | 1 | d |  | mon | wed | fri |  | MORN |  |  |  |
| Every day at 10am |  |  | 1 |  | 1 | d |  |  | 10:00 |  |  |  |  |
| Take once, at any time |  |  |  |  |  |  |  |  |  |  |  |  | 1 |

Many systems avoid the complexity of the Timing structure by using a text field for these things (e.g. "Dosage Text"). Those systems do not use the Timing data type. Other systems use a set of 'common' codes - including, but usually not limited to, widely understood acronyms such as "BID". If a code is provided, the code is understood to be a complete statement of whatever is specified in the structured timing data (except for Timing.repeat.bounds, which applies to the code), and either the code or the data may be used to interpret the Timing. A structured timing specification SHOULD be provided whenever possible, unless the code is BID, TID, QID, AM or PM, which have a ubiquitous meaning.

This table shows the relationship between the [codes provided as part of the base specification](file:///C:\temp\valueset-timing-abbreviation.html), and the structured data portions of the Timing type:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **description** | **duration** | **durationUnit** | **frequency** | **frequencyMax** | **period** | **periodUnit** | **periodMax** | **when** | **bounds[x]** |
| QOD |  |  | 1 |  | 2 | d |  |  |  |
| QD |  |  | 1 |  | 1 | d |  |  |  |
| BID |  |  | 2 |  | 1 | d |  |  |  |
| TID |  |  | 3 |  | 1 | d |  |  |  |
| QID |  |  | 4 |  | 1 | d |  |  |  |
| Q4H |  |  | 1 |  | 4 | h |  |  |  |
| Q6H |  |  | 1 |  | 6 | h |  |  |  |
| AM |  |  | 1 |  | 1 | d |  | MORN |  |
| PM |  |  | 1 |  | 1 | d |  | AFT or EVE |  |

These codes SHALL be understood as having the formal meanings documented in this table. Note that BID etc are defined as 'at institutionally specified times'. For example, an institution may choose that BID is "always at 7am and 6pm". If it is inappropriate for this choice to be made, the code BID should not be used. Instead, a distinct organization-specific code should be used in place of the HL7-defined BID code and/or the a structured representation should be used (in this case, timeOfDay).

**Constraints**

Note that these constraints still allow for non-sensical timing specifications such as "Once per day at 2:00 and 4:00" or "every 3 days on Friday". Implementers must take care to ensure that their configuration/data collection designs do not lead to these non-interpretable timing specifications. The elements dayOfWeek, timeOfDay, and when are particularly likely to be at issue here.

Timing is used in the following places:

**Signature**

See also [Examples](file:///C:\temp\datatypes-examples.html#Signature), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Signature), [Mappings](file:///C:\temp\datatypes-mappings.html#Signature), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Signature) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Signature).

A Signature holds an electronic representation of a signature and its supporting context in a FHIR accessible form. The signature may either be a cryptographic type (XML DigSig or a JWT), which is able to provide non-repudiation proof, or it may be a graphical image that represents a signature or a signature process.

**Constraints**

Note: One consequence of signing the document is that URLs, identifiers and internal references are frozen and cannot be changed. This might be a desired feature, but it may also cripple interoperability between closed ecosystems where [re-identification](file:///C:\temp\managing.html) frequently occurs. For this reason, it is recommended that systems consider carefully the impact of any signature processes. The impact of signatures on [Document bundles](file:///C:\temp\documents.html) and their related processes is the most well understood use of digital signatures.

**XML Signature rules**

When the signature is an XML Digital Signature (contentType = application/signature+xml), the following rules apply:

* The Signature.blob is base64 encoded XML-Signature
* The XML-Signature is a [Detached](http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/#def-SignatureDetached) Signature (where the content that is signed is separate from the signature itself)
* The Signature SHOULD conform to XAdES-X-L for support of Long Term signatures. The XAdES-X-L specification adds the timestamp of the signing, inclusion of the signing certificate, and statement of revocation
* When FHIR Resources are signed, the signature is accross the [Canonical XML form](file:///C:\temp\xml.html#canonical) of the resource(s)
* The Signature SHOULD use the hashing algorithm sha256. Signature validation policy will apply to the signature and determine acceptability
* The Signature SHALL include a "CommitmentTypeIndication" element for the Purpose(s) of Signature. The Purpose can be the action being attested to, or the role associated with the signature. The value shall come from ASTM E1762-95(2013). The Signature.type shall contain the same values as committmentTypeIndication

There are three levels of signature verification:

1. verifying that the Digital Signature block itself has integrity through verifying the signature across the XML-Signature,
2. confirming that the signer was authentic, not revoked, and appropriate to the signature purpose,
3. confirming that the signed content of interest is unmodified using the hash algorithm.

Deviations from these guidelines would need to be expressed in site policy, and would be enumerated in the XML-Signature block. For example some environments may choose a different XAdES profile, hashing algorithm, policy identifier, or signature purpose vocabulary.

Signature is used in the following places:

**Annotation**

See also [Examples](file:///C:\temp\datatypes-examples.html#Annotation), [Detailed Descriptions](file:///C:\temp\datatypes-definitions.html#Annotation), [Mappings](file:///C:\temp\datatypes-mappings.html#Annotation), [Profiles & Extensions](file:///C:\temp\datatypes-extras.html#Annotation) and [R2 Conversions](file:///C:\temp\datatypes-version-maps.html#Annotation).

A text note which also contains information about who made the statement and when.

Systems that do not have structured annotations simply communicate a single annotation with no author or time.

This element may need to be included in narrative because of the potential for modifying information.

Annotations **SHOULD NOT** be used to communicate "modifying" information that could be computable (this is a SHOULD because enforcing user behavior is nearly impossible).

Annotation is used in the following places:

**Open Type Element**

Some elements do not have a specified type. The type is represented by the wildcard symbol "\*". In these cases, the element type may be one of the following:

* [boolean](file:///C:\temp\datatypes.html#boolean)
* [integer](file:///C:\temp\datatypes.html#integer)
* [decimal](file:///C:\temp\datatypes.html#decimal)
* [base64Binary](file:///C:\temp\datatypes.html#base64Binary)
* [instant](file:///C:\temp\datatypes.html#instant)
* [string](file:///C:\temp\datatypes.html#string)
* [uri](file:///C:\temp\datatypes.html#uri)
* [date](file:///C:\temp\datatypes.html#date)
* [dateTime](file:///C:\temp\datatypes.html#dateTime)
* [time](file:///C:\temp\datatypes.html#time)
* [code](file:///C:\temp\datatypes.html#code)
* [oid](file:///C:\temp\datatypes.html#oid)
* [id](file:///C:\temp\datatypes.html#id)
* [unsignedInt](file:///C:\temp\datatypes.html#unsignedInt)
* [positiveInt](file:///C:\temp\datatypes.html#positiveInt)
* [markdown](file:///C:\temp\datatypes.html#markdown)
* [Annotation](file:///C:\temp\datatypes.html#Annotation)
* [Attachment](file:///C:\temp\datatypes.html#Attachment)
* [Identifier](file:///C:\temp\datatypes.html#Identifier)
* [CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept)
* [Coding](file:///C:\temp\datatypes.html#Coding)
* [Quantity](file:///C:\temp\datatypes.html#Quantity)
* [Range](file:///C:\temp\datatypes.html#Range)
* [Period](file:///C:\temp\datatypes.html#Period)
* [Ratio](file:///C:\temp\datatypes.html#Ratio)
* [SampledData](file:///C:\temp\datatypes.html#SampledData)
* [Signature](file:///C:\temp\datatypes.html#Signature)
* [HumanName](file:///C:\temp\datatypes.html#HumanName)
* [Address](file:///C:\temp\datatypes.html#Address)
* [ContactPoint](file:///C:\temp\datatypes.html#ContactPoint)
* [Timing](file:///C:\temp\datatypes.html#Timing)
* [Reference](file:///C:\temp\references.html#Reference) - a reference to another resource
* [Meta](file:///C:\temp\resource.html#Meta)

The element name ends with "[x]", and this is replaced with the Title cased name of the data type.

Open references are used in the following places:

**Other Types**

The following types are defined as part of the data types, but are documented elsewhere in the specification:

* [**Resource**](file:///C:\temp\resource.html#metadata) - the conceptual base class for all resources
* [**Reference**](file:///C:\temp\references.html#Reference) - for references from one resource to another
* [**Extension**](file:///C:\temp\extensibility.html) - used to convey additional data in a resource
* [**Narrative**](file:///C:\temp\narrative.html#Narrative) - conveys a human-readable representation of the content of a resource

**defining-extensions.html**

**Defining Extensions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

All extensions used in resources require a formal published definition which can be used by application developers, or the applications themselves, to help integrate extensions into the healthcare process they support.

Every extension in a resource refers directly to its definition, which is made available as a [StructureDefinition](file:///C:\temp\structuredefinition.html). A resource can be [profiled](file:///C:\temp\profiling.html) to specify where particular extensions are used.

Whenever resources containing extensions are exchanged, the definitions of the extensions SHALL be available to all the parties that share the resources. Each extension contains a URI that references the source of the definitions as a [StructureDefinition](file:///C:\temp\structuredefinition.html). The source SHOULD be a literal reference, such as an http: url that refers to an end-point that responds with the contents of the definitions - preferably a [FHIR RESTful server](file:///C:\temp\http.html) supporting the Resources Profile, or a logical reference (e.g. using a urn:) - for instance, to a national published standard. Extensions may be defined by any project or jurisdiction, up to and including international standards organizations such as HL7 itself.

Before defining a new extension, attempt to reuse existing extensions defined in one of the [shared registries described below](file:///C:\temp\main-pages.html#registration). As well, some concepts may be appropriate to adding as part of the core specification.

**Core Elements**

Elements are included as part of FHIR resources and data types principally on the basis of current world-wide usage patterns. Policy is that if a significant majority of systems throughout the world that would use a resource or data type would use an element, then that element will be included as part of the resource/data type. If not, it will be left to an extension. This holds even if the element is very common or even mandatory in one or two specific jurisdictions.

[Proposals](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemBrowse&tracker_id=677) suggesting a new core element can be raised by anyone. (Free registration is required.) However, given the [timelines](file:///C:\temp\timelines.html) for new FHIR releases as well as the uncertainties associated with vetting the specification through a ballot process, it may still be necessary to define extensions even for elements that are likely to be supported as part of the core specification in a future release.

**Context**

Extensions are always defined against some particular context - the type of element that they may be used to extend. The following are possible contexts for an extension:

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Context type** | **Context format** | **Examples** |
| resource | A particular element (including the root) in a single resource | The element path for that element, using the standard dotted notation | Condition Condition.code |
| datatype | A particular element (including the root) in a particular data type | The data type name for primitive types or the element path for complex data types. These extensions can be used anywhere the data type is used | Address.part.value string |
| mapping | A particular context in one of the mapped reference models | The name of the reference model followed by the mapping path. The details of the path depend on the named mapping | RIM: Act[moodCode="EVN"] |
| extension | Another extension | The profile URI of the extension followed by the extension code | http://myextensions.org/someExtension |

Note: For type 'resource' and 'datatype', if the context is an element that can have multiple types, then use the [x] qualified name (e.g. Observation.value[x]) if the extension works on all choice types, or otherwise an enumeration of explicitly named elements if not (e.g. Observation.valueQuantity)

In addition, the extension definition might apply additional constraints with regards to particular element values of the target that make its use appropriate. Extensions SHALL only be used on a target for which they are defined.

**Cardinality**

The cardinality constraints asserted by the extension definition itself apply to any contexts where the extension is used.

**Minimum Cardinality**

If the Extension minimum cardinality is 0, then the extension is optional anywhere it appears. A [profile](file:///C:\temp\profiling.html#resource) that defines the use of an extension may make the minimum cardinality any number up to maximum cardinality of the extension itself. Example: Example: [Patient birthplace](file:///C:\temp\extension-birthplace.html).

If the Extension minimum cardinality is > 1, then the extension must have a minimum cardinality of at least the minimum cardinality in any [profile](file:///C:\temp\profiling.html#resource) that defines the use of an extension. The the minimum cardinality may be any number up to maximum cardinality of the extension. Even with a minimum cardinality >0, the extension is only required to be present in instances if the inatances explicitly or implicitly conform to a profile that defines the use of the extension. Example: [Conformance Expection](file:///C:\temp\extension-capabilitystatement-expectation.html).

**Maximum Cardinality**

If the Extension maximum cardinality is 1, then the extension is only allowed once on any element on which it appears. A [profile](file:///C:\temp\profiling.html#resource) that defines the use of an extension can only make the maximum cardinality 1 (or zero if the minimum cardinality is 0, and the profile constrains another profile that allows the extension). Example: [Mother's Maiden Name](file:///C:\temp\extension-patient-mothersmaidenname.html).

If the Extension maximum cardinality is >1, then the extension is allowed up to the specified number of times on any element on which it appears. A [profile](file:///C:\temp\profiling.html#resource) that defines the use of an extension may make the maximum cardinality any value up to the specified maximum. Example: [Patient Disability](file:///C:\temp\extension-patient-disability.html).

**Use of ElementDefinition in Extension Definitions**

An extension is a wrapper for url and a value or other extensions. As such, some of the properties of the extension are defined on the extension itself, while others are defined on the Extension.value. This list provides guidance for the correct usage:

* Extension root element:
  + Cardinality
  + Short, Definition, Comments
  + IsModifier
  + MustSupport (is used on invocation of the extension)
  + Conditions & Constraints. These SHOULD never be on url/value[x]
  + Mappings. these SHALL never be on url/value[x]
* Extension.url
  + Cardinality = 1...1 (fixed)
  + value = canonical url (fixed)
* Extension.value[x]
  + Type
  + Cardinality for Simple extensions (not nested): 1...1. Use 0..0 if nested. Note that the actual ED cardinality is defined by the root element
  + Binding
  + MaxLength, DefaultValue, Pattern, Example, MinValue, MaxValue

Note: Extensions are always sliced by their url property. It is allowed to re-slice extensions by additional properties (see [Profiling/Slicing](file:///C:\temp\profiling.html#slicing)).

**Publishing Extension Definitions**

As well as defining the base element structure for resources, HL7 also publishes extensions, including as [part of this specification](file:///C:\temp\extensibility-registry.html). HL7 publishes such data definitions as extensions rather than as part of the base resource structure in order to keep the base resource structure simple and concise, and to allow implementers not to engage with an entire world's worth of functionality up front. Note that HL7 does not generally define "modifier" extensions - if HL7 publishes an element that modifies the meaning of other elements, it will mostly be part of the resource content itself, since everyone has to understand the extension anyway.

Before extensions can be used in instances, their definition SHALL be published. HL7 maintains two extension registries:

1. HL7 approved extensions, approved by an appropriate part of the HL7 community following a review process, and which have formal standing
2. Provided as a service to the community, where anyone can register an extension

Users are encouraged to register their extensions in the second registry, though this is not required. All that is required is that the extension is published in a context that is available for users of the extension. So, for example, if a particular extension is only used within a single institution, the definition of the extension can be placed on the institution's intranet. However since, by their nature, resources tend to travel well, it's always better to use the HL7 or other publicly accessible extension registries.

The HL7 FHIR registry can be found at <http://hl7.org/fhir/registry>.

HL7 extension definitions may be balloted alongside resource content as part of the FHIR specification or may be published as part of separate specifications. When HL7 publishes extension definitions as part of the FHIR specification, these extensions SHALL be used for this data whenever the data is represented in instances. Applications SHOULD use other HL7-defined extensions published to represent equivalent data in the interest of maximum interoperability.

To minimize complexity for implementers, HL7 will not elevate widely adopted extensions (defined by HL7 or other organizations) to be content defined in a core resource in future versions of the resource unless there is widespread endorsement of such a migration from the implementer community. This policy ensures that widespread adoption of an extension does not result in a forced migration to a core element. Extensions labeled as draft may be moved in either direction, but after extensions are finalised as normative they won't be moved.

In some cases, an HL7 work group or other body may publish a profile whose sole purpose is to define extensions expected to be needed by implementers in a particular context; e.g. extensions needed to map a particular set of [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) segments or a [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) model.

Implementations are encouraged to share their extensions with HL7 and register them with the HL7 extension registry. The domain committees will work to elevate the extensions into HL7 published extensions or, if adopted by a broad enough portion of the implementer community, into the base resource structure itself.

To avoid interoperability issues, extensions SHALL NOT change their definition once published. (Small clarifications to descriptions that do not affect interoperability are permitted.) Rather than modifying an existing extension, a new extension should be introduced. Revisions to an extension may extend the set of contexts in which the extension apply but may not remove or constrain any context previously listed

**diagnostics-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Diagnostic Medicine Module**

**Introduction**

The Diagnostics Module provides an overview and guide to the FHIR content that addresses ordering and reporting of clinical diagnostics including laboratory testing, imaging and genomics.

**Index**

The Diagnostics module covers the following resources:

|  |  |  |
| --- | --- | --- |
| * [Observation](file:///C:\temp\observation.html) * [DiagnosticReport](file:///C:\temp\diagnosticreport.html) * [ProcedureRequest](file:///C:\temp\procedurerequest.html) | * [ImagingStudy](file:///C:\temp\imagingstudy.html) * [ImagingManifest](file:///C:\temp\imagingmanifest.html) * [Sequence](file:///C:\temp\sequence.html) | * [Specimen](file:///C:\temp\specimen.html) * [BodySite](file:///C:\temp\bodysite.html) |

The diagnostic resources and their relationships are shown below. The arrows represent the direction of the references between resources (for example, DiagnosticReport references ProcedureRequest). See the [Workflow Module](file:///C:\temp\workflow-module.html) for information about the coordination of activities such as ordering and fulfilling of diagnostics.

*Note:* See the [Genomics Implementation Guidance](file:///C:\temp\genomics.html) for additional information about how to use the Diagnostic resources for Clinical Genomic Reporting and Analysis.

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |

**Security and Privacy**

The diagnostic resources often represent patient-related data, and as such are susceptible to data breaching. Necessary privacy and security provision must be in place for searching and fetching this information. For more general considerations, see [the Security and Privacy module](file:///C:\temp\secpriv-module.html).

**Common use Cases**

Diagnostic resources are commonly used to plan, recommend, order and report clinical diagnostics:

* Planning a diagnostic test such as in this [Careplan-example](file:///C:\temp\careplan-example-f203-sepsis.html)
* Recommending diagnostics such as is done by Clinical Decision Support tools. See the [Clinical Reasoning module](file:///C:\temp\clinicalreasoning-module.html) for more information.
* Reporting a Laboratory Test such as a [lipid panel](file:///C:\temp\diagnosticreport-example-lipids.html)
* Ordering a Laboratory Test such as a [lipid panel](file:///C:\temp\procedurerequest-example-lipid.html)
* Reporting a Laboratory Test such as a [lipid panel](file:///C:\temp\diagnosticreport-example-lipids.html)
* Ordering an imaging study such as a [CT study](file:///C:\temp\procedurerequest-example-di.html)
* Reporting an imaging study such as a [CT study](file:///C:\temp\imagingstudy-example.html)
* Ordering a genetics test such as a [gene mutation analysis](file:///C:\temp\procedurerequest-genetics-example-1.html)
* Reporting a genetics test such as [HLA genotyping](file:///C:\temp\diagnosticreport-hla-genetics-results-example.html)

There are many ways to use these resources independently as well. The Observation resource in particular is central to capturing many measurements and events in healthcare and is often used outside the context of diagnostic orders and reports.

* recording vital signs such as [temperature](file:///C:\temp\observation-example-f202-temperature.html)
* recording other observations such as patient's social history like tobacco use
* recording other characteristics of subject such as pregnancy status (note: The Condition resource may be used to represent these as well)
* recording clinical assessment tool scores such as an [Apgar score](file:///C:\temp\observation-example-5minute-apgar-score.html).
* identifying and describing a [specimen](file:///C:\temp\specimen-example-serum.html) or [body site](file:///C:\temp\bodysite-example-tumor.html)

**Related Resources and Modules**

The resources that represent the basic information about a patient and a clinical encounter can be found in the [Administration Module](file:///C:\temp\administration-module.html). Other resources that represent core clinical information generated by healthcare providers during the course of a patient encounter are detailed in the [Clinical Summary Module](file:///C:\temp\clinicalsummary-module.html) and the [Medications Module](file:///C:\temp\medications-module.html).

**Developmental Roadmap**

* The Observation and Diagnostic Report resources have been tested and used in production tooling and as such have reached a [maturity level (FMM)](file:///C:\temp\resource.html#maturity) where changes become less likely.
* The ImagingStudy resource has had some limited testing and use. Further use of this resource in production tooling in order for it to reach a more stable FMM level.

These other diagnostic resources are still under development:

* ProcedureRequest: used in the HL7 production tooling and in several implementation guides. As a result of the addition of [FHIR workflow](file:///C:\temp\workflow.html) for STU3, the ProcedureRequest resources has undergone substantial changes since DSTU2. It has been renamed from DiagnosticOrder and its structure and content updated to align with a general "request" pattern. Because of this, it remains an immature resource.
* Specimen: We expect to see more widespread use of this resource and future updates to its FMM level
* ImagingManifest: We expect to see more widespread use of this resource and future updates to its FMM level
* Sequence: This resource remains draft and we expect to see more widespread use of this resource and future updates to its FMM level
* BodySite: This resource remains draft and currently can only be referenced using extensions.

**diff.html**

**Differences from DSTU Release 2**

This is an automatically generated list of all the structural differences between FHIR DSTU2 and this version. The following possible changes are checked when preparing this list (and others throughout the specification):

* Addition or deletion of elements, or name change to elements
* Changes to minimum and maximum cardinality
* Changes to allowed types
* Changes to bindings for Extensible/Required bindings, and changes to codes if it's a required binding on a code element
* Changes to the is-modifier status of an element
* Changes to the default value for an element

This analysis only considers changes to data types and resources. Comparison of extensions, profiles and value sets will be added in a future version.

**directory.html**

**All Published Versions of FHIR**

The following versions of FHIR are available:

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| **Current Versions** | | |
| [Sept 30, 2014](file:///C:\temp\index.html) | 0.0.82 | Current Official Published Version *(Currently: DSTU1 + 2 technical errata)* |
| [(current)](http://build.fhir.org) | (last commit) | Current Development build (about 40min behind version control, may be incoherent and change rapidly) |
| **DSTU 2 sequence** | | |
| [August 31, 2015](file:///C:\temp\2015Sep\index.html) | 1.0.0 | DSTU 2 QA Preview + CQIF Ballot (Sep 2015) |
| [April 2, 2015](file:///C:\temp\2015May\index.html) | 1.0.0 | DSTU 2 Ballot version (May 2015 Ballot) |
| [Dec 12, 2014](file:///C:\temp\2015Jan\index.html) | 0.4.0 | Draft For Comment (January 2015 Ballot) |
| **DSTU 1 sequence** | | |
| [Sept 30, 2014](file:///C:\temp\DSTU1\index.html) | 0.0.82 | **DSTU 1 (Official version)** with 2 technical errata (Permanent home) |
| Sept ?, 2013 | 0.11 | DSTU 1 Ballot version (Missing - hunting for a copy) |
| [Dec 4, 2012](file:///C:\temp\2013Jan\index.htm) | 0.06 | 2nd Draft for Comment (January 2013 Ballot) |
| [Sep 9, 2012](file:///C:\temp\2012Sep\index.htm) | 0.05 | 1st Draft for Comment (Sept 2012 Ballot) |
| [May 14, 2012](file:///C:\temp\2012May\index.htm) | 0.01 | Oldest Archive version available |

Note: Subsequent to Sept 2013, the FHIR version policy was changed.

**documentation.html**

**Documentation Index**

This page provides an index to the key commonly used documentation pages for FHIR.

|  |  |  |
| --- | --- | --- |
| **Framework**   * [Conformance Rules](file:///C:\temp\conformance-rules.html) * [Resource Life Cycles](file:///C:\temp\lifecycle.html) * [References between Resources](file:///C:\temp\references.html) * [Compartments](file:///C:\temp\compartmentdefinition.html) * [Narrative](file:///C:\temp\narrative.html) * [Extensibility](file:///C:\temp\extensibility.html) * [Formats:](file:///C:\temp\formats.html) [XML](file:///C:\temp\xml.html), [JSON](file:///C:\temp\json.html), & [RDF](file:///C:\temp\rdf.html) * [Data Types (Base)](file:///C:\temp\datatypes.html) * [Data Types (Metadata)](file:///C:\temp\metadatatypes.html) * [FHIRPath](http://hl7.org/fluentpath) | **Exchanging Resources**   * [**RESTful API (HTTP)**](file:///C:\temp\http.html) * [Search](file:///C:\temp\search.html) + [Search Param Registry](file:///C:\temp\searchparameter-registry.html) * [Operations](file:///C:\temp\operations.html) * [Documents](file:///C:\temp\documents.html) * [Messaging](file:///C:\temp\messaging.html) * [Services](file:///C:\temp\services.html)   **Design Patterns**   * [Event](file:///C:\temp\event.html) * [Request](file:///C:\temp\request.html) * [Definition](file:///C:\temp\definition.html) | **Adopting & Using FHIR**   * [Profiling FHIR](file:///C:\temp\profiling.html) * [Downloads - Schemas, Code, Tools](file:///C:\temp\downloads.html) * [Validating Resources](file:///C:\temp\validation.html) * [Mapping Language](file:///C:\temp\mapping-language.html) ([tutorial](file:///C:\temp\mapping-tutorial.html)) * [Testing Implementations](file:///C:\temp\testing.html) * [Inter-version Compatibility](file:///C:\temp\compatibility.html) (also, [FHIR Timelines](file:///C:\temp\timelines.html))   **Safety & Security**   * [Security](file:///C:\temp\security.html) & [Security Labels](file:///C:\temp\security-labels.html) * [Clinical Safety](file:///C:\temp\safety.html) |
| **Base Types**   * [Resource](file:///C:\temp\resource.html) * [DomainResource](file:///C:\temp\domainresource.html) * [Element](file:///C:\temp\element.html) * [BackboneElement](file:///C:\temp\backboneelement.html) * [ElementDefinition](file:///C:\temp\elementdefinition.html) * + [Dosage](file:///C:\temp\dosage.html) (for medications) | **Terminologies**   * [Using Codes in Resources](file:///C:\temp\terminologies.html) * [Code Systems](file:///C:\temp\terminologies-systems.html) * [Value Set List](file:///C:\temp\terminologies-valuesets.html) * [Mappings between Value sets](file:///C:\temp\terminologies-conceptmaps.html) * [Identifier Registry](file:///C:\temp\identifier-registry.html) * + see the [Terminology Service](file:///C:\temp\terminology-service.html) API | **Implementation Advice**   * [Guide to Resources](file:///C:\temp\resourceguide.html) * [Variations between Submitted data and Retrieved data](file:///C:\temp\updates.html) * [Managing Resource Identity](file:///C:\temp\managing.html) * [Push vs Pull](file:///C:\temp\pushpull.html) * [Integrated Examples](file:///C:\temp\integrated-examples.html) * [Common Use Cases](file:///C:\temp\usecases.html) * [Version History](file:///C:\temp\history.html) * [Differences to DSTU 2](file:///C:\temp\diff.html) * [Transforms between DSTU 2 and STU 3](file:///C:\temp\r2maps.html) |

**documentinformation.html**

**Document Information**

The document information structure represents information that sets the context of a document, and carries key information such as the subject and author. This information is also generally used for searching.

Notes:

* The author and the attester are often the same person, but this may not be the case in some clinical workflows
* The attester attests to the collated narrative portions of the resources
* The custodian is responsible for the maintenance of the document. Principally, they are responsible for the policy regarding persistence of the documents. They are not required to actually retain a copy of the document, but they should do so.

**documents.html**

**FHIR Documents**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

FHIR resources can be used to build documents that represent a composition: a set of coherent information that is a statement of healthcare information, particularly including clinical observations and services. A document is an immutable set of resources with a fixed presentation that is authored and/or attested by humans, organizations and devices.

Documents built in this fashion may be exchanged between systems and also persisted in document storage and management systems, including systems such as IHE XDS.

Applications claiming conformance to this framework claim to be conformant to "FHIR documents" (see [Conformance](file:///C:\temp\conformance-rules.html)).

FHIR documents may be 'clinical' (focused on patient healthcare information) but may also serve non-clinical purposes (e.g. FHIR Implementation guides, practice guidelines, patient handouts, etc.) HL7 will develop profiles in the future giving additional guidance on appropriate representation of clinical documents in general as well as specific types of clinical documents (e.g. Consolidated CDA).

Note that FHIR defines both this document format and also a [document reference resource](file:///C:\temp\documentreference.html). FHIR documents are for documents that are authored and assembled in FHIR, while the document reference resource is for general references to pre-existing documents.

* [Example discharge summary](file:///C:\temp\document-example-dischargesummary.html): [XML](file:///C:\temp\document-example-dischargesummary.xml.html) or [JSON](file:///C:\temp\document-example-dischargesummary.json.html)

**Document Content**

All documents have the same structure: a [Bundle](file:///C:\temp\bundle.html) of resources of [type](file:///C:\temp\bundle-definitions.html#Bundle.type) "document" that has a [Composition](file:///C:\temp\composition.html) resource as the first resource in the bundle, followed by a series of other resources, referenced from the Composition resource that provide supporting evidence for the document. The bundle gathers all the content of the document into a single XML or JSON document which may be signed and managed as required. The resources include both human readable and computer processable portions. In addition, the bundle may include [CSS stylesheets](http://www.w3.org/Style/CSS/Overview.en.html), [provenance](file:///C:\temp\provenance.html) statements and a signature.

The composition resource is the foundation of the clinical document. It:

* provides identity and its purpose, sets the context of the document
* carries key information such as the subject and author, and who attests to the document
* divides the document up into a series of sections, each with their own narrative

Any resource referenced directly in the Composition SHALL be included in the bundle when the document is assembled. Specifically, this means the following resource references:

Other resources that these referenced resources refer to may also be included in the bundle if the document construction system chooses to do so. Including these additional resources will make the document bigger, but will save applications from needing to retrieve the linked resources if they need them while processing the document. Thus, whether these linked resources should be included or not depends on the implementation environment.

The document bundle SHALL include only:

1. The Composition resource, and any resources directly or indirectly (e.g. recursively) referenced from it
2. A Binary resource containing a stylesheet (as described below)
3. Provenance Resources that have a target which is a resource included in item #1

There are two key identifiers on the document:

* The document identifier (mandatory). This is found in *Bundle.identifier* and is globally unique for this instance of the document, and is never re-used anywhere, including for other documents derived from the same composition
* The Composition identifier (optional). This is found in *Composition.identifier*, and is the same for all documents that are derived from this composition

The document has several dates in it:

* The document date (mandatory). This is found in *Bundle.meta.lastUpdated* and identifies when the document bundle was assembled from the underlying resources
* The Composition date (mandatory). This is found in *Composition.date*, which is when the author wrote the document logically
* The Attestation dates (optional). This is found in *Composition.attester.time* and is when the document was witnessed by the attesters. This would usually be at the same time as the composition date or afterwards
* The Composition last modified time (optional). This is found in *Composition.meta.lastUpdated* for the composition, and is the last date of change of the composition. This must be >= the composition date

Document Bundles may be signed using digital signatures following the rules laid out in the [digital signatures](file:///C:\temp\signatures.html) page. The signature SHOULD be provided by a listed attester of the document and the signature SHOULD contain a [KeyInfo element](http://www.w3.org/TR/xmldsig-core/#sec-KeyInfo) that contains a KeyName element whose value is a URI that matches the [fullUri](file:///C:\temp\bundle-definitions.html#Bundle.entry.fullUri) for the matching attester resource.

Once assembled into a bundle, the document is immutable - its content can never be changed, and the document id can never be reused. Note that the document may be represented in either XML or JSON, and interconverted between these or have its character encoding changed, all the while remaining the same document. However the directly referenced content within the document and the presentation of the document cannot change substantially (such that it changes the clinical meaning of the content). Any additional documents derived from the same composition SHALL have a different document id.

**Document Presentation**

When the document is presented for human consumption, applications SHOULD present the collated narrative portions in order:

1. The [subject resource](file:///C:\temp\composition-definitions.html#Composition.subject) Narrative
2. The [Composition](file:///C:\temp\composition.html) resource Narrative
3. The [section.text](file:///C:\temp\composition-definitions.html#Composition.section.text) Narratives

The presentation of the document is called the 'attested content' of the document. Additional resources can be included in the bundle (e.g. resources referenced from the List that represent the section.content SHOULD be in the bundle, and other additional resources they reference can be included), but these (and any narrative) are not attested content. Specifically, the Composition.attester attests to the presented form of the document.

The Composition resource narrative should summarize the important parts of the document header that are required to establish clinical context for the document (other than the subject, which is displayed in its own right). To actually build the combined narrative, simply append all the narrative <div> fragments together.

If the document is presented in a different order from that given above, it may not represent the original attested content. Implementation Guides may restrict docuemnt narrative and document display behaviour further.

The [XML Tools reference implementation](file:///C:\temp\downloads.html#refimpl) includes a XSLT transform that converts an XML document into browser-ready XHTML.

In addition to the [basic style rules about Narratives](file:///C:\temp\narrative.html#css), which must be followed, a document can reference or contain one or more stylesheets that contains additional styles that apply to the collated narrative. This is done by asserting stylesheet links on the feed:

<Bundle xmlns="http://hl7.org/fhir">

<!-- metadata and type -->

<link>

<relation value="stylesheet"/>

<url value="[uri]"/>

</link>

</Bundle>

The url can be an absolute reference to a CSS stylesheet or a relative reference to a Binary resource that carries a CSS stylesheet. Stylesheet references can only refer to a CSS stylesheet - other forms of stylesheet are not acceptable.

Relative (internal) references SHOULD be used for stylesheets, because the viewer may be unable to resolve external content at the time of viewing, due to technical problems or local policy decisions.

Any stylesheet referenced or used SHALL NOT alter the presentation in such a way that it changes the clinical meaning of the content.

Unless otherwise agreed in local trading partner agreements, applications displaying the collated narrative SHOULD use the stylesheets specified by the document (see [security note](file:///C:\temp\security.html#stylesheet)). Parties entering into a trading agreement to do otherwise should consider the implications this action will have on their long term scope for document exchange very carefully. If the parties agree to use stylesheets that are not contained in the document, then it may be that they will never be able to share their documents safely in a more general context, such as a regional or national EHR or a global personal health record.

**Document Profiles**

[Document profiles](file:///C:\temp\profiling.html) are used to describe documents for a particular purpose. Document profiles may make rules about:

* The content of the Composition resource, including
* The structure of the sections in the composition
* Which resources are to be included in the bundle along with the resources that are directly referenced in the Document resource

Applications should consider publishing [Capability Statements](file:///C:\temp\capabilitystatement.html) that identify particular documents they support. Documents can identify a profile that they conform to by placing a profile identifier in the Bundle.meta.profile element - see [Profile Tags](file:///C:\temp\resource.html#meta) for a discussion of the utility of this.

**Document Handling Obligations**

The authors/constructors and processors of Clinical Documents, whether human or software, have obligations that they must satisfy.

**Author/Constructor Obligations**

A document constructor is an application that creates a document. An author is a human, organization or device that uses the constructor to create a document. Between them, the constructor and the author may create new content resources and/or assemble already existing content resources while performing their tasks. They also have the following responsibilities:

* To assure that the document SHALL contain valid composition that conforms to the rules described here and that only links to other valid resources
* To assure that the content of document SHALL conform to any declared [Profiles](file:///C:\temp\profiling.html) (see below).
* Ensure that the attesters are properly aware of the presentation of the document to which they are attesting

**Processor Obligations**

A document processor is an application and/or human user that receives documents, and extracts data from them, or makes decisions because of them. The documents may be received directly from a document constructor, accessed via a document management system or forwarded by a third party. The document processor is responsible for ensuring that received documents are processed and/or rendered in accordance to this specification. A document processor has the obligation to assure that the following rules are followed:

* When storing/transmitting a document, any method may be used as long as the bundled document can be (re-)assembled with sufficient integrity to validate a digital signature
* When presenting the narrative of the document, the rules described above SHALL be followed
* Resources or data from the document may be extracted for additional uses, but such data is no longer considered to be attested by the document author
* Wherever the data from the document is displayed to a user, there SHOULD always be a way for the user to access a presentation of the original document

In addition to these obligations, document receivers SHOULD carefully track the source of documents for new documents that supersede existing documents, particularly when the documents represent compositions that have been retracted. When documents have been replaced, they SHOULD either withdraw data extracted from superseded documents or warn users when they view the document or data taken from it.

**Document End-Points**

There are several different RESTful end-points used when working with documents. The use of the various end-points can be best be described by considering the consequences of posting to them:

the Bundle.id will change, but the Bundle.identifier will not

|  |  |  |
| --- | --- | --- |
| **End-Point** | **Type of Content** | **Description** |
| [baseurl]/Bundle | Document Bundle | This works like a normal end-point for managing a type of resource, but it works with whole document bundles - i.e. a read operation returns a bundle, an update gets a bundle and a search returns a bundle of bundles. Note that if documents are POSTed using a [create](file:///C:\temp\http.html#create) interaction |
| [baseurl]/Composition | Composition Resource | The normal end-point for managing composition resources. This can be used while building a document or after breaking a document up into its constituent resources or when using compositions separately from documents |
| [[baseurl]/Binary](file:///C:\temp\binary.html) | Document Bundle | Just store the entire document as a sequence of bytes and return exactly that sequence when requested. There is no way to find content in the /Binary end-point, so usually this would be associated with a [Document Reference](file:///C:\temp\documentreference.html) so that applications can find and process the document, though this is not required |
| [[baseurl]](file:///C:\temp\http.html#transaction) (e.g. a transaction) | Document Bundle | Ignore the fact that the bundle is a document and process all of the resources that it contains as individual resources. Clients SHOULD not expect that a server that receives a document submitted using this method will be able to reassemble the document exactly. (Even if the server can reassemble the document (see below), the result cannot be expected to be in the same order, etc. Thus a document signature will very likely be invalid.) |

Note: While these end-points are defined for use with document resources and document bundles, it is not necessary to use them. Documents may be transferred between systems using any method desired. In addition, servers and/or specifications may define additional [operations](file:///C:\temp\operations.html) for handling documents beyond the options described above.

**Generating a Document**

A client can ask a server to generate a fully bundled document from a composition resource. For details, see [Generate Document Operation](file:///C:\temp\composition-operations.html).

**dosage-definitions.html**

**Dosage Detailed Descriptions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Dosage**

**dosage-examples.html**

**Dosage Examples**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**dosage-mappings.html**

**Dosage Mappings**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page provides mappings for Dosage.

**dosage.html**

**Dosage**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The Dosage structure defines general dosage instruction information typically represented in medication requests, medication dispenses and medication statements.

**Content**

**Constraints**

Dosage is used in the following places:

**Interpretation of Dosage in different contexts**

**downloads.html**

**Downloads**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

|  |  |
| --- | --- |
| **Specification Downloads** | |
| FHIR Definitions | All the value sets, profiles, etc.defined as part of the FHIR specification, and the included implementation guides:   * [XML](file:///C:\temp\definitions.xml.zip) * [JSON](file:///C:\temp\definitions.json.zip) * [Forge](file:///C:\temp\definitions-r2.xml.zip): Special version of definitions for [Forge](http://fhir.furore.com/Forge) (temporary)   These is the master set of definitions that should be the first choice whenever generating any implementation artefacts. All the other forms below include only subsets of the information available in these definition files, and do not contain all of the rules about what makes resources valid. Implementers will still need to be familiar with the content of the specification and with any [profiles that apply to the resources](file:///C:\temp\profiling.html) in order to make a conformant implementation. |
|  | |
| XML | * [Examples](file:///C:\temp\examples.zip) - all the example resources in XML format * [Validation Schemas](file:///C:\temp\fhir-all-xsd.zip) (includes support schemas, resource schemas, modular & combined schemas, and Schematrons) * [Code Generation Schemas](file:///C:\temp\fhir-codegen-xsd.zip) (See [notes about code-generation schemas](file:///C:\temp\xml.html#schema-gen)) |
|  | |
| JSON | * [Examples](file:///C:\temp\examples-json.zip) - all the example resources in JSON format * [JSON Schemas](file:///C:\temp\fhir.schema.json.zip) (includes both modular and combined schema) |
|  | |
| RDF | * [Turtle Examples](file:///C:\temp\examples-ttl.zip) - all the example resources in Turtle format * [JSON-LD Examples](file:///C:\temp\examples-jsonld.zip) - all the example resources in JSON-LD format (experimental) * [ShEx Schemas](file:///C:\temp\fhir.schema.shex.zip) - [ShEx](https://www.w3.org/2001/sw/wiki/ShEx) definitions for validating RDF resources * [Definitions](file:///C:\temp\fhir.rdf.ttl.zip) - the formal definitions that define the predicates and classes used in the RDF format (not up to date) * [JSON-LD Definitions](file:///C:\temp\fhir.jsonld) - the JSON-LD @context for this version of FHIR |
|  | |
| FHIR Specification | [The whole specification](file:///C:\temp\fhir-spec.zip) so that you can host your own local copy (does not include the downloads) |
| **Implementation Tools** | |
| Validator | The [official FHIR validator](file:///C:\temp\validator.zip) - a Java jar file that can be used to validate resources. See [Validation Tools](file:///C:\temp\validation.html) for further information |
| IG Publisher | The [Implementation Guide Publishing Tool](file:///C:\temp\org.hl7.fhir.igpublisher.jar) (see [IG Publishing documentation](http://wiki.hl7.org/index.php?title=IG_Publisher_Documentation)) |
| IG Pack | The [Definitions File used by the IG Publisher](file:///C:\temp\igpack.zip). This is a zip file that contains all the conformance & example resources, a version marker, and a json file that lists all the pages and urls in the specification itself. This file can also be used by the validator |
| Translation File | [Translations of common FHIR names and messages](file:///C:\temp\translations.xml) into multiple languages (see [wiki](http://wiki.hl7.org/index.php?title=FHIR_Implementation_Page) for instructions on how to add to this) |
| Icon Pack | The [FHIR Icon at various resolutions](file:///C:\temp\icon-pack.zip). Any FHIR Implementation created by an organization that has attended a connectathon is allowed to use the FHIR icon in association with the application (this policy will be reviewed in the future). |
| Test Cases | A [Collection of Test Cases](file:///C:\temp\test-cases.zip). These are XML or JSON files that provide test cases for the various FHIR reference implementations to ensure correct functioning |
| **Reference Implementations** | |
| There are many open source reference implementations available to help implementers. Here are a list of the more common implementations used by implementers: | |
| Java | [HAPI-FHIR](http://jamesagnew.github.io/hapi-fhir/): Object Models, Parsers, Client + Server Framework, FHIR Validator, & Utilities. The specification is built with this Java code |
| C# | [HL7.FHIR](http://www.nuget.org/packages/Hl7.Fhir): Object models, Parsers/Serialisers, Utilities, and a Client. Source code on GitHub at <http://github.com/ewoutkramer/fhir-net-api> |
| Pascal | [FhirServer](http://github.com/grahamegrieve/fhirserver): Object models, Parsers/Serialisers, Validator, Utilities, Client, and the FHIR Reference server. Requires [Delphi](https://www.embarcadero.com/products/delphi) (unicode versions) |
| XML | [XML Tools](file:///C:\temp\fhir--XMLTools-0.01.zip): Document Rendering Stylesheet, supplementary implementation schemas and transforms |
| Javascript | [FHIR.js](https://github.com/FHIR/fhir.js): Javascript Client and Utilities |
| Swift | [Swift-FHIR](https://github.com/smart-on-fhir/Swift-FHIR): Object Model, Client and Utilities |

Note: These reference implementations are provided for implementer interest and assistance. While they may be used (and are) in production system, HL7 and their various contributors accept no liability for their use. Note that these reference implementations are provided to assist to implementers to adopt the specification, and some are maintained by the FHIR project team, but are not part of the specification, and implementations are not required to conform to these, nor are they subject to the formal standards process.

Full blown open source implementations for FHIR, some of which use these reference implementations, are listed on the [HL7 wiki](http://wiki.hl7.org/index.php?title=Open_Source_FHIR_implementations).

It is not necessary to use these particular implementations in order to be conformant. Any other approach may be used, including code generated from the schemas.

**ehr-fm.html**

**Appendix: HL7 EHR System Functional Model and FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

IDO/HL7 10781 EHR System Functional Model Release 2 provides a reference list of functions that may be present in an Electronic Health Record System. While FHIR is an implementation focused on exchange of information in healthcare, this often happens in the context of an EHR system and EHR record. This table briefly describes one way that FHIR can be used to meet the requirements described in the EHR-S FM and is provided to help readers of the FHIR specification understand how FHIR can be used. There are many other equally valid ways to implement the EHR-S FM and to make use of FHIR.

|  |  |  |
| --- | --- | --- |
| **EHR Function** | | **FHIR Implementation Notes** |
| TI.1 | Security | FHIR defines parts of the security infrastructure, and delegates others to standard web based security frameworks |
| TI.1.1 | Entity Authentication | FHIR assumes that the users are authenticated. OAuth is the preferred mechanism |
| TI.1.2 | Entity Authorization | FHIR does not currently provide any resources to describe or manage access-control permissions. By default, underlying web frameworks such as SAML would be used. See [the security section](file:///C:\temp\security.html#binding) for a discussion of binding between FHIR and SAML |
| TI.1.3 | Entity Access Control | See above about SAML / OAuth |
| TI.1.4 | Patient Access Management | See [Security Labels](file:///C:\temp\security-labels.html) |
| TI.1.5 | Non-Repudiation | The [provenance resource](file:///C:\temp\provenance.html) tracks the timestamps, actors, digital signatures associated with resources |
| TI.1.6 | Secure Data Exchange | TLS (https:) should be used for all production exchange of data. All conformant FHIR RESTful implementations SHALL be able to use TLS |
| TI.1.7 | Secure Data Routing | FHIR allows for brokers and various forms of messaging that support assured destinations and delivery (also see IN.2.2 below) |
| RI.1.1.4 | Information Attestation | See the [provenance resource](file:///C:\temp\provenance.html) |
| TI.1.8 | Patient Privacy and Confidentiality | FHIR does not include functionality related to this requirement, though implementations would be expected to provide this |
| RI.1.1 | Health Record Information and Management | This is a core application of the FHIR capabilities |
| RI.1.22 | Data Retention, Availability and Destruction | A FHIR RESTful server gives precise and fine-grained control of retention, availability and destruction of resources, all clearly described by the capability statement |
| RI.1.1.x.1 | Auditable Records | FHIR provides the [AuditEvent](file:///C:\temp\auditevent.html) resource for auditable records. |
| RI.2 | Synchronization | FHIR supports synchronization using standard web publication/subscription methods via [Bundles](file:///C:\temp\bundle.html). Bundle-based pub/sub may be push or pull based, and can include all resources of a particular type, or selected subsets of the resources. In addition, groups of resources can be exchanged in bundles, keeping a set of related resources in synchronization |
| RI.1.1.13 | Extraction of Health Record Information | FHIR does not provide report formats, but does provide extensive search and retrieval functions to assist with building such reports |
| RI.1.1.1 | Store and Manage Health Record Information | A FHIR RESTful server can store and manage health information persistently - see below for further information. |
| RI.1.2.1 | Manage Structured and Unstructured Health Record Information | The dual contents of FHIR resources - structured data and XHTML narrative - provide seamless support for dealing with a mix of structured and unstructured information |
| TI.3 | Registry and Directory Services | The FHIR [Administration resources](file:///C:\temp\resourcelist.html#administrative) provide a registry based access to patients, providers, etc. |
| TI.4 | Standard Terminologies and Terminology Services | FHIR encourages the use of standard terminologies wherever possible, and provides full support for their use through a variety of terminology related [data types](file:///C:\temp\datatypes.html). FHIR defines [a terminology service infrastructure](file:///C:\temp\terminology-service.html). Also, see [profiling](file:///C:\temp\profiling.html), which discusses how terminology is used in a FHIR context |
| TI.5 | Standards-based Interoperability | FHIR is a definition of a standard on which to base interoperability |
| TI.5.1 | Interchange Standards | This is the core focus of FHIR. See below for discussion of interaction modes |
| TI.5.2 | Interchange Standards Versioning and Maintenance | FHIR version maintenance is [described here](file:///C:\temp\resource.html#version) |
| TI.5.3 | Standards-based Application Integration | FHIR enables simple integration through use of an easy to understand, use and debug web-based infrastructure. The same framework used within an EHR for persistence can also offer a simple way to implement exchange |
| TI.5.4 | Interchange Agreements | The FHIR Conformance Statement and Resource Profile resources provide a registry based infrastructure for individual trading partner agreements, as well as for community based ones |
| TI.6 | Business Rules Management | FHIR does not address this requirement at this point in time |
| TI.7 | Workflow Management | FHIR does not address this requirement at this point in time, though the resources and services exist to support this functionality |

The EHR system functional model describes several modes for interaction between systems. Each of these can be implemented in several different ways using FHIR

|  |  |
| --- | --- |
| **Interaction Modes** | **FHIR Options** |
| Unsolicited Notifications e.g. a patient has arrived for a clinic appointment | * create/update new resource via HTTP * push resources using Bundle * Send FHIR [Message](file:///C:\temp\messaging.html) (if appropriate event is defined) |
| Query/Response e.g. Is Adam Everyman known to the system? Yes, MRN is 12345678. | * search with parameters * A query message (though not defined yet) |
| Service Request and Response e.g. Laboratory Order for Fasting Blood Sugar and a response containing the results of the test. | Could be supported either through Messaging or SOA solutions. Request/Response support is not yet defined |
| Information Interchange between organizations (e.g. in a RHIO, or in a National Health System) | * pub/sub using bundles (push or pull) * RESTful interface * FHIR messaging |
| Structured / Unstructured clinical document, e.g. dictated surgical note | See the [Documents](file:///C:\temp\documents.html) |

The combination of a properly secured and managed FHIR server, along with enforced use of the [AuditEvent](file:///C:\temp\auditevent.html) and [Provenance](file:///C:\temp\provenance.html) resources ensures that the core record management functions defined in the EHR-S FM are met (as follows). See the FHIR Record Lifecycle Event Implementation Guide for additional details.

* Lifespan/Lifecycle tracking, including capturing source, origination and authorship information, along with tracking of views and exchanges
* Attestation for accuracy and completeness, along with digital signature
* A full version history with content retention
* Retention and persistence

Additional functionality, not defined at this point in time in FHIR, is required to ensure non-repudiation, access control, and consent tracking.

**element-definitions.html**

**Element - Detailed Descriptions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Element**

**element.html**

**Element**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The base definition for all elements contained inside a resource. All elements, whether defined as a [Data Type](file:///C:\temp\datatypes.html) (including primitives) or as part of a resource structure, have this base content:

* [Extensions](file:///C:\temp\extensibility.html)
* An internal id

There are 3 kinds of descendent types that specialize Element:

* [Primitive data types](file:///C:\temp\datatypes.html#primitive), that add a primitive value property of the specified type
* [Complex data types](file:///C:\temp\datatypes.html#complex), that add their own children (all of which are also elements)
* [BackboneElement](file:///C:\temp\backboneelement.html), A specialization that adds modifierExtension, which is the super-type of all the element types defined in resource definitions (e.g. [Patient.contact](file:///C:\temp\patient.html#resource))

Note that resources themselves all specialise the base type [Resource](file:///C:\temp\resource.html).

**Content**

* [Structure](file:///C:\temp\main-pages.html#tabs-Element-struc)
* [UML](file:///C:\temp\main-pages.html#tabs-Element-uml)
* [XML](file:///C:\temp\main-pages.html#tabs-Element-xml)
* [JSON](file:///C:\temp\main-pages.html#tabs-Element-json)
* [All](file:///C:\temp\main-pages.html#tabs-Element-all)

**Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [**Name**](file:///C:\temp\formats.html#table) | [**Flags**](file:///C:\temp\formats.html#table) | [**Card.**](file:///C:\temp\formats.html#table) | [**Type**](file:///C:\temp\formats.html#table) | [**Description & Constraints**](file:///C:\temp\formats.html#table) |
| [Element](file:///C:\temp\element-definitions.html#Element) | I |  | n/a | Base for all elements *All FHIR elements must have a @value or children* |
| [id](file:///C:\temp\element-definitions.html#Element.id) |  | 0..1 | [id](file:///C:\temp\datatypes.html#id) | xml:id (or equivalent in JSON) |
| [extension](file:///C:\temp\element-definitions.html#Element.extension) |  | 0..\* | [Extension](file:///C:\temp\extensibility.html#Extension) | Additional Content defined by implementations |
| [Documentation for this format](file:///C:\temp\formats.html#table) | | | | |

**UML Diagram** ([Legend](file:///C:\temp\formats.html#uml))

Elementid : id 0..1extension : Extension 0..\*

**XML Template**

<[**[name]**](file:///C:\temp\element-definitions.html#Element) xmlns="http://hl7.org/fhir" id="Internal id (e.g. like xml:id) ([id](file:///C:\temp\datatypes.html#id))">

<[**extension**](file:///C:\temp\element-definitions.html#Element.extension)><!-- **0..\*** [Extension](file:///C:\temp\extensibility.html#Extension) [Additional Content defined by implementations](file:///C:\temp\terminologies.html#unbound) --></extension>

</[name]>

**JSON Template**

// complex types:

{

"[id](file:///C:\temp\element-definitions.html#Element.id)" : "<[id](file:///C:\temp\datatypes.html#id)>", // Internal Id

"[extension](file:///C:\temp\element-definitions.html#Element.extension)" : [{// [Additional Content defined by implementations](file:///C:\temp\extensibility.html#Extension)

// from Element: [extension](file:///C:\temp\extensibility.html) (recursive)

"[url](file:///C:\temp\extensibility-definitions.html#Extension.url)" : "<[uri](file:///C:\temp\datatypes.html#uri)>", // **R!** identifies the meaning of the extension

"[value[x]](file:///C:\temp\extensibility-definitions.html#Extension.value_x_)" : <[\*](file:///C:\temp\datatypes.html#open)> // identifies the meaning of the extension

}]

}

// primitive types:

{

"{name}" : "value", // The primitive value

"{name}\_" : { // Special syntax for the id and extensions

"[id](file:///C:\temp\element-definitions.html#Element.id)" : "<[id](file:///C:\temp\datatypes.html#id)>", // Internal Id

"[extension](file:///C:\temp\element-definitions.html#Element.extension)" : [{// [Additional Content defined by implementations](file:///C:\temp\extensibility.html#Extension)

// from Element: [extension](file:///C:\temp\extensibility.html) (recursive)

"[url](file:///C:\temp\extensibility-definitions.html#Extension.url)" : "<[uri](file:///C:\temp\datatypes.html#uri)>", // **R!** identifies the meaning of the extension

"[value[x]](file:///C:\temp\extensibility-definitions.html#Extension.value_x_)" : <[\*](file:///C:\temp\datatypes.html#open)> // identifies the meaning of the extension

}]

}

}

**Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [**Name**](file:///C:\temp\formats.html#table) | [**Flags**](file:///C:\temp\formats.html#table) | [**Card.**](file:///C:\temp\formats.html#table) | [**Type**](file:///C:\temp\formats.html#table) | [**Description & Constraints**](file:///C:\temp\formats.html#table) |
| [Element](file:///C:\temp\element-definitions.html#Element) | I |  | [Element](file:///C:\temp\element.html) | Base for all elements *All FHIR elements must have a @value or children* |
| [id](file:///C:\temp\element-definitions.html#Element.id) |  | 0..1 | [id](file:///C:\temp\datatypes.html#id) | xml:id (or equivalent in JSON) |
| [extension](file:///C:\temp\element-definitions.html#Element.extension) |  | 0..\* | [Extension](file:///C:\temp\extensibility.html#Extension) | Additional Content defined by implementations |
| [Documentation for this format](file:///C:\temp\formats.html#table) | | | | |

**UML Diagram** ([Legend](file:///C:\temp\formats.html#uml))

Elementid : id 0..1extension : Extension 0..\*

**XML Template**

<[**[name]**](file:///C:\temp\element-definitions.html#Element) xmlns="http://hl7.org/fhir" id="Internal id (e.g. like xml:id) ([id](file:///C:\temp\datatypes.html#id))">

<[**extension**](file:///C:\temp\element-definitions.html#Element.extension)><!-- **0..\*** [Extension](file:///C:\temp\extensibility.html#Extension) [Additional Content defined by implementations](file:///C:\temp\terminologies.html#unbound) --></extension>

</[name]>

**JSON Template**

// complex types:

{

"[id](file:///C:\temp\element-definitions.html#Element.id)" : "<[id](file:///C:\temp\datatypes.html#id)>", // Internal Id

"[extension](file:///C:\temp\element-definitions.html#Element.extension)" : [{// [Additional Content defined by implementations](file:///C:\temp\extensibility.html#Extension)

// from Element: [extension](file:///C:\temp\extensibility.html) (recursive)

"[url](file:///C:\temp\extensibility-definitions.html#Extension.url)" : "<[uri](file:///C:\temp\datatypes.html#uri)>", // **R!** identifies the meaning of the extension

"[value[x]](file:///C:\temp\extensibility-definitions.html#Extension.value_x_)" : <[\*](file:///C:\temp\datatypes.html#open)> // identifies the meaning of the extension

}]

}

// primitive types:

{

"{name}" : "value", // The primitive value

"{name}\_" : { // Special syntax for the id and extensions

"[id](file:///C:\temp\element-definitions.html#Element.id)" : "<[id](file:///C:\temp\datatypes.html#id)>", // Internal Id

"[extension](file:///C:\temp\element-definitions.html#Element.extension)" : [{// [Additional Content defined by implementations](file:///C:\temp\extensibility.html#Extension)

// from Element: [extension](file:///C:\temp\extensibility.html) (recursive)

"[url](file:///C:\temp\extensibility-definitions.html#Extension.url)" : "<[uri](file:///C:\temp\datatypes.html#uri)>", // **R!** identifies the meaning of the extension

"[value[x]](file:///C:\temp\extensibility-definitions.html#Extension.value_x_)" : <[\*](file:///C:\temp\datatypes.html#open)> // identifies the meaning of the extension

}]

}

}

**Constraints**

This constraint exists to reduce syntactical variation in resource contents. If an element has no children, then it is always omitted from the resource, as opposed to optionally present without any content.

**Representation of Element**

As the base type for all elements included in a resource, Element is an important structural element of FHIR. Even the primitive types inherit the base features and representation rules that apply to the Element type.

**XML Representation**

Elements are represented by an XML element. The name of the element comes from the context in which it is used, not from the type. The internal id is represented as an attribute (similar to xml:id, but see below about scope). Extensions are represented as XML elements. Here is the representation for an element 'code' of type [Coding](file:///C:\temp\datatypes.html#Coding):

<code id="[internal id]">

<extension url="..."/>

... if there is any extensions

<extension>

.. elements of Coding type...

</code>

Primitive types have the same representation; the actual primitive value appears as an XML attributed named value on the XML element. For example, a string property named "name" will be represented like this:

<code id="[internal id] value="[value of string]"">

<extension url="..."/>

... if there is any extensions

<extension>

</code>

**JSON Representation**

Elements (except for primitive types, see below) are represented by a json object property. The name of the property comes from the context in which it is used, not from the type. The internal id is represented as a JSON string property named "\_id". Extensions are represented in a JSON array of object named "extension". Here is the representation for a property 'code' of type [Coding](file:///C:\temp\datatypes.html#Coding):

{

"code" : {

"\_id" : "[internal id]",

"extension" : [

..extensions, if present...

],

.. properties of Coding type...

}

}

Primitive types are represented differently; the actual primitive value appears as a JSON string or number property. If an internal id or extensions are present, they appear in a json object with the name of the primitive value property with "\_" appended. For example, a string property named "name" will be represented like this:

{

"name" : "[value of string]",

"name\_" : {

"\_id" : "[internal id]",

"extension" : [

..extensions, if present...

]

}

}

The exact use of this pattern is [described here](file:///C:\temp\json.html#primitive).

**Internal Id Scope**

The id property of the element is defined to allow implementers to build implementation functionality that makes use of internal references inside the resource. This specification does not use the internal id on the element in any way.

The internal id is unique within the scope of the resource that contains it. Specifically, this means:

* The id SHALL be unique within a given resource
* the uniqueness boundary extends into contained resources. i.e. a contained resource cannot have the same id as any element in the resource that contains it or any other contained resource
* The uniqueness boundary is broken at Bundle.entry.resource and Parameters.parameter.resource, since these are elements that aggregate different resources

These rules ensure that there is no need to change internal identifiers while exchanging resources.

**elementdefinition-definitions.html**

**Element Definition Detailed Descriptions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**ElementDefinition**

**elementdefinition-examples.html**

**Element Definition Examples**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

For ElementDefinition examples, see the extensive [StructureDefinition example list](file:///C:\temp\structuredefinition-examples.html).

**elementdefinition-mappings.html**

**Element Definition Mappings**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page provides mappings for Element Definition.

**elementdefinition.html**

**Element Definition**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The definition of an element in a resource or extension. The definition includes:

* Path (name), Cardinality, and data type
* Definitions, usage notes, requirements
* Default or fixed values
* Constraints, Length limits, and other usage rules
* Terminology Binding
* Mappings to other specifications
* Structural Usage Information such as [Slicing](file:///C:\temp\profiling.html#slicing)

The ElementDefinition type is the core of the FHIR metadata layer, and is closely (conceptually) aligned to [ISO 11179](file:///C:\temp\elementdefinition-mappings.html#iso11179). The [DataElement](file:///C:\temp\dataelement.html) resource is a packaging wrapper around the ElementDefinition type. All the data elements defined in this specification are published as a collection of data elements ([XML](file:///C:\temp\dataelements.xml) or [JSON](file:///C:\temp\dataelements.json)).

**Content**

**Constraints**

**Use of ElementDefinition.path**

The path element is the most important property of the element definition. It both names the element, and locates the element within a heirarchy of defined within a particular context. Within the FHIR specification, there is only one original definition for each path (that is, an element definition with no base element). This is the master definition to which all the other definitions with the same path must conform.

All elements defined wihin the FHIR specification itself are defined within a [StructureDefinition](file:///C:\temp\structuredefinition.html) that defines a resource, or a datatype. This defines the identity of the element amnd provides the context in which the meaning of the element is understood. When Elements are defined, the following rules apply:

* Element names (the parts of a path delineated by the '.' character) SHALL not conain whitespace (i.e. unicode characters marked as whitespace)
* Element names SHALL not contain the characters ,:;'"/|?|!@#$%^&\*()[]{}
* Element names SHOULD not contain non-ASCII characters
* Element paths SHALL not exceed 64 characters in length
* Element paths cannot imply elements that are not explicitly defined i.e. a.b.c.d cannot be defined unless a.b.c is explicitly defined
* By convention, the paths starts with an uppercase letter (type) but all the element names that follow this are lowercase (not type names). All resources and data types (except for Primitive data types) follow this convention, but logical models are not required to do so

Elements may be defined in:

* StructureDefinitions of [kind](file:///C:\temp\structuredefinition-definitions.html#StructureDefinition.kind) = resource, complex-type or primitive-type, where [derivation](file:///C:\temp\structuredefinition-definitions.html#StructureDefinition.derivation) = specialization. These are either Resources or Data types defined in the specification, or in custom resource definitions (see [Custom Resource Policy](http://wiki.hl7.org/index.php?title=FHIR_Custom_Resources))
* StructureDefinitions of [kind](file:///C:\temp\structuredefinition-definitions.html#StructureDefinition.kind) = logical
* Data Elements

StructureDefinitions with [derivation](file:///C:\temp\structuredefinition-definitions.html#StructureDefinition.derivation) = constraint (i.e. Resource and Data Type profiles) are not allowed to define or include ElementDefinitions with a path not defined within the base resource definition from which they derive (e.g. in the FHIR specification).

**ElementDefinition.id**

In addition to the path, every ElementDefinition SHALL have a populated [id](file:///C:\temp\element.html#table), and the id SHALL have a unique value populated by following this algorithm:

* The id will be constructed as a dot separated string, each part corresponding to a token in the path
* For each token in the path, use the syntax pathpart:slicename/reslicename

Note that in a profile with no slices, this id will match the path exactly and entirely. id values constructed in this fashion are unique, and persistent, and may be used as the target of external references into the definition, where necessary.

**Interpretation of ElementDefinition in different contexts**

The data type ElementDefinition is used both in [StructureDefinition](file:///C:\temp\structuredefinition.html) and [DataElement](file:///C:\temp\dataelement.html), and within those resources, the way its elements are to be used and interpreted depend on the context where ElementDefinition used:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ElementDefinition field** | **DataElement** | **Type definition, first element** | **Type definition, following elements** | **Constraint Definition, first element** | **Constraint Definition, following elements** |
| sliceName | prohibited | prohibited | prohibited | prohibited | required for slices, else prohibited |
| label | optional | optional | optional | recommended | recommended |
| code | optional | optional | optional | optional | optional |
| slicing | prohibited | prohibited | prohibited | prohibited | optional |
| short/definition | optional | required | required | requiredÃ¢â‚¬Â¡ | requiredÃ¢â‚¬Â¡ |
| requirements/ comments/alias | optional | prohibited | optional | prohibitedÃ¢â‚¬Â¡ | optionalÃ¢â‚¬Â¡ |
| base | prohibited | prohibited | prohibited | required | required |
| type | required | required | required | optional | optional |
| nameReference | prohibited | prohibited | optional | prohibited | optional |
| min/max | optional | optional (irrelevant) | required | optional | optionalÃ¢â‚¬Â |
| defaultValue[x] | optional | prohibited | optional | prohibited | optionalÃ¢â‚¬Â |
| meaningWhenMissing | optional | prohibited | optional | prohibited | optionalÃ¢â‚¬Â |
| fixed[x] | prohibited | prohibited | prohibited | prohibited | optional |
| pattern[x] | optional | prohibited | prohibited | prohibited | optional |
| example[x] | optional | prohibited | optional | prohibited | optional |
| minValue[x] | optional | prohibited | prohibited | prohibited | optional |
| maxValue[x] | optional | prohibited | prohibited | prohibited | optional |
| maxLength | optional | prohibited | prohibited | prohibited | optional |
| mustSupport | optional | prohibited | prohibited | optional | optional |
| isModifier | prohibited | prohibited | optional | prohibited | optionalÃ¢â‚¬Â |
| isSummary | prohibited | prohibited | optional | prohibited | optionalÃ¢â‚¬Â |
| binding | optional | prohibited | optional | prohibited | optional |
| constraint | optional | optional | optional | optionalÃ¢Ë†â€ | optionalÃ¢Ë†â€ |
| condition | optional | prohibited | optional | prohibited | optionalÃ¢Ë†â€ |
| mapping | optional | optional | optional | optionalÃ¢Ë†â€ | optionalÃ¢Ë†â€ |

Notes:

* Type definition: A StructureDefinition without a base element, or where the derivation type is 'specialization'
* Constraint definition: A StructureDefinition with a base element and a derivation of 'constraint' - e.g. a definition of a structure that constrains another base structure, referring to the differential portion
* Ã¢â‚¬Â: The element's presence, and value, must match the definition in the base structure
* Ã¢â‚¬Â¡: The element content must be consistent with that matching element in the base structure
* Ã¢Ë†â€: Additional constraints and mappings can be defined, but they do not replace the ones in the base structure

The use of Path and type depends more deeply on the context where the ElementDefinition is used:

|  |  |  |  |
| --- | --- | --- | --- |
| **Context** | **path (1st element)** | **path (following elements)** | **type (1st element)** |
| Data Element | Any token | Any token (distinct in the DataElement) | Any type |
| Base definition of a data type  (example: [Quantity](file:///C:\temp\datatypes.html#Quantity) - [XML](file:///C:\temp\quantity.profile.xml.html), [JSON](file:///C:\temp\quantity.profile.json.html)) | Name of the type | Path inside the datatype | Element |
| A constrained data type  (example: [Money](file:///C:\temp\datatypes.html#Money) - [XML](file:///C:\temp\money.profile.xml.html), [JSON](file:///C:\temp\money.profile.json.html)) | Name of the base type | Path inside the datatype | Name of the base type |
| Base definition of a resource  (example: [Patient](file:///C:\temp\patient.html) - [XML](file:///C:\temp\patient.profile.xml.html), [JSON](file:///C:\temp\patient.profile.json.html)) | The name of the resource | Path inside the resource | DomainResource or sometimes Resource |
| Constraint on a resource  (example: [DAF Patient](file:///C:\temp\vitalsigns.html) - [XML](file:///C:\temp\vitalsigns.profile.xml.html), [JSON](file:///C:\temp\vitalsigns.profile.json.html)) | The name of the resource | Path inside the resource  (including into the data types) | The name of the resource |
| Base Extension (a standard data type)  (example: [Extension](file:///C:\temp\extensibility.html#Extension) - [XML](file:///C:\temp\extension.profile.xml.html), [JSON](file:///C:\temp\extension.profile.json.html)) | Extension | Extension.value[x] or Extension.extension | Extension |
| A defined Extension  (example: [Extension](file:///C:\temp\extension-geolocation.html) - [XML](file:///C:\temp\extension-geolocation.xml.html), [JSON](file:///C:\temp\extension-geolocation.json.html)) | Extension | Extension.value[x] or Extension.extension (for complex extensions) | Extension |

There are additional notes about the use of ElementDefinition when defining Extensions on the [Defining Extensions](file:///C:\temp\defining-extensions.html#ed) page.

**Rules about Slicing**

For a description of slicing, see [Slicing](file:///C:\temp\profiling.html#slicing)

* Slicing is only allowed when constraining an existing structure
* slicing can only be used on the first repeat of an element, this element is considered the slicing entry
* All elements following the first repeat that has a slicing SHALL have a sliceName
* The first entry (the one having the slicing information) is understood to be the set of constraints to be used for the "open" slice. i.e. when the slice is open, this definition is used as a constraint on elements in the instance that are not part of a slice and belong to the open portion of the slice. Its use follows the "normal case", except:
  + slicing must be present
  + min and max govern the number of total occurrences of the sliced element including the number of occurrences in the open portion of the slice.
* An element with a cardinality of 0..1 and a choice of multiple types can be sliced by type. This is in order to specify different constraints for different types. In this case, the discriminator SHALL be "@type"

**Rules about min and max**

In a [DataElement](file:///C:\temp\dataelement.html), min and max are always required.

In a [StructureDefinition](file:///C:\temp\structuredefinition.html):

* **If there is no StructureDefinition.base**: min and max are always required
* Else, in StructureDefinition.differential: min and max are always optional; if they are not present, they default to the base min and max
* and in StructureDefinition.snapshot: min and max are always required

**Missing Elements**

Most elements have a minimum cardinality of 0, which means that they may be missing from a resource when it is exchanged between systems. Generally, when an element is missing, all that an application processing the resource can say about the element is that the value is unknown - it may have a correct value, but it has not been provided for security or workflow reasons. On the other hand, it may not have a value at all. All the application can say is that the value is unknown.

However, for some elements, this specification makes specific rules about what it means if the element is missing. For some elements, this takes the form of a default value, a value that should be assumed if on element is present. For example, here are some default values:

|  |  |
| --- | --- |
| **Element** | **Default Value** |
| ElementDefinition.mustSupport | false |
| SampledData.factor | 1 |
| ValueSet.codeSystem.concept.abstract | false |
| List.entry.deleted | false |

Most default values are mathematical / numerical non-operations e.g. false, 0 for addition or 1 for multiplication. Default values are rarely used because:

* The value must be known by all implementations
* When an element has a default value, it can never be unknown - e.g. it is implicitly mandatory
* The default value can never be changed.

For other elements, the resource explicitly documents what the meaning of a missing element is, and the meaning is not represented by a default value. Some examples:

|  |  |
| --- | --- |
| **Element** | **Meaning when missing** |
| Period.end | If the end of the period is missing, it means that the period is ongoing |
| Quantity.comparator | If there is no comparator, then there is no modification of the value |
| MedicationAdministration.wasNotGiven | If this is missing, then the medication was administered |
| Substance.instance | If this element is not present, then the substance resource describes a kind of substance |

Like default values, and for the same reasons, assigning a meaning when an element is missing is something that is not done very often. Note that Profiles cannot change the default value or missing meaning of an element, since it can't be done in the base specification either.

Here is a list of all elements with a default value or a missing meaning:

**extensibility-definitions.html**

**Extension - Detailed Descriptions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**extensibility-examples.html**

**Extensibility Examples**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

In order to use an extension, there is a three step process:

1. Define the extension
2. Register the extension
3. Use it in the instance

This page contains examples of how this process executes.

**Patient Consent for Record Sharing**

The basic patient resource contains no information relating to patient consent, and/or the policy under which the patient consents to their registration details. A social web provider of personal healthcare record (PHR) services might be obliged to keep track of the particular policy under which a patient has created their relationship with the PHR provider, and share this with their participants via their FHIR API. If they wish, they can extend the patient resource to represent the patient's participation agreement. Note that other approaches to this problem are possible, but it suffices to demonstrate the extension process.

For the purposes of this example, we assume that the patient agrees to a participation policy as part of their sign up, and that as the provider has to change their policy, they ask patients to agree to new participation details. Each participation agreement has a URI by which it is identified, and the patient resource will carry this URI for each policy agreement that the patient has agreed to.

**Define the Extension**

For each extension, the first thing to do is to fill out the [definitional properties of the extension](file:///C:\temp\extensibility.html#define):

|  |  |
| --- | --- |
| Code | "participation-agreement" |
| Context | This extension is used in the patient resource |
| Short Defn | Agreed agreement/policy |
| Definition | A URI that identifies a participation agreement/policy to which the patient has agreed |
| Comment | URI is a literal reference to agreement text (html) |
| Cardinality | 1..\* (patient cannot participate without at least one agreement) |
| Type | uri |
| XPaths | no Invariants |
| is Modifier | No (The participation agreements do not affect that interpretation of the elements of the patient, though they will likely influence how the system interacts with the patient) |
| Binding | (No binding - not a coded value) |

**Write the Definition of the Extension**

From this table, we can build a formal extension definition. In this case, it looks like this:

<StructureDefinition xmlns="http://hl7.org/fhir">

<url value="http://example.org/fhir/StructureDefinition/participation-agreement" />

<name value="Example Extension Definition" />

<!-- snip other metadata -->

<kind value="datatype" />

<contextType value="resource" />

<context value="Patient" />

<type value="Extension" />

<baseDefinition value="http://hl7.org.fhir/StructureDefinition/Extension" />

<derivation value="constraint" />

<differential>

<element>

<path value="Extension"/>

<short value="Agreed agreement/policy" />

<definition value="A URI that identifies a participation agreement/policy

to which the patient has agreed" />

<comment value="URI is a literal reference to agreement text (html).

Systems SHALL conform to the policies as indicated.

For further information, see the partnership agreement..." />

<mustSupport value="true" />

<isModifier value="false" />

</element>

<element>

<path value="Extension.valueUri"/>

<short value="The URI value" />

<min value="1" />

<max value="\*" />

<type>

<code value="uri" />

</type>

<mustSupport value="true" />

</element>

</differential>

</StructureDefinition>

Note that usually you would build the actual profile using some tool. This example was built by hand for this example.

**Register the Extension**

This means the profile shown above that defines the extension is placed on the web somewhere. By preference, it will be hosted in a FHIR Profile endpoint, and the best location of all is the HL7 Profile registry (yet to be implemented).

For this example, we assume that it has been uploaded to the PHR provider's own website at http://example.org/phr/documents/fhir/extensions.

**Use it in the instance**

To use the extension in an instance, the extension is placed in the root of the resource. Note that the URL of the extension refers to the registered location, with the id of the extension as a fragment identifier.

<Patient xmlns="http://hl7.org/fhir">

<extension url="http://example.org/phr/documents/fhir/extensions/participation-agreement" >

<valueUri value="http://example.org/phr/documents/patient/general/v1" />

</extension>

<!-- ... -->

</Patient>

**Using it in a Patient Profile**

The profile definition above simply defines the extension "participation-agreement", and says that it is used with patient. But the profile above doesn't say that the server actually uses it. For the PHR provider to indicate that all the patients resources will use this resource, a StructureDefinition on the patient resource is used:

<StructureDefinition xmlns="http://hl7.org/fhir">

<id value="patient-profile" />

<!-- snip other metadata, narrative -->

<differential>

<!-- first, the patient root element

- can be copy/paste from the base patient profile -->

<element>

<path value="Patient"/>

<!-- snip definition -->

</element>

<!-- now, the general definition for extensions

- can be copy/paste from the base patient profile,

with changes for slicing -->

<element>

<path value="Patient.extension"/>

<!-- we're going to slice the extension element, and

one of the extensions is one we have defined -->

<slicing>

<!-- extension is always sliced on url -->

<discriminator>

<type value="value"/>

<path value="url"/>

</discriminator>

<!-- we don't care what the order of any extensions is -->

<ordered value="false"/>

<!-- Other extensions are allowed in addition to this one -->

<rules value="open"/>

</slicing>

<!-- snip definition -->

</element>

<!-- now, the slice that contains our extension -->

<element>

<path value="Patient.extension"/>

<sliceName value="base-extension"/>

<!-- clone information from the extension definition.

duplicative, but this duplication makes it over all simpler -->

<short value="Agreed agreement/policy"/>

<definition value="A URI that identifies a participation agreement/policy

to which the patient has agreed"/>

<!-- min has to be 1, since the extension itself has min = 1 -->

<min value="1"/>

<max value="\*"/>

<type>

<!-- obviously it has to be an extension -->

<code value="Extension"/>

<!-- and here is the link to the extension definition:

this extension has to conform to the rules laid down in its definition -->

<profile value="http://example.org/phr/documents/fhir/extensions#participation-agreement"/>

</type>

<isModifier value="false"/>

</element>

<!-- snip the rest of the profile -->

</differential>

</StructureDefinition>

Note - this step is optional.

**Patient Name Parts**

ISO 21090 (Healthcare Data Types) defines a concept called a "name part qualifier" that contains extra information about how a particular name part should be used or interpreted. In practice, this field is used rarely, except in particular cultural contexts, where certain part qualifiers are used as a matter of practice. Following the [FHIR design policy](file:///C:\temp\extensibility.html), such a field is not included in the overall definition of the core name data type, instead is it added as an extension.

In practice, for cases such as these in ISO 21090, HL7 provides common extensions, and these are defined at [location still to be finalized].

**Define the Extension**

For each extension, the first thing to do is to fill out the [definitional properties of the extension](file:///C:\temp\extensibility.html#define):

|  |  |
| --- | --- |
| Code | "name-qualifier" |
| Context | This extension can be used anywhere a HumanName.part appears |
| Short Defn | (one of the codes) LS | AC | NB | PR | HON | BR | AD | SP | MID | CL | IN |
| Definition | A set of codes each of which specifies a certain subcategory of the name part in addition to the main name part type |
| Comment | Used to indicate additional information about the name part and how it should be used |
| Cardinality | 0..\* (this is always optional, but more than one can be used if required) |
| Type | code |
| XPaths | N/A |
| Is Modifier | No (Qualifiers do not change the fact that the part is a given or family name) |
| RIM Mapping | ENXP.qualifier |
| v2 Mapping | N/A |
| Binding | Bound to a subset of the codes specified for [EntityNamePartQualifierR2 in ISO 21090](http://www.hl7.org/v3ballot/html/infrastructure/vocabulary/EntityNamePartQualifierR2.html) |

Not all the codes of the EntityNamePartQualifierR2 are required in this context, because prefix and suffix are explicitly part of the name types. Rather than simply refer to the OID for EntityNamePartQualifierR2 (2.16.840.1.113883.5.1122), in this case we enumerate the available codes, and set the type of the extension to code. The type of "code" is only allowed if the profile itself defines the codes that can be used. Here is a table of the codes (see the [EntityNamePartQualifierR2](http://www.hl7.org/v3ballot/html/infrastructure/vocabulary/EntityNamePartQualifierR2.html) reference for the full definitions):

|  |  |  |
| --- | --- | --- |
| LS | Legal status | For organizations a suffix... |
| AC | Academic | Indicates that a prefix like "D... |
| NB | Nobility | In Europe and Asia, there are s... |
| PR | Professional | Primarily in the British Im... |
| HON | Honorific | An honorific such as 'The Rig... |
| BR | Birth | A name that a person was given at ... |
| AD | Acquired | A name part a person acquired. ... |
| SP | Spouse | The name assumed from the partner... |
| MID | Middle Name | Indicates that the name par... |
| CL | Call me | Callme is used to indicate which... |
| IN | Initial | Indicates that a name part is ju... |

This is all then represented formally in a profile. Such profiles do not need to include constraint statements of resources; instead, they include just extension declarations and their associated bindings. In this case, the definition looks like this:

<StructureDefinition xmlns="http://hl7.org/fhir">

<url value="http://hl7.org/fhir/StructureDefinition/iso21090-EN-qualifier" />

<name value="Iso21090NameQualifier" />

<title value="iso-21090 Name Qualifier" />

<!-- snip other metadata, including definition of RIM Mapping -->

<kind value="datatype" />

<contextType value="datatype" />

<context value="HumanName.given" />

<context value="HumanName.prefix" />

<context value="HumanName.family" />

<context value="HumanName.suffix" />

<type value="Extension" />

<baseDefinition value="http://hl7.org.fhir/StructureDefinition/Extension" />

<derivation value="constraint" />

<differential>

<element>

<path value="Extension"/>

<short value="LS | AC | NB | PR | HON | BR | AD | SP | MID | CL | IN" />

<definition value="A set of codes each of which specifies a certain subcategory

of the name part in addition to the main name part type" />

<comment value="Used to indicate additional information about the

name part and how it should be used" />

<mustSupport value="false" />

<isModifier value="false" />

</element>

<element>

<path value="Extension.value"/>

<short value="LS | AC | NB | PR | HON | BR | AD | SP | MID | CL | IN" />

<min value="0" />

<max value="\*" />

<type>

<code value="code" />

</type>

<binding>

<strength value="required" />

<description value="A set of codes each of which specifies a certain subcategory

of the name part in addition to the main name part type" />

<valueSetReference>

<reference value="ValueSet/name-part-qualifier" />

</valueSetReference>

</binding>

<mapping>

<identity value="RIM" />

<map value="ENXP.qualifier" />

</mapping>

</element>

</differential>

</StructureDefinition>

Note that usually you would build the actual profile using some tool. This example was built from a spreadsheet definition by the FHIR build tooling.

**Register the Extension**

For this example, it is registered at <http://hl7.org/fhir/StructureDefinition/iso21090-EN-qualifier>. This is the URL that will appear in the definition element when the extension is used.

**Use it in the instance**

To use the extension in an instance, the extension is nested within the attribute that is extended. Note that the URL of the extension refers to the registered location, with the id of the extension as a fragment identifier.

<name>

<use value="official" />

<given value="Ãƒâ€“stlund">

<extension url="http://hl7.org/fhir/StructureDefinition/iso21090-EN-qualifier" >

<valueCode value="MID" />

</extension>

</given>

</name>

This particular example is a Scandinavian mellannamn. See [Datatypes examples for additional examples](file:///C:\temp\datatypes-examples.html).

**Complex Extension: Patient Clinical Trial**

Defining complex extensions is a little different. They have the same meta data - context etc., but differ in the internal structure. As an example, consider enrolling a patient in a clinical trial. There are 3 data items to collect:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Code** | **Name** | **Cardinality** | **Type** | **Content** |
| NCT | Clinical Trial number | 1..1 | string | The format for the US ClinicalTrials.gov registry number is "NCT" followed by an 8-digit number, e.g.: NCT00000419 |
| period | trialPeriod | 0..1 | Period | The start and end times of the participation of this patient in the clinical trial |
| reason | reason enrolled | 0..1 | CodableConcept | indication or reason the patient is part of this trial |

Like simple extensions, the first thing to do is to assign a URI to the extension. Since this one is part of the FHIR publication, it is assigned a URI where it will be published: http://hl7.org/fhir/StructureDefinition/patient-clinicalTrial.

Internally, in the extension, the cardinality of the value[x] element is set to 0, since it will not (and cannot) be used. Then, the Extension.extension element is sliced by url, and 3 slices are defined, each with a fixed relative URI which is the code from the table above. Here's the relevant parts of the definition of this extension:

<StructureDefinition xmlns="http://hl7.org/fhir">

<!-- metadata - setting up the base definition -->

<url value="http://hl7.org/fhir/StructureDefinition/patient-clinicalTrial"/>

<name value="ClinicalTrialParticipation"/>

<title value="The patient's participation in clinical trials"/>

<contextType value="resource"/>

<context value="Patient"/>

<type value="Extension" />

<baseDefinition value="http://hl7.org.fhir/StructureDefinition/Extension" />

<derivation value="constraint" />

<snapshot>

<element>

<path value="Extension"/>

<!-- etc.-->

</element>

<!-- set up the slicing -->

<element>

<path value="Extension.extension"/>

<slicing>

<discriminator>

<type value="value"/>

<path value="url"/>

</discriminator>

<ordered value="true"/>

<rules value="openAtEnd"/>

</slicing>

</element>

<!-- first slice, NCT -->

<element>

<path value="Extension.extension"/>

<sliceName value="NCT"/>

<short value="National Clinical Trial number"/>

<min value="1"/>

<max value="1"/>

</element>

<element>

<path value="Extension.extension.extension"/>

<min value="0"/>

<max value="0"/> <!-- not allowed to be used -->

</element>

<element>

<path value="Extension.extension.url"/>

<min value="1"/>

<max value="1"/>

<fixedUri value="NCT"/>

</element>

<element>

<path value="Extension.extension.valueString"/>

<min value="1"/>

<max value="1"/>

<type>

<code value="string"/>

</type>

</element>

<!-- second slice, period -->

<element>

<path value="Extension.extension"/>

<sliceName value="period"/>

<short value="The period of participation in the clinical trial"/>

<min value="0"/>

<max value="1"/>

</element>

</element>

<element>

<path value="Extension.extension.extension"/>

<min value="0"/>

<max value="0"/>

</element>

<element>

<path value="Extension.extension.url"/>

<min value="1"/>

<fixedUri value="period"/>

</element>

<element>

<path value="Extension.extension.valuePeriod"/>

<type>

<code value="Period"/>

</type>

</element>

<!-- third slice, reason -->

<element>

<path value="Extension.extension"/>

<sliceName value="reason"/>

<short value="The reason for participation in the clinical trial"/>

<min value="0"/>

<max value="1"/>

</element>

<element>

<path value="Extension.extension.extension"/>

<min value="0"/>

<max value="0"/>

</element>

<element>

<path value="Extension.extension.url"/>

<min value="1"/>

<fixedUri value="reason"/>

</element>

<element>

<path value="Extension.extension.valueCodeableConcept"/>

<type>

<code value="CodeableConcept"/>

</type>

</element>

<!-- last (for order reasons): the fixed URI -->

<element>

<path value="Extension.url"/>

<fixedUri value="http://hl7.org/fhir/StructureDefinition/patient-clinicalTrial"/>

</element>

<!-- and no value in the root -->

<element>

<path value="Extension.value[x]"/>

<min value="0"/>

<max value="0"/>

</element>

</snapshot>

</StructureDefinition>

**Use it in the instance**

With complex extensions, only the first URL is an absolute URL:

<Patient xmlns="http://hl7.org/fhir">

<extension url="http://hl7.org/fhir/StructureDefinition/patient-clinicalTrial" >

<extension url="NCT" >

<valueString value="NCT00000419" />

</extension>

<extension url="period" >

<valuePeriod>

<start value="200140105" />

<end value="20120105" />

</valuePeriod>

</extension>

<extension url="reason" >

<valueCodeableConcept>

<text value="NCT00000419" />

</valueCodeableConcept>

</extension>

</extension>

<!-- ... -->

</Patient>

**Using it in a Patient Profile**

This is the same as for a simple extension:

<StructureDefinition xmlns="http://hl7.org/fhir">

<id value="patient-profile" />

<!-- snip other metadata, narrative -->

<differential>

<!-- first, the patient root element

- can be copy/paste from the base patient profile -->

<element>

<path value="Patient"/>

<!-- snip definition -->

</element>

<!-- now, the general definition for extensions

- can be copy/paste from the base patient profile,

with changes for slicing -->

<element>

<path value="Patient.extension"/>

<!-- we're going to slice the extension element, and

one of the extensions is one we have defined -->

<slicing>

<!-- extension is always sliced on url -->

<discriminator>

<type value="value"/>

<path value="url"/>

</discriminator>

<!-- we don't care what the order of any extensions is -->

<ordered value="false"/>

<!-- Other extensions are allowed in addition to this one -->

<rules value="open"/>

</slicing>

<!-- snip definition -->

</element>

<!-- now, the slice that contains our extension -->

<element>

<max value="1"/>

<type>

<!-- obviously it has to be an extension -->

<code value="Extension"/>

<!-- and here is the link to the extension definition:

this extension has to conform to the rules laid down in its definition -->

<profile value="http://hl7.org/fhir/StructureDefinition/patient-clinicalTrial"/>

</type>

</element>

<!-- snip the rest of the profile -->

</differential>

</StructureDefinition>

**extensibility-registry.html**

**Extension Registry**

All extensions defined under http://hl7.org/fhir/StructureDefinition/. Additional extensions may be registered on the HL7 FHIR registry at <http://hl7.org/fhir/registry>.

**extensibility.html**

**Extensibility**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 3 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This exchange specification is based on generally agreed common requirements across healthcare - covering many jurisdictions, domains, and different functional approaches. It is common for specific implementations to have valid requirements that are not part of these agreed common requirements. Incorporating all of these requirements would make this specification very cumbersome and difficult to implement. Instead, this specification expects that these additional distinct requirements will be implemented as extensions.

As such, extensibility is a fundamental part of the design of this specification. Every element in a resource may have extension child elements to represent additional information that is not part of the basic definition of the resource. Applications should not reject resources merely because they contain extensions, though they may need to reject resources because of the specific contents of the extensions.

Note that, unlike in many other specifications, there can be no stigma associated with the use of extensions by any application, project, or standard - regardless of the institution or jurisdiction that uses or defines the extensions. The use of extensions is what allows the FHIR specification to retain a core simplicity for everyone.

In order to make the use of extensions safe and manageable, there is a strict governance applied to the definition and use of extensions. Though any implementer is allowed to define and use extensions, there is a set of requirements that must be met as part of their use and definition.

**Extension Element**

Every element in a resource or data type includes an optional "extension" child element that may be present any number of times. This is the content model of the extension as it appears in each resource:

Notes:

* The url is a mandatory attribute / property and identifies a retrievable [extension definition](file:///C:\temp\structuredefinition.html) that defines the content and meaning of the extension
* The url SHALL be a URL, not a URN (e.g. not an OID or a UUID), and it SHALL be the canonical URL of a StructureDefinition that defines the extension. Except for child extensions defined within a complex extensions, the URL SHALL be an absolute URL). The StructureDefinition that defines the extension SHOULD be available at the identified location so that resource processors can access the definition from the URL
* An extension SHALL have either a value (i.e. a value[x] element) or sub-extensions, but not both. If present, the value[x] element SHALL have content (value attribute or other elements)
* If it is not safe for an application processing the content of the resource to ignore the extension it must be represented differently, using [a Modifier Extension](file:///C:\temp\main-pages.html#isModifier)
* The *value[x]* element has an actual name of "value" and then the TitleCased name of one of these defined types, and its contents are those as defined for that type:
  + valueInteger : [integer](file:///C:\temp\datatypes.html#integer)
  + valueUnsignedInt : [unsignedInt](file:///C:\temp\datatypes.html#unsignedInt)
  + valuePositiveInt : [positiveInt](file:///C:\temp\datatypes.html#positiveInt)
  + valueDecimal : [decimal](file:///C:\temp\datatypes.html#decimal)
  + valueDateTime : [dateTime](file:///C:\temp\datatypes.html#dateTime)
  + valueDate : [date](file:///C:\temp\datatypes.html#date)
  + valueTime : [time](file:///C:\temp\datatypes.html#time)
  + valueInstant : [instant](file:///C:\temp\datatypes.html#instant)
  + valueString : [string](file:///C:\temp\datatypes.html#string)
  + valueUri : [uri](file:///C:\temp\datatypes.html#uri)
  + valueOid : [oid](file:///C:\temp\datatypes.html#oid)
  + valueUuid : [uuid](file:///C:\temp\datatypes.html#uuid)
  + valueId : [id](file:///C:\temp\datatypes.html#id)
  + valueBoolean : [boolean](file:///C:\temp\datatypes.html#boolean)
  + valueCode : [code](file:///C:\temp\datatypes.html#code) (only if the extension definition provides a [fixed](file:///C:\temp\terminologies.html#code) binding to a suitable set of codes)
  + valueMarkdown : [markdown](file:///C:\temp\datatypes.html#markdown)
  + valueBase64Binary : [base64Binary](file:///C:\temp\datatypes.html#base64Binary)
  + valueCoding : [Coding](file:///C:\temp\datatypes.html#Coding)
  + valueCodeableConcept : [CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept)
  + valueAttachment : [Attachment](file:///C:\temp\datatypes.html#Attachment)
  + valueIdentifier : [Identifier](file:///C:\temp\datatypes.html#Identifier)
  + valueQuantity : [Quantity](file:///C:\temp\datatypes.html#Quantity)
  + valueSampledData : [SampledData](file:///C:\temp\datatypes.html#SampledData)
  + valueRange : [Range](file:///C:\temp\datatypes.html#Range)
  + valuePeriod : [Period](file:///C:\temp\datatypes.html#Period)
  + valueRatio : [Ratio](file:///C:\temp\datatypes.html#Ratio)
  + valueHumanName : [HumanName](file:///C:\temp\datatypes.html#HumanName)
  + valueAddress : [Address](file:///C:\temp\datatypes.html#Address)
  + valueContactPoint : [ContactPoint](file:///C:\temp\datatypes.html#ContactPoint)
  + valueTiming : [Timing](file:///C:\temp\datatypes.html#Timing)
  + valueReference : [Reference](file:///C:\temp\references.html#Reference) - a reference to another resource
  + valueAnnotation : [Annotation](file:///C:\temp\datatypes.html#Annotation)
  + valueSignature : [Signature](file:///C:\temp\datatypes.html#Signature)
  + valueMeta : A [Meta](file:///C:\temp\resource.html#Meta) heading from a resource

Here is an example of an extension in XML:

<name>

<extension url="http://hl7.org/fhir/StructureDefinition/iso-21090-name-use" >

<valueCode value="I" />

</extension>

<text value="Chief Red Cloud"/>

</name>

In this example, the name with text = "Chief Red Cloud" is extended to have a name use code of "Indigenous" (defined in ISO 21090, but very rarely used in practice).

In JSON, extensions are represented similarly:

{

"extension" : [{

"url" : "http://hl7.org/fhir/StructureDefinition/iso-21090-name-use",

"valueCode" : "I"

}],

"text" : "Chief Red Cloud"

}

Making the types explicit in the representation means that all systems can read and write (and therefore store and/or exchange) extensions correctly without needing to access the definition of the extension.

Note that the JSON represenation for extensions on primitive data types is handled differently. See [Representing primitive types in JSON](file:///C:\temp\json.html#primitive) for further information.

Extensions can also contain extensions, either because the extension definition itself defines complex content - that is, a nested tree of values in the extension, or because the extension is extended with an additional extension defined separately.

In the first case, the identity of the parts of the extension are local/relative to the reference to the extension definition.

As an example, consider extending a patient with an opt-in status for a clinical trial, with 3 fields: clinical trial number, period of involvement, and a reason for enrollment. In XML:

<Patient>

<extension url="http://hl7.org/fhir/StructureDefinition/patient-clinicalTrial" >

<extension url="NCT" >

<valueString value="123456789" />

</extension>

<extension url="period" >

<valuePeriod>

<start value="2009-03-14" />

</valuePeriod>

</extension>

<extension url="reason" >

<valueCodeableConcept>

<coding>

<system value="http://acme.org/codes/general" />

<code value="tt14j" />

</coding>

</valueCodeableConcept>

</extension>

</extension>

<!-- other data for patient -->

</Patient>

Or in JSON:

{

"resourceType" : "Patient",

"extension" : [{

"url" : "http://hl7.org/fhir/StructureDefinition/patient-clinicalTrial",

"extension" : [{

"url" : "NCT",

"valueString" : "123456789"

}, {

"url" : "period",

"valuePeriod" : {

"start" : "2009-03-14"

}

}, {

"url" : "reason",

"valueCodeableConcept" : {

"coding" : [{

"system" : "http://acme.org/codes/general",

"code" : "tt14j"

}]

}

}]

}]

}

As the URL suggests, [this extension is defined as part of this specification](file:///C:\temp\extension-patient-clinicaltrial.html).

This can be extended again, by adding a "registrar" extension:

The registrar is defined as a separate extension (e.g. by an implementing organization), rather than part of the official clinical-trial extension. The URL of the extension is different:

<Patient>

<extension url="http://hl7.org/fhir/StructureDefinition/patient-clinicalTrial" >

<extension url="NCT" >

<valueString value="123456789" />

</extension>

<extension url="period" >

<valuePeriod>

<start value="2009-03-14" />

</valuePeriod>

</extension>

<extension url="reason" >

<valueCodeableConcept>

<coding>

<system value="http://acme.org/codes/general" />

<code value="tt14j" />

</coding>

</valueCodeableConcept>

</extension>

</extension>

<extension url="http://acme.org/fhir/StructureDefinition/registrar" >

<valueReference>

<reference value="Practitioner/example" />

</valueReference>

</extension>

<!-- other data for patient -->

</Patient>

or in JSON:

{

"resourceType": "Patient",

"extension": [{

"url": "http://hl7.org/fhir/StructureDefinition/patient-clinicalTrial",

"extension": [{

"url": "NCT",

"valueString": "123456789"

},

{

"url": "period",

"valuePeriod": {

"start": "2009-03-14"

}

},

{

"url": "reason",

"valueCodeableConcept": {

"coding": [{

"system": "http://acme.org/codes/general",

"code": "tt14j"

}]

}

},

{

"url": "http://acme.org/fhir/StructureDefinition/registrar",

"valueReference": {

"reference": "Practitioner/example"

}

}]

}]

}

**Modifier Extensions**

There are some cases where the information provided in an extension modifies the meaning of the element that contains it. Typically, this means information that qualifies or negates the primary meaning of the element that contains it. Some examples:

* An anti-prescription: recording an instruction **not** to take a medication
* Using the [Condition](file:///C:\temp\condition.html) resource to record an assertion that a patient has a family history of the condition rather than the condition itself
* Asserting that a performer was **not** actually involved in a [Procedure](file:///C:\temp\procedure.html)
* Asserting an additional subsumption relationship on a concept in a [value set](file:///C:\temp\valueset.html)

Implementers should avoid the use of modifier extensions where possible. Any use should be carefully considered against its possible downstream consequences. However, implementers are often forced into these situations by the business arrangements around the use of resources, so this specification creates a framework for handling such cases. If modifier extensions are present, an application cannot safely process the resource unless it knows what the extension means for its own use of the data.

This specification allows for such modifier elements to be included at the base of a resource or in any elements that do not have a data type (e.g. the elements that correspond to classes in the resource UML diagrams). Elements that are data types, or that are inside data types SHALL NOT have modifier extensions, nor can extensions have modifier extensions internally.

Note that complex extensions are allowed to have elements in the complex extension that are marked is [modifier = true](file:///C:\temp\conformance-rules.html#isModifier), which mean that these elements modify the extension value itself.

Extensions marked "is modifier" are still represented using the "extension" element , not "modifierExtension" because the impact of the modifier element is expected to be known by applications that understand the containing extension.

Any modifierExtension defined outside an extension cannot be placed in another extension.

**STU Note:** Is this a problem? Input from implementers identiying places where modifier extensions might be needed inside other extensions is welcome [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

In XML, these modifier elements are represented using an element named "modifierExtension", which has same content as the *extension* element documented above:

Example: There's no element on [MedicationRequest](file:///C:\temp\medicationrequest.html) to write an "anti-prescription" - an instruction not to take a medication for a particular time. Classical clinical recording systems do not record this as a prescription - but one particular system does, and these "anti-prescription" records need to be shared within the institution where this happens, as they are an important part of the workflow. Hence, applications are allowed to extend a resource with data like this:

<MedicationRequest>

<modifierExtension url="http://example.org/fhir/StructureDefinition/anti-prescription">

<valueBoolean value="true"/>

</modifierExtension>

<!-- ... other content ... -->

</MedicationRequest>

Or in JSON:

{

"resourceType" : "MedicationRequest",

"modifierExtension" : [{

"url" : "http://example.org/fhir/StructureDefinition/anti-prescription",

"valueBoolean" : "true"

}],

.. other content ...

}

Implementations processing the data in resources SHALL check for modifiers anywhere they may appear, and if a modifier extension is present, SHALL do one of these things:

1. understand the impact of the extension when using the data
2. refuse to process the data
3. carry a warning concerning the data along with any action or output that results from processing the data to inform users that it has not fully understood the source information

Processing the data of a resource typically means copying or filtering data out of a resource for use in another context (display to a human, decision support, exchange in another format where not all information is included, or storing it for this kind of use). Servers and background processes that simply move whole resources around unchanged are not "processing the data of the resource", and therefore these applications are not required to check for unknown modifier extensions.

**#1**: When an application understands this extension, it means that some developer has provided appropriate instructions for what to do with the data contained in it because of the existence of the modifier extension.

**#2**: This means that implementations are not inherently required to "support" a modifier extension in any meaningful way - they may achieve this understanding by rejecting instances that contain this extension (a server, for instance, could return a HTTP 422 status code with an [OperationOutcome](file:///C:\temp\operationoutcome.html) if a client PUTs or POSTs a modifier extension it does not know. Applications may also be able to ignore a modifier extension if they can know that this is safe to do in its own context, though this would not usually be the case.

Note that implementations may be able to be sure, due to their implementation environment (e.g. specific trading partner agreement), that modifier extensions will never occur, and can therefore meet the requirement to check for modifiers at the design stage. However, since integration and deployment options often change subsequently, applications SHOULD always check for modifier extensions when processing resources anyway.

**#3**: One way to warn the user is to download the extension definition from the given URL, and then use the defined display name to present the extension to the user. An error message could look something like this:

Note that the narrative of the resource SHALL contain this qualifying information, so it is safe to show this to the user as an expression of the resource's content. A warning dialog box could be extended to offer the user the choice to see the original narrative.

Here is the prescription example from above with narrative:

<MedicationRequest xmlns="http://hl7.org/fhir">

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml">

<p><b>Note: This prescription is an instruction NOT to take a medication</b></p>

<!-- snip actual narrative -->

</div>

</text>

<!-- ...data... -->

<modifierExtension url="http://example.org/fhir/StructureDefinition/anti-prescription">

<valueBoolean value="true"/>

</modifierExtension>

<!-- ...data... -->

</MedicationRequest>

An application only needs to concern itself with modifierExtensions on elements that it processes. Take, for example, the case above where a procedure resource has a modifierExtension on one of the performer elements indicating that they did not participate in the procedure. In this case, if an application is not using the performer details at all, the fact that one of the performers has a modifierExtension is irrelevant and the application is free to ignore it. If it does process the performers, and it sees the modifier extension, it must act in one of the ways outlined above.

**Summary: Conformance Rules for Modifier Extensions**

* Modifier Extensions SHALL only modify the element which contains it and/or that element's children
* It SHALL always be safe to show the narrative to humans; any modifier extension SHALL be represented in the narrative
* Applications SHALL always check for modifier extensions when processing the data from any element that may carry one
* If a modifier Extension they do not understand is present, the application SHALL either refuse to process the resource or affected element, or provide an appropriate warning to its users

**Special Case: Missing data**

In some cases, implementers may find that they do not have appropriate data for an element with minimum cardinality = 1. In this case, the element must be present, but unless the resource or a profile on it has made the actual value of the primitive data type mandatory, it is possible to provide an extension that explains why the primitive value is not present:

<uri>

<extension url="http://hl7.org/fhir/StructureDefinition/data-absent-reason">

<valueCode="unknown"/>

</extension>

</uri>

In this example, instead of a value, a [data missing code](file:///C:\temp\general-extensions.html) is provided. Note that it is not required that this particular extension be used. This extension is **not** a modifier extension, because the primitive data type has no value.

It is not valid to create a fictional piece of data for the primitive value, and then to add an extension indicating that the data has been constructed to meet the data rules. This would be both a bad idea, and also a modifier extension, which is not allowed on data types.

**Exchanging Extensions**

Extensions are a way of allowing local requirements to be reflected in a resource using a common information based approach so that all systems can confidently process them using the same tools. However, when it comes to processing the information, applications will be constrained in their ability to handle extensions by the degree to which they are informed about them.

While the structured definition of an extension should always be available (see below for details), the mere availability of a definition does not automatically mean that applications know how to handle them correctly - generally, human decisions are required to be made around how the data in extensions contain should be handled, along with the implicit obligations that surround the information.

For this reason, local requirements that manifest as extensions are an obstacle to integration. The more the requirements are shared (i.e. regional or national scale), the less impact they will have. The consistent representation, definition and registration of extensions that this specification defines cannot resolve that problem - it only provides a framework within which such local variations can be handled more easily.

When it comes to deploying applications that support local requirements, situations will very likely arise where different applications exchanging information with each other are supporting different sets of extensions. This specification makes some basic rules that are intended to make management of these situations easier, but they cannot resolve them.

* When exchanging resources, systems SHOULD retain unknown extensions when they are capable of doing so (just as they SHOULD retain core elements when they are capable of doing so)
* If a system modifies a resource it SHOULD remove any extensions that it does not understand from the modified element and its descendants, because it cannot know whether the modifications it has made might invalidate the value of the unknown extension
* Systems that drop existing elements are considered to be "processing the resource"
* A system SHALL NOT modify a resource or element that contains "modifier" extensions it doesn't understand
* Applications SHOULD ignore extensions that they do not recognize if they are not "modifier" extensions
* Systems that do not accept unknown extensions SHALL declare so in their Conformance resource instances

The degree to which a system can retain unknown extensions is a function of the type of system it is: a general purpose FHIR server, or a middleware engine would be expected to retain all extensions, while an application that manages patient registration through a user interface can only retain extensions to the degree that the information in them is part of the set managed by the user. Other applications will fall somewhere between these two extremes.

**Summary: Handling extensions**

Use the following rules as a guideline for handling resources:

* When writing extensions, make sure they are defined and published
* When reading, navigating through or searching on elements that can have modifier extensions, check whether there are any
* When reading elements, read and process the extensions you know and use, and ignore other extensions
* Retain extensions whenever you can

**fhirpatch.html**

**FHIRPath Patch**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 0 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page documents how the [Parameters](file:///C:\temp\parameters.html) resource is used to define a set of [FHIRPath](http://hl7.org/fhirpath) based patch operations.

**Conceptual FHIRPath Operations**

This table documents the FHIRPath operations that may be used to specify changes to a resource:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Type** | **Path** | **Name** | **Value** | **Index** | **source** | **destination** | **Details** |
| add | Path at which to add the content | Name of the property to add | Data type to add at nominated place |  |  |  | The content will be appended to the element identified in the path, using the name specified |
| insert | Path of the collection in which to insert the content |  | value (Data type) to add at nominated place | index at which to insert |  |  | The content will be inserted into the nominated list at the index specified (0 based). The index must be equal or less than the number of elements in the list |
| delete | Path of the element to delete |  |  |  |  |  | Only a single element can be deleted |
| replace | Path of the element to replace |  | value (Data type) to replace it with |  |  |  |  |
| move | Path of the collection in which to move the content |  |  |  | list index to move from | list index to move to | Move an element within a single list |

There are a few base rules that apply for all operations:

* The FHIRPath statement must return elements from the resource being modified. e.g. a FHIRPath could cross resources like Observation.subject.resolve().identifier. Servers SHALL not allow patch operations to alter other resources than the nominated target, and SHOULD return an error if the patch operation tries
* The type of the value must be correct for the place at which it will be added/inserted. Servers SHALL return an error if the type is wrong
* Servers SHALL return an error if the outcome of the patch operation is a not a valid resource

**Parameters Format**

The FHIRPath patch operations are encoded in a [Parameters](file:///C:\temp\parameters.html) resource as follows:

* Each operation is a Parameter named "operation"
* Each operation has a series of parts, using the following paramter names from the table above with these types:

|  |  |
| --- | --- |
| Parameter | Type |
| Type | code |
| Path | string |
| Name | string |
| Value | [\*](file:///C:\temp\datatypes.html#open) |
| index | integer |
| source | integer |
| destination | integer |

Here is an example of adding an element:

<Parameters xmlns="http://hl7.org/fhir">

<parameter>

<name value="operation"/>

<part>

<name value="type"/>

<valueCode value="add"/>

</part>

<part>

<name value="path"/>

<valueString value="Patient"/>

</part>

<part>

<name value="name"/>

<valueString value="birthDate"/>

</part>

<part>

<name value="value"/>

<valueDate value="1930-01-01"/>

</part>

</parameter>

</Parameters>

**Anonymous Types**

Only some named data types (see [the list](file:///C:\temp\datatypes.html#open)) are allowed to be used directly in parameters. In order to add or insert other kinds of types - including anonymous elements (e.g. Observation.component, Timing.repeat), the content is defined by defining the name as described above, and instead of providing a value, a set of parts that are values are provided. Here is an example:

<Parameters xmlns="http://hl7.org/fhir">

<parameter>

<name value="operation"/>

<part>

<name value="type"/>

<valueCode value="add"/>

</part>

<part>

<name value="path"/>

<valueString value="Patient"/>

</part>

<part>

<name value="name"/>

<valueString value="contact"/>

</part>

<part>

<name value="value"/>

<part>

<name value="name"/>

<valueHumanName>

<text value="a name"/>

</valueHumanName>

</part>

</part>

</parameter>

</Parameters>

This pattern repeats as deep as necessary.

There is a set of test cases for implementers as part of the [Test Case Downloads](file:///C:\temp\downloads.html).

**financial-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Financial Module**

**Introduction**

The Financial module covers the resources and services provided by FHIR to support the costing, financial transactions and billing which occur within a healthcare provider as well as the eligibility, enrollment, authorizations, claims and payments which occur between healthcare providers and insurers and the reporting and notification between insurers and subscribers and patients.

See also the [Administration](file:///C:\temp\administration-module.html) and [WorkFlow](file:///C:\temp\workflow-module.html) modules.

**Index**

|  |  |  |
| --- | --- | --- |
| * [Account](file:///C:\temp\account.html) * [Coverage](file:///C:\temp\coverage.html) * [EligibilityRequest](file:///C:\temp\eligibilityrequest.html) * [EligibilityResponse](file:///C:\temp\eligibilityresponse.html) | * [EnrollmentRequest](file:///C:\temp\enrollmentrequest.html) * [EnrollmentResponse](file:///C:\temp\enrollmentresponse.html) * [Claim](file:///C:\temp\claim.html) * [ClaimResponse](file:///C:\temp\claimresponse.html) | * [PaymentNotice](file:///C:\temp\paymentnotice.html) * [PaymentReconciliation](file:///C:\temp\paymentreconciliation.html) * [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html) |

**Support**

Administrative

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |

**Billing**

Claims, processing and responses

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |

**Payment**

Used to support service payment processing and reporting

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |

**Other**

Patient reporting and other purposes

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |

Additional Resources will be added in the future. A list of hypothesized resources can be found on the [HL7 wiki](http://wiki.hl7.org/index.php?title=FHIR_Resource_Types). Feel free to add any you think are missing or engage with one of the [HL7 Work Groups](http://www.hl7.org/Special/committees/index.cfm) to submit a [proposal](http://wiki.hl7.org/index.php?title=Category:FHIR_Resource_Proposal) to define a resource of particular interest.

**Security and Privacy**

Financial information in general and in particular when related to or including health information, such as claims data, are typically considered Protected Health Information and as such must be afforded the same protection and safeguards as would be afforded to purely clinical identified health data.

The Security and Privacy measures associated with FHIR, such as the use of Security labels and tags in the resource.meta, is encouraged as are the use of whatever measures for authorization and encryption are supported by the chosen exchange model, eg. FHIR REST, Web Services, Direct, MLLP, SMTP and others.

For more general considerations, see [the Security and Privacy module](file:///C:\temp\secpriv-module.html).

**Common use Cases**

The table below details various common business activities which occur in the financial realm, and the focal resources which may be exchanged, along with supporting resources, to accomplish the business activities. Whether or not the resources specified are actually needed requires consideration of the business itself and the exchange methodology and transport being used.

For example: If a content model is not requred to obtain the appropriate status element then a SEARCH (GET) may be used, however if a content model is required to support the request for information then the content model will need to be CREATEd (POST). Alternately, if FHIR Operations are being used then the specified focal resource may be employed as one of the Operation parameters or may not be required.

|  |  |  |
| --- | --- | --- |
| **Business Activity** | **Request Resource** | **Response Resource** |
| Eligibility Check | [EligibilityRequest](file:///C:\temp\eligibilityrequest.html) | [EligibilityResponse](file:///C:\temp\eligibilityresponse.html) |
| Enrollment Update | [EnrollmentRequest](file:///C:\temp\enrollmentrequest.html) | [EnrollmentResponse](file:///C:\temp\enrollmentresponse.html) |
| Claim | [Claim](file:///C:\temp\claim.html) (type={discipline}, use=complete) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Pre-determination | [Claim](file:///C:\temp\claim.html) (type={discipline}, use=exploratory) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Pre-Authorization | [Claim](file:///C:\temp\claim.html) (type={discipline}, use=proposed) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Reversal | [ProcessRequest](file:///C:\temp\processrequest.html) (action=cancel, nullify=false) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Nullify | [ProcessRequest](file:///C:\temp\processrequest.html) (action=cancel, nullify=true) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Re-adjudication | [ProcessRequest](file:///C:\temp\processrequest.html) (action=reprocess) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Status Check | [ProcessRequest](file:///C:\temp\processrequest.html) (action=status) | [ProcessResponse](file:///C:\temp\processresponse.html) |
| Pended Check (Polling) | [ProcessRequest](file:///C:\temp\processrequest.html) (action=poll) | [{Resource}](file:///C:\temp\resourcelist.html) or [ProcessResponse](file:///C:\temp\processresponse.html) |
| Payment Notice | [PaymentNotice](file:///C:\temp\paymentnotice.html) | [ProcessResponse](file:///C:\temp\processresponse.html) |
| Payment Reconciliation | [ProcessRequest](file:///C:\temp\processrequest.html) (action=poll, include=[PaymentReconciliation](file:///C:\temp\paymentreconciliation.html)) | [PaymentReconciliation](file:///C:\temp\paymentreconciliation.html) |
| Send Attachments | [Communication](file:///C:\temp\communication.html) | [ProcessResponse](file:///C:\temp\processresponse.html) |
| Request Attachments | [ProcessRequest](file:///C:\temp\processrequest.html) (action=poll, include=[CommunicationRequest](file:///C:\temp\communicationrequest.html)) | [CommunicationRequest](file:///C:\temp\communicationrequest.html) |
| Request an Explanation of Benefits | [ProcessRequest](file:///C:\temp\processrequest.html) (action=poll, include=[ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html)) | [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html) |

**{discipline}** means the type of claim: OralHealth, Vision, Pharmacy, Professional or Institutional.

**{Resource}** means any pended or undelivered resource subject to the selection details specified in the request.

**Relative Order of Use**

The table below details the relative order of events and use of financial resources for patient care during the care cycle. Not all steps or information exchanges may occur, and supporting information may be required more frequently than has been depicted below.

|  |  |
| --- | --- |
| **Business Activity** | **Focal Resource** |
| **Patient visits Provider** |  |
| Provider checks for valid insurance coverage | [EligibilityRequest](file:///C:\temp\eligibilityrequest.html) |
| Insurer responds with coverage status and optional plan details | [EligibilityResponse](file:///C:\temp\eligibilityresponse.html) |
| **Provider examines Patient and reviews treatment options** |  |
| Provider submits Pre-determination(s) for treatment options to determine potential reimbursement | [Claim](file:///C:\temp\claim.html) {use=exploratory} |
| Insurer responds with potential reimbursement | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| **Provider and Patient determine treatment plan** |  |
| Treament plan submitted to Insurer to reserve funds | [Claim](file:///C:\temp\claim.html) {use=proposed} |
| Insurer acknowledges receipt of pre-authorization | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Insurer requests additional information | [CommunicationRequest](file:///C:\temp\communicationrequest.html) |
| Provider submits supporting information | [Communication](file:///C:\temp\communication.html) |
| Insurer provides adjudicated response to pre-authorization | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Provider checks on status of pre-authorization processing | [ProcessRequest](file:///C:\temp\processrequest.html) {action=status} |
| Insurer responds indicating adjudication is ready | [ProcessResponse](file:///C:\temp\processresponse.html) |
| Provider retrieves pre-authorization adjudication | READ or [ProcessRequest](file:///C:\temp\processrequest.html) {action=poll} |
| **Provider provides treatment** |  |
| Provider submits patient's claim for reimbursement | [Claim](file:///C:\temp\claim.html) {use=complete} |
| Insurer responds with claim adjudication | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| **Patient leaves treatment setting** |  |
| Patient requests an Explanation of Benefit for their Personal Health Record application | READ or [ProcessRequest](file:///C:\temp\processrequest.html) {action=poll} |
| Insurer responds with Explanation of Benefit | [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html) |
| Provider requests the payment details associated with a bulk payment | SEARCH or [ProcessRequest](file:///C:\temp\processrequest.html) {action=poll} |
| Insurer responds with a Payment Reconciliation | [PaymentReconciliation](file:///C:\temp\paymentreconciliation.html) |
| Insurer notifies provider that payment has been issued | [PaymentNotice](file:///C:\temp\paymentnotice.html) |
| Insurer notifies parties that payment funds have been received | [PaymentNotice](file:///C:\temp\paymentnotice.html) |

**Secondary Uses**

In addition to their primary use of conveying patient billing information to insurers for reimbursement either to the subscriber or the provider (assignment of benefit), many of the financial resources, such as [Claim](file:///C:\temp\claim.html) and [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html), may be used to export data to other agencies to support reporting and analytics.

**Attachments - Supporting Information**

There is often a need to provide supporting information, commonly referred to as **attachments**, to document the need for a service and/or to confirm that the good or service was actually authorized or rendered. This information may be in many forms, including: scanned documents, PDFs, word processing files, XRays, images, CDAs and FHIR Resources.

Supporting information may be provided, as a reference or the actual material, to support the [Claim](file:///C:\temp\claim.html) (complete claim or Pre-Authorization) or [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html) in a variety of manners:

* **Included**: in the *{resource}*.information section;
* **Unsolicited**: in a [Communication](file:///C:\temp\communication.html) which refers to the Claim or Explanation Of Benefit; or
* **Solicited**: in a [Communication](file:///C:\temp\communication.html) in response to a [CommunicationRequest](file:///C:\temp\communicationrequest.html) from the insurer requesting more information; or
* **Input**: in the input parameters of a FHIR operation or Task.input element if a [Task](file:///C:\temp\task.html) resource is used.

**Developmental Roadmap**

The Financial Management Work Group (FM) is responsible for two subdomains:

Financial Accounts and Billing (FIAB) - resources for accounts, charges (internal costing transactions) and patient billing, and  
Financial Claims and Reimbursement (FICR) - insurance information, enrollement, eligibility, pre-authorization, claims, patient reporting and payments.

To date FM has been focusing on the resources required to support the exchange of claims and related information between healthcare providers and insurers. The first draft of this work is nearing completion with the release of the first Financial Standard for Trial Use in STU3 of FHIR. Over the next year further refinements will be expected as we begin developing regional profiles and begin live pilots with resources.

Once the above is well underway FM can then look to developing the Enrollment-related resources and the resources to support the FIAB functions.

**Terminology**

In many cases an **example** valueset has been provided in this release. Financal Management wil be devoting effort in the preparation to Release 4 of FHIR to develop more representative example sets and to determine where global codesets exist such that some of the valuesets may be elevated in strength to extensible or required.

**financial.html**

**Financial Resources**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The table below details various common business activities which occur in the financial realm, and the focal resources which may be exchanged, along with supporting resources, to accomplish the business activities. Whether or not the resources specified are actually needed requires consideration of the business itself and the exchange methodology and transport being used.

For example, if a definitive 'Request' does not need to be documented and communicating parties are using REST then a GET may be used in place of a CREATE of the request resource. Alternately, if FHIR Operations are being used then the specified focal resource may not be required or may be employed as one of the Operation parameters.

**Note:** The SupportingDocumentation resource has been deprecated in favour of DocumentManifest for the exchange of attachments and supporting materials, PDF, Images, XRays, etc.

|  |  |  |
| --- | --- | --- |
| **Business Activity** | **Request Resource** | **Response Resource** |
| Eligibility Check | [EligibilityRequest](file:///C:\temp\eligibilityrequest.html) | [EligibilityResponse](file:///C:\temp\eligibilityresponse.html) |
| Enrollment Update | [EnrollmentRequest](file:///C:\temp\enrollmentrequest.html) | [EnrollmentResponse](file:///C:\temp\enrollmentresponse.html) |
| Claim | [Claim](file:///C:\temp\claim.html) (type={discipline}, use=complete) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Pre-determination | [Claim](file:///C:\temp\claim.html) (type={discipline}, use=exploratory) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Pre-Authorization | [Claim](file:///C:\temp\claim.html) (type={discipline}, use=proposed) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Reversal | [ProcessRequest](file:///C:\temp\processrequest.html) (action=cancel, nullify=false) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Nullify | [ProcessRequest](file:///C:\temp\processrequest.html) (action=cancel, nullify=true) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Re-adjudication | [ProcessRequest](file:///C:\temp\processrequest.html) (action=reprocess) | [ClaimResponse](file:///C:\temp\claimresponse.html) |
| Status Check | [ProcessRequest](file:///C:\temp\processrequest.html) (action=status) | [ProcessResponse](file:///C:\temp\processresponse.html) |
| Pended Check (Polling) | [ProcessRequest](file:///C:\temp\processrequest.html) (action=poll) | [{Resource}](file:///C:\temp\resourcelist.html) or [ProcessResponse](file:///C:\temp\processresponse.html) |
| Payment Notice | [PaymentNotice](file:///C:\temp\paymentnotice.html) | [ProcessResponse](file:///C:\temp\processresponse.html) |
| Payment Reconciliation | [ProcessRequest](file:///C:\temp\processrequest.html) (action=poll, include=[PaymentReconciliation](file:///C:\temp\paymentreconciliation.html)) | [PaymentReconciliation](file:///C:\temp\paymentreconciliation.html) |
| Send Attachments | [DocumentManifest](file:///C:\temp\documentmanifest.html) | [ProcessResponse](file:///C:\temp\processresponse.html) |
| Send Attachments \*NEW | [Communication](file:///C:\temp\communication.html) | [ProcessResponse](file:///C:\temp\processresponse.html) |
| Request Attachments \*NEW | [ProcessRequest](file:///C:\temp\processrequest.html) (action=poll, include=[CommunicationRequest](file:///C:\temp\communicationrequest.html)) | [CommunicationRequest](file:///C:\temp\communicationrequest.html) |
| Request an Explanation of Benefits | [ProcessRequest](file:///C:\temp\processrequest.html) (action=poll, include=[ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html)) | [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html) |

**{discipline}** means the type of claim: OralHealth, Vision, Pharmacy, Professional or Institutional.

**{Resource}** means any pended or undelivered resource subject to the selection details specified in the request.

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |

Additional Resources will be added in the future. A list of hypothesized resources can be found on the [HL7 wiki](http://wiki.hl7.org/index.php?title=FHIR_Resource_Types). Feel free to add any you think are missing or engage with one of the [HL7 Work Groups](http://www.hl7.org/Special/committees/index.cfm) to submit a [proposal](http://wiki.hl7.org/index.php?title=Category:FHIR_Resource_Proposal) to define a resource of particular interest.

**fluentpath.html**

**FHIRPath**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\publish\resource.html#maturity): 0 | [Ballot Status](file:///C:\temp\help.html#status): [Draft](file:///C:\temp\history.html#pubs) |

Note: this content is a joint development between the FHIR project and the CQL project. It is intended that this will be published as its own standard. As a temporary measure, it is included in FHIR until publishing questions are resolved.

FHIRPath is a path based navigation and extraction language, somewhat like XPath. Operations are expressed in terms of the logical content of hierarchical data models, and support traversal, selection and projection of data. Its design was influenced by the needs for path navigation, selection and formulation of invariants in both HL7 Fast Healthcare Interoperability Resources (FHIR) and HL7 Clinical Quality Language (CQL).

In both FHIR and CQL, this means that expressions can be written that deal with the contents of the resources and data types as described in the Logical views, or the UML diagrams, rather than against the physical representation of those resources. JSON and XML specific features are not visible to the FHIRPath language (such as comments and the split representation of primitives).

The expressions can in theory be converted to equivalent expressions in XPath, OCL, or another similarly expressive language.

**1. Navigation model**

FHIRPath navigates and selects nodes from a tree that abstracts away and is independent of the actual underlying implementation of the source against which the FHIRPath query is run. This way, FHIRPath can be used on in-memory Java POJOs, Xml data or any other physical representation, so long as that representation can be viewed as classes that have properties. In somewhat more formal terms, FHIRPath operates on a directed acyclic graph of classes as defined by a MOF-equivalent type system.

Data is represented as a tree of labelled nodes, where each node may optionally carry a primitive value and have child nodes. Nodes need not have a unique label, and leaf nodes must carry a primitive value. For example, a (partial) representation of a FHIR Patient resource in this model looks like this:

The diagram shows a tree with a repeating identifier node, which represents repeating members of the FHIR object model. Leaf nodes such as use and family carry a (string) value. It is also possible for internal nodes to carry a value, as is the case for the node labelled active: this allows the tree to represent FHIR "primitives", which may still have child extension data.

**2. Path selection**

FHIRPath allows navigation in the tree by composing a path of concatenated labels, e.g.

name.given

This would result in a set of nodes, one with the value "Wouter" and one with the value "Gert". In fact, each step in such a path results in a collection of nodes by selecting nodes with the given label from the step before it. The focus at the beginning of the evaluation contained all elements from Patient, and the path name selected just those named name. Since the name element repeats, the next step given along the path, will contain all nodes labeled given from all nodes name in the preceding step.

The underlying type of the entry point (here: Patient) can be represented, but is optional. To illustrate this point, the path name.given above can be evaluated as an expression on a set of data of any type. However, for natural human use, expressions may be prefixed with the name of the type:

Patient.name.given

The two expressions have the same outcome, but when evaluating the second, the evaluation will only produce results when used on data of type Patient.

Syntactically, FHIRPath defines identifiers as any sequence of characters consisting only of letters, digits, and underscores, beginning with a letter or underscore. Paths may use double quotes to include characters in path parts that would otherwise be interpreted as operators, e.g.:

Message."PID-1"

**3.1 Paths and polymorphic items**

In the underlying representation of data, nodes may be typed and represent polymorphic items. Paths may either ignore the type of a node, and continue along the path or may be explicit about the expected node and filter the set of nodes by type before navigating down child nodes:

Observation.value.unit - all kinds of value

Observation.value.as(Quantity).unit - only values that are of type Quantity

The is function can be used to determine whether or not a given value is of a given type:

Observation.value.is(Quantity) - returns true if the value is of type Quantity

The list of available types that can be passed as a parameter to the as and is functions is determined by the underlying data model.

**3.2 Referring to the current item**

It is sometimes useful to refer to the current item under evaluation when writing an expression, especially within where() when the value of the current item needs to be passed as a function parameter. This can be done using the special path $this:

Patient.name.given.where(substring($this.length()-3)) = 'out'

**3.3 Order nodes and traversal**

Collections of nodes are inherently ordered, and implementations must retain the original order of a collection. There are two special cases: the outcome of operations like children() and descendants() cannot be assumed to be in any meaningful order, and first(), last(), tail(), skip() and take() should not be used on collections derived from these paths. Note that some implementations may follow the standard order, and some may not, and some may be different depending on the underlying source.

**4. Expressions**

In addition to paths, FHIRPath expressions may contain *literals* and *function invocations*. FHIRPath supports the following types of literals:

boolean: true, false

string: 'test string', 'urn:oid:3.4.5.6.7.8'

integer: 0, 45

decimal: 0.0, 3.141592653587793236

datetime: @2015-02-04T14:34:28Z - @ followed by ISO8601 compliant date/time

time: @T14:34:28Z @ followed by ISO8601 compliant time (beginning with T)

Unicode is supported in both string literals and quoted identifiers. String literals are surrounded by single quotes and may use \-escapes to escape quotes and represent Unicode characters:

* Unicode characters may be escaped using \u followed by four hex digits.
* Additional escapes are those supported in JSON:
  + \\ (backslash),
  + \/ (slash),
  + \f (form feed - \u000c),
  + \n (newline - \u000a),
  + \r (carriage return - \u000d),
  + \t (tab - \u0009)

Numbers can be integers and may optionally have decimal parts. Numbers cannot use exponential notation.

Expressions can also contain *operators*, like those for mathematical operations and boolean logic:

Appointment.minutesDuration / 60 > 5

MedicationAdministration.wasNotGiven.exists() implies MedicationAdministration.reasonNotGiven.exists()

name.given | name.last

'sir ' + name.given

Finally, FHIRPath supports the notion of functions, which all take a collection of values as input and produce another collection as output. For example:

(name.given | name.last).distinct()

identifier.where(use = 'official')

Since all functions work on collections, constants will first be converted to a collection when functions are invoked on constants:

(4+5).count()

will return 1, since this is implicitly a collection with one constant number 9.

Note: There is no concept of null in FHIRPath. This means that when, in an underlying instance a member is null, there will simply be no corresponding node for that member in the tree. This means that for example Patient.children() will contain only non-null members and Patient.name will return an empty collection if there are no name elements in the instance.

**4.1 Boolean evaluation of collections**

Collections can be evaluated as booleans in logical tests in criteria. When a collection is implicitly converted to a boolean then:

* IF the collection contains a single node AND the node's value is a boolean THEN
  + the collection evaluates to the value of that single boolean node
* ELSE IF the collection is empty THEN
  + the collection evaluates to an empty collection
* ELSE
  + the collection evaluates to true

This same principle applies when using the path statement in invariants.

Note: Because the path language is side effect free, it does not matter whether implementations use short circuit boolean evaluation or not. However with regard to performance, implementations are encouraged to use short circuit evaluation, and authors of path statements should pay attention to short circuit evaluation when designing statements for optimal performance.

**4.2 Propagation of empty results**

FHIRPath functions and operators both propagate empty results. This means in general that if any input to a function or operator is empty, then the result will be empty as well. More specifically:

* If a function operates on an empty collection, the result is an empty collection
* If a function is passed an empty collection as argument, the result is an empty collection
* If any operand to an operator is an empty collection, the result is an empty collection.

When functions behave differently (for example the count() and empty() functions), this is clearly documented in the next sections.

**5. Functions**

Functions are distinguished from path navigation names by the fact that they are followed by a () with zero or more parameters. Functions always take a collection as input and produce another collection as output, even though these may be collections of just a single item. Correspondingly, arguments to the functions can be any FHIRPath expression, though some functions require these expressions to evaluate to a collection containing a single item of a specific type.

The following list contains all functions supported in FHIRPath, detailing the expected kind of parameters and kind of collection returned by the function:

* If the function expects a parameter to be a single value (e.g. item(index: integer) and it is passed an argument that evaluates to a collection with multiple items or a collection with an item that is not of the required type, the evaluation of the expression will end and an error will be signaled to the calling environment.
* If the function takes an expression as a parameter, the function will evaluate this parameter with respect to each of the items in the input collection. These expressions may refer to the special $this element, which represents the item from the input collection currently under evaluation. For example, in:
* name.given.where($this > 'ba' and $this < 'bc')

the where() function will iterate over each item in the input collection (elements named given) and $this will be set to each item when the expression passed to where() is evaluated.

* Optional parameters are enclosed in square brackets in the definition of a function.
* All functions return a collection, but if this is a single item of a predefined type, the description of the function will specify its output type explicitly, instead of just stating collection, e.g. all(...) : boolean

**5.1 Existence**

**empty() : boolean**

Returns true if the input collection is empty ({ }) and false otherwise.

**not() : boolean**

Returns true if the input collection evaluates to false, and false if it evaluates to true. Otherwise, the result is empty ({ }):

|  | **not** |
| --- | --- |
| true | false |
| false | true |
| empty ({ }) | empty ({ }) |

**exists() : boolean**

Returns the opposite of empty(), and as such is a shorthand for empty().not()

**all() : boolean**

Returns true if every element in the input collection evaluates to true, and false if any element evaluates to false. Otherwise, the result is empty ({ }).

**subsetOf(other : collection) : boolean**

Returns true if all items in the input collection are members of the collection passed as the other argument. Membership is determined using the equals (=) operation (see below).

**supersetOf(other : collection) : boolean**

Returns true if all items in the collection passed as the other argument are members of the input collection. Membership is determined using the equals (=) operation (see below).

**isDistinct() : boolean**

Returns true if all the items in the input collection are distinct. To determine whether two items are distinct, the equals (=) operator is used, as defined below.

**distinct() : collection**

Returns a collection containing only the unique items in the input collection. To determine whether two items are the same, the equals (=) operator is used, as defined below.

**count() : integer**

Returns a collection with a single value which is the integer count of the number of items in the input collection. Returns 0 when the input collection is empty.

**5.2 Filtering and projection**

**where(criteria : expression) : collection**

Filter the input collection to only those elements for which the stated criteria expression evaluates to true.

**select(projection: expression) : collection**

Evaluates the given expression for each item in the input collection. The result of each evaluation is added to the output collection. If the evaluation results in a collection with multiple items, all items are added to the output collection (collections resulting from evaluation of projection are *flattened*).

**repeat(projection: expression) : collection**

A version of select that will repeat the projection and add it to the output collection, as long as the projection yields new items (as determined by the Equals operator).

This operation can be used to traverse a tree and selecting only specific children:

ValueSet.expansion.repeat(contains)

Will repeat finding children called contains, until no new nodes are found.

Questionnaire.repeat(group | question).question

Will repeat finding children called group or question, until no new nodes are found.

Note that this is slightly different from

Questionnaire.descendants().select(group | question)

which would find **any** descendants called group or question, not just the ones nested inside other group or question elements.

**is(type : identifier) : boolean**

Returns true if the collection contains a single element of the given type or a subclass thereof.

**as(type : identifier) : collection**

Returns a collection that contains all items in the input collection that are of the given type or a subclass thereof.

**5.3 Subsetting**

***name*[ index : integer ] : collection**

This indexer operation returns a collection with only the index-th item (0-based index). If the index lies outside the boundaries of the input collection, an empty collection is returned.

Example:

Patient.name[0]

**single() : collection**

Will return the single item in the input if there is just one item. If there are multiple items, an error is signaled to the evaluation environment.

**first() : collection**

Returns a collection containing just the first item in the list. Equivalent to item(0), so it will return an empty collection if the input collection has no items.

**last() : collection**

Returns a collection containing the last item in the list. Will return an empty collection if the input collection has no items.

**tail() : collection**

Returns a collection containing all but the first item in the list. Will return an empty collection if the input collection has no or just one item.

**skip(num : integer) : collection**

Returns a collection containing all but the first num items in the list. Will return an empty collection if there are no items remaining after the indicated number of items have been skipped.

**take(num : integer) : collection**

Returns a collection containing the first num items in the list, or less if there are less then num items. Will return an empty collection if the input collection is empty.

**5.5 Conversion**

The functions in this section operate on collections with a single item. If there is more than one item, or an incompatible item, the evaluation of the expression will end and signal an error to the calling environment.

To use these functions over a collection with multiple items, one may use filters like where() and select():

Patient.name.given.select(substring(1))

**iif(criterium: expression, true-expression [, otherwise-expression]) : collection**

If the input collection contains a single item, this function evaluates the criterium expression. If this expression evaluates to true, the function evaluates the true-expression on the input and returns that as a result.

If the evaluation resulted in false or an empty collection, the otherwise-expression is evaluated on the input and returned, unless the optional otherwise-expression is not given, in which case the function returns an empty collection.

If the input has multiple items or is empty, the function will return an empty collection.

**toInteger() : integer**

If the input collection contains a single item, this function will return a single integer if:

* the item in the input collection is an integer
* the item in the input collection is a string and is convertible to an integer
* the item is a boolean, where true results in a 1 and false results in a 0.

In all other cases, the function will return an empty collection.

**toDecimal() : decimal**

If the input collection contains a single item, this function will return a single decimal if:

* the item in the input collection is a decimal
* the item in the input collection is a string and is convertible to a decimal
* the item is a boolean, where true results in a 1 and false results in a 0.

In all other cases, the function will return an empty collection.

**toString() : string**

If the input collection contains a single item, this function will return a single string if:

* the item in the input collection is a string
* the item in the input collection is a number, the output will contain its string representation
* the item is a boolean, where true results in "true" and false in "false".

In all other cases, the function will return an empty collection.

**5.6 String manipulation**

The functions in this section operate on collections with a single item. If there is more than one item, or an incompatible item, the evaluation of the expression will end and signal an error to the calling environment.

**substring(start : integer [, length : integer]) : collection**

If the input collection contains a single item of type string, it returns a collection with the part of the string starting at position start (zero-based). If length is given, will return at most length number of characters from the input string.

If start lies outside the length of the string, the function returns an empty collection. If there are less remaining characters in the string than indicated by length, the function returns just the remaining characters.

**startsWith(string : string) : boolean**

If the input collection contains a single item of type string, the function will return true when the value starts with the specified content.

**endsWith(string : string) : boolean**

If the input collection contains a single item of type string, the function will return true when the value ends with the specified content.

**matches(regex : string) : boolean**

If the input collection contains a single item of type string, the function will return true when the value matches the given regular expression. Regular expressions are supposed to work culture invariant, case-sensitive and in 'single line' mode and allow Unicode characters.

**replaceMatches(regex : string, substitution: string) : string**

If the input collection contains a single item of type string, the function will match the input using the regular expression in regex and replace each match with the substitution string. The substitution may refer to identified match groups in the regular expression.

This example of replaceMatches() will convert a string with a date formatted as MM/dd/yy to dd-MM-yy:

'11/30/1972'.replaceMatches('\\b(?<month>\\d{1,2})/(?<day>\\d{1,2})/(?<year>\\d{2,4})\\b',

'${day}-${month}-${year}')

Note: All platforms will use their native regular expression implementations, which will commonly be close to the regular expressions in Perl 5, however there are always small differences. I don't think we can prescribe any "common" dialect for FHIRPath.

**contains(string : string) : boolean**

A simpler variation of matches() that returns a boolean when the given string is a substring of the single string in the input collection.

**replace(pattern : string, substitution : string) : string**

A simpler variation of replaceMatches that returns the input string with all instances of pattern replaced with substitution.

**length() : integer**

If the input collection contains a single item of type string, the function will return the length of the string.

**5.6 Tree navigation**

**children() : collection**

Returns a collection with all immediate child nodes of all items in the input collection.

**descendants() : collection**

Returns a collection with all descendant nodes of all items in the input collection. The result does not include the nodes in the input collection themselves. Is a shorthand for repeat(children()).

Note: Many of these functions will result in a set of nodes of different underlying types. It may be necessary to use as() as described in the previous section to maintain type safety. See section 8 for more information about type safe use of FHIRPath expressions.

**5.7 Utility functions**

**memberOf(valueset : string) : boolean**

If the input collection contains a single string item, it is taken to be a code and valueset membership is tested against the valueset passed as the argument. The valueset argument is a uri used to resolve to a valueset.

**trace(name : string) : collection**

Add a string representation of the input collection to the log, using the parameter name as the name in the log. This log should be made available to the user in some appropriate fashion. Does not change the input, so returns the input collection as output.

**today() : datetime**

Returns a datetime containing the current date.

**now() : datetime**

Returns a datetime containing the current date and time, including timezone.

**6. Operations**

Operators are allowed to be used between any kind of path expressions (e.g. expr op expr). Like functions, operators will generally propagate an empty collection in any of their operands. This is true even when comparing two empty collections using the equality operators.

**6.1 Equality**

**= (Equals)**

Returns true if the left collection is equal to the right collection:

If both operands are collections with a single item:

* For primitives:
  + If the value is a string, the strings must be exactly the same
  + For numbers, equality requires that the 2 values have the same precision
  + For boolean values, the values must be the same
* For complex types, equality requires all child properties to be equal, recursively.

If both operands are collections with at least one item:

* Each item must be equal
* Comparison is order dependent

Otherwise, equals returns false.

Note that this implies that if both collections have a different number of items to compare, the result will be false.

Typically, this operator is used with single fixed values as operands. This means that Patient.telecom.system = 'phone' will return false if there is more than one telecom with a use element. Typically, you'd want Patient.telecom.where(system = 'phone')

If one or both of the operands is the empty collection, this operation returns an empty collection.

**~ (Equivalent)**

Returns true if the collections are the same.

If both operands are collections with a single item:

* For primitives
  + If the value is a string, the strings must be the same while ignoring case and accents
  + For numbers, precision is ignored (exact rules to be discussed)
  + For boolean values, the values must be the same
* For complex types, equivalence requires all child properties to be equivalent, recursively.

If both operands are collections with multiple items:

* Each item must be equivalent
* Comparison is not order dependent

Note that this implies that if both collections have a different number of items to compare, the result will be false.

If one or both of the operands is the empty collection, this operation returns an empty collection.

**!= (Not Equals)**

The inverse of the equals operator.

**!~ (Not Equivalent)**

The inverse of the equivalent operator.

**6.2 Comparison**

* The comparison operators are defined for strings, numbers, datetimes, and times.
* unless there is only one item in each collection (left and right) of the same type, the comparisons return false
* string evaluation is strictly lexical, not based on any defined meaning of order
* If one or both of the arguments are an empty collection, the comparisons return an empty collection.

**> (Greater Than)**

**< (Less Than)**

**<= (Less or Equal)**

**>= (Greater or Equal)**

**6.3 Types**

**is**

If the left operand is a collection with a single item and the second operand is an identifier, this operator returns true if the type of the left operand is the type specified in the second operand, or a subclass thereof. In all other cases this function returns the empty collection.

Patient.contained.all($this is Patient implies age > 10)

**as**

If the left operand is a collection with a single item and the second operand is an identifier, this function returns the value of the left operand, or a subclass thereof. Otherwise, this operator returns the empty collection.

**6.4 Collections**

**| (union collections)**

Merge the two collections into a single collection, eliminating any duplicate values (using equals (=)) to determine equality).

**in (membership)**

If the left operand is a collection with a single item, this operator returns true if the item is in the right operand using equality semantics. This is the inverse operation of contains.

**contains (containership)**

If the right operand is a collection with a single item, this operator returns true if the item is in the left operand using equality semantics. This is the inverse operation of in.

**6.5 Boolean logic**

For all boolean operators, the collections passed as operands are first evaluated as booleans (as described in 4.1). The operators then use three-valued logic to propagate empty operands.

**and**

Returns true if both operands evaluate to true, false if either operand evaluates to false, and empty collection ({ }) otherwise:

|  | **true** | **false** | **empty ({ })** |
| --- | --- | --- | --- |
| true | true | false | empty ({ }) |
| false | false | false | false |
| empty ({ }) | empty ({ }) | false | empty ({ }) |

**or**

Returns false if both operands evaluate to false, true if either operand evaluates to true, and empty ({ }) otherwise:

|  | **true** | **false** | **empty ({ })** |
| --- | --- | --- | --- |
| true | true | true | true |
| false | true | false | empty ({ }) |
| empty ({ }) | true | empty ({ }) | empty ({ }) |

**xor**

Returns true if exactly one of the operands evaluates to true, false if either both operands evaluate to true or both operands evaluate to false, and the empty collection ({ }) otherwise:

|  | **true** | **false** | **empty ({ })** |
| --- | --- | --- | --- |
| true | false | true | empty ({ }) |
| false | true | false | empty ({ }) |
| empty ({ }) | empty ({ }) | empty ({ }) | empty ({ }) |

**implies**

If the left operand evaluates to true, this operator returns the boolean evaluation of the right operand. If the left operand evaluates to false, this operator returns true. Otherwise, this operator returns true if the right operand evaluates to true, and the empty collection ({ }) otherwise.

|  | **true** | **false** | **empty ({ })** |
| --- | --- | --- | --- |
| true | true | false | empty ({ }) |
| false | true | true | true |
| empty ({ }) | true | empty ({ }) | empty ({ }) |

**6.6 Math**

The math operators require each operand to be a single element. Both operands must be of the same type, each operator below specifies which types are supported.

If there is more than one item, or an incompatible item, the evaluation of the expression will end and signal an error to the calling environment.

As with the other operators, the math operators will return an empty collection if one or both of the operands are empty.

**\* (multiplication)**

Multiplies both arguments (numbers only)

**/ (division)**

Divides the left operand by the right operand (numbers only).

**+ (addition)**

For numbers, add the numbers. For strings, concatenates the right operand to the left operand.

**- (subtraction)**

Subtracts the right operand from the left operand (numbers only).

**div**

Performs truncated division of the left operand by the right operand (numbers only).

**mod**

Computes the remainder of the truncated division of its arguments (numbers only).

**6.5 Operator precedence**

Precedence of operations, in order from high to low:

#01 . (path/function invocation)

#02 [] (indexer)

#03 unary + and -

#04: \*, /, div, mod

#05: +, -,

#06: |

#07: >, <, >=, <=

#08: is, as

#09: =, ~, !=, !~

#10: in, contains

#11: and

#12: xor, or

#13: implies

As customary, expressions may be grouped by parenthesis (()).

**7. Environment variables**

A token introduced by a % refers to a value that is passed into the evaluation engine by the calling environment. Using environment variables, authors can avoid repetition of fixed values and can pass in external values and data.

The following environmental values are set for all contexts:

%sct - (string) url for snomed ct

%loinc - (string) url for loinc

%ucum - (string) url for ucum

%"vs-[name]" - (string) full url for the provided HL7 value set with id [name]

%"ext-[name]" - (string) full url for the provided HL7 extension with id [name]

%context - The original node that was passed to the evaluation engine before starting evaluation

Note how the names of the `vs-` and `ext-` constants are escaped (just like paths) to allow "-" in the name.

Implementers should note that using additional environment variables is a formal extension point for the language. Implementation Guides are allowed to define their own externals, and implementers should provide some appropriate configuration framework to allow these constants to be provided to the evaluation engine at run time. E.g.:

%us-zip = '[0-9]{5}(-[0-9]{4}){0,1}'

Authors of Implementation Guides should be aware that adding specific environment variables restricts the use of the FHIRPath to their particular context.

Note that these tokens are not restricted to simple types, and they may not have defined fixed values that are known before evaluation at run-time, though there is no way to define these kind of values in implementation guides.

**8. Type safety and strict evaluation**

Strongly typed languages are intended to help authors avoid mistakes by ensuring that expressions written describe valid operations. For example, a strongly typed language would typically disallow the expression:

1 + 'John'

because it performs an invalid operation, namely adding numbers and strings. However, there are cases where the author knows that a particular invocation may be safe, but the compiler is not aware of, or cannot infer, the reason. In these cases, type-safety errors can become an unwelcome burden, especially for experienced developers.

As a result, FHIRPath defines a *strict* option that allows an execution environment to determine how much type safety should be applied. With *strict* enabled, FHIRPath behaves as a traditional strongly-typed language, whereas without *strict*, it behaves as a traditional dynamically-typed language.

For example, since some functions and most operators will only accept a single item as input, and throw an exception otherwise:

Patient.name.given + ' ' + Patient.name.family

will work perfectly fine, as long as the patient has a single name, but will fail otherwise. It is in fact "safer" to formulate such statements as either:

Patient.name.select(given + ' ' + family)

which would return a collection of concatenated first and last names, one for each name of a patient. Of course, if the patient turns out to have multiple given names, even this statement will fail and the author would need to choose the first name in each collection explicitly:

Patient.name.first().select(given.first() + ' ' + family.first())

It is clear that, although more robust, the last expression is also much more elaborate, certainly in situations where, because of external constraints, the author is sure names will not repeat, even if the unconstrained data model allows repetition.

Apart from throwing exceptions, unexpected outcomes may result because of the way the equality operators are defined. The expression

Patient.name.given = 'Wouter'

will return false as soon as a patient has multiple names, even though one of those may well be 'Wouter'. Again, this can be corrected:

Patient.name.where(given = 'Wouter').exists()

but is still less concise than would be possible if constraints were well known in advance.

The strict option provides a mode in which the author of the FHIRPath statement is protected against such cases by employing strict typing. Based on the definition of the operators and functions and given the type of input, a compiler can trace the statement and determine whether "unsafe" situations can occur.

Unsafe uses are:

* A function that requires an input collection with a single item is called on an output that is not guaranteed to have only one item.
* A function is passed an argument that is not guaranteed to be a single value.
* A function is passed an input value or argument that is not of the expected type
* An operator that requires operands to be collections with a single item is called with arguments that are not guaranteed to have only one item.
* An operator has operands that are not of the expected type
* Equality operators are used on operands that are not both collections or collections of single items.

There are a few constructs in the FHIRPath language where the compiler cannot trace the type, and should issue a warning to the user when doing "strict" evaluation:

* The children() and descendants() functions
* The resolve() function
* A member which is polymorphic (e.g. a choice[x] type in FHIR)

Authors can use the as() function directly after such constructs to inform the compiler of the expected type, so that strict type-checking can continue.

In strict mode, when the compiler finds places where a collection of multiple items can be present while just a single item is expected, the author will need to make explicit how repetitions are dealt with. Depending on the situation one may:

* Use first(), last() or indexer ([ ]) to select a single item
* Use select() and where() to turn the expression into one that evaluates each of the repeating items individually (as in the examples above)
* Use single() to return either the single item or else an empty collection. This is especially useful when using FHIRPath to formulate invariants: in cases where single items are considered the "positive" or "true" situation, single() will return an empty collection, so the invariant will evaluate to the empty collection (or false) in any other circumstance.

**Appendix A - Use of FHIRPath in HL7 FHIR**

FHIRPath is used in five places in the FHIR specifications:

* search parameter paths - used to define what contents the parameter refers to
* slicing discriminator - used to indicate what element(s) define uniqueness
* invariants in ElementDefinition, used to apply co-occurrence and other rules to the contents
* error message locations in OperationOutcome
* URL templates in Smart on FHIR's cds-hooks
* may be used for Patch in the future

As stated in the introduction, FHIRPath uses a tree model that abstracts away the actual underlying datamodel of the data being queries. For FHIR, this means that the contents of the resources and data types as described in the Logical views (or the UML diagrams) are used as the model, rather than the JSON and XML formats, so specific xml or json features are not visible to the FHIRPath language (such as comments and the split representation of primitives).

More specifically:

* A FHIRPath may optionally start with a full resource name
* Elements of datatypes and resources are used as the name of the nodes which can be navigated over, except for choice elements (ending with '[x]'), see below.
* The contained element node does not have the name of the Resource as its first and only child (instead it directly contains the contained resource's children)
* There is no difference between an attribute and an element
* Repeating elements turn into multiple nodes with the same name

**A.1 Polymorphism in FHIR**

FHIR has the notion of choice elements, where elements can be one of multiple types, e.g. Patient.deceased[x]. In actual instances these will be present as either Patient.deceasedBoolean or Patient.deceasedDateTime. In FHIRPath choice elements are labeled according to the name without the '[x]' suffix, and children can be explicitly filtered using the asType operation:

Observation.value.as(Quantity).unit

**A.2 Using FHIR types in expressions**

The evaluation engine will automatically convert the value of FHIR types representing primitives to FHIRPath types when they are used in expression in the following fashion:

| **FHIR primitive type** | **FHIRPath type** |
| --- | --- |
| boolean | boolean |
| string, uri, code, oid, id, uuid, sid, markdown, base64Binary | string |
| integer, unsignedInt, positiveInt | integer |
| decimal | decimal |
| date, dateTime, instant | datetime |
| time | time |

Note that FHIR primitives may contain extensions, so that the following expressions are *not* mutually exclusive:

Patient.name.given = 'Ewout' // value of Patient.name.given as a string

Patient.name.given.extension.first().value = true // extension of the primitive value

**A.3 Additional functions**

FHIR adds (backwards compatible) functionality to the common set of functions:

**extension(string : string) : collection**

Return any extension with the given URL. This is a syntactical shortcut for .extension.where(url = string), but is simpler to write.

**trace(name : string) : collection**

When FHIRPath statements are used in an invariant, the log contents should be added to the error message constructed when the invariant is violated. For example:

"SHALL have a local reference if the resource is provided inline (url: height; ids: length,weight)"

from

"reference.startsWith('#').not()

or ($context.reference.substring(1).log('url') in $resource.contained.id.log('ids'))"

**resolve() : collection**

For each item in the collection, if it is a string, locate the target of the reference, and add it to the resulting collection. If the item is not a string, the item is ignored and nothing is added to the output collection.

The items in the collection may also represent a Reference, in which case the Reference.reference is resolved.

If fetching the resource fails, the failure message is added to the output collection.

**as(type : identifier) : collection**

In FHIR, only concrete core types are allowed as an argument. All primitives are considered to be independent types (so markdown is **not** a subclass of string). Profiled types are not allowed, so to select SimpleQuantity one would pass Quantity as an argument.

**A.4 Changes to operators**

**~ (Equivalence)**

Equivalence works in exactly the same manner, but with the addition that for complex types, equality requires all child properties to be equal, **except for "id" elements**.

**A.5 Environment variables**

The FHIR specification specified one additional variable:

%resource - The original resource current context is part of.

When evaluating a datatype, this would be the resource the element is part of. Do not go past a root resource into a bundle, if it is contained in a bundle

**Appendix B - Use of FHIRPath in Clinical Quality Language (CQL)**

Clinical Quality Language is being extended to use FHIRPath as its core expression language, in much the same way that XQuery uses XPath to represent expressions within queries. In particular, the following extensions to CQL are proposed:

**Path Traversal**

When a path expression involves an element with multiple cardinality, the expression is considered short-hand for an equivalent query invocation. For example:

Patient.name

is allowed, and is considered a short-hand for the following query expression:

Patient.name X where X.name is not null return X.name

Note that the restriction is required as it ensures that the resulting list will not contain any null elements.

**Constants and Contexts**

FHIRPath has the ability to reference contexts (using the $ prefix) and environment-defined variables (using the % prefix). Within CQL, these contexts and environment-defined variables are added to the appropriate scope (global for environment-variables, local for contexts) with the prefix included. This allows them to be referenced like any other variable within CQL, but preserves the prefix as a namespace differentiator.

**Additional Operators**

The following additional operators are being added to CQL:

* ~, !~ - Equivalent operators (formerly matches in CQL)
* != - As a synonym for <>
* implies - Logical implication
* | - As a synonym for union

**Method-style Invocation**

One of the primary syntactic features of FHIRPath is the ability to Ã¢â‚¬Å“invokeÃ¢â‚¬Â a function on a collection. For example:

Patient.name.given.substring(3)

The CQL syntax is being extended to support this style of invocation, but as a short-hand for an equivalent CQL statement for each operator. For example:

stringValue.substring(3, 5)

is allowed, and is considered a short-hand for the following CQL expression:

Substring(stringValue, 3, 5)

For most functions, this short-hand is a simple rewrite, but for contextual functions such as where() and select(), this rewrite must preserve the context semantics:

Patient.name.where(given = 'John')

is short-hand for:

Patient.name N where N.given = 'John'

**Strict Evaluation**

Because CQL is a type-safe language, embedded FHIRPath expressions should be compiled in *strict* mode. However, to enable the use of FHIRPath in *loose* mode, an implicit conversion from a list of elements to an element is added. This implicit conversion is implemented as an invocation of singleton from, ensuring that if the list has multiple elements at run-time an error will be thrown.

In addition, the underlying Expression Logical Model (ELM) is being extended to allow for dynamic invocation. A Dynamic type is introduced with appropriate operators to support run-time invocation where necessary. However, these operators are introduced as an additional layer on top of core ELM, and CQL compiled with the *strict* option will never produce expressions containing these elements. This avoids placing additional implementation burden on systems that do not need dynamic capabilities.

**footer.html**

**Warning: FHIR is a draft specification that is still undergoing development prior to balloting as a full HL7 standard**   
Implementers are welcome to experiment with the content defined here, but should note that the contents are subject to change without prior notice.  
®© HL7.org 2011 - 2012. FHIR v generated on . [License](file:///C:\temp\index.html#license)

**footer1.html**

**Warning: FHIR is a draft specification that is still undergoing development prior to balloting as a full HL7 standard**   
Implementers are welcome to experiment with the content defined here, but should note that the contents are subject to change without prior notice.  
© HL7.org 2011 - 2012. FHIR v generated on . [License](file:///C:\index.html#license)

**footer2.html**

**Warning: FHIR is a draft specification that is still undergoing development prior to balloting as a full HL7 standard**   
Implementers are welcome to experiment with the content defined here, but should note that the contents are subject to change without prior notice.  
© HL7.org 2011 - 2012. FHIR v generated on . [License](file:///C:\..\index.html#license)

**footer3.html**

**Warning: FHIR is a draft specification that is still undergoing development prior to balloting as a full HL7 standard**   
Implementers are welcome to experiment with the content defined here, but should note that the contents are subject to change without prior notice.  
© HL7.org 2011 - 2012. FHIR v generated on . [License](file:///C:\..\..\index.html#license)

**formats-definitions.html**

**Resource Format - Detailed Descriptions**

**formats-examples.html**

**Resource Format Examples**

This page includes additional examples of the resource format, based on common usages and questions

Todo

**formats.html**

**Resource Formats**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 3 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page documents how the content of the resources are described. In actual exchange, resources can be represented in the following formats: [XML](file:///C:\temp\xml.html), [JSON](file:///C:\temp\json.html) and [Turtle](file:///C:\temp\rdf.html). Other representations are allowed, but are not described by this specification.

**Resource Definition**

The resources are described in several different ways:

* a hierarchical table that presents a logical view of the content
* a UML diagram that summarizes the content graphically
* a pseudo-XML syntax that provides a visual sense of what the end resource instances will look like in XML
* a pseudo-JSON syntax that provides a visual sense of what the end resource instances will look like in JSON
* a pseudo-Turtle syntax that provides a visual sense of what the end resource instances will look like in Turtle

In addition to this descriptive syntax, other definitional forms are available, including W3C schema and Schematron, and the [StructureDefinition](file:///C:\temp\structuredefinition.html) syntax defined internally.

**Logical table**

The Logical View shows the resources as a tree structure with the following columns:

|  |  |
| --- | --- |
| **Column** | **Content** |
| Name | The name of the element in the resource (manifests as XML element name, or JSON property name. Some names finish with [x] - the meaning of this is discussed below. In addition, this column contains an icon that denotes the underlying type of the content. The icons are described below |
| Flags | A set of information about the element that impacts how implementers handle them. The flags are described below |
| Card. | The lower and upper bounds on how many times this element is allowed to appear in the resource |
| Type | The type of the element (hyperlinked to the definition of the type). Note that the type of the element has one of two meanings, depending on whether the element has defined children. If the element has children, then the element has an anonymous type that specializes the given type. If the element has no children, then the element has properties and children as specified by the nominated type |
| Description & Constraints | A description of the element, and details about constraints that are applied to it. Particularly, for coded elements, information about which codes can be used |

Here's an example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Flags** | **Card.** | **Type** | **Description & Constraints** |
| [Resource Name](file:///C:\temp\main-pages.html) |  |  | [Base Type](file:///C:\temp\resource.html) | Definition |
| [nameA](file:///C:\temp\main-pages.html) | ÃŽÂ£ | 1..1 | [type](file:///C:\temp\main-pages.html) | description of content |
| [nameB[x]](file:///C:\temp\main-pages.html) | ?! ÃŽÂ£ | 0..1 |  | description SHALL at least have a value |
| nameBType1 |  | 0..1 | [type1](file:///C:\temp\main-pages.html) |  |
| nameBType2 | I | 0..1 | [type2](file:///C:\temp\main-pages.html) |  |
| [nameC](file:///C:\temp\main-pages.html) |  | 1..\* | Element | Definition |
| [nameD](file:///C:\temp\main-pages.html) |  | 1..1 | [type](file:///C:\temp\main-pages.html) | Relevant Records |

Key to Type Icons and Flags

* : The base element for a resource (see [Resources](file:///C:\temp\resource.html))
* : An element that is part of the resource and has elements within it defined in the same resource or profile
* : An element which can have one of several different types (see below)
* : An element of a data type which describes an element that has a value attribute/property
* : An element of a data type which describes an element that has other elements
* : An element that contains a reference to another resource (see [references](file:///C:\temp\references.html))
* : This element has the same content as another element defined within this resource or profile
* : Introduction of a set of slices (see [Slicing](file:///C:\temp\profiling.html#slicing))
* : An extension (see [Extensibility](file:///C:\temp\extensibility.html))
* : A complex extension - one with nested extensions (see [Extensibility](file:///C:\temp\extensibility.html#complex))
* : An extension that has a value and no nested extensions (see [Extensibility](file:///C:\temp\extensibility.html))
* : A complex modifier extension - one with nested extensions (see [Extensibility](file:///C:\temp\extensibility.html#complex))
* : A modifier extension that has a value and no nested extensions (see [Extensibility](file:///C:\temp\extensibility.html))
* : The root of a logical profile
* ?!: This element is a modifying element - see [Modifier Elements](file:///C:\temp\conformance-rules.html#isModifier)
* S: This element is an element that must be supported - see [Must-Support Elements](file:///C:\temp\conformance-rules.html#mustSupport)
* ÃŽÂ£: This element is an element that is part of the summary set - see [Summary Searches](file:///C:\temp\search.html#summary)
* I: This element defines or is affected by constraints - see [Constraints](file:///C:\temp\conformance-rules.html#constraints)
* NE: This element cannot have extensions (some infrastructural elements only)

Notes:

* Resource and Element names are case-sensitive (though duplicates that differ only in case are never defined)
* Any elements that have a [primitive type](file:///C:\temp\datatypes.html#primitive) will have a value attribute/property to contain the actual value of the element
* This value attribute/property can never be empty. Either it is absent, or it is present with at least one character of non-whitespace content
* Elements are assigned a cardinality that specifies how many times the element may or must appear.
* Unless elements have children defined directly (as nameC does above) they are assigned one or more types. Most of the types are defined in [the data types](file:///C:\temp\datatypes.html). All the type names are hyperlinked to the source definition
* Element reuse: Some data types that have children have the same set of children as some other element defined in the resource. In that case, the type of that element has a "see [name]" where [name] is the name of the element that has the defined children
* Each element name is also a hyperlink to the formal definition of the element in the data dictionary that underlies the exchange formats.
* Any of the elements may have an id attribute to serve as [the target of an internal reference](file:///C:\temp\references.html#id). The id attribute is not shown in this format. Extensions are not always shown, but may appear except where the flag NE appears
* FHIR elements can never be empty. If an element is present in the resource, it SHALL have either a value, child elements as defined for its type, or 1 or more [extensions](file:///C:\temp\extensibility.html)
* Infrastructural elements that are common to all resources are not shown in the logical representation. These are described in the common base classes [Resource](file:///C:\temp\resource.html), and [DomainResource](file:///C:\temp\domainresource.html)

**Choice of Data Types**

A few elements have a choice of more than one data type for their content. All such elements have a name that takes the form nnn[x]. The "nnn" part of the name is constant, and the "[x]" is replaced with the title-cased name of the type that is actually used. The table view shows each of these names explicitly.

Elements that have a choice of data type cannot repeat. I.e. They must have a maximum cardinality of 1. When constructing an instance of an element with a choice of types, the authoring system must create a single element with a data type chosen from among the list of permitted data types.

Note: In object-orientated based implementations, this is naturally represented as a polymorphic property. However this is not necessary and the correct implementation varies according to the particular features of the language. In XML schema, these become an xs:choice of element.

**UML**

The UML diagrams represent the same content as a series of classes that represent the elements of a resource.

NameAelement : [type] [0..\*]nameB : CodeableConcept [0..1] Ã‚Â« Value Set Name? Ã‚Â»NameCvalue[x] : Type [0..1] Ã‚Â« Type1|Type2|Type3 Ã‚Â»reference : Reference [0..1] Ã‚Â« Resource1|Resource2 Ã‚Â»nameC[0..1]

The elements and the data types are hyperlinks to the formal definitions of the parts. The UML diagrams also show the vocabulary bindings. These are hyperlinks to the value set details.

Where an element can have a choice of data types, or is a [Reference](file:///C:\temp\references.html) these are represented by showing the common type (Reference or Type), and then showing the applicable data type names or resource types in a stereotype, separated by the | character. Type is not formally otherwise defined by this specification, but is a super type of all the data types.

The actual order of the elements in XML cannot be determined from the diagram, nor whether a UML property becomes an element or an attribute in the XML representation.

Bindings to value sets are indicated by a stereotype on the element. The stereotype has 2 parts: the value set name, and a symbol that denotes the strength of the binding:

* ??: [Example](file:///C:\temp\terminologies.html#example) Binding
* ?: [Preferred](file:///C:\temp\terminologies.html#example) Binding
* +: [Extensible](file:///C:\temp\terminologies.html#example) Binding
* !: [Required](file:///C:\temp\terminologies.html#example) Binding

**Serialization Format Representations**

This specification defines the following ways to represent resources when they are exchanged:

* [XML](file:///C:\temp\xml.html)
* [JSON](file:///C:\temp\json.html)
* [RDF](file:///C:\temp\rdf.html) *(Under development)*

Clients and servers can choose what syntax(s) to implement. In the interests of interoperability, servers SHOULD support all formats (though support for RDF should best wait until it is more fully developed). Systems SHALL declare which format(s) they support in their [Capability Statement](file:///C:\temp\capabilitystatement.html). If a server receives a request for its Capability Statement in a format it does not otherwise support, it SHALL return a 415 Unsupported Media Type.

**foundation-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Foundation Module**

The Foundation Module is responsible the overall infrastructure of the FHIR specification. Every implementer works with the content in the foundation module however they use FHIR.

|  |  |  |
| --- | --- | --- |
| **Overview**  Tutorial and Introductory Information.   * [Overview & Roadmap](file:///C:\temp\overview.html) * [Developer's Introduction](file:///C:\temp\overview-dev.html) * [Clinical Introduction](file:///C:\temp\overview-clinical.html) * [Architect's Introduction](file:///C:\temp\overview-arch.html) * [1 page Summary (Glossy)](file:///C:\temp\summary.html) * [**FHIR Timelines**](file:///C:\temp\timelines.html)   **Administration**  Documentation Guidance / Background.   * [Documentation Guidance & Glossary](file:///C:\temp\help.html) * [License and Legal Terms](file:///C:\temp\license.html) * [Community & Credits](file:///C:\temp\credits.html) * [Version History](file:///C:\temp\history.html) * [Differences to DSTU 2](file:///C:\temp\diff.html) * [Transforms between DSTU 2 and STU 3](file:///C:\temp\r2maps.html) * [Outstanding Issues](file:///C:\temp\todo.html)   **Appendices**:   * [Coming Challenges Driving Change](file:///C:\temp\change.html) | **Technical Documentation**  General Implementer Documentation.   * [Conformance Rules](file:///C:\temp\conformance-rules.html) * [Resource Life Cycles](file:///C:\temp\lifecycle.html) * [References between Resources](file:///C:\temp\references.html) * [Narrative](file:///C:\temp\narrative.html) * [Extensibility](file:///C:\temp\extensibility.html) & [Extension Registry](file:///C:\temp\extensibility-registry.html) * [Compartments (Patient based access)](file:///C:\temp\compartmentdefinition.html) * [Formats](file:///C:\temp\formats.html): [XML](file:///C:\temp\xml.html), & [JSON](file:///C:\temp\json.html) * [FHIRPath](http://hl7.org/fhirpath) * [Data Types](file:///C:\temp\datatypes.html) * [Element](file:///C:\temp\element.html) * [BackboneElement](file:///C:\temp\backboneelement.html) * [ElementDefinition](file:///C:\temp\elementdefinition.html)   **Exchange**  Tutorial and Introductory Information.   * [RESTful API](file:///C:\temp\http.html) + [Search](file:///C:\temp\search.html) * [Operations](file:///C:\temp\operations.html) * [Documents](file:///C:\temp\documents.html) * [Messaging](file:///C:\temp\messaging.html) * [Services](file:///C:\temp\services.html) | **Resources**  Foundation Framework   * [Resource](file:///C:\temp\resource.html) * [DomainResource](file:///C:\temp\domainresource.html) * [Basic](file:///C:\temp\basic.html) * [Binary](file:///C:\temp\binary.html) * [Bundle](file:///C:\temp\bundle.html) * [MessageHeader](file:///C:\temp\messageheader.html) * [OperationOutcome](file:///C:\temp\operationoutcome.html) * [Parameters](file:///C:\temp\parameters.html)   **Infrastructure**  Support Resources.   * [Questionnaire](file:///C:\temp\questionnaire.html) * [QuestionnaireResponse](file:///C:\temp\questionnaireresponse.html) * [List](file:///C:\temp\list.html) * [Media](file:///C:\temp\media.html) * [Composition](file:///C:\temp\composition.html) * [DocumentReference](file:///C:\temp\documentreference.html) * [DocumentManifest](file:///C:\temp\documentmanifest.html) |

**Relationships with other modules**

* All the other modules depend on the foundation module
* The [Linked Data module](file:///C:\temp\linked-data-module.html) builds on the foundation model by adding [RDF](file:///C:\temp\rdf.html) representations, and strengthening the definitions by facilitating linkage to external ontologies
* The [Terminology module](file:///C:\temp\terminology-module.html) provides the formal basis for using Concepts defined in Code Systems in the definitions
* The [Conformance module](file:///C:\temp\conformance-module.html) provides the basis for extending the foundation for national and local use
* The [Security & Privacy](file:///C:\temp\secpriv-module.html) provides the linking framework to external standards for security and privacy
* The [Implementation Support module](file:///C:\temp\implsupport-module.html) builds on the foundation to provide testing and reference implementations

**Common use Cases: Exchanging Data**

FHIR supports 4 different paradigms for exchange: the [RESTful API](file:///C:\temp\http.html), [Messaging](file:///C:\temp\messaging.html), [Documents](file:///C:\temp\documents.html), and [Services](file:///C:\temp\services.html). Each of these approaches can be used to exchange information, and each has it's own strengths and weaknesses.

**RESTful API**

Most implementers focus on [RESTful API](file:///C:\temp\http.html). This is a client/server API designed to follow the [principles of RESTful design](https://en.wikipedia.org/wiki/Representational_state_transfer) for **C**reate, **R**ead, **U**pdate and **D**elete operations, along with [**S**earch](file:///C:\temp\search.html) and [**E**xecute (Operations)](file:///C:\temp\operations.html) support.

The RESTful API is a general purpose interface that can be used to [push and pull](file:///C:\temp\pushpull.html) data between systems. Which is appropriate depends on architecture and deployment considerations.

**Messaging**

In addition to the RESTful API, a [Messaging exchange framework](file:///C:\temp\messaging.html) is documented, which supports exchange between systems by exchanging content by sending routed messages from system to system. This exchange can be implemented on the RESTful API, or using some other messaging stack. The messaging paradigm is provided to support implementers who wish to use such a messaging paradigm.

Implementers should note that the messaging framework is not provided to fill any functional deficiency in the RESTful API (or vice versa), these frameworks are provided to allow implementers to choose how to exchange content based on their own architectural and deployment considerations.

**Documents**

This specification also defines a [Document based exchange framework](file:///C:\temp\documents.html), where content to be exchanged is wrapped by a [Composition](file:///C:\temp\composition.html) that provides the context of the content, and that has a fixed presentation for a human reader. The document framework is provided to help with computer-assisted human to human communication uses - which are not uncommon in healthcare.

Typically, exchanging documents is associated with exchanging clinical information across clinical governance borders, while data based exchange using the RESTful API is appropriate within where there are well established clinical governance arrangements.

**Services / SOA**

In addition, this specification describes the use of FHIR in a [services framework](file:///C:\temp\services.html)(e.g. a SOA). Note that any use of any of the above alternatives in production is a 'service' by some or many definitions. The services description provides context regarding the use of FHIR (and particularly the RESTful API) in a wider enterprise architecture.

**Developmental Roadmap**

Much of the foundation module is well advanced through the maturity model process. Specifically, the following parts of the specification:

* [Element](file:///C:\temp\element.html) / [Resource](file:///C:\temp\resource.html) + [ElementDefinition](file:///C:\temp\elementdefinition.html)
* The [XML](file:///C:\temp\xml.html), & [JSON](file:///C:\temp\json.html) formats
* Many of the [Data Types](file:///C:\temp\datatypes.html)
* The [RESTful API](file:///C:\temp\http.html) + [Search](file:///C:\temp\search.html)
* [Binary](file:///C:\temp\binary.html), [Bundle](file:///C:\temp\bundle.html), [OperationOutcome](file:///C:\temp\operationoutcome.html) & [Parameters](file:///C:\temp\parameters.html)

The focus over the next 18 months or so as the 4th release of FHIR is prepared is to focus on stability and move these artefacts to normative status. However some parts of the foundation module are not as well explored, and are not as far advanced with regard to their development. Specifically [Documents](file:///C:\temp\documents.html), [Messaging](file:///C:\temp\messaging.html) and [Services](file:///C:\temp\services.html) are areas that still need further testing with regard to interoperability. HL7 expects to focus on testing these things in connectathons over the next 18 months.

**genomics.html**

**Genomics Implementation Guidance**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 1 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Table of Contents**

1. [Background](file:///C:\temp\main-pages.html#background)
2. [Overview](file:///C:\temp\main-pages.html#overview)
3. [Sequence Resource](file:///C:\temp\main-pages.html#sequence)
4. [Observation-genetics Profile](file:///C:\temp\main-pages.html#observation-genetics)
5. [DiagnosticReport-genetics Profile](file:///C:\temp\main-pages.html#diagnosticreport-genetics)
6. [ProcedureRequest-genetics Profile](file:///C:\temp\main-pages.html#procedurerequest-genetics)
7. [HLA genotyping results Profile](file:///C:\temp\main-pages.html#hla)
8. [Relationship among Sequence resource and genetics profiles](file:///C:\temp\main-pages.html#resource_vs_profiles)
9. [HL7 Domain Analysis Model Use Cases](file:///C:\temp\main-pages.html#dam)
10. [Other Genomics Use Cases](file:///C:\temp\main-pages.html#other_examples)
11. [Clinical Genomics Apps](file:///C:\temp\main-pages.html#apps)
12. [Appendix 1: From DSTU2 Genomics to STU3](file:///C:\temp\main-pages.html#history)

**Background**

Rapid advances in sequencing technologies, clinical genetics tests for whole genome and exome sequencing are allowing sophisticated genetics testing to be used by providers and patients in clinical decisions. Results from such tests are used to identify distinct genetic variants that may contribute to syndromes, conditions and/or predictive responses to treatments. The implementation of precision medicine will depend upon having such data to diagnose patients, choose medications, and predict the course of disease and care, but will require standards and effective user interfaces.

A current technical challenge exists in interoperability, or allowing the ability to access and share clinical and genetics data. This interoperability includes collection, coding, and retrieval to scale. An individual's genetic data set is large, complex and requires curation. Unfortunately, incompatible systems and nomenclatures are already in use. A standards based ontology that could be adopted to integrate both genetic data and clinical information systems will be crucial to accelerating the integration of precision medicine and to make sense of genetic testing results in a complete clinical context.

One approach for collecting, coding, and retrieving genetics data comes from the [Global Alliance for Genomics and Health](http://genomicsandhealth.org) (GA4GH). The GA4GH organization has built and is refining an API and data model for the exchange of full sequence genomic information across multiple research organizations and platforms. The GA4GH focuses on the needs of researchers.

A second approach is evolving from [Health Level 7](http://hl7.org) (HL7) through its standards family called [Fast Healthcare Interoperability Resources](http://hl7.org/fhir/) (FHIR). FHIR focuses on clinical needs and systems and has principally focused on standardizing traditional clinical data of the sort in EMR systems using very modern web technology-based ideas. FHIR is attractive because it is relatively easy to implement because it is comprised of a set of modular components called resources, which can be easily and incrementally assembled into working systems. The clinical requirements for genetics data is, relative to genomics research needs, utilitarian and reductive because it is about distilling and extracting particular genetics data produced by ever more sophisticated testing for use at the point-of-care. This has made FHIR a very functional framework to initiate an interoperable clinical genetics data standardization to which multiple stakeholders have contributed to this guide.

Starting in 2014, the HL7 Clinical Genomics Working Group (CGWG) has been drafting the FHIR framework for clinical genetics testing for general adoption. This effort reached its first major milestone on September 23, 2015, when HL7 released its second FHIR Draft Standard for Trial Use (DSTU2). DSTU2 introduced a Standard genetics profile that applies to the FHIR Observation resource. The profile was created according to the FHIR Clinical Genomics Project Scope Statement that authorized its community development. Using this profile, Observation payloads can return genetic testing results in a standardized manner.

Extending Observations rather than creating a new, dedicated FHIR genetics data resource is consistent with the FHIR community mandate because it only adds new resources to an existing resource library when there is a compelling case to do so. We provide the case to add a Sequence resource in this document.

**Overview**

Given the clinical significance of genetic sequence data and the wealth of information that may be retrieved for an individual's health and with metadata for the betterment of public health, we propose to move beyond FHIR DSTU2 Standard Genetics profile on Observation and allow increased granularity and less ambiguity by creating a new resource to be called **Sequence**. This resource will be used to hold clinically relevant sequence data in a manner that is both efficient and versatile integrating new and as yet undefined types of genomic and other -omics data that will soon be commonly entered into health records for clinical use. Sequence will be leveraged by other FHIR resources, including Observation. This is consistent with how all FHIR resources are designed and used.

The FHIR Clinical Genomics Working Group (CGWG) began discussing such a dedicated resource in 2014. The September 2014 Informative Ballot (Ã¢â‚¬Å“HL7 Clinical Genomics, Domain Analysis Model: Clinical Sequencing Release 1Ã¢â‚¬Â) provided guiding use cases, which initially informed development of the initial Standard Genetics profile that is found in FHIR DSTU2. The same use cases also led to a second Project Scope Statement in May 2015 (approved by HL7 DESD/OO/FMG) to develop a Sequence resource (Ã¢â‚¬Å“Develop FHIR sequence resource for Clinical GenomicsÃ¢â‚¬Â). A preliminary effort to address these issues has been explored and published in context of the [Substitutable Medical Applications and Reusable Technologies](http://smarthealthit.org/an-app-platform-for-healthcare/about/) (SMART) Platforms Project and described in an article (Ã¢â‚¬Å“[SMART on FHIR Genomics: Facilitating standardized clinico-genomic apps](http://jamia.oxfordjournals.org/content/early/2015/07/21/jamia.ocv045.long)Ã¢â‚¬Â).

Sequence is designed to hold genetic sequences in blocks relevant to actionable clinical decision-making. Extensions to Sequence address complex cases and can associate it with repositories for retrieving a patientÃ¢â‚¬â„¢s full sequence data, such as those defined by GA4GH. Other changes include a suite of genetics profiles for other FHIR resources. In addition, the Observation-genetics profile adds new references so that an Observation can report genetics test results to be integrated into the EHR. There are also new genetics-extension profiles for DiagnosticReport, ProcedureRequest and FamilyMemberHistory, respectively, to extend them to report genetics results. We have given all of these FHIR genetics profiles the suffix Ã¢â‚¬Å“-geneticsÃ¢â‚¬Â (e.g. Ã¢â‚¬Å“DiagnosticReport-genetics profileÃ¢â‚¬Â). New profiles on top of DiagnosticReport have been created for reporting HLA genotyping results.

On the following pages, we elaborate upon the rationale for the proposed design, introducing in some detail the following resource and profiles:

* A Sequence resource
* An Observation-genetics profile on Observation
* A DiagnosticReport-genetics profile on DiagnosticReport
* A ProcedureRequest-genetics profile on ProcedureRequest
* An HLA-genotyping-results profile on DiagnosticReport

With these resource and profiles, FHIR can support a large set of clinical use cases (see [Section 9](file:///C:\temp\main-pages.html#dam), [10](file:///C:\temp\main-pages.html#other_examples) , and [11](file:///C:\temp\main-pages.html#apps)) and is thus positioned to address all emergent -omics use cases, including Next-Generation Sequencing (NGS). These tools are simple to implement, will optimize payload sizes, and help developers avoid redundant retrieval of data. [Appendix 1](file:///C:\temp\main-pages.html#history) of this document shows how DSTU 2.0 can be mapped to the new additions to the resource.

**Sequence Resource**

**Structure Diagram**

**Description**

The [Sequence resource](file:///C:\temp\sequence.html) is designed for next-generation sequencing data. PatientsÃ¢â‚¬â„¢ observed sequences should be represented by recording reference sequence id/string and detected variants. To specify how it proceed, here is a picture below:

Sequence.coordinateSystem: This element shall be constrained into only two possible values: 0 for 0-based system and 1 for 1-based system. Below is the picture that could explain whatÃ¢â‚¬â„¢s the difference between these two systems:

Here are two examples that clarify the usage in both cases (they represent same segment part):

* 0-based example: [here](file:///C:\temp\coord-0base-example.html) Source:[XML](file:///C:\temp\coord-0base-example.xml.html),[JSON](file:///C:\temp\coord-0base-example.json.html)
* 1-based example: [here](file:///C:\temp\coord-1base-example.html) Source:[XML](file:///C:\temp\coord-1base-example.xml.html),[JSON](file:///C:\temp\coord-1base-example.json.html)

Sequence.referenceSeq: Four optional ways are provided to represent reference sequence in Sequence resource:

1. Sequence.referenceSeq.referenceSeqId: Reference sequence id in public database;
2. Sequence.referenceSeq.referenceSeqString: Reference sequence string;
3. Sequence.referenceSeq.referenceSeqPointer: Reference to observed sequence stored in another sequence entity;
4. Sequence.referenceSeq.genomeBuild, Sequence.reference.chromosome: The combination of genome build and chromosome.

The window selects a range from the reference sequence (or genome) that is used to define building block of a current sequence (e.g. Sequence resource instance 1).

Sequence.referenceSeq.strand: Only two possible values can be made by strand, +1 for plus strand while -1 for minus strand. Since the directionality of the sequence string might be represented in different word in different omics scenario, below are simple example of how to map other expressions into its correlated value:

|  |  |
| --- | --- |
| **Map to +1** | **Map to -1** |
| 5Ã¢â‚¬Â²-to-3Ã¢â‚¬Â² direction | 3Ã¢â‚¬Â²-to-5Ã¢â‚¬Â² direction |
| Watson | Crick |
| Sense | Antisense |
| Positive | Negative |

Sequence.quality: Quality scores for bases in the sequence. Is is intended to be compliant with emerging regulatory needs needs (eg: those found at [PrecisionFDA](https://precision.fda.gov)).

Sequence.variant: This complex element is used for encoding sequence. When the information of reference sequence and variants are provided the observed sequence will be derived.

**Internal Pointers**

Sequence.patient: This element points to a Patient identifier to show that this sequence is related to the same patient.

Sequence.specimen: A pointer to specimen identifier, if needed.

Sequence.device: A pointer to Device identifier which is used for describing sequencing method (such as chip id, chip manufacturer etc.)

Sequence.pointer: A pointer to a Sequence instance for the next sequence block to build a sequence graph.

**External Pointers**

Sequence.repository: This complex element is used to provide a clarifying structure, a base URL, and/or relevant IDs when referring to an external repository.

GA4GH Repository Example. If the Sequence resource refers to a GA4GH repository for read info, references to a GA4GH full sequence dataset should conform to GA4GH data models and accessed via the GA4GH API. The URL of a GA4GH repository, ids of a GA4GH variant and read group are contained in the Sequence resource. The URL of a GA4GH repository is an api\_base of a GA4GH server that could be called for sequence data. The GA4GH variant set is a collection of call sets and the GA4GH call set is a collection of variant calls, typically for one sample. A variant call represents a determination of genotype with respect to that variant.

VariantSet definition: A VariantSet is a collection of variants and variant calls intended to be analyzed together.

CallSet definition: A CallSet is a collection of calls that were generated by the same analysis of the same sample.

A read group is a collection of reads produced by a sequencer. A read group set typically models reads corresponding to one sample, sequenced one way, and aligned one way. The API reference of [Google Genomics](https://cloud.google.com/genomics/) is a GA4GH repository built by Google and provides details of the data models, such as the resource representations.

**Usage Examples**

**Different way to represent sequence record with variations**

We provide a detailed example to show how sequence resource can be used to represent record of observed sequence by different method.

The diagram above describes 4 optional ways provided in the Sequence resource to encode sequencing data. Here is the corresponding examples

* [By referenceSeqId](file:///C:\temp\sequence-graphic-example-1.html) Source:[XML](file:///C:\temp\sequence-graphic-example-1.xml.html),[JSON](file:///C:\temp\sequence-graphic-example-1.json.html),[TURTLE](file:///C:\temp\sequence-graphic-example-1.ttl.html)
* [By referenceSeqString](file:///C:\temp\sequence-graphic-example-2.html) Source:[XML](file:///C:\temp\sequence-graphic-example-2.xml.html),[JSON](file:///C:\temp\sequence-graphic-example-2.json.html),[TURTLE](file:///C:\temp\sequence-graphic-example-2.ttl.html)
* [By referenceSeqPointer](file:///C:\temp\sequence-graphic-example-3.html) Source:[XML](file:///C:\temp\sequence-graphic-example-3.xml.html),[JSON](file:///C:\temp\sequence-graphic-example-3.json.html),[TURTLE](file:///C:\temp\sequence-graphic-example-3.ttl.html)
* [By Genomebuild+chromosome](file:///C:\temp\sequence-graphic-example-4.html) Source:[XML](file:///C:\temp\sequence-graphic-example-4.xml.html),[JSON](file:///C:\temp\sequence-graphic-example-4.json.html),[TURTLE](file:///C:\temp\sequence-graphic-example-4.json.html)

**Example from precision FDA**

We provide one [example](file:///C:\temp\sequence-example-fda.html) to show how precision FDA vcf data can be upload and comform FHIR specification, and how quality in sequence resource is represented by comparision between a reference sequence and the observed sequence.

Source: [XML](file:///C:\temp\sequence-example-fda.xml.html), [JSON](file:///C:\temp\sequence-example-fda.json.html), [Turtle](file:///C:\temp\sequence-example-fda.ttl.html).

**Complex Variation example**

We provide one [example](file:///C:\temp\sequence-complex-variant.html) to show how a complex variant can be represented with the help of cigar. The deletion, insertion and mutation will represented in charaters along with the number of repitition.

Source: [XML](file:///C:\temp\sequence-complex-variant.xml.html), [JSON](file:///C:\temp\sequence-complex-variant.json.html), [Turtle](file:///C:\temp\sequence-complex-variant.ttl.html).

**Observation-genetics Profile**

**Structure Diagram**

**Description**

The [Observation-genetics profile](file:///C:\temp\observation-genetic.html) Observation-genetics profile is used to interpret variants from sequence resource. Clinical usage may need more specific representation of variant at locus or structural variant in whole genome.

Some of the attributes of the profile follow:

* The observation-geneticsSequence extension will refer to the Sequence resource for sequence information related to this variant.
* The observation-geneticsInterpretation extension will refer to an Observation instance which contains clinical interpretations for the variant described.
* The code, effective[x], issued, performer, method, specimen elements can be used to describe how the genetic observation (variant and sequence data) is obtained.
* Other extensions are used to describe attributes of this variant such as Genomics Source Class, Amino Acid Change Type, etc. These are mappings from v2 and lonic code reference with details can be found in this [list](file:///C:\temp\observation-genetic.html#content).

Additional Observations instance will be created for variant's further analysis. For example, Observation.component element will be used for knowledge-based interpretations of the sequence variant. Here are some examples for the component.code.

|  |  |  |
| --- | --- | --- |
| **LOINC Code** | **LOINC Element Name** | **Comments** |
| 51963-7 | Medication Assessed | A coded medication accessed in a pharmacogenetic test (recommend RxNorm). |
| 51967-8 | Genetic disease assessed | A coded disease that is associated with the region of DNA covered by the genetic test (recommend SNOMED). |
| 53037-8 | Genetic Disease Sequence Variant Interpretation | Interpretation of the pathogenicity of the DNA Sequence Variant in the context of the assessed genetic disease. |
| 53040-2 | Drug Metabolism Sequence Variant Interpretation | Predicted phenotype for drug efficacy. A sequence variant interpretation value known to allow (responsive) or prevent (resistant) the drug to perform. |
| 51961-1 | Drug Efficacy Sequence Variant Interpretation | Predicted phenotype for ability of drug to bind to intended site in order to deliver intended effect. A Sequence Variant interpretation value known to allow (responsive) or prevent (resistant) the drug to perform. |

In the meantime, the related element in this Observation instance will point to Observaiton-genetics profile to show these clinical interpretations are further analysis for the variant. (For example, sequence variant has its pointer back to the observation, see [PGx example](file:///C:\temp\main-pages.html#PGx-example))

**Usage Examples**

**An Observation-genetics instance and its two related Observation instances for interpretations**

We provide an [example](file:///C:\temp\observation-example-genetics-1.html) of an Observation-genetics instance which records a variant detected in the patient. (we call this example A)

Source: [XML](file:///C:\temp\observation-example-genetics-1.xml.html), [JSON](file:///C:\temp\observation-example-genetics-1.json.html), [Turtle](file:///C:\temp\observation-example-genetics-1.ttl.html).

An [example](file:///C:\temp\observation-example-genetics-2.html) of an Observation instance which records knowledge-based clinical interpretations for the variant represented in A.

Source: [XML](file:///C:\temp\observation-example-genetics-2.xml.html), [JSON](file:///C:\temp\observation-example-genetics-2.json.html), [Turtle](file:///C:\temp\observation-example-genetics-2.ttl.html).

An [example](file:///C:\temp\observation-example-genetics-3.html) of an Observation instance which records PCR validation test for the variant in A.

Source: [XML](file:///C:\temp\observation-example-genetics-3.xml.html), [JSON](file:///C:\temp\observation-example-genetics-3.json.html), [Turtle](file:///C:\temp\observation-example-genetics-3.ttl.html).

**Phenotype, Diplotype and Haplotye**

We provide following examples to reveal how PGx data can be harmoinzed with specfication in FHIR. Detailed discussion will be put [here.](file:///C:\temp\main-pages.html#PGx-example)

* [Phenotype](file:///C:\temp\observation-example-phenotype.html) Source:[XML](file:///C:\temp\observation-example-phenotype.xml.html),[JSON](file:///C:\temp\observation-example-phenotype.json.html),[TURTLE](file:///C:\temp\observation-example-phenotype.ttl.html)
* [Diplotype](file:///C:\temp\observation-example-diplotype1.html) Source:[XML](file:///C:\temp\observation-example-diplotype1.xml.html),[JSON](file:///C:\temp\observation-example-diplotype1.json.html),[TURTLE](file:///C:\temp\observation-example-diplotype1.ttl.html)
* [Haplotype](file:///C:\temp\observation-example-haplotype1.html) Source:[XML](file:///C:\temp\observation-example-haplotype1.xml.html),[JSON](file:///C:\temp\observation-example-haplotype1.json.html),[TURTLE](file:///C:\temp\observation-example-haplotype1.ttl.html)

**DiagnosticReport-genetics Profile**

**Structure Diagram**

This [DiagnosticReport-genetics](file:///C:\temp\diagnosticreport-genetic.html) is built on top of DiagnosticReport. The new profile is used to describe a genetics test report. The result element in DiagnosticReport will refer to the Observation resource that can lead to a bundle of genetic observations. And the element of code, effective[x], issued, performer, request, specimen are be used to describe the details of the genetic test. Extensions about AssessedCondition and FamilyMemberHistory are added. Overall, this profile extends the DiagnosticReport resource to enable reporting of structured genetic test results. In addition, it denotes condition context for genetic testing, which may influence reported variants and interpretations for large genomic testing panels.

* The DiagnosticReport-genetics profile contains two extensions referring to other resources/profiles and one complex extension.
* AssessedCondition is used to denote condition context for genetic testing, which may influence reported variants and interpretation for large genomic testing panels. It refers to the Condition resource.
* FamilyMemberHistory are significant health events and conditions for one or more persons related to the patient. It refers to the FamilyMemberHistory-Genetic profile.
* Analysis provides knowledge-based interpretation of overall results of tests performed on patients.

Examples for analysis.code:

|  |  |  |
| --- | --- | --- |
| **LOINC Code** | **LOINC Element Name** | **Comments** |
| 51968-6 | Genetic Disease Analysis Overall Interpretation | Interpretation of all identified DNA Sequence variants along with any known clinical information for the benefit of aiding clinicians in understanding the results overall in either the context of diagnosis or increased risk of disease. |
| 51964-5 | Drug Efficacy Analysis Overall Interpretation | Overall predicted phenotype for drug efficacy for all DNA Sequence Variants identified in a single case. |
| 51971-0 | Drug metabolism analysis overall interpretation | Overall predicted phenotype for drug metabolism for all DNA Sequence Variants identified in a single case. |

**Usage Examples**

An [example](file:///C:\temp\diagnosticreport-genetics-example-2-familyhistory.html) of a genetic test report for a patient with FamilyHistory.

Source: [XML](file:///C:\temp\diagnosticreport-genetics-example-2-familyhistory.xml.html), [JSON](file:///C:\temp\diagnosticreport-genetics-example-2-familyhistory.json.html), [Turtle](file:///C:\temp\diagnosticreport-genetics-example-2-familyhistory.ttl.html).

An [example](file:///C:\temp\diagnosticreport-genetics-comprehensive-bone-marrow-report.html) of a comprehensive bone marrow report.

Source: [XML](file:///C:\temp\diagnosticreport-genetics-comprehensive-bone-marrow-report.xml.html), [JSON](file:///C:\temp\diagnosticreport-genetics-comprehensive-bone-marrow-report.json.html), [Turtle](file:///C:\temp\diagnosticreport-genetics-comprehensive-bone-marrow-report.ttl.html).

**ProcedureRequest-genetics profile**

**Structure Diagram**

A complex extension is added on top of the ProcedureRequest resource. Here is the structure of the extension:

**Description**

To describe an order requested sequence variants detection. User must set up the code for the request and they can also refer to the corresponding sequence instance for that variant.

**Usage Example**

**An order for genetics test**

[Here](file:///C:\temp\procedurerequest-genetics-example-1.html) is a diagnostic request for testing 185delAG variant. The mother of the patient received results from a mutation panel (eg. MyRisk from Myriad) and she has a BRCA1 185delAG mutation. The clinician the would like to request to test the patient only for an 185delAG mutation. In this case, the diagnostic request for the patient will specify the sequence variant - 185delAG.

**HLA genotyping results Profile**

Human leukocyte antigen (HLA) genotyping is fundamental for research and clinical practice in immunogenetics and histocompatibility. Pointers to external locations refer to registered methods, raw NGS reads, and reference standards can be conveyed in this profile. Information about allele assignment including ambiguous results and the allele database used for assignments is stored in extensions.

The structure of the [HLA typing report](file:///C:\temp\hlaresult.html) in this profile attempts to follow the principles outlined in the Minimum Information for ReportIng Next-generation sequence Genotyping (MIRING). These principles were identified through a series of meetings with international group of stakeholders in the application of Next Generation Sequencing (NGS) technology for genotyping the HLA and KIR loci as well as other immune-related loci (http://igdawg.org/ngs.html). MIRING describes eight principles, described in detail in [Human Immunology. 2015 Dec; 76(12):954-962](http://www.ncbi.nlm.nih.gov/pubmed/26319908). These include detailed metadata about:

1. MIRING Annotation
2. Reference Context
3. Full Genotype
4. Consensus Sequence
5. Novel Polymorphisms
6. Platform Documentation
7. Read Processing Documentation
8. Primary Data

(see the publication above for details about each)

These principles were implemented in a technical specification by extending an existing XML based format for exchanging histocompatibility and immunogenetic genotyping data called Histoimmunogenetics Markup Language (HML) to include results from NGS methodologies (https://bioinformatics.bethematchclinical.org/hla-resources/hml/). The resulting schema may be found in https://schemas.nmdp.org/. The National Marrow Donor Program (NMDP)/Be The Match uses this format for reporting HLA genotyping from potential donors and for patients needing stem cell transplants.

Both MIRING and HML were used to inform mapping data elements to FHIR resources such as Patient, Specimen, Sequence, Observation-Genetic Profile, and DiagnosticReport where possible. Several additional data elements were needed specifically for this use case, resulting in the development of a specific profile for reporting HLA genotyping results (Diagnostic Report Profile for HLA Genotyping Results).

* Minimum information for reporting next generation sequence genotyping (MIRING): Guidelines for reporting HLA and KIR genotyping via next generation sequencing (see [here](http://www.ncbi.nlm.nih.gov/pubmed/26407912))
* Histoimmunogenetics Markup Language 1.0: Reporting Next Generation Sequencing-based HLA and KIR Genotyping (see [here](http://www.ncbi.nlm.nih.gov/pubmed/26319908))

**Structure Diagram of HLA genotyping results Profile**

Both hla-genotyping-resultsGlstring and hla-genotyping-resultsHaploid are complex extensions.

The development of these extensions were informed from the allele-assignment structure found in HML. While allele assignment for individual loci can be reported in an Observation, here they are used to summarize the assignments at a report level. Two methods may be used for reporting HLA allele-assignments: Haploid and GL String.

Structure of hla-genotyping-resultsHaploid:

Haploid reporting is a method that is used to report an individual allele, or a list of possible alleles if the results are ambiguous, using NMDP Multiple Allele Codes (MAC) which is a shorthand for describing allele ambiguity (https://bioinformatics.bethematchclinical.org/hla-resources/allele-codes/). MACs are widely used for reporting HLA typing results because they are able to report long allele lists in a small amount of space, but the system has a number of shortcomings ([Tissue Antigens. 2013 Aug;82(2):106-12](http://www.ncbi.nlm.nih.gov/pubmed/23849068)). When reporting data using haploid, typical use is one or two haploid elements for a particular locus, but possibly more if multiple loci are covered (ex: two HLA-DRB1 haploids + one HLA-DRB3 haploid).

Within Haploid, Method indicates whether the general methodology is DNA based typing (e.g., Sequence Specific Primers (SSP), Sequence Specific Oligonucleotide Probes (SSOP), or Sequence Based Typing (SBT)), or SER which indicates serology based methods.

Example of using two hla-genotyping-resultsHaploid extensions to report a HLA-A genotype of HLA-A\*01:AB and HLA-A\*02:MN which expands to HLA-A\*01:01/HLA-A\*01:02 and HLA-A:02:01/HLA-A:02:02/HLA-A:02:03

<extension url="http://hl7.org/fhir/StructureDefinition/hla-genotyping-resultsHaploid" >

<extension url=Ã¢â‚¬ÂlocusÃ¢â‚¬Â>

<valueCodeableConcept>

<coding>

<system value="http://www.genenames.org"/>

<code value="4931"/>

<display value="HLA-A"/>

</coding>

<text value="HLA-A"/>

</valueCodeableConcept>

</extension>

<extension url=Ã¢â‚¬ÂtypeÃ¢â‚¬Â>

<valueCodeableConcept>

<coding>

<system value="https://bioinformatics.bethematchclinical.org/hla-resources/allele-codes/"/>

<code value="AB"/>

<display value="01/02"/>

</coding>

<text value="HLA-A\*01:AB"/>

</valueCodeableConcept>

</extension>

<extension url=Ã¢â‚¬ÂmethodÃ¢â‚¬Â>

<valueCodeableConcept>

<text value=Ã¢â‚¬ÂDNAÃ¢â‚¬Â/>

</valueCodeableConcept>

</extension>

</extension>

<extension url="http://hl7.org/fhir/StructureDefinition/hla-genotyping-resultsHaploid" >

<extension url=Ã¢â‚¬ÂlocusÃ¢â‚¬Â>

<valueCodeableConcept>

<coding>

<system value="http://www.genenames.org"/>

<code value="4931"/>

<display value="HLA-A"/>

</coding>

<text value="HLA-A"/>

</valueCodeableConcept>

</extension>

<extension url=Ã¢â‚¬ÂtypeÃ¢â‚¬Â>

<valueCodeableConcept>

<coding>

<system value="https://bioinformatics.bethematchclinical.org/hla-resources/allele-codes/"/>

<code value="AB"/>

<display value="01/02/03"/>

</coding>

<text value="HLA-A\*02:MN"/>

</valueCodeableConcept>

</extension>

<extension url=Ã¢â‚¬ÂmethodÃ¢â‚¬Â>

<valueCodeableConcept>

<text value=Ã¢â‚¬ÂDNAÃ¢â‚¬Â/>

</valueCodeableConcept>

</extension>

</extension>

Because of limitations with MAC, another method called GL Strings was developed that encodes the results in a text string with hierarchical set of operators to describe the relationships between alleles, lists of possible alleles, phased alleles, genotypes, lists of possible genotypes, and multilocus unphased genotypes, without losing typing information or increasing typing ambiguity. ([Tissue Antigens. 2013 Aug;82(2):106-12](http://www.ncbi.nlm.nih.gov/pubmed/23849068)).

The structure of HLA-genotyping-resultsGlstring:

While the string format is easily parsed into separate components to be rendered for user viewing, GL Strings by themselves are potentially quite long and difficult to read. It often advantageous to point to a URI which may return the GL String on demand to avoid manual data entry. This is available through the URI element in this extension.

Example of using the hla-genotyping-resultsGlstring to report a GL String in both text and URI formats

<extension url="http://hl7.org/fhir/StructureDefinition/hla-genotyping-resultsGlstring">

<extension url=Ã¢â‚¬ÂtextÃ¢â‚¬Â>

<valueString value="HLA-A\*01:01:01:01/HLA-A\*01:02+HLA-A\*02:01:02/HLA-A\*02:02:02/HLA-A\*02:03:01"/>

</extension>

<extension url=Ã¢â‚¬ÂuriÃ¢â‚¬Â>

<valueUri value="https://gl.nmdp.org/imgt-hla/3.23.0/genotype/1h"/>

</extension>

</extension>

Whether reporting in Haploid or GL String formats, it is important to identify the version of the IMGT/HLA allele database, as new HLA alleles are constantly being discovered and allele assignment is based on the known alleles at the time, and so results may need to be reinterpreted later. This is done through the Allele Database element.

Structure of hla-genotyping-resultsAlleleDatabase

Example of using hla-genotyping-resultsAlleleDatabase to report using the IMGT/HLA database, version 3.23

<extension url="http://hl7.org/fhir/StructureDefinition/hla-genotyping-resultsAlleleDatabase">

<valueCodeableConcept>

<coding>

<system value="http://www.ebi.ac.uk/ipd/imgt/hla/"/>

<version value="3.23"/>

</coding>

<text value="IMGT/HLA 3.23"/>

</valueCodeableConcept>

</extension>

Overall methodology may be reported using hla-genotyping-resultsMethod. Here a codeable concept may be used to refer to method entered into a public registry, such as the NCBI Genetic Test Registry, or a local private registry.

Structure of hla-genotyping-resultsMethod

Example of hla-genotyping-resultsMethod to report a lab test registered in the NCBI Genetic Test Registry

<extension uri="http://hl7.org/fhir/StructureDefinition/hla-genotyping-resultsMethod">

<valueCodeableConcept>

<coding>

<system value="http://www.ncbi.nlm.nih.gov/gtr/"/>

<code value="GTR000000000.0"/>

</coding>

<text value=Ã¢â‚¬ÂNGS based HLA-A genotypingÃ¢â‚¬Â

</valueCodeableConcept>

</extension>

**Usage Example**

[Example](file:///C:\temp\diagnosticreport-hla-genetics-results-example.html)of a HLA genotyping report

* that was requested in a Diagnostic Request,
* with results for HLA-A, HLA-B, HLA-C loci,
* by Sequence Based Typing (SBT) of exons 2 and 3,
* using Next Generation Sequencing technologies,
* allele assignment based on the IMGT/HLA database version 3.23,
* reported as a multilocus unphased genotype using GL Strings,
* and with genotyping results for individual loci available in separate Observations

Source: [XML](file:///C:\temp\diagnosticreport-hla-genetics-results-example.xml.html), [JSON](file:///C:\temp\diagnosticreport-hla-genetics-results-example.json.html), [Turtle](file:///C:\temp\diagnosticreport-hla-genetics-results-example.ttl.html).

In the above example, three Observations are referenced, each containing the details surrounding the genotyping of each gene/locus. Those observations would further refer to other observations (using Observation.related) that point to phased exon sequences which are the basis for individual allele assignment. This strategy is illustrated in the following figure for one gene, and sequencing two exons for each allele. Note that references from DiagnosticReport and Observation back to Patient/Donor and Specimen are not shown for clarity, although we would include them in the resource instances.

**Relationship among Sequence resource and genetics profiles**

**Pointers to Sequence resource**

Sequence resource is used to transmit and represent sequencing data. In FHIR Genomics, there are several genetics profiles containing internal pointers to Sequence resource for genetic data reporting. Here is the summary:

Sequence.pointer: A bundle of Sequence instances can be connected by Sequence.pointer to build sequence graph described in Section 3. Sequence.pointer will point to the next sequence block.

Sequence extension in Observation-genetics profile: Observation-genetics profile is used to report a genetic variant found in patients. Sequence extension contains a pointer to Sequence identifier which has related sequencing read info. It will provide reference/observed allele information and quality scores for each base/sequence block.

**Pointer to Observation resource (based on genetics profile)**

Observation-genetics based Observation resource is used for interpretative genetic data. Sequence resource and genetics profiles will use internal pointers to Observation-genetics-profile based Observation instance for variant report.

Sequence.observation: A pointer to genetics Observation instance which is used for interpretations of this sequence block. Interpretations are generally about genetic variant found in this sequence block.

DiagnosticReport.result: A pointer to Observation instance. This element is used for reporting genetic result.

Item extension in ProcedureRequest-genetics profile: It is used to describe the genetic test order.

FamilyMemberHistory: A pointer to genetics Observation instance to report genetic test results of family member

**Observation vs. Sequence**

Sequence is used for raw sequencing data while genetic-profile-based Observation resource is used for reporting interpretative genetic information (eg. DNA/amino acid variant detected in the patient). Although both Sequence and Observation-genetics profiles contain variant info, they are used differently: Sequence.variant is used for encoding the sequence block; the Observation-genetics profile is mainly used for a variants test result or interpretations of raw data in Sequence resource.

**Usage example**

This picture depicts the logical relationship among these resources. And the definition of pointers are described above:

Here are the details and examples mentioned above.

|  |  |  |
| --- | --- | --- |
| **Resource** | **Details of data structures and pointers** | **Links of resources example** |
| Sequence | [Sequence Data Structure](file:///C:\temp\main-pages.html#sequence) | [Example Here](file:///C:\temp\sequence-example.html) |
| Observation | [Observation Data Structure](file:///C:\temp\main-pages.html#observation-genetics) | [Example Here](file:///C:\temp\observation-example.html) |
| ProcedureRequest | [ProcedureRequest Data Structure](file:///C:\temp\main-pages.html#procedurerequest-genetics) | [Example Here](file:///C:\temp\procedurerequest-example-lipid.html) |
| DiagnosticReport | [DiagnosticReport Data Structure](file:///C:\temp\main-pages.html#diagnosticreport-genetics) | [Example Here](file:///C:\temp\diagnosticreport-example.html) |

Here is a specific use case to depict the whole story of FHIR in the clinical genomics setting: a [ProcedureRequest](file:///C:\temp\procedurerequest-example-pgx.html) is requested, then a [Diagnosticreport](file:///C:\temp\diagnosticreport-example-pgx.html) are reported. Two [Sequence](file:///C:\temp\sequence-example-pgx-1.html) instance, two [Observation-haplotype](file:///C:\temp\observation-example-haplotype1.html) instance, a [Observation-diplotype](file:///C:\temp\observation-example-diplotype1.html) and a [Observation-phenotype](file:///C:\temp\observation-example-phenotype.html) for further interpretation are also created. (You can see the xml code and json code if you click the link.)

* **Haplotype:** A haplotype is a set of DNA variations, or polymorphisms, that tend to be inherited together. A haplotype can refer to a combination of alleles or to a set of single nucleotide polymorphisms (SNPs) found on the same chromosome.
* **Diplotype:** A diplotype is defined as a specific combination of two haplotypes.
* **Genotype:** A genotype is an individualÃ¢â‚¬â„¢s collection of genes.
* **Phenotype:** A phenotype is the composite of an organismÃ¢â‚¬â„¢s observable characteristics or traits, such as its morphology, development, biochemical or physiological properties, behavior and products of behavior.

You can see the relationship among these resources in following picture:

**HL7 Domain Analysis Use Cases**

Simulated use of the proposed Sequence resource and profiles on other FHIR resources are shown for many of the examples in Section 5 Ã¢â‚¬â€œ Use Case Scenarios in the [HL7 Domain Analysis Model (DAM): Clinical Genomics, Release 1, September 2014 Informative Ballot](https://www.hl7.org/documentcenter/public_temp_649E1787-1C23-BA17-0CA4C354EB3E41D5/wg/clingenomics/docs/V3DAM_CG_CLINSEQ_R1_O1_2013JAN.pdf).

We show use of the FHIR design we are proposing for these cases:

1. Specimen Identification
   1. Get references to all variants obtained from germline analysis.
   2. Explore antibacterial drug resistance over time (non-patient DNA)
2. Clinical Sequencing Ã¢â‚¬â€œ Germline Testing
3. Cancer Profiling
4. Decision Making Tools Ã¢â‚¬â€œ Family History and Drug Dosage Calculators
   1. Today clinicians translate (i.e. manually re-enter) genetic data into tools for decision making. This includes family history tools and drug dosage calculators. In the future, this data will automatically be incorporated into clinical decision making tools.
   2. Get pharmacogenomic reporting and disease risk or diagnosis associated with variant
5. Public Health
6. Clinical and Research Data Warehouses

**Specimen Identification**

* Germline testing for biomarkers/mutations (usually inherited)
* Tumor testing for biomarkers/mutations (somatic / tumor specific)
* Pediatric Testing
* Infectious Disease Testing

**Get references to all variants obtained from germline analysis**

GET /Observation?

\_profile=http://hl7.org/fhir/StructureDefinition/observation-genetics&

GenomicSourceClass=http://loinc.org|LA6683-2

\*Return a bundle of genetics-profile-based Observation instances

**Clinical Sequencing Ã¢â‚¬â€œ Germline Testing**

For a chart review, get references to all DNA sequences related to mutations with an interpretation Ã¢â‚¬Å“Unknown SignificanceÃ¢â‚¬Â

GET /Observation?

subject=123&

component-code-value-[x]= http://loinc.org|53037-8$LA6682-4

53037-8: LOINC code for Ã¢â‚¬Å“Genetic disease sequence variant interpretationÃ¢â‚¬Â

LA6682-4: LOINC answer code for Ã¢â‚¬Å“Unknown SignificanceÃ¢â‚¬Â.

**Cancer Profiling**

The goal of this profile methodology is to get references from all variants obtained from somatic analysis. Changes in the population of cells with particular mutations will change overtime as well as in conjunction with events such as therapy. For instance, targeted chemotherapy may kill a specific population of cancer cells with specific mutations and other cancer cell populations may survive and continue to divide. Therefore, clearly annotating these specimens as somatic variants and capturing annotations related to a time relevant to a treatment timeline may be critical for analysis.

GET /Observation?

\_profile=http://hl7.org/fhir/StructureDefinition/observation-genetics&

GenomicsSourceClass=http://hl7.org/fhir/LOINC-48002-0-answerlist|LA6684-0&

date=2015-07-04&

subject=123

**Decision Making Tools Ã¢â‚¬â€œ Family History and Drug Dosage Calculators**

Today clinicians translate (i.e. manually re-enter) genetic data into tools for decision making. This includes family history tools and drug dosage calculators. In the future, this data will automatically be incorporated into clinical decision making tools.

Get family history related to one observation:

GET /DiagnosticReport?

\_profile=http://hl7.org/fhir/StructureDefinition/diagnosticreport-genetic&

subject=123

**Get pharmacogenomic reporting and disease risk or diagnosis associated with variant**

GET /Observation?

\_profile=http://hl7.org/fhir/StructureDefinition/observation-genetics&

subject=123&

Interpretation.component-code= http://loinc.org|51963-7

51963-7 : LOINC code for Ã¢â‚¬Å“Medication Assessed Ã¢â‚¬Â

**Public Health**

Today Registrars manually translate clinical data into public health reporting systems. This data is used to monitor and improve public health (e.g. surveillance and clinical research). In the future, this data will be extracted from the EHR in an automated (or semi-automated) fashion.

For a breast cancer clinical genomic study, get all genetic-profile-based observations of patients with breast cancer:

GET /DiagnosticReport?

\_profile=http://hl7.org/fhir/StructureDefinition/diagnosticreport-genetic&

AssessedCondition.code=http://snomed.info/sct|254837009

**Clinical and Research Data Warehouses**

Health data warehousing should persist data in its standardized formats, while allowing users to export subsets of the data in the warehouse into multiple Ã¢â‚¬Ëœdata martsÃ¢â‚¬â„¢, optimized for specific use cases, analysis type or reporting needs.

Get all genetic-profile-based observations of patients with the variant c.181T>G

GET /Observation?

\_profile=http://hl7.org/fhir/StructureDefinition/observation-genetics&

DNAVariantID=http://www.ncbi.nlm.nih.gov/projects/SNP|rs58238560

**Other Genomics Use Cases**

During the development of the FHIR Genomics design, CGWG participants have commented on the earlier use cases and/or proposed new use cases. In this section, we list describe of these new cases and demonstrate how the proposed design will address them.

**Get a patientÃ¢â‚¬â„¢s family members genetics report**

This example is proposed by Kevin Hughes. Family history is useful for clinicians to know more about the condition of the patient.

Get /FamilyMemberHistory?

\_profile=http://hl7.org/fhir/StructureDefinition/familymemberhistory-genetic&

patient=123

**Panel**

Search for results from nephrotic syndrome panel

Get /DiagnosticReport&

\_profile=http://hl7.org/fhir/StructureDefinition/diagnosticreport-genetic&

code=N0336&

patient=123

**Sequence Test Metrics**

**Clinical Genomic Apps**

The idea for a Sequence resource grew out, in part, the SMART Platforms Project, which explored creating clinical genomic apps to integrated traditional EMR clinical data and genomic data to show data visualization and analysis, including CDS that depended upon both types of data. Below are a couple of examples. Several apps have already been designed including Genomics Advisor, SMART Precision Cancer Medicine, and Diabetes Bear EMR. Below, one of these apps will be described. To include other apps in this section, please feel free to add a note on it and how it uses FHIR/Genomics calls.

**Genomics Advisor**

The SMART on FHIR Genomics Advisor was an app incorporating genomics data to show risk of disease, drug susceptibility, and related conditions based upon genotype. Technically, this app was architected (see below) by combining data from independent data services, a SMART on FHIR clinical server for clinical information and one for a SMART on FHIR Genomics data server for genomic data. The set of FHIR API calls that are necessary to support this app are shown below:

The presentation of the app looks like this:

**Appendix 1: From DSTU2 Genomics to STU3**

The table shows how the data elements in the DSTU2 Observation resource as extended by the Standard Genetics Profile would be mapped to the new Sequence resource.

|  |  |  |
| --- | --- | --- |
| **DSTU2 Standard Genetics profile on Observation** | **Observation-genetics profile** | **Maps to Sequence (Proposed)** |
| geneticsGenomeBuild | Sequence extension | referenceSeq.genomicsBuild |
| geneticsChromosome | Sequence extension | referenceSeq.chromosome |
| geneticsGenomicsStart | Sequence extension | variant.start |
| geneticsGenomicsStop | Sequence extension | variant.end |
| geneticsReferenceAllele | Sequence extension | variant.referenceAllele |
| geneticsObservedAllele | Sequence extension | variant.observedAllele |
| geneticsSpecies | Sequence extension | specimen |
| geneticsAllelicState | Allelic State extension |  |
| geneticsAllelicFrequency | Allelic Frequency extension |  |
| geneticsReadCoverage | Sequence extension | readCoverage |
| geneticsCopyNumberEvent | Copy number event extension |  |
| geneticsTranscriptReferenceSequenceId | Transcript reference sequence identifier extension |  |
| geneticsProteinReferenceSequenceId | Protein reference sequence identifier |  |
| geneticsCIGAR | Sequence extension | variant.cigar |
| geneticsVariationId | DNA variation id extension |  |
| geneticsVariationType | DNA sequence variation type extension |  |
| geneticsAminoAcidChange | Amino acid change extension |  |
| geneticsAminoAcidChangeType | Amino acid change type extension |  |
| geneticsGene | Gene extension |  |
| geneticsDNARegionName | DNA region name extension |  |
| geneticsGenomicSourceClass | Genomic source class extension |  |

**header.html**

**help.html**

**Documentation Assistance And Glossary**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

On this page:

* Ballot Status Help
* Glossary
* Documentation Style Guidelines

**history.html**

See [below](file:///C:\temp\main-pages.html#history) for version history details.

**Version History**

This is the developmental version of FHIR. The only changes tracked here are the changes made after the publication of the first DSTU release. For earlier changes, see the [DSTU #1 Version History](http://hl7.org/fhir/history.html). Note that a full archive history of everything is available [through the HL7 gForge SVN archives](http://wiki.hl7.org/index.php?title=FHIR).

**How FHIR Versioning works**

The FHIR version policy is based on [Semantic versioning](http://semver.org/), but with some differences due to the fact that FHIR is a specification, not a software API.

There is a single development version of FHIR. This undergoes cycles of development as managed by HL7. Each major cycle of development is concluded by a formal ballot, and then a new specification is published. In version control terms, each published specification is a branch off the development trunk, which may then itself undergo further change as HL7 maintains the published specification (though such changes are usually extremely minimal).

The following kinds of changes may be made to the specification:

* Non-substantive changes do not cause changes in any conformant application. For example, section renumbering, broken links, style corrections, typos, and clarifications that do not change the meaning. Some corrections may be judged not to create any expectation of change to a conformant application.
* Substantive changes that are not breaking. These introduce new functionality - changes to the specification that create new capabilities - but would not render existing applications non-conformant if they do not change.
* Breaking changes would mean that previously conformant applications are no longer conformant

Note that the following are, by definition, considered non-breaking changes, even though some implementations (including the reference implementations) may not be able to handle some consequences of these changes without error:

* creation of new resources
* adding new optional elements in an existing resource
* defining new data types
* Adding new API operations

Also, the examples are never substantive. Every effort is made to ensure that FHIR examples are correct, but changes to the examples in the specification are not considered substantive.

Each FHIR version is identified by a string composed from 4 parts: publication.major.minor.revision.

|  |  |
| --- | --- |
| publication | * Incremented when HL7 publishes FHIR as an updated specification, e.g. a STU or normative version of FHIR * HL7 plans to do this every 1 to 2 years * The first STU was version 0 |
| major | * Increments every time a breaking change is made * When a new publication is made, this is reset to 0 in the publication, and 1 in the development branch * Since HL7 does not make breaking changes as technical corrections to a published specification, these versions of FHIR always have a version number X.0.n.r * Because the development version is the subject of ongoing analysis, debate, ballot and repeated alterations, breaking changes are to be expected |
| minor | * Increments every time a substantive change is made * Resets to 0 any time the major version changes |
| revision | * Incremented any time a change is made to the specification * At present, this is the SVN version number (this allows anyone to reconstruct the source from which the version was built) |

Additional notes:

* Changes to a formally published specification (except for minor publishing corrections, such as correcting broken links) are only made via announced technical corrections
* The reference implementations have 2 versions - the version of the specification that they implement, and their own version. Consult the reference implementation documentation for policy regarding this version number
* The specification published by the continuous integration service (<http://build.fhir.org/>) build may not conform to this version policy, but the versions published on the HL7 web site will (see [Directory of published versions](http://hl7.org/fhir/directory.html))
* The first DSTU was published prior to these rules being agreed as v0.80-2286. This has been updated to 0.0.81.2382 as a technical correction to align with this policy on 9-May 2014

**Version History since DSTU #1**

Major Milestones:

|  |  |  |
| --- | --- | --- |
| [Oct 24, 2015](http://hl7.org/fhir/DSTU2/index.html) | 1.0.2 | DSTU2 (Second Draft Standard for Trial Use) |
| [Sept 30, 2014](http://hl7.org/fhir/DSTU2/index.html) | 0.0.82 | DSTU1 (First Draft Standard for Trial Use) |

This table lists substantive changes only.

|  |  |
| --- | --- |
| **Version** | **Changes** |
| 1.9.0 | **FHIR Rolling Build, 2016 Dec onwards** |
| 1.8.0 | **FHIR QA Release, Dec 6, 2016**  STU 3 Candidate, and also the basis for several implementation guide ballots and the [San Antonio Jan 2017 Connectathon](http://wiki.hl7.org/index.php?title=FHIR_Connectathon_14)  **Breaking Changes Summary:**   * Change [HumanName](file:///C:\temp\datatypes.html#HumanName).family from 0..\* to 0..1 * Rename MedicationOrder and DiagnosticOrder to [MedicationRequest](file:///C:\temp\medicationrequest.html) and DiagnosticRequest and make a series of changes to align with the [Request Pattern](file:///C:\temp\request.html) * Add cross resource search + rework composite parameter expressions * Harmonise resource metadata across the Terminology, Conformance, and Knowledge resources * Change the way [versioned references to canonical URLs](file:///C:\temp\references.html#canonical) are handled * REST - servers must [return a location header](file:///C:\temp\http.html#create) on a create * [Condition](file:///C:\temp\condition.html) status elements reworked   These are only the more significant changes, there were many 100s of changes made in response to ballot comments and ongoing implementation experience. These include breaking changes to may resources. Structural changes are [summarised here](file:///C:\temp\diff.html), and [Transforms between DSTU 2 and STU 3](file:///C:\temp\r2maps.html) are provided for many resources. |
| 1.6.0 | **FHIR STU3 ballot, Aug 11, 2016**  STU 3 ballot version, and also the basis for the [Baltimore Sept 2016 Connectathon](http://wiki.hl7.org/index.php?title=FHIR_Connectathon_13)  **Publication Changes:**   * Move the US Realm implementation guides out of the specification (see <http://www.fhir.org/guides/registry>) * Rework the home page, and introduce modules as a way to navigate the specification + add significant new implementer advice (e.g. "[Getting Started](file:///C:\temp\modules.html)") * Change the Navigation menu to focus on indexes to the content in the specification * Move FHIRPath expression language to it's [long term home](http://hl7.org/fluentpath)   **Technical Changes:**   * RESTful API   + Change the [FHIR mime type](file:///C:\temp\http.html#mime-type) to application/fhir+xml|json instead of application/xml|json+fhir (**breaking change**, for conformance to W3C+IETF rules)   + Add new uses for the Prefer header (return OperationOutcome, and manage behavior related to unknown / unsupported search parameters   + Deprecate use of the OPTIONS command to retrieve the CapabilityStatement   + Add support for conditional references to the transaction interaction   + Add reverse chaining * Formats: No change to XML + JSON formats. Add [Turtle Format](file:///C:\temp\rdf.html), and related validation tools * For Data Types and Resources, there is [formal difference](file:///C:\temp\diff.html) analysis from DSTU2 (also found throughout the specification), and [transforms between DSTU 2 and STU 3](file:///C:\temp\r2maps.html) are provided for many resources |
| 1.5.0 | **FHIR Current Build Update, July 8 2016**  Update current version to 1.5 to prevent confusion with implementations still running the May 2016 version.   * There are many changes in this version from 1.4.0, but no formal list is provided. A formal list will be provided for 1.6 (as a diff to 1.4) |
| 1.4.0 | **FHIR Connectathon 12 Snapshot, Mar 30 2016**  Frozen base for Connectathon 12 & For Comment ballots:   * FHIR API and Serialization Format Enhancment   + The [\_sort parameter](file:///C:\temp\search.html#sort) has been reworked   + New or extended search parameters for read, search or history: \_type, \_at, \_summary, \_elements   + Change [RDF representation](file:///C:\temp\rdf.html) - now ready for trial * Conformance resources with significant breaking changes and behavior:   + [StructureDefinition](file:///C:\temp\structuredefinition.html) - Type Handling: Changes to how the structure definition types work: **baseType** *replaces* **constrainedType** & **baseDefinition** *replaces* **base**   + [CodeSystem](file:///C:\temp\codesystem.html) now separated out from [ValueSet](file:///C:\temp\valueset.html): CodeSystem is now a first class resource type in order to support use throughout the FHIR eco-system, such as to support value set expansion and validation. It is intended to be used for distributing the smaller ad-hoc code systems that are ubiqutiously encountered through out the healthcare process. * Other resources with significant breaking changes and behavior:   + [AllergyIntolerance](file:///C:\temp\allergyintolerance.html)   + [Appointment](file:///C:\temp\appointment.html)   + [AuditEvent](file:///C:\temp\auditevent.html)   + [Claim](file:///C:\temp\claim.html)   + [ClaimResponse](file:///C:\temp\claimresponse.html)   + [Device](file:///C:\temp\device.html)   + [ConceptMap](file:///C:\temp\conceptmap.html)   + [Conformance](file:///C:\temp\capabilitystatement.html) (now CapabilityStatement)   + [Contract](file:///C:\temp\contract.html)   + [Device](file:///C:\temp\device.html)   + DiagnosticRequest   + [EligibilityRequest](file:///C:\temp\eligibilityrequest.html)   + [EligibilityResponse](file:///C:\temp\eligibilityresponse.html)   + [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html)   + [ImagingManifest](file:///C:\temp\imagingmanifest.html)   + [Library](file:///C:\temp\library.html)   + [Measure](file:///C:\temp\measure.html)   + [Medication](file:///C:\temp\medication.html)   + [MedicationAdministration](file:///C:\temp\medicationadministration.html)   + [MedicationStatement](file:///C:\temp\medicationstatement.html)   + [Observation](file:///C:\temp\observation.html): the introduction of the Vital Signs Profile (proposed to be mandatory)   + [OperationDefinition](file:///C:\temp\operationdefinition.html)   + [Patient](file:///C:\temp\patient.html)   + [PaymentNotice](file:///C:\temp\paymentnotice.html)   + [PaymentReconciliation](file:///C:\temp\paymentreconciliation.html)   + [Practitioner](file:///C:\temp\practitioner.html)   + [ProcessRequest](file:///C:\temp\processrequest.html)   + [ProcessResponse](file:///C:\temp\processresponse.html)   + [Provenance](file:///C:\temp\provenance.html)   + [ReferralRequest](file:///C:\temp\referralrequest.html)   + [Schedule](file:///C:\temp\schedule.html)   + [Sequence](file:///C:\temp\sequence.html)   + [Slot](file:///C:\temp\slot.html)   + [StructureDefinition](file:///C:\temp\structuredefinition.html)   + [TestScript](file:///C:\temp\testscript.html)   + [ValueSet](file:///C:\temp\valueset.html) * Add new **draft** resources:   + [CareTeam](file:///C:\temp\careteam.html)   + [CodeSystem](file:///C:\temp\codesystem.html)   + [CompartmentDefinition](file:///C:\temp\compartmentdefinition.html)   + [Linkage](file:///C:\temp\linkage.html)   + [MeasureReport](file:///C:\temp\measurereport.html)   + [PractitionerRole](file:///C:\temp\practitionerrole.html)   + Protocol   + [StructureMap](file:///C:\temp\structuremap.html)   + [Task](file:///C:\temp\task.html) |
| 1.2.0 | **FHIR Connectathon 11 Snapshot, Dec 11 2015**  Frozen base for Connectathon 11:   * Remove GuidanceRequest * Add new **draft** resources: [Sequence](file:///C:\temp\sequence.html), [ExpansionProfile](file:///C:\temp\expansionprofile.html) * Modifications to Financial Resource & TestScript resource   Note: this version is temporary, and will be removed after Connectathon 11 is complete |
| 1.1.0 | **GAO Ballot + technical corrections, Dec 2 2015**  A ballot publication for the GAO Ballot that also includes:   * Various technical corrections to the generated snapshots * Populate FHIRPath expressions in extensions for all search parameters and invariants * Add new **draft** resources: GuidanceRequest, [GuidanceResponse](file:///C:\temp\guidanceresponse.html), ModuleMetadata, ModuleDefinition, [Library](file:///C:\temp\library.html), DecisionSupportServiceModule, DecisionSupportRule, OrderSet, [Measure](file:///C:\temp\measure.html) * Major restructure of [Questionnaire](file:///C:\temp\questionnaire.html) |
| 1.0.2 | **Technical Correction 1, Oct 24 2015**  A series of technical corrections to the specification following extensive review:   * Corrections to Extension cardinalities in implementation guides * Corrections in the conformance resources that support the specifications * Correct several erroneous invariants * Various typos, broken links, and fixes in examples * For a comprehensive list of corrections, see the [Task list for FHIR DSTU2 Technical Correction 1](http://wiki.hl7.org/index.php?title=Task_list_for_FHIR_DSTU2_Technical_Correction_1) |
| 1.0.1 | **DSTU 2, Sept 23 2015**  Changes of significance during the QA process:   * Remove the Clinical Quality Improvement Framework (CQIF) from this published version * made fixes to generated schematrons * updated generated comformance resources (StructureDefinitions and SearchParameters) so they were consistent with the specification * Many spelling / grammar / broken link fixes |
| 1.0.0 | **DSTU 2 QA Preview, Aug 31 2015**  This version had extensive change as a result of the May 2015 DSTU ballot, ongoing testing, and the open change proposals (1317 gForge tasks). The extent of the changes is best illustrated by the number of the [list of changes labelled 'breaking change'](http://wiki.hl7.org/index.php?title=FHIR_Breaking_changes_between_DSTU_2_ballot_and_final) - 158 changes of 1317 total tasks. Below is a list of the most important changes:   * General: introduced the [maturity framework](file:///C:\temp\resource.html#maturity) * [RESTful API](file:///C:\temp\http.html): add batch, several clarifications around versioning & transactional integrity, changed Bundle URL resolution rules * [Search](file:///C:\temp\search.html): changed the way <> etc.works, added \_list parameter, changed rules around contained and included resources * Formats: added a [note about whitespace in XML](file:///C:\temp\xml.html), added [code generation schemas](file:///C:\temp\xml.html#schema-gen) * Data Types:   + New data types: [markdown](file:///C:\temp\datatypes.html#markdown), [Annotation](file:///C:\temp\datatypes.html#Annotation),   + changed data types: [Coding](file:///C:\temp\datatypes.html#Coding), [Quantity](file:///C:\temp\datatypes.html#Quantity), [Signature](file:///C:\temp\datatypes.html#Signature), [Timing](file:///C:\temp\datatypes.html#Timing), [Address](file:///C:\temp\datatypes.html#Address), [ContactPoint](file:///C:\temp\datatypes.html#ContactPoint)   + changes to [ElementDefinition](file:///C:\temp\elementdefinition.html): add base, make type.profile repeat, remove invariant.name and replace with invariant.requirements, remove binding.name, add min/max value * Resources:   + New resources: [Account](file:///C:\temp\account.html), [ImplementationGuide](file:///C:\temp\implementationguide.html), [TestScript](file:///C:\temp\testscript.html)   + renamed: Contraindication -> [DetectedIssue](file:///C:\temp\detectedissue.html), MedicationPrescription -> [MedicationRequest](file:///C:\temp\medicationrequest.html), QuestionnaireAnswers -> [QuestionnaireResponse](file:///C:\temp\questionnaireresponse.html)   + removed: Supply   + changed: almost all resources - too many to list (1317 tasks worth of changes) - add, remove elements, change types, references, definitions & value sets, re-order elements, provide much more documentation and new examples * Implementation Guide:   + Move [Argonaut content](http://argonautwiki.hl7.org/index.php?title=Main_Page) out |
| 0.5.0 | **DSTU Ballot, May 2015**  This version had extensive change as a result of the January 2015 Draft ballot, ongoing testing, and the open change proposals (over 800 gForge tasks). The list below is a summary of the major changes to resource content. It shows only a limited number of the overall changes.  **Enumerations**   * All spaces removed * Extensive content changes not noted here   **New Data Types**   * [unsignedInt](file:///C:\temp\datatypes.html#unsignedInt) * [positiveInt](file:///C:\temp\datatypes.html#positiveInt) * [Signature](file:///C:\temp\datatypes.html#Signature) * [Meta](file:///C:\temp\resource.html#Meta)   **Changed Data Types**   * [Coding](file:///C:\temp\datatypes.html#Coding) - remove valueSet property * [Attachment](file:///C:\temp\datatypes.html#Attachment) - add creation * [Identifier](file:///C:\temp\datatypes.html#Identifier) - replace label with type * [Timing](file:///C:\temp\datatypes.html#Timing) - major rework of content * [ElementDefinition](file:///C:\temp\elementdefinition.html) - add label, code, rename 'formal' to definition, rename synonym to alias, add language to mapping, remove conformance and isExtensible and replace with strength   **New Resources**   * [BodySite](file:///C:\temp\bodysite.html) * [Claim](file:///C:\temp\claim.html) * [ProcessRequest](file:///C:\temp\processrequest.html) * [ProcessResponse](file:///C:\temp\processresponse.html)   **Removed Resources**   * CarePlan2 -> collapsed into CarePlan * FamilyHistory -> broken up into FamilyMemberHistory * InstitutionalClaim, OralHealthClaim, PharmacyClaim, ProfessionalClaim, VisionClaim -> collapsed into Claim * Other - use Basic instead * PendedRequest,Readjudicate, Reversal, StatusRequest, StatusResponse - use ProcessRequest/Response instead * SupportingDocumentation - use DocumentManifest instead   **Renamed Resources**   * Alert -> Flag: 'alert' made people think it was an action like an alarm * SecurityEvent -> AuditEvent: it wasn't just for security purposes * ClinicalAssessment -> ClinicalImpression: people got confused with 'assessment' tools like APGAR score * Profile -> StructureDefinition: 'Profile' is the process, a package of statements   **Changes Inside Resources**   * [Parameters](file:///C:\temp\parameters.html) - allow parameter.part to contain a resource * [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) - rename subject to patient * [Appointment](file:///C:\temp\appointment.html) - remove lastModifiedBy/lastModified, add location * [AppointmentResponse](file:///C:\temp\appointmentresponse.html) - remove lastModifiedBy/lastModified, add rename individual to actor * [AuditEvent](file:///C:\temp\auditevent.html) - add .event.purposeOfEvent, participant.location, .policy, and .purposeOfUse * [Bundle](file:///C:\temp\bundle.html) - major reorganization * [CarePlan](file:///C:\temp\careplan.html) - pull goal out + other reorganization * [ClinicalImpression](file:///C:\temp\clinicalimpression.html) - add status, replace careplan & referral with trigger, rename diagnosis to finding, make plan 0..\*, * [Composition](file:///C:\temp\composition.html) - change .section.content to refer to List only, not any * [ConceptMap](file:///C:\temp\conceptmap.html) - change identifier to url, add useContext, change telecom to contact, * [Condition](file:///C:\temp\condition.html) - rename subject to patient, rename status to clinicalStatus, change to bodySite = code or Reference(BodySite), rename .codeableConcept to .code * [Conformance](file:///C:\temp\capabilitystatement.html) (now CapabilityStatement) - change identifier to url, add useContext, change telecom to contact, add requirements and copyright, add support for conditional operations, * [Contract](file:///C:\temp\contract.html) - extensive rewrite * [Coverage](file:///C:\temp\coverage.html) - add bin, subscriberId * [DataElement](file:///C:\temp\dataelement.html) - total rewrite to use ElementDefinition * [Device](file:///C:\temp\device.html) - add status, manufactureDate * [DeviceMetric](file:///C:\temp\devicemetric.html) - rename operationalState to operationalStatus, add measurementMode, rename calibrationInfo to calibration, change color to an enumerations * [DeviceRequest](file:///C:\temp\devicerequest.html)/[DeviceUseStatement](file:///C:\temp\deviceusestatement.html) - change to bodySite = code or Reference(BodySite) * DiagnosticRequest- change to bodySite = code or Reference(BodySite) * [DiagnosticReport](file:///C:\temp\diagnosticreport.html) - add encounter * [DocumentManifest](file:///C:\temp\documentmanifest.html) - add options for how content is referred to * [DocumentReference](file:///C:\temp\documentreference.html) - add format, remove policyManager, make content : Attachment, and remove several related attributes, remove service reference and add context.practiceSetting, sourcePatientInfo, and related * [Encounter](file:///C:\temp\encounter.html) - add incomingReferralRequest, allow reason to repeat, rename diet to dietPreference * [EpisodeOfCare](file:///C:\temp\episodeofcare.html) - rename currentStatus to status, allow referralRequest to repeat, * [Flag](file:///C:\temp\flag.html) - rename subject to patient, change from note to code * [Goal](file:///C:\temp\goal.html) - add targetDate, statusDate, author, priority * [HealthcareService](file:///C:\temp\healthcareservice.html) - extensive rewrite * ImagingObjectSelection - remove retrieveAETitle, rename retrieveUrl to url, add frames * [ImagingStudy](file:///C:\temp\imagingstudy.html) - add laterality, change url to attachment * [Immunization](file:///C:\temp\immunization.html) - add encounter, rename subject to patient, rename refusedIndicator to wasNotGiven, rename refusalReason to reasonNotGiven * [ImmunizationRecommendation](file:///C:\temp\immunizationrecommendation.html) - rename subject to patient * [List](file:///C:\temp\list.html) - add title, status, change ordered to orderedBy, add note * [Location](file:///C:\temp\location.html) - remove status * [Media](file:///C:\temp\media.html) - remove created (-> Attachment) * [Medication](file:///C:\temp\medication.html) - add batch * [MedicationAdministration](file:///C:\temp\medicationadministration.html) - add reasonGiven, note, text. remove timing & maxDosePerPeriod * [MedicationDispense](file:///C:\temp\medicationdispense.html) - collapse to a single dispense, add daysSupply, note and substitution, change quantity to allow range * [MedicationRequest](file:///C:\temp\medicationrequest.html) - add note, change quantity to allow range, * [MedicationStatement](file:///C:\temp\medicationstatement.html) - add informationSource, status, dateAsserted, replace whenGiven with effective[x], remove device, add dosage.text * [NamingSystem](file:///C:\temp\namingsystem.html) - add date, publisher, * [NutritionOrder](file:///C:\temp\nutritionorder.html) - extensive rewrite * [Observation](file:///C:\temp\observation.html) - change name to code, allow more types of value[x], change type of dataAbsentReason, change to bodySite = code or Reference(BodySite), allow identifier to repeat, add device, * [OperationDefinition](file:///C:\temp\operationdefinition.html) - change identifier to url, add useContext, change telecom to contact, change name to title, add reuqirements, idempotent, * [OperationOutcome](file:///C:\temp\operationoutcome.html) - change type of .issue.type * OrderResponse - rename code to orderStatus * [Organization](file:///C:\temp\organization.html) - remove location and contact.gender * [Patient](file:///C:\temp\patient.html) - communication to allow 'preferred' * [Person](file:///C:\temp\person.html) - rename other to target * [Practitioner](file:///C:\temp\practitioner.html) - change type of birthDate, allow multiple roles per practitioner * [Procedure](file:///C:\temp\procedure.html) - add status and category, change to bodySite = code or Reference(BodySite), allow date to be period too, add location, change followUp to code 0..\*, add device tracking * [ProcedureRequest](file:///C:\temp\procedurerequest.html) - change to bodySite = code or Reference(BodySite) * [Provenance](file:///C:\temp\provenance.html) - change integritySignature to signature & make it a type, allow reference by Reference as well as URI * [Questionnaire](file:///C:\temp\questionnaire.html) - add telecom * [Schedule](file:///C:\temp\schedule.html) - move lastModified * [SearchParameter](file:///C:\temp\searchparameter.html) - change telecom to contact, add status, experimental, date, * [Slot](file:///C:\temp\slot.html) - move lastModified * [Specimen](file:///C:\temp\specimen.html) - change source to parent, change to bodySite = code or Reference(BodySite) * [StructureDefinition](file:///C:\temp\structuredefinition.html) - complete rewrite * [Subscription](file:///C:\temp\subscription.html) - change type of tag, reanme url to endPoint, * [ValueSet](file:///C:\temp\valueset.html) - change identifier to url, add useContext, change telecom to contact, replace purpose with useContext, add requirements, rename stableDate to lockedDate, change type of expansion.identifier, add expansion parameters |
| 0.4.0 | **Draft For Comment, January 2015 Ballot**  Breaking Changes (full list):   * Replace atom and taglist with a native [Bundle](file:///C:\temp\bundle.html) format ([3728](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3728), [3558](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3558), [2889](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=2889)) (and also [Binary](file:///C:\temp\binary.html)) * JSON: change how extensions are represented ([3471](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3471)) * RESTful API: change how version specific upgrades work ([3451](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3451)) * DataTypes: * Rename Schedule to [Timing](file:///C:\temp\datatypes.html#Timing) ([3536](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3536), [3236](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3236)) * Rename Contact to [ContactPoint](file:///C:\temp\datatypes.html#ContactPoint) ([3533](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3533)) and swap order of elements ([3108](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3108))) * [Address](file:///C:\temp\datatypes.html#Address) - change zip to postCode ([2888](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=2888)) * [Quantity](file:///C:\temp\datatypes.html#Quantity): Correct schema spelling for "QuantityCompararator" ([3531](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3531)) * Change allowable values for the [id](file:///C:\temp\datatypes.html#id) type to include capital letters, and allow up to 64 chars ([3750](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3750)) * Restructure [Profile](file:///C:\temp\profile.html) - only one structure, and pull [ExtensionDefinition](file:///C:\temp\structuredefinition.html) out of Profile (3647, 3498), and pull [SearchParameter](file:///C:\temp\searchparameter.html) out ([3626](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3626)) * [Profile](file:///C:\temp\profile.html): allow 0..\* discriminator ([3131](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3131)), and change the way discriminators work across resource boundaries ([3124](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3124)) + generate multiple types properly ([2856](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=2856)) * remove \_validate interaction, and replace with $validate operation ([3686](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3686)) * [Patient](file:///C:\temp\patient.html): separate birth time from birthDate ([3731](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3731)), Change Administrative Gender from a CodableConcept to a Code. Also fixed the values as male|female|other|unknown with mappings to v2 and v3 ([3070](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3070)) * [DocumentReference](file:///C:\temp\documentreference.html): change encoding of Hash to Base64 ([3291](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3291)) * [Group](file:///C:\temp\group.html): rename header to title ([3126](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3126)) * [Condition](file:///C:\temp\condition.html): split relatedItem into two ([3111](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3111)) * [Questionnaire](file:///C:\temp\questionnaire.html): drop questionnaire.group.question.remarks ([3255](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3255)) and move omitReason from extension to base resource ([3260](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3260)) * [QuestionnaireResponse](file:///C:\temp\questionnaireresponse.html): allow multiple answers ([3146](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3146)) * [ValueSet](file:///C:\temp\valueset.html): replace ValueSet.compose.include.code with ValueSet.compose.include.concept ([3258](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3258)), added new rules about expansion content ([3138](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3138)) * [Media](file:///C:\temp\media.html): Rename element 'dateTime' to 'created' ([3174](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=3174)) and length to duration ([2866](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=2866)) * Remove DeviceObservationReport and Query * Collapse AdverseReaction into [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) * [Appointment](file:///C:\temp\appointment.html) changes - individual field renamed to actor, and added mappings to [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) and [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) * [FamilyMemberHistory](file:///C:\temp\familymemberhistory.html) combined with [List](file:///C:\temp\list.html) replaces FamilyHistory (with corresponding updates to related profiles) * [Flag](file:///C:\temp\flag.html) replaces Alert including improved clarification of how it is used and replacement of "note" with "code" * [CarePlan](file:///C:\temp\careplan.html) significantly refactored including splitting Goal out as a distinct resource, moving elements between activity and detail, introduction of several new elements and supported relationship types   New Resources:   * [Appointment](file:///C:\temp\appointment.html) * [AppointmentResponse](file:///C:\temp\appointmentresponse.html) * [Basic](file:///C:\temp\basic.html) * [ClaimResponse](file:///C:\temp\claimresponse.html) * [ClinicalImpression](file:///C:\temp\clinicalimpression.html) * [Communication](file:///C:\temp\communication.html) * [CommunicationRequest](file:///C:\temp\communicationrequest.html) * [Contract](file:///C:\temp\contract.html) * [Contraindication](file:///C:\temp\detectedissue.html) * [Coverage](file:///C:\temp\coverage.html) * [DataElement](file:///C:\temp\dataelement.html) * [DeviceComponent](file:///C:\temp\devicecomponent.html) * [DeviceMetric](file:///C:\temp\devicemetric.html) * [DeviceRequest](file:///C:\temp\devicerequest.html) * [DeviceUseStatement](file:///C:\temp\deviceusestatement.html) * [EligibilityRequest](file:///C:\temp\eligibilityrequest.html) * [EligibilityResponse](file:///C:\temp\eligibilityresponse.html) * [EnrollmentRequest](file:///C:\temp\enrollmentrequest.html) * [EnrollmentResponse](file:///C:\temp\enrollmentresponse.html) * [EpisodeOfCare](file:///C:\temp\episodeofcare.html) * [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html) * [StructureDefinition](file:///C:\temp\structuredefinition.html) * [Goal](file:///C:\temp\goal.html) * [HealthcareService](file:///C:\temp\healthcareservice.html) * ImagingObjectSelection * [InstitutionalClaim](file:///C:\temp\claim.html) * [NamingSystem](file:///C:\temp\namingsystem.html) * [NutritionOrder](file:///C:\temp\nutritionorder.html) * [OperationDefinition](file:///C:\temp\operationdefinition.html) * [OralHealthClaim](file:///C:\temp\claim.html) * [PaymentNotice](file:///C:\temp\paymentnotice.html) * [PaymentReconciliation](file:///C:\temp\paymentreconciliation.html) * [Person](file:///C:\temp\person.html) * [PharmacyClaim](file:///C:\temp\claim.html) * [ProcedureRequest](file:///C:\temp\procedurerequest.html) * [ProfessionalClaim](file:///C:\temp\claim.html) * [QuestionnaireResponse](file:///C:\temp\questionnaireresponse.html) * [ReferralRequest](file:///C:\temp\referralrequest.html) * [RiskAssessment](file:///C:\temp\riskassessment.html) * [SearchParameter](file:///C:\temp\searchparameter.html) * [Schedule](file:///C:\temp\schedule.html) * [Slot](file:///C:\temp\slot.html) * [Subscription](file:///C:\temp\subscription.html) * SupportingDocumentation * [VisionClaim](file:///C:\temp\claim.html) * [VisionPrescription](file:///C:\temp\visionprescription.html)   New Implementation Guides (see discussion of status)   * Argonaut Project * Data Access Framework * Structured Data Capture & Structured Data Capture - Data Element Exchange * US Laboratory Order, Report & Report to Public Health * [EHRS Functional model - Record Lifecycle Events](file:///C:\temp\ehrsrle\ehrsrle.html) |
| 0.3.0 | * Renamed Namespace to NamingSystem * Split [QuestionnaireResponse](file:///C:\temp\questionnaireresponse.html) from [Questionnaire](file:///C:\temp\questionnaire.html) and significantly revamped the [Questionnaire](file:///C:\temp\questionnaire.html) in response to feedback from the Connectathon. * Added [DataElement](file:///C:\temp\dataelement.html) resource (has been previously discussed as ObservationDefinition) * Defined [Subscription](file:///C:\temp\subscription.html) resource for evaluation * Add [time](file:///C:\temp\datatypes.html#time) data type * Define RPC-type [operations](file:///C:\temp\operations.html) on the RESTful interface, and add [Operation Definition](file:///C:\temp\operationdefinition.html) * Defined operations on several resources including [ConceptMap](file:///C:\temp\conceptmap.html), [Questionnaire](file:///C:\temp\questionnaire.html) and [ValueSet](file:///C:\temp\valueset.html) * [Conformance](file:///C:\temp\capabilitystatement.html) (now CapabilityStatement): Change Conformance.rest.operation to Conformance.rest.interaction and Conformance.rest.resource.operation to Conformance.rest.resource.interaction, and add Conformance.rest.operation to point to [Operation Definition](file:///C:\temp\operationdefinition.html) * [Profile](file:///C:\temp\profile.html): add Profile.url, Profile.structure.snapshot and Profile.structure.differential, and remove query definition (now in OperationDefinition) * Add pages for [LOINC](file:///C:\temp\loinc.html), [RxNorm](file:///C:\temp\rxnorm.html), and [SNOMED CT](file:///C:\temp\snomedct.html) * Significant rework for [ConceptMap](file:///C:\temp\conceptmap.html) so it can be used to map between structures as well as value sets * Add [Contraindication](file:///C:\temp\detectedissue.html) and [Risk Assessment](file:///C:\temp\riskassessment.html) and added examples to other resources in support of these * Add [Referral Request](file:///C:\temp\referralrequest.html) * Add supportingInformation to DiagnosticRequest * Add fulfills to [Encounter](file:///C:\temp\encounter.html) * Add date and age[x] to [FamilyHistory](file:///C:\temp\familymemberhistory.html) * Change cardinality of [Location](file:///C:\temp\location.html).identifier to 0..\* * Change cardinality of [Practitioner](file:///C:\temp\practitioner.html).address to 0..\* * Add [Observation](file:///C:\temp\observation.html).encounter to 0..\* * Temporarily added content for the [Structured Data Capture implementation guide](http://hl7.org/fhir/us/sdc/index.html), including profiles on several resources, conformance instances, etc. This content will be moved to an external specification once the necessary tooling is in place to support that form of publication. |
| 0.2.1 | * Minor new optional elements on value set for metadata, new extensions for all the rest of the VSD project metadata, formal profile to express basic minimum metadata for value set |
| **0.2.0** | * Namespace: adjustments based on Grahame's feedback |
| 0.1.0 | * Add [Appointment](file:///C:\temp\appointment.html), [Appointment Response](file:///C:\temp\appointmentresponse.html), [Schedule](file:///C:\temp\schedule.html) and [Slot](file:///C:\temp\slot.html) * Add *Namespace* |

Note: a useful tool for displaying the differences between pages is the [W3C HTML Diff engine](http://services.w3.org/htmldiff).

**http.html**

**RESTful API**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Each "resource type" has the same set of interactions defined that can be used to manage the resources in a highly granular fashion. Applications claiming conformance to this framework claim to be conformant to "RESTful FHIR" (see [Conformance](file:///C:\temp\conformance-rules.html)).

Note that in this RESTful framework, transactions are performed directly on the server resource using an HTTP request/response. The API does not directly address authentication, authorization, and audit collection - for further information, see the [Security Page](file:///C:\temp\security.html).

The API describes the FHIR resources as a set of operations (known as "interactions") on resources where individual resource instances are managed in collections by their type. Servers can choose which of these interactions are made available and which resource types they support. Servers SHALL provide a [Capability Statement](file:///C:\temp\capabilitystatement.html) that specifies which interactions and resources are supported.

The following logical interactions are defined:

|  |  |
| --- | --- |
| **Instance Level Interactions** |  |
| [read](file:///C:\temp\main-pages.html#read) | Read the current state of the resource |
| [vread](file:///C:\temp\main-pages.html#vread) | Read the state of a specific version of the resource |
| [update](file:///C:\temp\main-pages.html#update) | Update an existing resource by its id (or create it if it is new) |
| [patch](file:///C:\temp\main-pages.html#patch) | Update an existing resource by posting a set of changes to it |
| [delete](file:///C:\temp\main-pages.html#delete) | Delete a resource |
| [history](file:///C:\temp\main-pages.html#history) | Retrieve the change history for a particular resource |
| **Type Level Interactions** | |
| [create](file:///C:\temp\main-pages.html#create) | Create a new resource with a server assigned id |
| [search](file:///C:\temp\main-pages.html#search) | Search the resource type based on some filter criteria |
| [history](file:///C:\temp\main-pages.html#history) | Retrieve the change history for a particular resource type |
| **Whole System Interactions** | |
| [capabilities](file:///C:\temp\main-pages.html#capabilities) | Get a capability statement for the system |
| [batch/transaction](file:///C:\temp\main-pages.html#transaction) | Update, create or delete a set of resources in a single interaction |
| [history](file:///C:\temp\main-pages.html#history) | Retrieve the change history for all resources |
| [search](file:///C:\temp\main-pages.html#search) | Search across all resource types based on some filter criteria |

In addition to these interactions, there is an [operations framework](file:///C:\temp\operations.html), which includes endpoints for [validation](file:///C:\temp\operation-resource-validate.html), [messaging](file:///C:\temp\messaging.html#mailbox) and [Documents](file:///C:\temp\documents.html#bundle).

**Style Guide**

The interactions on this page are defined like this:

VERB [base]/[type]/[id] {?\_format=[mime-type]}

* The first word is the HTTP verb used for the interaction
* Content surrounded by [] is mandatory, and will be replaced by the string literal identified. Possible insertion values:
  + base: The [Service Base URL](file:///C:\temp\main-pages.html#root)
  + mime-type: The [Mime Type](file:///C:\temp\main-pages.html#mime-type)
  + type: The name of a resource type (e.g. "Patient")
  + id: The [Logical Id](file:///C:\temp\resource.html#id) of a resource
  + vid: The [Version Id](file:///C:\temp\resource.html#metadata) of a resource
  + compartment: The name of a [compartment](file:///C:\temp\compartmentdefinition.html)
  + parameters: URL parameters as defined for the particular interaction
* Content surrounded by {} is optional

Implementations constructing URLs using these patterns SHOULD conform to [RFC 3986 Section 6 Appendix A](https://tools.ietf.org/html/rfc3986#appendix-A) which requires percent-encoding for a number of characters that occasionally appear in the URLs (mainly in search parameters).

This specification uses the underscore as a prefix to disambiguate reserved names from other names in 3 cases:

* To differentiate system wide history and search interactions from interactions on Resource Types
* To differentiate search, history and similar interactions from instances of a resource type
* To differentiate search parameters defined for all resources from those defined for specific resource types

In addition, the character $ is used as a prefix to operation names that are RPC-like additions to the base API defined, either by this specification or by implementers.

**Service Base URL**

The Service Base URL is the address where all of the resources defined by this interface are found. The Service Base URL takes the form of

http(s)://server{/path}

The path portion is optional, and does not include a trailing slash. Each resource type defined in this specification has a manager (or "entity set") that lives at the address /[type] where the [type] is the name of the resource type. For instance, the resource manager for the type Patient will live at:

https://server/path/Patient

All the logical interactions are defined relative to the service root URL. This means that if the address of any one FHIR resource on a system is known, the address of other resources may be determined.

Note: All URLs (and ids that form part of the URL) defined by this specification are case sensitive. Clients SHOULD encode URLs using UTF-8, and servers SHOULD decode them assuming they are UTF-8 (for background, [see here](http://stackoverflow.com/questions/912811/what-is-the-proper-way-to-url-encode-unicode-characters)).

Note that a server may use a path of the form http://server/...[xx]... where the [xx] is some variable portion that identifies a particular instantiation of the FHIR API. Typically, the variable id identifies a patient or a user, and the underlying information is completely compartmented by the logical identity associated with [xx]. In this case, the FHIR API presents a patient or user centric view of a record, where authentication/authorization is explicitly granted to the URL, on the grounds that some identifiable user is associated with the logical identity. It is not necessary to explicitly embed the patient id in the URL - implementations can associate an FHIR end-point with a particular patient or provider by using an OAuth login. See [Compartments](file:///C:\temp\compartmentdefinition.html) for the logical underpinning.

**Identity**

Systems often need to compare two URLs to determine whether they refer to the same underlying object or not. For the purposes of this specification, the following rules apply:

* The query part of the URL (anything after ?) is ignored
* The comparison of the document portion of the URL (i.e. not the server/port) is case sensitive
* The protocols http: and https: SHALL NOT be used to refer to different underlying objects
* If a port is specified, then the ports must be identical or the objects are different (due to the prevalence of port mapping and/or interface engines running on different ports). Ports should only be explicit when they have explicit meaning to the server

For example: http://myserver.com/Patient/1 and https://myserver.com/Patient/1 refer to the same underlying object, while http://myserver.com:81/Patient/1 is a distinct entity from either of the above. This does not mean that the two addresses need to be treated the same, or that a server must serve both addresses, or that the content from the two addresses must be identical, but just that if these two addresses have the same identity, and if both are served, they must both represent the same underlying object. Systems are not required to check that this is true. Note: the identity comparison for protocols other than http:/https: is undefined.

**Resource Metadata and Versioning**

Each resource has an associated set of [resource metadata elements](file:///C:\temp\resource.html#metadata). These map to the HTTP request and response using the following fields:

|  |  |
| --- | --- |
| **Metadata Item** | **Where found in HTTP** |
| [Logical Id (.id)](file:///C:\temp\resource.html#id) | The Id is represented explicitly in the URL |
| [Version Id (.meta.versionId)](file:///C:\temp\resource.html#metadata) | The Version Id is represented in the ETag header |
| [Last modified (.meta.lastUpdated)](file:///C:\temp\resource.html#metadata) | HTTP Last-Modified header |

Note that the Version Id is considered a "weak" ETag and ETag headers should be prefixed with W/ and enclosed in quotes, for example:

ETag: W/"3141"

**Security**

Using HTTPS is optional, but all production exchange of healthcare data SHOULD use SSL and additional security as appropriate. See [HTTP Security](file:///C:\temp\security.html#http) for further information. Most operations will require user authentication, and all operations that do are subject to [RBAC](https://en.wikipedia.org/wiki/Role-based_access_control) and/or [ABAC](https://en.wikipedia.org/wiki/Attribute-based_access_control), and some operations may depend on appropriate consent being granted.

See the [HTTP Security](file:///C:\temp\security.html#AccessDenied) guidance on how to handle access denial response.

Note: to support browser-based client applications, recommend that servers SHOULD implement [cross-origin resource sharing](http://enable-cors.org/) for the interactions documented here. Experience shows that this is an area where ongoing issues may be expected as security holes are found and closed in an ongoing basis.

**HTTP Status Codes**

This specification makes rules about the use of specific HTTP status codes in particular circumstances where the status codes SHALL map to particular states correctly, and only where the correct status code is not obvious. Other HTTP status codes may be used for other states as appropriate, and this particularly includes various authentication related status codes and redirects. Authentication redirects should not be interpreted to change the location of the resource itself (a common web programming error).

FHIR defines an [OperationOutcome resource](file:///C:\temp\operationoutcome.html) that can be used to convey specific detailed processable error information. For a few combinations of interactions and specific return codes, an OperationOutcome is required to be returned as the content of the response. The OperationOutcome may be returned with any HTTP 4xx or 5xx response, but is not required - many of these errors may be generated by generic server frameworks underlying a FHIR server.

**Managing Return Content**

In the interests of managing band-width, this specification allows clients to specify what kind of content to return.

**conditional read**

Clients may use the If-Modified-Since, or If-None-Match HTTP header on a read request. If so, they MUST accept either a 304 Not Modified as a valid status code on the response (which means that the content is unchanged since that date) or full content (either the content has changed, or the server does not support conditional request).

Servers can return 304 Not Modified where content is unchanged since the If-Modified-Since date-time or the If-None-Match ETag specified or they can return the full content as normal. This optimisation is relevant in reducing bandwidth for caching purposes and servers are encouraged but not required to support this. If servers don't support conditional read, they just return the full content.

**create/update/patch/transaction**

These interactions are performed using POST, PUT or PATCH, and it may be appropriate for a server to return either only a status code, or also return the entire resource that is the outcome of the create or update (which may be different to that provided by the client). In the case of transactions this means returning a Bundle with just the Bundle.entry.response populated for each entry, and not the Bundle.entry.resource values.

The client can indicate whether the entire resource is returned using the [HTTP return preference](https://tools.ietf.org/html/rfc7240#section-4.2):

Prefer: return=minimal

Prefer: return=representation

Prefer: return=OperationOutcome

The first of these two asks to return no body. The second asks to return the full resource. The third asks the server to return an [OperationOutcome](file:///C:\temp\operationoutcome.html) resource containing hints and warnings about the operation rather than the full resource. Servers SHOULD honor this header. In the absence of the header, servers may choose whether to return the full resource or not (but not the OperationOutcome; that should only be returned if explicitly requested). Note that this setting only applies to successful interactions. In case of failure, servers SHOULD always return a body that contains an [OperationOutcome](file:///C:\temp\operationoutcome.html) resource.

**Content Types and encodings**

The formal MIME-type for FHIR resources is application/fhir+xml or application/fhir+json. The correct mime type SHALL be used by clients and servers:

* XML: application/fhir+xml
* JSON: application/fhir+json
* RDF: text/turtle (only the Turtle format is supported)

Servers SHALL support server-driven content negotiation as described in [section 12](http://www.w3.org/Protocols/rfc2616/rfc2616-sec12.html#sec12) of the HTTP specification.

Note: between FHIR DSTU2 and STU3, the correct mime type was changed from application/xml+fhir and application/json+fhir to application/fhir+xml and application/fhir+json. Servers MAY also support the older mime types, and are encouraged to do so to smooth the transtion process.

In order to support various implementation limitations, servers SHOULD support the optional \_format parameter to specify alternative response formats by their MIME-types. This parameter allows a client to override the accept header value when it is unable to set it correctly due to internal limitations (e.g. XSLT usage). For the \_format parameter, the values xml, text/xml, application/xml, and application/fhir+xml SHALL be interpreted to mean the [XML format](file:///C:\temp\xml.html), the codes json, application/json and application/fhir+json SHALL be interpreted to mean the [JSON format](file:///C:\temp\json.html), and the codes ttl and text/turtle SHALL be interpreted to mean the [Turtle RDF format](file:///C:\temp\rdf.html). In addition, the values html and text/html are allowed.

FHIR uses UTF-8 for all request and response bodies. Since the HTTP specification (section 3.7.1) defines a default character encoding of ISO-8859-1, requests and responses SHALL explicitly set the character encoding to UTF-8 using the charset parameter of the MIME-type in the Content-Type header. Requests MAY also specify this charset parameter in the Accept header and/or use the Accept-Charset header.

Note: the \_format parameter does not override the content-type header.

The content types application/x-www-form-urlencoded is also accepted for posting search requests.

If neither the accept header nor the \_format parameter are specified, the MIME-type of the content returned by the server is undefined and may vary.

Implementation Note: If a client provides a generic mime type in the Accept header (application/xml, text/json, or application/json), the server SHOULD respond with the requested mime type, using the XML or JSON formats described in this specification as the best representation for the named mime type (though see the [note on the Binary resource](file:///C:\temp\binary.html#rest)).

**Support for Versions**

Servers that support this API SHOULD provide full version support - that is, populate and track versionId correctly, support vread, and implement [version aware updates](file:///C:\temp\main-pages.html#versionaware). Supporting versions like this allows for related systems to track the correct version of information, and to keep integrity in clinical records. However, many current operational systems do not do this, and cannot easily be re-engineered to do so.

For this reason, servers are allowed to not provide versioning support and this API does not enforce that versioning is supported. Clients may elect to only interact with servers that do provide full versioning support. Systems declare their support for versioning in their [Capability Statements](file:///C:\temp\capabilitystatement-definitions.html#CapabilityStatement.rest.resource.versioning). Where they can indicate one of three levels for versioning support:

* **no-version**: Versioning and meta.version is not supported (server) or used (client)
* **versioned**: Versioning and meta.version is supported (server) or used (client)
* **versioned-update**: Versioning and meta.version is supported, and version aware updates are used - Version ID must be correct for updates (server) or will be specified (If-match header) for updates (client)

Server should always return the default timezone for [date searches](file:///C:\temp\search.html#date) in the HTTP Response headers using the Date header. Note: Servers are not required to have a default timezone.

**read**

The read interaction accesses the current contents of a resource. The interaction is performed by an HTTP GET command as shown:

GET [base]/[type]/[id] {?\_format=[mime-type]}

This returns a single instance with the content specified for the resource type. This url may be accessed by a browser. The possible values for the [Logical Id](file:///C:\temp\resource.html#id) ("id") itself are described in the [id type](file:///C:\temp\datatypes.html#id). The returned resource SHALL have an id element with a value that is the [id]. Servers SHOULD return an ETag header with the versionId of the resource (if versioning is supported) and a Last-Modified header.

Note: Unknown resources and deleted resources are treated differently on a read: A GET for a deleted resource returns a 410 status code, whereas a GET for an unknown resource returns 404. Systems that do not track deleted records will treat deleted records as an unknown resource. Since deleted resources may be brought back to life, servers MAY include an ETag on the error response when reading a deleted record to allow version contention management when a resource is brought back to life.

In addition, the search parameter \_summary can be used when reading a resource:

GET [base]/[type]/[id] {?\_summary=text}

This requests that only a subset of the resource content be returned, as specified in the \_summary parameter, which can have the values [true, false, text & data](file:///C:\temp\search.html#summary). Note that a resource that only contains a subset of the data is not suitable for use as a base to update the resource, and may not be suitable for other uses. The same applies to the [\_elements](file:///C:\temp\search.html#elements) parameter - both that it should be supported, and the subset implications. Servers SHOULD define a Resource.meta.tag with the [SUBSETTED](file:///C:\temp\v3\ObservationValue\cs.html#v3-ObservationValue-SUBSETTED) as a [Simple Tag](file:///C:\temp\resource.html#simple-tags) to explicitly mark such resources.

**vread**

The vread interaction preforms a version specific read of the resource. The interaction is performed by an HTTP GET command as shown:

GET [base]/[type]/[id]/\_history/[vid] {?\_format=[mime-type]}

This returns a single instance with the content specified for the resource type for that version of the resource. The returned resource SHALL have an id element with a value that is the [id], and a meta.versionId element with a value of [vid]. Servers SHOULD return an ETag header with the versionId (if versioning is supported) and a Last-Modified header.

The [Version Id](file:///C:\temp\resource.html#metadata) ("vid") is an opaque identifier that conforms to the same [format requirements](file:///C:\temp\datatypes.html#id) as a [Logical Id](file:///C:\temp\resource.html#id). The id may have been found by performing a history interaction (see below), by recording the version id from a content location returned from a read or from a version specific reference in a content model. If the version referred to is actually one where the resource was deleted, the server should return a 410 status code.

Servers are encouraged to support a version specific retrieval of the current version of the resource even if they do not provide access to previous versions. If a request is made for a previous version of a resource, and the server does not support accessing previous versions (either generally, or for this particular resource), it should return a 404 Not Found error, with an operation outcome explaining that history is not supported for the underlying resource type or instance.

**update**

The update interaction creates a new current version for an existing resource or creates an initial version if no resource already exists for the given id. The update interaction is performed by an HTTP PUT command as shown:

PUT [base]/[type]/[id] {?\_format=[mime-type]}

The request body SHALL be a [Resource](file:///C:\temp\resource.html) with an id element that has an identical value to the [id] in the URL. If no id element is provided, or the value is wrong, the server SHALL respond with a HTTP 400 error code, and SHOULD provide an operation outcome identifying the issue. If the request body includes a [meta](file:///C:\temp\resource.html#meta), the server SHALL ignore the provided versionId and lastUpdated values. If the server supports versions, it SHALL populate the meta.versionId and meta.lastUpdated with the new correct values. Servers are allowed to review and alter the other metadata values, but SHOULD refrain from doing so (see [metadata description](file:///C:\temp\resource.html#meta) for further information). Note that there is no support for updating past versions - see notes on the [history](file:///C:\temp\main-pages.html#history) interaction.

A server SHOULD accept the resource as submitted when it accepts the update, and return the same content when it is subsequently read. However systems may not be able to do this; see the note on [transactional integrity](file:///C:\temp\main-pages.html#transactional-integrity) for discussion. Also, see [Variations between Submitted data and Retrieved data](file:///C:\temp\updates.html) for additional discussion around update behavior. Note that update genreally updates the whole content of the resource. For partial updates, see [patch](file:///C:\temp\main-pages.html#patch) below.

If the interaction is successful, the server SHALL return either a 200 OK HTTP status code if the resource was updated, or a 201 Created status code if the resource was created, with a Last-Modified header, and an ETag header which contains the new versionId of the resource. If the resource was created (i.e. the interaction resulted in a 201 Created), the server SHOULD return a Location header (this is for HTTP conformance; it's not otherwise needed).

Note: Servers MAY choose to preserve XML comments, instructions, and formatting or JSON whitespace when accepting updates, but are not required to do so. The impact of this on digital signatures may need to be considered.

Note that servers MAY choose to allow clients to PUT a resource to a location that does not yet exist on the server - effectively, allowing the client to define the id of the resource. Whether a server allows this is a deployment choice based on the nature of its relationships with the clients. While many servers will not allow clients to define their ids, there are several reasons why it may be necessary in some configurations:

* client is reproducing an existing data model on the server, and needs to keep original ids in order to retain ongoing integrity
* client is a server doing push based pub/sub (this is a special case of the first reason)
* multiple clients doing push in the context of agreed data model shared across multiple servers where ids are shared across servers

Alternatively, clients may be sharing an agreed identification model (e.g. key server, scoped identifiers, or UUIDs) where clashes do not arise.

Servers can choose whether or not to support client defined ids, and indicate such to the clients using [CapabilityStatement.rest.resource.updateCreate](file:///C:\temp\capabilitystatement-definitions.html#CapabilityStatement.rest.resource.updateCreate).

**Rejecting Updates**

Servers are permitted to reject update interactions because of integrity concerns or other business rules, and return HTTP status codes accordingly (usually a 422).

Common HTTP Status codes returned on FHIR-related errors (in addition to normal HTTP errors related to security, header and content type negotiation issues):

* **400 Bad Request** - resource could not be parsed or failed basic FHIR validation rules (or multiple matches were found for conditional criteria)
* **401 Not Authorized** - authorization is required for the interaction that was attempted
* **404 Not Found** - resource type not supported, or not a FHIR end-point
* **405 Method Not allowed** - the resource did not exist prior to the update, and the server does not allow client defined ids
* **409/412** - version conflict management - see [below](file:///C:\temp\main-pages.html#concurrency)
* **422 Unprocessable Entity** - the proposed resource violated applicable FHIR profiles or server business rules

Any of these errors SHOULD be accompanied by an [OperationOutcome](file:///C:\temp\operationoutcome.html) resource providing additional detail concerning the issue.

For additional information on how systems may behave when processing updates, refer to the [Variations between Submitted data and Retrieved data](file:///C:\temp\updates.html) page.

**Conditional update**

The conditional update interaction allows a client to update an existing resource based on some identification criteria, rather than by [logical id](file:///C:\temp\resource.html#meta). To accomplish this, the client issues a PUT as shown:

PUT [base]/[type]?[search parameters]

When the server processes this update, it performs a search using its standard [search facilities](file:///C:\temp\search.html) for the resource type, with the goal of resolving a single logical id for this request. The action it takes depends on how many matches are found:

* **No matches**: The server performs a [create](file:///C:\temp\main-pages.html#create) interaction
* **One Match**: The server performs the update against the matching resource
* **Multiple matches**: The server returns a 412 Precondition Failed error indicating the client's criteria were not selective enough

This variant can be used to allow a stateless client (such as an interface engine) to submit updated results to a server, without having to remember the logical ids that the server has assigned. For example, a client updating the status of a lab result from "preliminary" to "final" might submit the finalized result using PUT path/Observation?identifier=http://my-lab-system|123

Note that transactions and conditional create/update/delete are complex interactions and it is not expected that every server will implement them. Servers that don't support the conditional update should return an HTTP 400 error and an operation outcome.

**Managing Resource Contention**

[Lost Updates](http://www.w3.org/1999/04/Editing/), where two clients update the same resource, and the second overwrites the updates of the first, can be prevented using a combination of the [ETag](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.19) and [If-Match](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.24) header. This is also known as 'Optimistic Locking'.

To support this usage, servers SHOULD always return an ETag header with each resource:

HTTP 200 OK

Date: Sat, 09 Feb 2013 16:09:50 GMT

Last-Modified: Sat, 02 Feb 2013 12:02:47 GMT

ETag: W/"23"

Content-Type: application/fhir+json

If provided, the value of the ETag SHALL match the value of the version id for the resource. Servers are allowed to generate the version id in whatever fashion that they wish, so long as they are valid according to the [id](file:///C:\temp\datatypes.html#id) data type, and are unique within the address space of all versions of the same resource. When resources are returned as part of a bundle, there is no ETag, and the versionId of the resource is used directly.

If the client wishes to request a version aware update, it submits the request with an If-Match header that quotes the ETag from the server:

PUT /Patient/347 HTTP/1.1

If-Match: W/"23"

If the version id given in the If-Match header does not match, the server returns a 409 Conflict status code instead of updating the resource.

Servers can require that clients provide an If-Match header by returning 412 Pre-condition failed status codes when no If-Match header is found.

**patch**

As an alternative to updating an entire resource, clients can perform a patch operation. This can be useful when a client is seeking to minimise it's bandwidth utilization, or in scenarios where a client has only partial access or support for a resource. The patch interaction is performed by an HTTP PATCH command as shown:

PATCH [base]/[type]/[id] {?\_format=[mime-type]}

The body of a PATCH operation SHALL be either:

* a [JSON Patch](http://jsonpatch.com/) document with a content type of application/json-patch+json
* an [XML Patch](http://tools.ietf.org/html/rfc5261) document with a content type of application/xml-patch+xml
* a [FHIRPath Patch](file:///C:\temp\fhirpatch.html) parameters resource with [FHIR Content Type](file:///C:\temp\main-pages.html#mime-type)

In either case, the server SHALL process it's own copy of the resource in the format indicated, applying the operations specified in the document, following the relevant PATCH specification. When the operations have all been processed, the server processes the resulting document as an [Update](file:///C:\temp\main-pages.html#update)operation; all the version and error handling etc applies as specified, as does the [Prefer Header](file:///C:\temp\main-pages.html#prefer).

Processing PATCH operations may be very version sensitive. For this reason, servers SHALL support conditional PATCH, which works exactly the same as specified for update in [Concurrency Management](file:///C:\temp\main-pages.html#versionaware). Clients SHOULD always consider using version specific PATCH operations so that inappropriate actions are not executed. In addition, servers SHALL support Conditional PATCH, which works exactly as described for [Conditional Update](file:///C:\temp\main-pages.html#cond-update).

The server SHALL ensure that the narrative in a resource is not clinically unsafe after the PATCH operation is performed. Exactly how this is defined and can be achieved depends on the context, and how narrative is being maintained, but servers may wish to consider:

* If the existing narrative has a status != generated, the server could reject the PATCH operation
* The server could regenerate the narrative once the operation has been applied to the data
* In some limited circumstances, an XML PATCH operation could update the narrative
* The server could delete the narrative, in the basis that some later process will be able to populate it correctly

Processing XML Patch documents is tricky because of namespace handling. Servers SHALL handle namespaces correctly, but note that FHIR resources only contain two XML namespaces, for FHIR (http://hl7.org/fhir) and XHTML (http://www.w3.org/1999/xhtml).

For PATCH Examples, see (link todo).

Patch operations may be performed as part of Batch or Transaction Operations using the FHIRPath Patch format.

**delete**

The delete interaction removes an existing resource. The interaction is performed by an HTTP DELETE command as shown:

DELETE [base]/[type]/[id]

A delete interaction means that subsequent [non-version specific reads](file:///C:\temp\main-pages.html#read) of a resource return a 410 HTTP status code and that the resource is no longer found through [search](file:///C:\temp\main-pages.html#search) interactions. Upon successful deletion, or if the resource does not exist at all, the server should return either a 200 OK if the response contains a payload, or a 204 No Content with no response payload.

Whether to support delete at all, or for a particular resource type or a particular instance is at the discretion of the server based on the business rules that apply in its context. If the server refuses to delete resources of that type as a blanket policy, then it should return the 405 Method not allowed status code. If the server refuses to delete a resource because of reasons specific to that resource, such as referential integrity, it should return the 409 Conflict status code. Performing this interaction on a resource that is already deleted has no effect, and the server should return either a 200 OK if the response contains a payload, or a 204 No Content with no response payload. Resources that have been deleted may be "brought back to life" by a subsequent [update](file:///C:\temp\main-pages.html#href) interaction using an HTTP PUT.

Many resources have a status element that overlaps with the idea of deletion. Each resource type defines what the semantics of the deletion interactions are. If no documentation is provided, the deletion interaction should be understood as deleting the record of the resource, with nothing about the state of the real-world corresponding resource implied.

For servers that maintain a version history, the delete interaction does not remove a resource's version history. From a version history respect, deleting a resource is the equivalent of creating a special kind of history entry that has no content and is marked as deleted. Note that there is no support for deleting past versions - see notes on the [history](file:///C:\temp\main-pages.html#history) interaction.

Since deleted resources may be brought back to life, servers MAY include an ETag on the delete response to allow version contention management when a resource is brought back to life.

**Conditional delete**

The conditional delete interaction allows a client to delete an existing resource based on some selection criteria, rather than by a specific [logical id](file:///C:\temp\resource.html#meta). To accomplish this, the client issues an HTTP DELETE as shown:

DELETE [base]/[type]/?[search parameters]

When the server processes this delete, it performs a search as specified using the standard [search facilities](file:///C:\temp\search.html) for the resource type. The action it takes depends on how many matches are found:

* **No matches** or **One Match**: The server performs an ordinary delete on the matching resource
* **Multiple matches**: Servers may choose to delete all the matching resources, or it may choose to return a 412 Precondition Failed error indicating the client's criteria were not selective enough. A server indicates whether it can delete multiple resources in its [Capability Statement (.rest.resource.conditionalDelete)](file:///C:\temp\capabilitystatement-definitions.html#CapabilityStatement.rest.resource.conditionalDelete). if there are multiple matches, either all must deleted, or the server SHALL return an error

This variant can be used to allow a stateless client (such as an interface engine) to delete a resource on a server, without having to remember the logical ids that the server has assigned. For example, a client deleting a lab atomic result might delete the resource using DELETE /Observation?identifier=http://my-lab-system|123.

Note that transactions and conditional create/update/delete are complex interactions and it is not expected that every server will implement them. Servers that don't support the conditional delete should return an HTTP 400 error and an operation outcome.

**create**

The create interaction creates a new resource in a server-assigned location. If the client wishes to have control over the id of a newly submitted resource, it should use the [update](file:///C:\temp\main-pages.html#update) interaction instead. The create interaction is performed by an HTTP POST command as shown:

POST [base]/[type] {?\_format=[mime-type]}

The request body SHALL be a FHIR Resource. The resource does not need to have an id element (this is the one of the few case where a resource exists without an id element). If an id is provided, the server SHALL ignore it. If the request body includes a [meta](file:///C:\temp\resource.html#meta), the server SHALL ignore the existing versionId and lastUpdated values. The server SHALL populate the id, meta.versionId and meta.lastUpdated with the new correct values. Servers are allowed to review and alter the other metadata values, but SHOULD refrain from doing so (see [metadata description](file:///C:\temp\resource.html#meta) for further information).

A server SHOULD otherwise accept the resource as submitted when it accepts the create, and return the same content when it is subsequently read. However some systems may not be able to do this; see the note on [transactional integrity](file:///C:\temp\main-pages.html#transactional-integrity) for discussion.

The server returns a 201 Created HTTP status code, and SHALL also return a Location header which contains the new [Logical Id](file:///C:\temp\resource.html#metadata) and [Version Id](file:///C:\temp\resource.html#metadata) of the created resource version:

Location: [base]/[type]/[id]/\_history/[vid]

where [id] and [vid] are the newly created id and version id for the resource version. Servers SHOULD return an ETag header with the versionId (if versioning is supported) and a Last-Modified header.

When the resource syntax or data is incorrect or invalid, and cannot be used to create a new resource, the server returns a 400 Bad Request HTTP status code. When the server rejects the content of the resource because of business rules, the server returns a 422 Unprocessible Entity error HTTP status code. In either case, the server SHOULD include a response body containing an [OperationOutcome](file:///C:\temp\operationoutcome.html) with detailed error messages describing the reason for the error.

Common HTTP Status codes returned on FHIR-related errors (in addition to normal HTTP errors related to security, header and content type negotiation issues):

* **400 Bad Request** - resource could not be parsed or failed basic FHIR validation rules
* **404 Not Found** - resource type not supported, or not a FHIR end-point
* **422 Unprocessable Entity** - the proposed resource violated applicable FHIR profiles or server business rules. This should be accompanied by an [OperationOutcome](file:///C:\temp\operationoutcome.html) resource providing additional detail

Note: Servers MAY choose to preserve XML comments, instructions, and formatting or JSON whitespace when accepting creates, but are not required to do so. The impact of this on digital signatures may need to be considered.

For additional information on how systems may behave when processing updates, refer to the [Variations between Submitted data and Retrieved data](file:///C:\temp\updates.html) page.

**Conditional create**

The conditional create interaction allows a client to create a new resource only if some equivalent resource does not already exist on the server. The client defines what equivalence means in this case by supplying a FHIR search query using an HL7 defined extension header "If-None-Exist" as shown:

If-None-Exist: [search parameters]

The parameter just contains the search parameters (what would be in the URL following the "?").

When the server processes this create, it performs a search as specified using its standard [search facilities](file:///C:\temp\search.html) for the resource type. The action it takes depends on how many matches are found:

* **No matches**: The server processes the create as above
* **One Match**: The server ignore the post and returns 200 OK
* **Multiple matches**: The server returns a 412 Precondition Failed error indicating the client's criteria were not selective enough

This variant can be used to avoid the risk of two clients creating duplicate resources for the same record. For example, a client posting a new lab result might specify If-None-Exist: identifier=http://my-lab-system|123 to ensure it is does not create a duplicate record.

Note that transactions and conditional create/update/delete are complex interactions and it is not expected that every server will implement them. Servers that don't support the conditional create should return an HTTP 412 error and an operation outcome.

**search**

This interaction searches a set of resources based on some filter criteria. The interaction can be performed by several different HTTP commands.

GET [base]/[type]{?[parameters]{&\_format=[mime-type]}}

This searches all resources of a particular type using the criteria represented in the parameters.

Because of the way that some user agents and proxies treat GET and POST requests, in addition to the get based search method above, servers that support *search* SHALL also support a POST based search:

POST [base]/[type]/\_search{?[parameters]{&\_format=[mime-type]}}

This has exactly the same semantics as the equivalent GET command. All these search interactions take a series of parameters that are a series of name=value pairs encoded in the URL (or as an application/x-www-form-urlencoded submission for a POST). (See [W3C HTML forms](http://www.w3.org/TR/REC-html40/interact/forms.html#form-content-type)).

Note: application/x-www-form-urlencoded is supported for POST so that invoking a search by GET or POST can be done from HTML forms in a browser (though considerable active content might be required in the browser), although this is not the main usage.

Searches are processed as specified for the [Search handling mechanism](file:///C:\temp\search.html).

If the search succeeds, the server SHALL return a 200 OK HTTP status code and the return content SHALL be a [Bundle](file:///C:\temp\bundle.html) with [type](file:///C:\temp\bundle-definitions.html#Bundle.type) = searchset containing the results of the search as a collection of zero or more resources in a defined order. The result collection can be long, so servers may use paging. If they do, they SHALL use the method [described below](file:///C:\temp\main-pages.html#paging) (adapted from [RFC 5005 (Feed Paging and Archiving](https://tools.ietf.org/html/rfc5005)) for breaking the collection into pages if appropriate. The server MAY also return an [OperationOutcome](file:///C:\temp\operationoutcome.html) resource within the searchset Bundle entries that contains additional information about the search; if one is sent it SHALL NOT include any issues with a fatal or error [severity](file:///C:\temp\valueset-issue-severity.html), and it SHALL be marked with a [Bundle.entry.search.mode](file:///C:\temp\valueset-search-entry-mode.html) of outcome.

If the search fails (cannot be executed, not that there is no matches), the return value is a status code 4xx or 5xx with an [OperationOutcome](file:///C:\temp\operationoutcome.html).

Common HTTP Status codes returned on FHIR-related errors (in addition to normal HTTP errors related to security, header and content type negotiation issues):

* **400 Bad Request** - search could not be processed or failed basic FHIR validation rules
* **401 Not Authorized** - authorization is required for the interaction that was attempted
* **404 Not Found** - resource type not supported, or not a FHIR end-point

**Variant Searches**

To search a [compartment](file:///C:\temp\compartmentdefinition.html), for either all possible resources or for a particular resource type, respectively:

GET [base]/[Compartment]/[id]/{\*?[parameters]{&\_format=[mime-type]}}

GET [base]/[Compartment]/[id]/[type]{?[parameters]{&\_format=[mime-type]}}

For example, to retrieve all the observation resources for a particular LOINC code associated with a particular encounter:

GET [base]/Encounter/23423445/Observation?code=2951-2 {&\_format=[mime-type]}

Note that there are specific operations defined to support fetching [an entire patient record](file:///C:\temp\patient-operations.html#everything) or [all record for an encounter](file:///C:\temp\encounter-operations.html#everything).

It's also possible to search across multiple resource types:

GET [base]?\_type=Condition,Observation&[parameters]{&\_format=[mime-type]}

This is a request to search on both Condition and Observation. In this case, the only parameters that can be used are those defined for both Condition and Observation (using [SearchParameter.base](file:///C:\temp\searchparameter-definitions.html#SearchParameter.base) - see [Cross-resource Search Parameters](file:///C:\temp\searchparameter-registry.html#common)), or the [parameters defined for all resources](file:///C:\temp\resource.html#search). If a search lists types not listed SearchParameter.base for any of the parameters, this is an error, and a server SHOULD return a 400 status. It's also possible to search on all types at once:

GET [base]?[parameters]{&\_format=[mime-type]}

When searching all resources at once, only the [parameters defined for all resources](file:///C:\temp\resource.html#search) can be used.

**capabilities**

The capabilities interaction retrieves the server's Capability Statement that defines how it supports resources. The interaction is performed by an HTTP GET command as shown:

GET [base]/metadata {?\_format=[mime-type]}

In addition, there is another method to get the capability statement, using the HTTP OPTIONS command:

OPTIONS [base] {?\_format=[mime-type]}

However as of STU3, this method is deprecated, and will be removed in a future version. Using OPTIONS like this is not conformant with HTTP, and creates challenges for [cross-origin resource sharing](http://enable-cors.org/) support.

Applications SHALL return a [Capability Statement](file:///C:\temp\capabilitystatement.html) that specifies which resource types and interactions are supported for the GET command. If a 404 Unknown is returned from the GET, FHIR is not supported on the nominated service url. An ETag header SHOULD be returned with the CapabilityStatement. The value of the header SHALL change if the CapabilityStatement itself changes. Additional parameters that are required to be returned with the OPTIONS command are defined in the [OMG hData RESTful Transport](file:///C:\temp\main-pages.html#hdata) specification.

The Capability statement returned typically has an arbitrary id, and no meta element, though it is not prohibited. Capability statements can become quite large; Servers are encouraged to support the [\_summary](file:///C:\temp\search.html#summary) and [\_elements](file:///C:\temp\search.html#elements) parameters on the capabilities interaction, though this is not required. In addition, servers are encouraged to implement the [$subset](file:///C:\temp\capabilitystatement-operations.html#subset) and [$implements](file:///C:\temp\capabilitystatement-operations.html#implements) operations to make it easy for a client to check conformance.

In addition to this capabilities interaction, a server may also choose to provide the standard set of interactions (read, search, create, update) defined on this page for the [CapabilityStatement Resource](file:///C:\temp\capabilitystatement.html) end-point. This is different from the capabilities interaction:

|  |  |
| --- | --- |
| capabilities interaction | returns a capability statement describing the server's current operational functionality |
| CapabilityStatement end-point | manages a repository of capability statements (e.g. the HL7 capability statement registry) |

All servers are required to support the capabilities interaction, but servers may choose whether they wish to support the CapabilityStatement end-point, just like any other end-point.

**Implementation Note:** In DSTU 2 and earlier, the resource that this interaction returned was named "Conformance". Clients often connect to a server, and use the capabilities interaction to check whether they are version and/or feature compatible with the server. Such clients should be able to process either a Conformance or a CapabilityStatement resource.

**batch/transaction**

The batch and transaction interactions submit a set of actions to perform on a server in a single HTTP request/response. The actions may be performed independently as a "batch", or as a single atomic "transaction" where the entire set of changes succeed or fail as a single entity. Multiple actions on multiple resources of the same or different types may be submitted, and they may be a mix of other interactions defined on this page (e.g. read, search, create, update, delete, etc.), or using the [operations](file:///C:\temp\operations.html) framework.

The transaction mode is especially useful where one would otherwise need multiple interactions, possibly with a risk of loss of referential integrity if a later interaction fails (e.g. when storing a Provenance resource and its corresponding target resource, or, on document repositories, a document index entry and its accompanying document).

Note that transactions and conditional create/update/delete are complex interactions and it is not expected that every server will implement them.

A batch or transaction interaction is performed by an HTTP POST command as shown:

POST [base] {?\_format=[mime-type]}

The content of the post submission is a [Bundle](file:///C:\temp\bundle.html) with [Bundle.type](file:///C:\temp\bundle-definitions.html#Bundle.type) = [batch](file:///C:\temp\valueset-bundle-type.html#batch) or [transaction](file:///C:\temp\valueset-bundle-type.html#transaction). Each entry carries request details ([Bundle.entry.request](file:///C:\temp\bundle-definitions.html#Bundle.entry.request)) that provides the HTTP details of the action in order to inform the system processing the batch or transaction what to do for the entry (note: the request is [optional](file:///C:\temp\bundle.html#transaction), but SHOULD be present). If the HTTP command is a PUT or POST, then the entry SHALL contain a resource for the body of the action. The resources in the bundle are each processed separately as if they were an individual interactions or operations as otherwise described on this page, or the [Operations framework](file:///C:\temp\operations.html). The actions are subject to the normal processing for each, including the [meta element](file:///C:\temp\resource.html#meta), verification and version aware updates, and [transactional integrity](file:///C:\temp\main-pages.html#transactional-integrity).

Examples:

* [Transaction Example](file:///C:\temp\bundle-transaction.html) with [Matching Response](file:///C:\temp\bundle-response.html)
* [Batch request to fetch Meds & Allergies](file:///C:\temp\bundle-request-medsallergies.html) with [Response](file:///C:\temp\bundle-response-medsallergies.html)
* [Batch request to fetch simple Patient Summary](file:///C:\temp\bundle-request-simplesummary.html) with [Response](file:///C:\temp\bundle-response-simplesummary.html)

**Batch Processing Rules**

For a batch, there SHALL be no interdependencies between the different entries in the Bundle that cause change on the server. The success or failure of one change SHOULD not alter the success or failure or resulting content of another change. Servers SHOULD validate that this is the case. Note that it is considered that servers execute the batch in the same order as that specified below for transactions, though the order of execution should not matter given the previous rule.

Static references within a Bundle.entry.resource another to Bundle.entry.resource that is being created within the Batch are be non-conformant.

**Transaction Processing Rules**

For a transaction, servers SHALL either accept all actions and return a 200 OK, along with a response bundle (see below), or reject all resources and return an HTTP 400 or 500 type response. It is not an error if the submitted bundle has no resources in it. The outcome of the processing the transaction SHALL NOT depend on the order of the resources in the transaction. A resource can only appear in a transaction once (by identity).

Because of the rules that a transaction is atomic where all actions pass or fail together and the order of the entries doesn't matter, there is a particular order in which to process the actions:

1. Process any DELETE interactions
2. Process any POST interactions
3. Process any PUT interactions
4. Process any GET interactions

If any resource identities (including resolved identities from conditional update/delete) overlap in steps 1-3, then the transaction SHALL fail.

**STU Note:** clients are able to request that operations be executed as part of a transaction. Some transactions can cause side effects, such as the creation of new resources or other actions that may be difficult to fit into a transaction framework. Input regarding this issue is sought during the STU period.

Provide feedback [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

A transaction may include references from one resource to another in the bundle, including circular references where resources refer to each other. If the server assigns a new id to any resource in the bundle as part of the processing rules above, it SHALL also update any references to that resource in the same bundle as they are processed. References to resources that are not part of the bundle are left untouched. Version-specific references should remain as version-specific references after the references have been updated. Servers SHALL replace all matching links in the bundle, whether they are found in the resource ids, resource references, url elements, or <a href="" & <img src="" in the narrative.

When processing a "POST" (create), the full URL is treated as the id of the resource on the source, and is ignored; the server generates an id for the resource. For updates, the server performs a mapping between the fullUrl specified and the local URL the server knows that instance as, if possible. If the server does not have a mapping for the fullUrl, the server ignores the base URL and attempts an update assuming the base is the same as the server base. This allows the same transaction bundle to be sent to multiple systems without changing the fullUrls for each target.

When processing a batch or transaction, a server MAY choose to honor existing logical ids (e.g. Observation/1234 remains as Observation/1234 on the server), but since this is only [safe in controlled circumstances](file:///C:\temp\main-pages.html#upsert), servers may choose to assign new ids to all submitted resources, irrespective of any claimed logical id in the resource, or fullUrl on entries in the batch/transaction.

**Note:** this behavior is subject to verification based on implementation experience and may change.

**Conditional References**

When constructing the bundle, the client may not know the logical id of a resource, but it may know identifying information - e.g. an identifier. This situations arises commonly when building transactions from v2 messages. The client could resolve that identifier to a logical id using a search, but that would mean that the resolution to a logical id does not occur within the same transaction as the commit (as well as significantly complicating the client). Because of this, in a transaction (and only in a transaction), references to resources may be replaced by a search URI that describes how to find the correct reference:

<Bundle xmlns="http://hl7.org/fhir">

<id value="20160113160203" />

<type value="transaction" />

<entry>

<fullUrl value="urn:uuid:c72aa430-2ddc-456e-7a09-dea8264671d8" />

<resource>

<Observation>

<subject>

<reference value="Patient?identifier=12345"/>

</subject>

<--! rest of resource omitted -->

</Observation>

</resource>

<request>

<method value="POST" />

</request>

</entry>

<Bundle>

The search URI is relative to the servers [base] path, and always starts with a resource type: [type]:?parameters.... Only filtering parameters are allowed; none of the parameters that control the return of resources are relevant.

When processing transactions, servers SHALL:

* check all references for search URIs
* For search URIs, use the search to locate matching resources
* if there are no matches, or multiple matches, the transaction fails, and an error is returned to the user
* if there is a single match, the server replaces the search URI with a reference to the matching resource

**Batch/Transaction Response**

For a batch, or a successful transaction, the response the server SHALL return a [Bundle](file:///C:\temp\bundle.html) with [type](file:///C:\temp\bundle-definitions.html#Bundle.type) set to batch-response or transaction-response that contains one entry for each entry in the request, in the same order, with the outcome of processing the entry. For a failed transaction, the server returns a single [OperationOutcome](file:///C:\temp\operationoutcome.html) instead of a Bundle.

A client may use the return bundled to track the outcomes of processing the entry, and the identities assigned to the resources by the server. Each entry element SHALL contain a response element which details the outcome of processing the entry - the HTTP status code, and the location and ETag header values, which are used for identifying and versioning the resources. In addition, a resource may be included in the entry, as specified by the [Prefer](file:///C:\temp\main-pages.html#prefer) header.

**Accepting Other bundle types**

A server may choose to accept bundle types other than batch or transaction when POSTed to the [base] URL.

Bundles of type history inherently have the same structure as a transaction, and can be treated as either a transaction or batch, so servers SHOULD accept a history bundle - this makes it possible to replicate data from one server to another easily using a pub/sub model. Note, however, that the original transaction boundaries may not represented in a history list, and a resource may occur more than once in a history list, so servers processing history bundles must have some strategy to manage this.

For other bundle types, should the server choose to accept them, there will be no request element (note that every entry will have a resource). In this case, the server treats the entry as either a create or an update interaction, depending on whether it recognises the identity of the resource - if the identity of the resource refers to a valid location on the server, it should treat it as an update to that location. Note: this option allows a client to delegate the matching process to the server.

**history**

The history interaction retrieves the history of either a particular resource, all resources of a given type, or all resources supported by the system. These three variations of the history interaction are performed by HTTP GET command as shown:

GET [base]/[type]/[id]/\_history{?[parameters]&\_format=[mime-type]}

GET [base]/[type]/\_history{?[parameters]&\_format=[mime-type]}

GET [base]/\_history{?[parameters]&\_format=[mime-type]}

The return content is a [Bundle](file:///C:\temp\bundle.html) with [type](file:///C:\temp\bundle-definitions.html#Bundle.type) set to history containing the specified version history, sorted with oldest versions last, and including deleted resources. Each entry SHALL minimally contain either a resource which holds the resource as it is at the conclusion of the interaction, or a request with entry.request.method The request provides information about the interaction that occurred to cause the new version, and allows, for instance, subscriber system to differentiate between create and update interactions. The principal reason a resource might be missing is that the resource was changed by some other channel rather than via the RESTful interface. If the entry.request.method is a PUT or a POST, the entry SHALL contain a resource.

The interactions [create](file:///C:\temp\main-pages.html#create), [update](file:///C:\temp\main-pages.html#update), and [delete](file:///C:\temp\main-pages.html#delete) create history entries. Other interactions do not (note that these operations may produce side-effects such as new AuditEvent resources; these are represented as create interactions in their own right). New resources or updates to existing resources that are triggered by operations also appear in the history, as do updates to the resources that result from interactions outside the scope of the RESTful interface.

A create interaction is represented in a history interaction in the following way:

<entry>

<resource>

<Patient>

<!-- the id of the created resource -->

<id value="23424"/>

<!-- snip -->

</Patient>

</resource>

<request>

<!-- POST: this was a create -->

<method value="POST"/>

<url value="Patient"/>

</request>

</entry>

Note that conditional creates, updates and deletes are converted to direct updates and deletes in a history list.

In addition to the standard \_format parameter, the parameters to this interaction may also include:

|  |  |  |
| --- | --- | --- |
| \_count : [integer](file:///C:\temp\datatypes.html#integer) | single | Number of return records requested. The server is not bound to return the number requested, but cannot return more |
| \_since : [instant](file:///C:\temp\datatypes.html#integer) | single | Only include resource versions that were created at or after the given instant in time |
| \_at : [dateTime](file:///C:\temp\datatypes.html#integer) | single | Only include resource versions that were current at some point during the time period specified in the date time value (may be more than one) |

The history list can be restricted to a limited period by specifying a \_since parameter which contains a full date time with time zone. Clients should be aware that due to timing imprecision, they may receive notifications of a resource update on the boundary instant more than once. Servers are not required to support a precision finer than by second.

The updates list can be long, so servers may use paging. If they do, they SHALL use the method [described below](file:///C:\temp\main-pages.html#paging) for breaking the list into pages if appropriate, and maintain the specified \_count across pages.

The history interaction can be used to set up a subscription from one system to another, so that resources are synchronized between them. Refer to the [Subscription resource](file:///C:\temp\subscription.html) for an alternate means of system synchronization.

Additional Notes about maintaining a history of resources:

* The history is a record version history on a per-resource basis. It is not intended to support concurrent versions, or multi-branch version history
* Accordingly, there is no way to update or delete past versions of the record, except that the [metadata can be modified](file:///C:\temp\resource-operations.html) (mainly for access control purposes)
* All past versions of a resource are considered to be superceded, and no longer active, but retained for audit/integrity purposes
* In the case that a past version of a resource needs to be explicitly documented as ['entered-in-error'](file:///C:\temp\lifecycle.html#error), use as provenance resource pointing to the past version of the resource
* When tracing the history of a specific resource, applications should retrieve any provenance resources relating to the resource or it's past versions
* If a request is made for a history that is not available (e.g. the system does not keep a history for the type, or the particular instance), the server should return a 404 Not Found along with an [OperationOutcome](file:///C:\temp\operationoutcome.html) explaining the problem

**Transactional Integrity**

When processing [create](file:///C:\temp\main-pages.html#create) and [update](file:///C:\temp\main-pages.html#update) interactions, a FHIR server is not obliged to accept the entire resource as it is; when the resource is retrieved through a [read](file:///C:\temp\main-pages.html#read) interaction subsequently, the resource may be different. The difference may arise for several reasons:

* The server merged updated content with existing content
* The server applied business rules and altered the content
* The server does not fully support all the features or possible values of the resource

Note that there is no general purpose method to make merging with existing content or altering the content by business rules safe or predictable - what is possible, safe and/or required is highly context dependent. These kinds of behaviors may be driven by security considerations. With regard to incomplete support, clients can consult the server's base Capability statement profile references to determine which features or values the server does not support.

To the degree that the server alters the resource for any of the 3 reasons above, the FHIR server will create implementation consequences for the eco-system that it is part of, which will need to be managed (i.e. it will cost more). For this reason, servers SHOULD change the resource as little as possible, given the constraints of the system exposing the FHIR resource. However due to the variability that exists within healthcare, this specification allows that servers MAY alter the resource on create/update.

Similarly, to the degree that an implementation context makes special rules about merging content or altering the content, that context will become more expensive to maintain.

Although these rules are stated with regard to servers, a similar concept applies to clients - to the degree that different client systems interacting with the server do not support the same feature set, the clients and/or the server will be forced to implement custom logic to prevent information from being lost or corrupted.

Some of these problems can be mitigated by following a pattern built on top of version-aware updates. In this pattern:

* The server provides a [read](file:///C:\temp\main-pages.html#read) interaction for any resource it accepts [update](file:///C:\temp\main-pages.html#update) interactions on
* Before updating, the client [reads](file:///C:\temp\main-pages.html#read) the latest version of the resource
* The client applies the changes it wants to the resource, leaving other information intact (note the [extension related rules](file:///C:\temp\extensibility.html#exchange) around this)
* The client writes the result back as an [update](file:///C:\temp\main-pages.html#update) interaction, and is able to handle a 409 or 412 response (usually by trying again)

If clients follow this pattern, then information from other systems that they do not understand will be maintained through the update.

Note that it's possible for a server to choose to maintain the information that would be lost, but there is no defined way for a server to determine whether the client omitted the information because it wasn't supported (perhaps in this case) or whether it wishes to delete the information.

**Conformance**

Both client and server systems SHOULD clearly document how transaction integrity is handled.

**STU Note:** For now, the only way to document how transaction integrity is handled is as text in the narrative portions of the [CapabilityStatement](file:///C:\temp\capabilitystatement.html) resource. Feedback during the Trial use period on what - if any - of this information should be computable is welcome.

Provide feedback [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**Paging**

If servers provide paging for the results of a [search](file:///C:\temp\main-pages.html#search) or [history](file:///C:\temp\main-pages.html#history) interaction, they SHALL conform to this method (adapted from [RFC 5005 (Feed Paging and Archiving)](https://tools.ietf.org/html/rfc5005) for sending continuation links to the client when returning a [Bundle](file:///C:\temp\bundle.html) (e.g. with history and search). If the server does not do this then there is no way to continue paging.

This example shows the third page of a search result:

<Bundle xmlns="http://hl7.org/fhir">

<!-- snip metadata -->

<!-- This Search. url starts with base search, and adds the effective

parameters, and additional parameters for search state. All searches

SHALL return this value.

In this case, the search continuation method is that the server

maintains a state, with page references into the stateful list.

-->

<link>

<relation value="self">

<url value="http://example.org/Patient?name=peter&stateid=23&page=3"/>

</link>

<!-- 4 links for navigation in the search. All of these are optional, but recommended -->

<link>

<relation value="first"/>

<url value="http://example.org/Patient?name=peter&stateid=23&page=1"/>

</link>

<link>

<relation value="previous"/>

<url value="http://example.org/Patient?name=peter&stateid=23&page=2"/>

</link>

<link>

<relation value="next"/>

<url value="http://example.org/Patient?name=peter&stateid=23&page=4"/>

</link>

<link>

<relation value="last"/>

<url value="http://example.org/Patient?name=peter&stateid=23&page=26"/>

</link>

<!-- then the search results... -->

</Bundle>

The server need not use a stateful paging method as shown in this example - it is at the discretion of the server how to best ensure that the continuation retains integrity in the context of ongoing changes to the resources. An alternative approach is to use version specific references to the records on the boundaries, but this is subject to continuity failures when records are updated.

A server MAY add additional state tracking parameters to the links, as shown in the example above. The client must use the server supplied links in order to traverse the pages. A server MAY inform the client of the total number of resources returned by the interaction for which the results are paged using the [Bundle.total](file:///C:\temp\bundle-definitions.html#Bundle.total).

Note that for search, where \_include can be used to return additional related resources, the total number of resources in the feed may exceed the number indicated in totalResults.

**Intermediaries**

The HTTP protocol may be routed through an HTTP proxy such as squid. Such proxies are transparent to the applications, though implementers should be alert to the effects of caching, particularly including the risk of receiving stale content. See the [HTTP specification](http://tools.ietf.org/html/rfc2616#page-74) for further detail

Interface engines may also be placed between the consumer and the provider. These differ from proxies because they actively alter the content and/or destination of the HTTP exchange and are not bound the rules that apply to HTTP proxies. Such agents are allowed, but SHALL mark the HTTP header to assist with troubleshooting.

Any agent that modifies an HTTP request or response content other than under the rules for HTTP proxies SHALL add a stamp to the HTTP headers like this:

request-modified-[identity]: [purpose]

response-modified-[identity]: [purpose]

The identity SHALL be a single token defined by the administrator of the agent that will sufficiently identify the agent in the context of use. The header SHALL specify the agent's purpose in modifying the content. End point systems SHALL not use this header for any purpose. Its aim is to assist with system troubleshooting.

**OMG hData RESTful Transport**

This RESTful specification described here is based on the [OMG Health RESTful specification (HData)](http://www.omg.org/spec/HData/). In this regard, FHIR functions as a Record Format Profile as described in that specification. Note the following significant factors to be aware of:

* FHIR maps the hData sections to resource types, and hData documents to resource instances. There are no subsections, and client systems are not able to create new sections, though [compartments](file:///C:\temp\compartmentdefinition.html) behave somewhat like sections
* Because clients cannot submit new sections (POST to service URL), POST to the service URL has been re-used for [the transaction interaction](file:///C:\temp\main-pages.html#transaction) (difference under review)
* FHIR does not (yet) define a root document. When defined, it will contain information about what the FHIR server has done (as opposed to a Capability Statement, which describes what it is capable of doing)
* Note that this specification does support the hData RESTful Transport OPTIONS command on the service URL

**Summary**

These tables present a summary of the interactions described here. Note that *all* requests may include an optional Accept header to indicate the format used for the response (this is even true for DELETE since an OperationOutcome may be returned).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Interaction** | **Path** | **Request** | | | | |
|  | | **Verb** | **Content-Type** | **Body** | **Prefer** | **Conditional** |
| read | /[type]/[id] | GET | N/A | N/A | N/A | O: ETag, If-Modified-Since, If-None-Match |
| vread | /[type]/[id]/\_history/[vid] | GET | N/A | N/A | N/A | N/A |
| update | /[type]/[id] | PUT | R | Resource | O | O: If-Match |
| delete | /[type]/[id] | DELETE | N/A | N/A | N/A | N/A |
| create | /[type] | POST | R | Resource | O | O: If-None-Exist |
| search | /[type]? | GET | N/A | N/A | N/A | N/A |
| /[type]/\_search? | POST | application/x-www-form-urlencoded | form data | N/A | N/A |
| search-all | ? | GET | N/A | N/A | N/A | N/A |
| capabilities | /metadata | GET | N/A | N/A | N/A | N/A |
| transaction | / | POST | R | Bundle | O | N/A |
| history | /[type]/[id]/\_history | GET | N/A | N/A | N/A | N/A |
| history-type | /[type]/\_history | GET | N/A | N/A | N/A | N/A |
| history-all | /\_history | GET | N/A | N/A | N/A | N/A |
| (operation) | /$[name], /[type]/$[name] or /[type]/[id]/$[name] | POST | R | Parameters | N/A | N/A |
| GET | N/A | N/A | N/A | N/A |
| POST | application/x-www-form-urlencoded | form data | N/A | N/A |

Notes:

* N/A = not present, R = Required, O = optional
* For operations defined on all resources, including direct access to the meta element, see [Resource Operations](file:///C:\temp\resource-operations.html)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Interaction** | **Response** | | | | | |
|  | **Content-Type** | **Body** | **Location** | **Versioning** | **Status Codes** |  |
| read | R | R: Resource | N/A | R: ETag, Last-Modified | 200, 404, 410 |  |
| vread | R | R: Resource | N/A | R: ETag, Last-Modified | 200, 404 |  |
| update | R if body | O: Resource (Prefer) | R on create | R: ETag, Last-Modified | 200, 201, 400, 404, 405, 409, 412, 422 |  |
| delete | R if body | O: OperationOutcome | N/A | N/A | 200, 204, 404, 405, 409, 412 |  |
| create | R if body | O : Resource (Prefer) | R | R: ETag, Last-Modified | 201, 400, 404, 405, 422 |  |
| search | R | R: Bundle | N/A | N/A | 200, 401? |  |
| search-all | R | R: Bundle | N/A | N/A | 200, 401? |  |
| capabilities | R | R: CapabilityStatement | N/A | N/A | 200, 404 |  |
| transaction | R | R: Bundle | N/A | N/A | 200, 400, 404, 405, 409, 412, 422 |  |
| history | R | R: Bundle | N/A | N/A | 200 |  |
| history-type | R | R: Bundle | N/A | N/A | 200 |  |
| history-all | R | R: Bundle | N/A | N/A | 200 |  |
| (operation) | R | R: Parameters/Resource | N/A | N/A | 200 |  |

Note: this table lists the status codes described here, but other status codes are possible as described by the HTTP specification. Additional codes that are likely are server errors and various codes associated with authentication protocols.

**icd.html**

**Using ICD-[X] Codes with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

ICD is a family of code systems maintained by [WHO](http://www.who.int/classifications/icd/en/), with many countries publishing their own variants.

The are two principal revisions of ICD in use - ICD-10 and ICD-9 (note that while US usage has recently updated to ICD-10, there is still a lot of legacy data coded in ICD-9). Though these can be referred to as different versions of ICD, they are entirely distinct sets of codes, with significant differences in organization and coding rules. Plus, if period ('.') characters are disregarded (as occurs in some systems), a few of the codes are overlapping between ICD-9 and ICD-10. Given these considerations, ICD-9 and ICD-10 are represented as entirely separate code systems. The next revision, ICD-11, is scheduled for release in 2017.

**Summary**

|  |  |  |
| --- | --- | --- |
|  | ICD-10 | ICD-9 |
| Source | [WHO](http://www.who.int/classifications/icd/en/) or see below | ([CDC](http://www.cdc.gov/nchs/icd/icd9.htm)) |
| System | http://hl7.org/fhir/sid/icd-10 and  http://hl7.org/fhir/sid/icd-10-[x] (see below) | http://hl7.org/fhir/sid/icd-9-cm |
| OID | 2.16.840.1.113883.6.3 | 2.16.840.1.113883.6.42 |
| Version | The versioning convention and frequency may vary between the various ICD revisions and variants. WHO publishes a new version of ICD-10 annually in October (with minor updates annually, and major updates, if required, every 3 years - the versions are identified as 'YYYY', e.g. '2016'). The national variants (which in some cases are used multi-nationally) are also typically published on an annual basis (refer to the documentation for a particular variant for details). | |
| Code | Either an ICD-10 or ICD-9 code, or a dual coding expression - see below | |
| Display | There are no specific conventions required or described for obtaining the complete display string for use for each code. | |
| Inactive | ICD-9 and ICD-10 do not have codes that are identified as inactive (although in some cases previously included codes may have been removed or changed in meaning in subsequent versions). | |
| Hierarchy | The tabular representations are organized with headings and multiple levels of codes (typically based on the numbers of digits contained in each code), but an explicit hierarchy is not defined. | |
| Filter Properties | None are described yet. | |

**ICD-10 variants**

Variants (other variants exist which are not listed here):

|  |  |  |
| --- | --- | --- |
| Germany | http://hl7.org/fhir/sid/icd-10-de | 2.16.840.1.113883.6.3.2 |
| Netherlands | http://hl7.org/fhir/sid/icd-10-nl | 1.2.276.0.76.5.409 |
| United States | http://hl7.org/fhir/sid/icd-10-cm | 2.16.840.1.113883.6.90 |

**Dual Coding**

Note that for dual coding (see volume 2 [ICD-10 Manual](http://www.who.int/entity/classifications/icd/ICD-10_2nd_ed_volume2.pdf)), section 3.1.3 *Two codes for certain conditions*), simply use the two ICD-10 codes separated by a space, e.g. "J21.8 B95.6".

Dual coding in ICD-10 is used for the following (and potentially other) reasons:

* Coding for diagnostic statements containing information about both (1) an underlying generalized disease and (2) a manifestation in a particular organ or site which is a clinical problem in its own right. This is referred to as the "dagger and asterisk" system, as it marks the primary code for the underlying disease with a dagger (Ã¢â‚¬Â ) and an optional additional code for the manifestation with an asterisk (\*).
* Other optional dual coding (noted in the tabular representation as Ã¢â‚¬Å“Use additional code, if desired ...Ã¢â‚¬Â)
  + For local infections, classifiable to the Ã¢â‚¬Å“body systemsÃ¢â‚¬Â chapters, codes from Chapter I may be added to identify the infecting organism, where this information does not appear in the title of the rubric.
  + For neoplasms with functional activity. To the code from Chapter II may be added the appropriate code from Chapter IV to indicate the type of functional activity.
  + For neoplasms, the morphology code from Volume 1, although not part of the main ICD, may be added to the Chapter II code to identify the morphological type of the tumour.
  + For conditions classifiable to F00-F09 (Organic, including symptomatic, mental disorders) in Chapter V, where a code from another chapter may be added to indicate the cause, i.e. the underlying disease, injury or other insult to the brain.
  + Where a condition is caused by a toxic agent, a code from Chapter XX may be added to identify that agent.
  + Where two codes can be used to describe an injury, poisoning or other adverse effect: a code from Chapter XIX, which describes the nature of the injury, and a code from Chapter XX, which describes the cause.

For "dagger and asterisk" coding of an underlying generalized disease and a specific manifestation, it is recommended that the primary code for the underlying disease is listed first.

Dual Coding Example: "Staph aureus bronchiolitis" is coded using ICD-10 codes J21.8 "Acute bronchiolitis due to other specified organisms" and B95.6 "Staphylococcus aureus as the cause of diseases classified to other chapters" as

<coding>

<system value="http://hl7.org/fhir/sid/icd-10"/>

<code value="J21.8 B95.6"/>

</coding>

**Copyright/License Issues**

ICD-10 is Ã‚Â©Copyright World Health Organization (WHO). WHO licenses its published material widely, in order to encourage maximum use and dissemination. See [Licensing WHO classifications](http://www.who.int/about/licensing/classifications/en/) for details. The ICD variants have their own separate copyright and licensing (refer to the documentation for the particular variant for details).

**ICD-X Filter Properties**

No need for filters identified yet.

**Implicit Value Sets**

No need for implicit value sets identified yet.

**identifier-registry.html**

**Known Identifier Systems**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The following names (URIs) may be used in the *system* element of the [Identifier](file:///C:\temp\datatypes.html#Identifier) data type. If a URI is defined here, it SHALL be used in preference to any other identifying mechanism. If an identifier system is not listed here, the correct URI may be determined by working through the following list, in order:

* the HL7 OID Registry
* the documentation associated with the identifier
* consulting the owner of the identifier
* asking on the HL7 vocabulary mailing list

See also the [list of known coding systems](file:///C:\temp\terminologies-systems.html) that can be used in the *system* element of the [Coding](file:///C:\temp\datatypes.html#Coding) data type. Additional identifier systems may be registered on the HL7 FHIR registry at <http://hl7.org/fhir/registry>.

**Important Note:** This list of URIs is incomplete and subject to change. Some values may be dropped and others will likely be added in the coming months as HL7 institutes formal processes around URIs in vocabulary.

The URI column indicates the correct value to use in the [Identifier](file:///C:\temp\datatypes.html#Identifier).system. The OID is provided for compatibility with [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185)/[CDA](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7) based systems. Note that any URL in http://example.org is reserved for testing and documentation purposes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **identifier** | **URI** | **OID (for non-FHIR systems)** | **Type** | **Comment** |
| URIs ([W3C](http://www.w3.org)): when the identifier is a URI | urn:ietf:rfc:3986 | 2.16.840.1.113883.4.873 |  | As defined by [RFC 3986](http://www.ietf.org/rfc/rfc3986.txt) (with many schemes defined in many RFCs). For OIDs and UUIDs, use the URN form ([urn:oid:](http://www.ietf.org/rfc/rfc3001.txt) (note: lowercase) and [urn:uuid:](http://www.ietf.org/rfc/rfc4122.txt) |

**ig-profiles.html**

**Profiles Defined by this Implementation Guide**

This table contains a list of all the profiles defined as part of the Implementation Guide.

**ig-terminologies-valuesets.html**

**Value Sets Defined in**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\resource.html#maturity): | [Ballot Status](file:///C:\help.html#status): |

This table contains a list of all the value sets defined as part of the . See also the list of value sets [defined by the FHIR specification](file:///C:\terminologies-valuesets.html).

**ig-valuesets.html**

**Value Sets Defined by this Implementation Guide**

This table contains a list of all the value sets defined as part of the Implementation Guide. Some of these value sets include codes defined elsewhere, some define their own codes, and some do both.

**iglist.html**

**Implementation Guides defined as part of FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

An Implementation Guide is a collection of [Profiles](file:///C:\temp\profiling.html), that make [Conformance](file:///C:\temp\conformance-rules.html) rules (technially, [CapabilityStatement](file:///C:\temp\capabilitystatement.html), [StructureDefinition](file:///C:\temp\structuredefinition.html), [ValueSet](file:///C:\temp\valueset.html), etc.) that, together with textual guidance, provides instruction on how to make use of FHIR in a particular problem space. Implementation guides may be developed and published as part of a FHIR release or as separate specifications. Additional support and tooling for implementation guides is being developed, so the content provided here should be considered "preliminary" from a publication perspective.

The implementation guides include artifacts with a set of representations generated from the conformance resources. All the representation forms are equally normative. Many examples are also provided, and considerable care is taken over them. However, unless indicated otherwise, the examples have no formal standing (see [General Notes about examples](file:///C:\temp\conformance-rules.html#examples)).

Published Implementation Guides:

* [Published by HL7, Affiliates & FHIR Foundation](http://www.fhir.org/guides/registry)
* [Oher IGs (FHIR Wiki)](http://wiki.hl7.org/index.php?title=FHIR_Profiles_from_other_Organizations)

**implsupport-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Implementation Support Module**

**Introduction**

This section provides information which will be useful for FHIR implementors, including information about available libraries, tools, and other similar resources, as well as where to seek help.

**Index**

In addition to the content below, a number of implementation resources can be found on the [Downloads Page](file:///C:\temp\downloads.html).

This module also contains some specific documentation that relates to issues commonly encountered by developers:

|  |  |  |
| --- | --- | --- |
| * [Testing FHIR](file:///C:\temp\testing.html) + [TestScript](file:///C:\temp\testscript.html) + [TestReport](file:///C:\temp\testreport.html) * [Validating Resources](file:///C:\temp\validation.html) * [Mapping Language](file:///C:\temp\mapping-language.html) ([Tutorial](file:///C:\temp\mapping-tutorial.html)) * [Common Usages](file:///C:\temp\usecases.html) * [Inter-version Compatibility](file:///C:\temp\compatibility.html) | * [Clinical Safety Considerations](file:///C:\temp\safety.html) * [How FHIR fits into an EHR](file:///C:\temp\ehr-fm.html) * [Managing Resource Identity](file:///C:\temp\managing.html) * [Interaction Patterns](file:///C:\temp\pushpull.html) * [Update Rules](file:///C:\temp\updates.html) * [Clinical Examples](file:///C:\temp\integrated-examples.html) | * [Comparisons:](file:///C:\temp\comparison.html)   + [v2](file:///C:\temp\comparison-v2.html) &   + [v3 Messaging](file:///C:\temp\comparison-v3.html)   + [CDA](file:///C:\temp\comparison-cda.html) + [CDA on FHIR](file:///C:\temp\cda-intro.html)   + [Other Specifications](file:///C:\temp\comparison-other.html) |

**Security and Privacy**

For more general considerations, see [the Security and Privacy module](file:///C:\temp\secpriv-module.html).

**Common use Cases**

**For Client Developers and Testers: Reference Servers**

The following reference servers have been created by the FHIR team, and are provided made available to help implemeters to test their code. While the reference servers are not considered to be a normative part of the FHIR specification, the maintainers make every effort to ensure that they are fully compliant.

Note that there are a large number of servers available for testing that are not listed here. A full list is available on the HL7 wiki [here](http://wiki.hl7.org/index.php?title=Publicly_Available_FHIR_Servers_for_testing).

| **Server Name** | **Maintainer** | **Link** |
| --- | --- | --- |
| Healthintersections | Grahame Grieve | <http://fhir3.healthintersections.com.au/> |
| Spark | Furore Informatica | <http://spark.furore.com/> |
| HAPI | University Health Network / James Agnew | <http://fhirtest.uhn.ca/> |
| HealthConnex | HealthConnex / Brian Postlethwaite | <http://sqlonfhir-may.azurewebsites.net/fhir> |

**For Developers: Reference Implementations (Libraries)**

The following reference implementations are made available under an open-source license. These libraries may be used by developers to quickly add FHIR capabilities to their applications.

| **Language** | **Library** | **Link** | **License** |
| --- | --- | --- | --- |
| .NET / C# | FHIR .NET API | <https://github.com/ewoutkramer/fhir-net-api> | BSD-3 |
| Java | HAPI FHIR | <http://hapifhir.io> | Apache 2.0 |
| Swift | Swift FHIR | <https://github.com/smart-on-fhir/Swift-FHIR> | Apache 2.0 |
| JavaScript | fhir.js | <https://github.com/smart-on-fhir/fhir.js> | MIT |
| Python | Client Py | <https://github.com/smart-on-fhir/client-py> | Apache 2.0 |
| Pascal | FHIR Pascal | <http://hl7.org/fhir/downloads.html> | BSD-3 |

**For Profilers**

The following tools are available to profilers wishing to create profiles for use in their implementations. See the [conformance module](file:///C:\temp\conformance-module.html) for information on profiling.

| **Tool** | **Link** |
| --- | --- |
| **Forge** | [**http://fhir.furore.com/Forge**](http://fhir.furore.com/Forge) |

**For Testers**

The following tools are available to solution testers who want to test FHIR implementations for conformance to the FHIR specification.

| **Tool** | **Link** |
| --- | --- |
| Touchstone | <http://www.aegis.net/touchstone.html> |
| Crucible | <https://projectcrucible.org/> |

**Getting Help**

The following are a few ways that implementers can seek help as they work with FHIR:

* [FHIR Community Forum](http://community.fhir.org/) (maintained by fhir.org)
* [FHIR Chat Channel / Zulip](http://chat.fhir.org/) (maintained by fhir.org)
* [StackOverflow](http://stackoverflow.com/questions/tagged/hl7_fhir) (General tech community, use the tag hl7\_fhir)

**Developmental Roadmap**

The reference servers and reference implementations generally try to keep up to date with recent changes to the FHIR specification. Each server may have multiple endpoints which are held to a specific version of the specification, but generally there will also be endpoints available for testing which conform to a very recent build.

Efforts are now underway to create a curated collection of quality test data which can be used by FHIR implementers to help test their applcations. This collection will be made available when it is ready.

**index.html**

**Welcome to FHIR®**

**First time here?**  
See the [executive summary](file:///C:\temp\summary.html), the [developer's introduction](file:///C:\temp\overview-dev.html), [clinical introduction](file:///C:\temp\overview-clinical.html), or [architect's introduction](file:///C:\temp\overview-arch.html), and then the FHIR [overview / roadmap](file:///C:\temp\overview.html) & [Timelines](file:///C:\temp\timelines.html). See also the [open license](file:///C:\temp\license.html) (and don't miss the full [Table of Contents](file:///C:\temp\toc.html) or you can [search this specification](file:///C:\temp\search.cfm)).

**External Links:**

|  |  |  |
| --- | --- | --- |
| **Implementation Guides**  Specifications based on the FHIR standard   * [Published by HL7, Affiliates & FHIR Foundation](http://www.fhir.org/guides/registry) * [Other IGs (FHIR Wiki)](http://wiki.hl7.org/index.php?title=FHIR_Profiles_from_other_Organizations) | [**FHIR Foundation**](http://fhir.org)  Enabling health interoperability through FHIR   * [Community Forum](http://community.fhir.org/) + [FHIR Chat](http://char.fhir.org/) * [Public Test Servers & Software](http://wiki.hl7.org/index.php?title=Publicly_Available_FHIR_Servers_for_testing) * [Blogs that cover FHIR](http://wiki.hl7.org/index.php?title=FHIR_Blogs) * [FHIR Wiki](http://wiki.hl7.org/index.php?title=FHIR) | **Translations**  Note that translations are not always up to date   * [Russian](http://fhir-ru.github.io/index.html) * [Chinese](https://github.com/wanghaisheng/fhir-cn/blob/source/README.md) * [Japanese](https://sites.google.com/site/fhirjp/) |

Note: This specification requires a browser that is SVG compatible (Microsoft Internet Explorer 10+, Firefox 3.0+, Chrome, or Safari), and uses the browser's session storage to remember which tabs are active.

**infrastructure.html**

**Infrastructure Resources**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

|  |  |  |
| --- | --- | --- |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |
| **Name** | **Aliases** | **Description** |

Additional Resources will be added in the future. A list of hypothesized resources can be found on the [HL7 wiki](http://wiki.hl7.org/index.php?title=FHIR_Resource_Types). Feel free to add any you think are missing or engage with one of the [HL7 Work Groups](http://www.hl7.org/Special/committees/index.cfm) to submit a [proposal](http://wiki.hl7.org/index.php?title=Category:FHIR_Resource_Proposal) to define a resource of particular interest.

**integrated-examples.html**

**Integrated Examples**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

These integrated examples are the result of choosing several real world healthcare records from a Dutch healthcare institution and representing them using FHIR resources. They are made published here for several reasons:

* To test that the resources are fit for purpose using real cases
* So that the available resource examples include real cases
* So that applications have real world clinical cases to use in development and prototyping

**Patient case 1**

The 69 year old Pieter van den Heuvel is diagnosed with several serious diseases over the last few years. In the spring and summer of 2011, Pieter had complains of fatigue, dyspnea and even fainting. After visiting his physician, Pieter was referred to the cardiologist for further examination. At the hospital a malfunction of the heart valve was diagnosed and eventually replaced. Pieter recovered well in the next 2 months, but probably needs a life time prescription on blood pressure reducing medication. In the summer of 2012, Pieter again visited his physician for a chest injury after he fell from the stairs. By coincidence, the doctor noticed a suspicious stain on the X-thorax and performed some additional tests on Pieter. After the results, Pieter was diagnosed with early stage non-small cell lung cancer. Fortunately, the tumor didn't spread to other parts of the body. Tumor type and location made it possible to perform a partial lobectomy in the right lung. After his recovery, Pieter was submitted to a high risk control group for yearly screening.

In 2013, Pieter was charged with an increasing neck swelling, a stiff neck and difficulty swallowing. Because of breathing problems, Pieter was urgently admitted to the hospital. An upper respiratory infection caused a retropharyngeal abscess in the back of the throat. Surgery was needed to remove the abscess. The surgeons recommended a tracheotomy during the surgery, so there was less change of complications during the abscess removal. A tracheotomy was intubated before the surgery. After the removal of the retropharyngeal abscess, the tracheotomy was extubated as the swelling was reduced and normal breathing became reinstated.

**Patient demographic information**

Patient name: [Pieter van den Heuvel](file:///C:\temp\patient-example-f001-pieter.html)

Patient number: 0108173

Date of birth: 17-11-1944

Gender: Male

**Patient contact information**

Address: van Egmondkade 23

Zip code: 1024 RJ

City: Amsterdam

Phone: +31648352638

email: p.heuvel@gmail.com

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Intakes | | | | | | | | | |
| **intake date** | **surgery date** | **practitioner** | **specialism** | **diagnose** | **procedure** | **care plan** | **surgery code** | **SNOMED CT code** | **related encounter** |
| 26-6-2011 | 27-6-2011 | [P. Voigt](file:///C:\temp\practitioner-example-f002-pv.html) | [CTC (Cardio-thoracale Chirurgie)](file:///C:\temp\organization-example-f002-burgers-card.html) | [Heart valve disorder](file:///C:\temp\condition-example-f001-heart.html) | [Heart valve replacement](file:///C:\temp\procedure-example-f001-heart.html) | [CP2903](file:///C:\temp\careplan-example-f001-heart.html) | 1000263502 | 64915003 | [v1451](file:///C:\temp\encounter-example-f001-heart.html) |
| 6-7-2012 | 7-7-2012 | [M.I.M. Versteegh](file:///C:\temp\practitioner-example-f003-mv.html) | [CTC (Cardio-thoracale Chirurgie)](file:///C:\temp\organization-example-f002-burgers-card.html) | [NSCLC - Non-small cell lung cancer](file:///C:\temp\condition-example-f002-lung.html) | [partial lobectomy](file:///C:\temp\procedure-example-f002-lung.html) | [CP2934](file:///C:\temp\careplan-example-f002-lung.html) | 1000263813 | 173171007 | [v3251](file:///C:\temp\encounter-example-f002-lung.html) |
| 22-3-2013 | 22-3-2013 | [A.P.M. Langeveld](file:///C:\temp\practitioner-example-f005-al.html) | [ENT](file:///C:\temp\organization-example-f003-burgers-ENT.html) | [Retropharyngeal abscess](file:///C:\temp\condition-example-f003-abscess.html) | [Trachea-tracheotomy](file:///C:\temp\procedure-example-f003-abscess.html) | CP2938 | 1000050465 | 48387007 | [v6751](file:///C:\temp\encounter-example-f003-abscess.html) |
| 24-3-2013 | 24-3-2013 | [E.M.J.M. van den broek](file:///C:\temp\practitioner-example-f001-evdb.html) | [ENT](file:///C:\temp\organization-example-f003-burgers-ENT.html) | [Retropharyngeal abscess](file:///C:\temp\condition-example-f003-abscess.html) | [retropharyngeal abscess](file:///C:\temp\procedure-example-f003-abscess.html) | [CP3953](file:///C:\temp\careplan-example-f003-pharynx.html) | 1000049161 | 172960003 | [v6751](file:///C:\temp\encounter-example-f003-abscess.html) |
| 27-3-2013 | 27-3-2013 | [R.J.P. Briet](file:///C:\temp\practitioner-example-f004-rb.html) | [ENT](file:///C:\temp\organization-example-f003-burgers-ENT.html) | [Retropharyngeal abscess](file:///C:\temp\condition-example-f003-abscess.html) | Trachea-extubatie | CP1283 | 1000050159 | 309812005 | [v6751](file:///C:\temp\encounter-example-f003-abscess.html) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Device | | | | | | |
| **device** | **hopital/home** | **dose** | **admission route** | **start date** | **presciber** | **SNOMED CT code** |
| [Feeding tube](file:///C:\temp\device-example-f001-feedingtube.html) | hopital | N.A. | tube | 10-4-2013 | R.A. van den Berk | 61420007 |

|  |
| --- |
| Lab results |
| diagnostic blood report |
| **date** | **substance** | **value** | **status** |
| 2013-04-02 | [glucose](file:///C:\temp\observation-example-f001-glucose.html) | 6,3 mmol/l | abnormal |
| 2013-04-02 | [base excess](file:///C:\temp\observation-example-f001-glucose.html) | 12,6 mmol/l | abnormal |
| 2013-04-02 | [carbon dioxide](file:///C:\temp\observation-example-f001-glucose.html) | 6,2 mm[Hg] | abnormal |
| 2013-04-02 | [erytrocyten](file:///C:\temp\observation-example-f001-glucose.html) | 18,7 g/dl | abnormal |
| 2013-04-02 | [Hemoglobin](file:///C:\temp\observation-example-f001-glucose.html) | 7,5 g/dl | abnormal |

**Anamnese**

***date: 8-6-2012*** Bronchoscopy; several biopsy specimen were taken from pathological mucosa, right main bronchus specimen send for pathologic analysis. Bronchoscopy because of atelectasis right. X-thorax; increase in atelectasis and pleural liquid. Bronchoscopy;   
-fluids drained from right main bronchus   
-pathalogic mucous membrane right bronchus, easily bleeding   
-left bronchial system open ***date: 18-3-2013*** Antiobiotic policy with retropharyngeal abscess with prolapse to the mediatinum. No surgical possibility to fully drain the mediatinum.

**Physical investigation**

***date: 18-3-2013*** Neck; swelling and redness pretracheal extending to chest. No fluctation, however induration is present. Swelling back pharynx, also present in postcricoid area. Light stridor sound when breathing. Overall condition is good. Scoop; little supraglottic swelling, vocal chords not judgable.

**Additional research**

CT thorax: no mediastinal/retropharyngal collering of the fluid collections. Diffuse edema retro/parapharyngeal and mediastinitis. Preoperative culture shows; S pyogenes, sensitivity for peneciline and E. cloacae, sensitivity for vancomycin .

**Policy**

Stop vancomycin prescription. Continue penicilin and start vancomycin supplementation. Total treatment time is approximatly 6 weeks. In consultation with the IC/anesthesia; intubation. Analyse lower respiratory tract.

**Conclusion**

Extubation on OR. Normal voice, no stridor. However, hypoxic at low respiratory tract obstruction with 84% O2 and rapid breathing.

**Patient case 2**

Patient name: [Roel Bor](file:///C:\temp\patient-example-f201-roel.html)

Patient number: 123456789

Date of birth: 1960-03-13

Gender: Male

**Patient contact information**

Address: Bos en Lommerplein 280

Zip code: 1055 RW

City: Amsterdam

Phone: +31612345678

Roel Bor is in his mid-fifties, works for an IT-company and has two healthy children and a wife. His uncle unfortunately was less healthy and died of cancer. At the end of 2012, Roel was also diagnosed with a tumor in the Erasmus Medical Center. His tumor is located in the head-neck area. The standard treatment for that is TPF-chemotherapy. The tumor is not fully curable due to its position, but it was optimally minimized with the therapy in the AUMC. Two severe complications followed the treatment, namely bacterial sepsis (streptococcus aureus) and renal failure. Both were stabilized within a short period through medication (see care plan). Roel thereafter had his ups and downs with the last known condition of a severe fever, for which he received among other treatment (paracetamol). On a side-note and as a final statement: the patient suffers from house dust allergy.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Intakes | | | | | | | | | | |
| **practitioner** | **organization** | **careplan** | **start date** | **substance** | **condition** | **other participant** | **procedure** | **encounter ID** | **surgery code** | **SNOMED CT code** |
| [A. Bronsig](file:///C:\temp\practitioner-example-f201-ab.html) | [Artis University Medical Center](file:///C:\temp\organization-example-f201-aumc.html) | [CP3928](file:///C:\temp\careplan-example-f201-renal.html) | 2013-03-11 | [potassium](file:///C:\temp\substance-example-f203-potassium.html) | [fever](file:///C:\temp\condition-example-f201-fever.html) | [Carla Espinosa](file:///C:\temp\practitioner-example-f204-ce.html) |  | [20130404](file:///C:\temp\encounter-example-f201-20130404.html) | 100028475 | 64915003 |
| [A. Bronsig](file:///C:\temp\practitioner-example-f201-ab.html) | [Artis University Medical Center](file:///C:\temp\organization-example-f201-aumc.html) | [CP7364](file:///C:\temp\careplan-example-f202-malignancy.html) |  |  | [malignant tumor](file:///C:\temp\condition-example-f202-malignancy.html) |  | [chemotherapy](file:///C:\temp\procedure-example-f201-tpf.html) | [20130128](file:///C:\temp\encounter-example-f202-20130128.html) | 100028475 | 363346000 |
| [A. Bronsig](file:///C:\temp\practitioner-example-f201-ab.html) | [Artis University Medical Center](file:///C:\temp\organization-example-f201-aumc.html) | [CP8766](file:///C:\temp\careplan-example-f203-sepsis.html) | 2013-03-11 |  | [Bacterial sepsis](file:///C:\temp\condition-example-f203-sepsis.html) | [Luigi Maas](file:///C:\temp\practitioner-example-f202-lm.html) | observation | [20130311](file:///C:\temp\encounter-example-f203-20130311.html) | 100028475 | 363346000 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Diagnoses | | | | | | | | | |
| **practitioner** | **date** | **encounter** | **reason** | **condition** | **medication** | **medication prescription** | **procedure** | **diagnostic report** | **issued by** |
| [A. Bronsig](file:///C:\temp\practitioner-example-f201-ab.html) | 28-1-2013 | [20130128](file:///C:\temp\encounter-example-f202-20130128.html) | chemotherapy | [malignant tumor](file:///C:\temp\condition-example-f202-malignancy.html) |  |  | [chemotherapy](file:///C:\temp\procedure-example-f201-tpf.html) | [BUMC](file:///C:\temp\organization-example-f203-bumc.html) |  |
| [A. Bronsig](file:///C:\temp\practitioner-example-f201-ab.html) | 4-3-2013 | [20130404](file:///C:\temp\encounter-example-f201-20130404.html) | fever | [fever](file:///C:\temp\condition-example-f201-fever.html) |  |  |  |  |  |
| [A. Bronsig](file:///C:\temp\practitioner-example-f201-ab.html) | 11-3-2013 | [20130311](file:///C:\temp\encounter-example-f203-20130311.html) | sepsis | [Bacterial sepsis](file:///C:\temp\condition-example-f203-sepsis.html) |  | [AUMC](file:///C:\temp\organization-example-f201-aumc.html) |  |  |  |
| [A. Bronsig](file:///C:\temp\practitioner-example-f201-ab.html) | 11-3-2013 | [20130311](file:///C:\temp\encounter-example-f203-20130311.html) | renal | [Renal insufficiency](file:///C:\temp\condition-example-f204-renal.html) |  | [AUMC](file:///C:\temp\organization-example-f201-aumc.html) |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Observations | | | | | |
| **practitioner** | **date** | **observation** | **value** | **status** | **SNOMED CT code** |
| [Luigi Maas](file:///C:\temp\practitioner-example-f202-lm.html) | 11-3-2013 | [Blood culture for bacteria](file:///C:\temp\observation-example-f206-staphylococcus.html) | Gram-positive bacteria | positive | 8745002 |
| [A. Bronsig](file:///C:\temp\practitioner-example-f201-ab.html) | 4-4-2013 | [temperature taken](file:///C:\temp\observation-example-f202-temperature.html) | 39 degrees Celsius1 | high | 89003005 |
| [A. Bronsig](file:///C:\temp\practitioner-example-f201-ab.html) | 4-4-2013 | [Serum bicarbonate measurement](file:///C:\temp\observation-example-f203-bicarbonate.html) | 28mmol/L | normal | 271239003 |
| [Luigi Maas](file:///C:\temp\practitioner-example-f202-lm.html) | 4-4-2013 | [Serum creatinine raised](file:///C:\temp\observation-example-f204-creatinine.html) | 122 umol/L | high | 166717003 |
| [Luigi Maas](file:///C:\temp\practitioner-example-f202-lm.html) | 4-4-2013 | [Epidermal Growth Factor Receptor](file:///C:\temp\observation-example-f205-egfr.html) | Negative for EGF receptor expression (Non-small cell lung cancer) | negative | 427038005 |

**json.html**

**JSON Representation of Resources**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 5 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The JSON representation for a resource is described using this format:

{

"resourceType" : "[**[Resource Type]**](file:///C:\temp\resourcelist.html)",

// from [Source](file:///C:\temp\json.html): [element #1](file:///C:\temp\json.html)

"[property1](file:///C:\temp\json.html)" : "<[[primitive]](file:///C:\temp\datatypes.html)>", // short description

"[property2](file:///C:\temp\json.html)" : { [[Data Type]](file:///C:\temp\datatypes.html) }, // short description

"[property3](file:///C:\temp\json.html)" : { // Short Description

"[propertyA](file:///C:\temp\json.html)" : { [CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept) }, // [Short Description](file:///C:\temp\json.html) ([Example](file:///C:\temp\terminologies.html#example))

},

"[property4](file:///C:\temp\json.html)" : [{ // Short Description

"[propertyB](file:///C:\temp\json.html)" : { [Reference](file:///C:\temp\references.html#Reference)([ResourceType](file:///C:\temp\resourcelist.html)) } // **R!** Short Description

}]

}

Using this format:

* To build a valid JSON instance of a resource, replace the contents of the property values with valid content as described by the type rules and content description found in the property value for each element
* In this example:
  1. property1 has a primitive data type; the value of the property will be as described for the stated type
  2. property2 has a complex data type; the value of the property is an object that has the content as described for the stated type
  3. property3 is an object property that contains additional properties (e.g. propertyA; the allowable properties are listed (but also include extensions as appropriate)
  4. property4 is an array property that contains items which are objects themselves. The items may have any of the types already encountered in points 1-3
  5. propertyA is an example of an object property that has a binding to a value set - the Short description is a link to the value set. In addition, the binding strength is shown
  6. propertyB is an example of an object property that has a reference to a particular kind of resource
* Property names are case-sensitive (though duplicates that differ only in case are never defined)
* Properties can appear in any order
* XHTML is represented as an escaped string
* Objects are never empty. If an element is present in the resource, it SHALL have properties as defined for its type, or 1 or more [extensions](file:///C:\temp\extensibility.html)
* String property values can never be empty. Either the property is absent, or it is present with at least one character of content
* The **R!** denotes that an element is mandatory - it must be present (or in an array, at least one item must be present)
* In this format, // is used for comments but these can't be in the JSON instances
* The character encoding is always UTF-8
* The MIME-type for this format is application/fhir+json.

Given the way [extensions](file:///C:\temp\extensibility.html) work, applications reading JSON resources will never enocunter unknown properties. However once an application starts trading with other appplications that conform to later versions of this specification, unknown properties may be encountered. Applications MAY choose to ignore unknown properties in order to foster forwards compatibility in this regard, but may also choose not to. Applications declare their behavior with regard to unknown elements using [CapabilityStatement.acceptUnknown](file:///C:\temp\capabilitystatement-definitions.html#CapabilityStatement.acceptUnknown).

**Comparison with XML**

The JSON format is similar to the XML format:

* The names for the JSON object members are the same as the names of the elements and attributes in XML, including elements that may repeat. Property names are case sensitive
* Just as in XML, JSON objects and arrays are never empty, and properties never have null values (except for a special case documented below). Omit a property if it is empty
* JSON whitespace is not part of the contents of a resource. Applications MAY preserve the whitespace when handling resources, but are not required to do so. Note that digital signatures may depend on the whitespace

There are differences to XML:

* There are no namespaces in the JSON representation
* The type of the resource is represented differently in JSON - instead of being the name of the base object (there is none in JSON), it is carried as the property resourceType
* The order of properties of an object is not significant in the JSON representation, though order within an array SHALL be maintained
* JSON does not have a notion of attributes versus elements, so attributes (e.g. id, value) are handled differently (see below)
* JSON has the array notation, which is used to represent repeating elements. Note that arrays are used when the item might repeat, even if it does not repeat in a specific instance
* The XHTML <div> element in the [Narrative](file:///C:\temp\narrative.html) datatype is represented as a single escaped string of XHTML. This is to avoid problems in JSON with mixed content, etc. The XHTML SHALL still conform to the rules described for [the Narrative](file:///C:\temp\narrative.html)

The JSON format for the resources follows the standard XML format closely to make interconversion easy, and so that XPath queries can easily be mapped to query the JSON structures. However the differences - particularly the repeating element one, which cannot be avoided - mean that generic XML --> JSON converters are not able to perform correctly. The [reference platforms](file:///C:\temp\downloads.html#refimpl) provide XML <--> JSON conversion functionality that accommodates these FHIR-specific characteristics.

**JSON Representation for repeating elements**

An element that has a maximum cardinality of >1 (e.g. x..\* in the definitions) may occur more than once in the instance. In XML, this is simply done by repeating the XML element multiple times. In JSON, this is done by using an array type. Note that:

* The name of the array is singular - the same as the XML element
* An item that may repeat is represented as an array even in the case that it doesn't repeat so that the process of parsing the resource is the same either way

<**coding**>

<**system** value="http://snomed.info/sct"/>

<**code** value="104934005"/>

</**coding**/>

<**coding**>

<**system** value="http://loinc.org"/>

<**code** value="2947-0"/>

</**coding**/>

is represented in JSON like this:

"coding": [

{

"system" : "http://snomed.info/sct",

"code" : "104934005"

},

{

"system" : "http://loinc.org",

"code" : "2947-0"

}

]

**JSON representation of primitive elements**

FHIR elements with primitive data types are represented in two parts:

* A JSON property with the name of the element, which has a JSON type of number, boolean, or string
* a JSON property with \_ prepended to the name of the element, which, if present, contains the value's id and/or extensions

The FHIR types [integer](file:///C:\temp\datatypes.html#integer) and [decimal](file:///C:\temp\datatypes.html#decimal) are represented as a JSON number, the FHIR type [boolean](file:///C:\temp\datatypes.html#boolean) as a JSON boolean, and all other types are represented as a JSON string which has the same content as that specified for the relevant data type. Whitespace is always significant (i.e. no leading and trailing spaces for non-strings).

<**code** value="abc"/> <!-- code -->

<**date** value="1972-11-30"/> <!-- dateTime -->

<**deceased** value="false" /> <!-- boolean -->

<**count** value="23" /> <!-- integer -->

is represented in JSON as

"code" : "abc",

"date" : "1972-11-30",

"deceased" : false,

"count" : 23

When using a JavaScript JSON.parse() implementation, note that JavaScript natively supports only one numeric datatype, which is a floating point number. This can cause loss of precision for FHIR numbers. In particular, trailing 0s after a decimal point will be lost e.g. 2.00 will be converted to 2. The FHIR decimal data type is defined such that precision, including trailing zeros, is preserved for presentation purposes, and this is widely regard as critical for correct presentation of clinical measurements. Implementations should consider using a custom parser and big number library (e.g. <https://github.com/jtobey/javascript-bignum>) to meet these requirements.

If the value has an id attribute, or extensions, then this is represented as follows:

<**birthDate** id="314159" value="1970-03-30" >

<extension url="http://example.org/fhir/StructureDefinition/text">

<valueString value="Easter 1970"/>

</extension>

</**birthDate**>

is represented in JSON as:

"birthDate": "1970-03-30",

"\_birthDate": {

"id": "314159",

"extension" : [ {

"url" : "http://example.org/fhir/StructureDefinition/text",

"valueString" : "Easter 1970"

}]

}

Note: If the primitive has an id attribute or extension, but no value, only the property with the \_ is rendered.

In the case where the primitive element may repeat, it is represented in two arrays. JSON null values are used to fill out both arrays so that the id and/or extension are aligned with the matching value in the first array, as demonstrated in this example:

<**code** value="au"/>

<**code** value="nz">

<extension url="http://hl7.org/fhir/StructureDefinition/display">

<valueString value="New Zealand a.k.a Kiwiland"/>

</extension>

</**code**>

is represented in JSON as:

"code": [ "au", "nz" ],

"\_code": [

null,

{

"extension" : [ {

"url" : "http://hl7.org/fhir/StructureDefinition/display",

"valueString" : "New Zealand a.k.a Kiwiland"

}]

}

]

Note: when one of the repeating elements has no value, it is represented in the first array using a null. When an element has a value but no extension/id, the second array will have a null at the position of that element.

Design Note: The representation of primitive data types has been split into two parts like this in order to simplify the representation of simple primitive values without id or extensions. This does have the cost of making the representation of the id attribute and extensions more ungainly, but these are both rarely used with primitive data types.

**JSON representation of Elements, and Complex Data types**

Elements, and complex [datatypes](file:///C:\temp\datatypes.html) (types that contain named elements of other types) are represented using a JSON object, containing a member for each element in the datatype. Composites can have id attributes, which are converted to JSON member values, in the same manner as described for primitives. For example:

<person>

<text>

<status value="generated" />

<div xmlns="http://www.w3.org/1999/xhtml"><p>...</p></div>

</text>

<name>

<use value="official" />

<given value="Karen" />

<family id="a2" value="Van" />

</name>

</person>

is represented in JSON as:

{

"person" : {

"name" : [{

"use" : "official" ,

"given" : [ "Karen" ],

"family" : [ "Van" ]

"\_family" : [ {"id" : "a2"} ]

}],

"text" : {

"status" : "generated" ,

"div" : "<div xmlns=\"http://www.w3.org/1999/xhtml\"><p>...</p></div>"

}

}

Things to note here are:

* Both given and family are repeating XML elements, so they are serialised as an Array whether or not they repeat in this instance
* In the family part of name, the id is added represented in \_family as described above
* The XHTML content in the div element which is in the Narrative element text is represented as an escaped string in the value property in JSON. The xhtml's root element needs to be a <div> in the xhtml namespace

**JSON representation of Resources**

A resource is a JSON object with a property resourceType which informs the parser which resource type this is:

{

"resourceType" : "Patient",

"text" : {

"status" : "generated" ,

"div" : "<div xmlns=\"http://www.w3.org/1999/xhtml\"><p>...</p></div>"

}

etc...

}

Note that parsers cannot assume that the resourceType property will come first.

**Design Note**: This is a problem for several JSON -> Object serialisers that assume that the resourceType property does come first, including [Json.NET](http://james.newtonking.com/json). However some JSON generators do not give the authoring application control of the order of the property values, and so these implementations cannot inter-operate with implementations that make assumptions about order. Given that JSON says that the property values are an unordered map of name/value pairs, this specification cannot require that properties come in any particular order, though implementers may choose to fix the property order if they are able (and the reference platforms provided with this specification do so).

There is [a sample file](file:///C:\temp\json-edge-cases.json) with many edge cases to help test JSON parsers.

**Canonical JSON**

Resources and/or Bundles may be digitally signed (see [Bundle](file:///C:\temp\bundle.html) and [Provenance](file:///C:\temp\provenance.html)).

This specification defines the following method for canonicalizing FHIR resources, when represented as JSON:

* No whitespace other than single spaces in property values and in the xhtml in the [Narrative](file:///C:\temp\narrative.html)
* Order properties alphabetically
* Omit all properties that have a default value, if a default value is defined

This canonicalization method is identified by the URL http://hl7.org/fhir/canonicalization/json. The following additional canonicalization URLS are also defined:

|  |  |
| --- | --- |
| http://hl7.org/fhir/canonicalization/json#data | The narrative (Resource.text) is omitted prior to signing (note the deletion is at Resource.text, not Resource.text.div) |
| http://hl7.org/fhir/canonicalization/json#static | In addition to narrative (Resource.text), the Resource.meta element is removed. This makes the signature robust as the content is moved from server to server, or workflow and access tags are added or removed |
| http://hl7.org/fhir/canonicalization/json#narrative | This method only retains the Resource.id and Narrative elements |
| http://hl7.org/fhir/canonicalization/json#document | The signs everything in a Bundle, except for the Bundle.id and Bundle.metadata on the root Bundle (allows for a document to copied from server to server) |

These canonicalization methods allow system the flexibility to sign the various portions of the resource that matter for the workflow the signature serves.

Note: One consequence of signing the document is that URLs, identifiers and internal references are frozen and cannot be changed. This might be a desired feature, but it may also cripple interoperability between closed ecosystems where [re-identification](file:///C:\temp\managing.html) frequently occurs. For this reason, it is recommended that systems consider carefully the impact of any signature processes. The impact of signatures on [Document bundles](file:///C:\temp\documents.html) and their related processes is the most well understood use of digital signatures.

**license.html**

**License and Legal Terms**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Disclaimer and Warning of Use**

FHIR Resource definitions developed by HL7 are derived from the considerable collective experience of the HL7 membership and wide community feedback from the development and application of a spectrum of health care interoperability solutions. However, Resource definitions are generalized to support multiple contexts of use. It is the responsibility of the persons or organizations using these Resources to ensure their use is fit for the particular purpose in which they are used, including validation for clinical and operational use.

See also the specific warnings associated with [use of the STU](file:///C:\temp\todo.html).

**FHIR License**

Copyright © 2011+ HL7.

This specification (specifically the set of materials included in the fhir-spec.zip file available from the Downloads page of this specification) is produced by HL7 under the terms of HL7Ã‚Â® [Governance and Operations Manual](http://www.hl7.org/documentcenter/public_temp_4108B35F-1C23-BA17-0C38BD44A97683FB/membership/HL7_Governance_and_Operations_Manual.pdf) relating to Intellectual Property (Section 16), specifically its copyright, trademark and patent provisions.

This document is licensed under Creative Commons "No Rights Reserved" ([CC0](http://creativecommons.org/publicdomain/zero/1.0/)).

HL7Ã‚Â®, HEALTH LEVEL SEVENÃ‚Â®, FHIRÃ‚Â® and the FHIR Ã‚Â® are trademarks owned by Health Level Seven International, registered with the United States Patent and Trademark Office.

**Additional information about the license**

* FHIR is © and ® HL7. The right to maintain FHIR remains vested in HL7
* You can redistribute FHIR
* You can create derivative specifications or implementation-related products and services
* You can't claim that HL7 or any of its members endorses your derived [thing] because it uses content from this specification
* Neither HL7 nor any of the contributors to this specification accept any liability for your use of FHIR
* You cannot publish an altered version of the FHIR specification unless it clearly identifies that it is a derivative specification, not FHIR itself
* Derivative Specifications cannot redefine what conformance to FHIR means
* HL7 is not responsible for either identifying patents for which a license may be required to implement FHIRÃ‚Â® or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention (section 16.03.04, [Governance and Operations Manual](http://www.hl7.org/documentcenter/public_temp_4108B35F-1C23-BA17-0C38BD44A97683FB/membership/HL7_Governance_and_Operations_Manual.pdf))"

While HL7 places no restrictions on the right to copy the FHIR specification or to create derivative works per the Creative Commons license, HL7 protects the "FHIR" trademark carefully. This means that:

* When referencing the FHIRÃ‚Â® standard in a web site, document, presentation, or otherwise, please in a place of prominence refer to it as the "HL7Ã‚Â® FHIRÃ‚Â® standard". In subsequent uses, please refer to it as the "FHIRÃ‚Â® standard" or "FHIRÃ‚Â®", using the Ã‚Â® symbol as often as is practical, at least once on each page of printed matter, generally in connection with the first or dominant usage.
* When using the  logo, always include the Ã‚Â® symbol
* Use of Health Level Seven International trademarks in URL domains or to brand your product or service without the express written consent of Health Level Seven International is strictly prohibited
* All non-trademark uses of the foregoing marks should include the Ã‚Â® symbol and indicate in text that:  
  "HL7, FHIR and the FHIR [FLAME DESIGN] are the registered trademarks of Health Level Seven International."

Trademark FAQs are posted on the [HL7 International website](http://www.hl7.org/legal/trademarks.cfm). Questions? Please contact [HL7trademarks@HL7.org](mailto:HL7trademarks@HL7.org) for more information.

**Third-party artifacts and terminologies**

The HL7 FHIR specification contains and references intellectual property owned by third parties ("Third Party IP"). Acceptance of these License Terms does not grant any rights with respect to Third Party IP. The licensee alone is responsible for identifying and obtaining any necessary licenses or authorizations to utilize Third Party IP in connection with the specification or otherwise.

Following is a non-exhaustive list of third-party artifacts and terminologies that may require a separate license:

American Medical Association ([AMA](http://www.ama-assn.org/))  
CPT copyright 2014 American Medical Association. All rights reserved.

|  |  |
| --- | --- |
| **Artifact/Terminology** | **Statement/Owner/Contact** |
| SNOMED CT | International Healthcare Terminology Standards Developing Organization ([IHTSDO](http://ihtsdo.org)). This specification includes content from SNOMED CT, which is copyright Ã‚Â© 2002+ International Health Terminology Standards Development Organisation (IHTSDO), and distributed by agreement between IHTSDO and HL7. Implementer use of SNOMED CT is not covered by this agreement |
| Logical Observation Identifiers Names & Codes (LOINC) | This material contains content from LOINC® (<http://loinc.org>). The LOINC table, LOINC codes, and LOINC panels and forms file are copyright © 1995-2014, 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>. |
| International Classification of Diseases (ICD) codes | Consult the World Health Organization ([WHO](http://who.int)) |
| Current Procedures Terminology (CPT) code set |  |

**lifecycle.html**

**FHIR Life Cycle Page**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 3 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page describes several issues around lifecycle management for the resources and the content they contain. Specifically, this page describes:

* [Resource Status](file:///C:\temp\main-pages.html#status): how resource status codes work
* [Current List](file:///C:\temp\main-pages.html#current): issues associated with retrieving "current X list" of resources
* [Entered in Error](file:///C:\temp\main-pages.html#error): information about how erroneous entry is handled for the resources

**Resource Status**

Many FHIR resources have a status element that represents the lifecycle state of the resource or the clinical process represented by the resource. Work groups can specify status values appropriate to the individual resource. Although consistency between resources is not the primary objective, it is helpful to users and developers to have well-crafted value sets that cover all possible states (since the value sets are typically required and non-extensible).

To understand existing status elements, and to help create extensions and resources involving resource states, we note that status value sets follow one of the following life cycles:

* Clinical workflow process life cycle
* Request/Order life cycle
* Entity status life cycle
* Clinical status life cycle

**Clinical Workflow Process Life Cycle**

Describes the lifecycle states of complex activities common in healthcare. Typically, these states follow a chronological life cycle that leads from initiation to the conclusion of the action. A characteristic (but non-exhaustive) set of states for the clinical workflow process life cycle include:

* planned - resources for the activity are being allocated but the activity has not begun
* cancelled - the planned activity did not start and will not take place
* in-progress - the activity has begun
* on-hold (suspended) - the activity has been temporarily interrupted
* stopped (aborted, failed) - the activity has not been completed but no future action is planned
* completed (finished) - the activity has been completed

Examples of the clinical workflow life cycle:

* Communication.status:
* Encounter.status:
* Goal.status:
* MedicationAdministration.status:
* MedicationDispense.status:
* Procedure.status:

**Request/Order Life Cycle**

Some resources in FHIR represent orders or requests. The request lifecycle can be generalized in terms of four stages: creating the request, sending the request, receiving acceptance or refusal of the request, and fulfillment of the request. A characteristic (but non-exhaustive) set of states for the request/order pattern include:

* proposed: An actor (e.g. a clinical decision support system) has proposed an action to be requested
* draft: The request is in preliminary form, prior to being requested
* requested: The request has been been made
* rejected: The request receiver has declined the request
* accepted: The request receiver has accepted the request
* in-progress: Work to fulfill the request has begun
* on-hold (suspended): Work on the request has been interrupted
* stopped (aborted): The activity has not been completed but no future action is planned
* completed: Work on the requested task has been completed, and no further action is required
* cancelled: The request has been withdrawn

Examples of the request/order life cycle:

* CommunicationRequest.status:
* DeviceRequest.status:
* MedicationRequest.status:
* ProcedureRequest.status:
* ReferralRequest.status:

**Entity Availability Life Cycle**

The entity availability life cycle indicates if the resource, or the entity described by the resource, is ready for use, not yet ready for use, or has been retired from use. A characteristic (but non-exhaustive) set of states for the entity availability life cycle include:

* draft: The entity is being prepared but is not yet in use
* active: The entity is in use
* suspended: The entity is not in use at the moment, but may return to active status
* amended: The entity has undergone a revision but is still active
* retired (superseded): The entity is no longer in use.

Examples of the entity availability life cycle:

* DiagnosticReport.status:
* MedicationStatement.status: . (note: in-progress and completed are states reflecting the administration of the medication)
* DocumentManifest.status:
* CapabilityStatement.status:
* StructureDefinition.status:
* DataElement:
* Questionnaire.status:
* DocumentReference.status:
* QuestionnaireResponse.status:
* Flag.status:
* Location.status:
* Organization.active:
* Patient.active:

**Clinical Status Life Cycle**

Clinical status is somewhat different than the previous status values, since it does not deal with workflow or lifecycle. Instead, it indicates how evidence is affecting a clinical interpretation. Here are two examples:

* AllergyIntolerance.clinicalStatus:
* Condition.clinicalStatus:

**Current Resource Lists**

Many clinical systems maintain current lists of some kind of resources for a patient. Some of the commonly maintained lists include:

* Current Problem List: a list of the problems that are of concern for care of the patient
* Current Medication List: a list of the medications that a patient is known to be on at the current time

Because of the way that resources are used, there is no simple way to determine, from examination of a resource, whether it is 'current' or not. Take, as an example, the [Condition](file:///C:\temp\condition.html) resource. In a typical EHR, condition resources might be published on the RESTful interface for the following reasons:

* to represent an item in a patient's curated problem list
* to represent a complaint or a diagnosis from an encounter record
* to represent a problem for investigation provided by a diagnostic system as part of a [ProcedureRequest](file:///C:\temp\procedurerequest.html)/[DiagnosticReport](file:///C:\temp\diagnosticreport.html) pair
* the resources were received from another system as part of a referral package, and were current for that system when they were received

There is no element on the Condition resource than can convey the difference between these usages. In particular, there can be no way to differentiate between current and past resources without having to retrospectively alter resources, which is problematic with regard to integrity and digital signatures.

One consequence of this is that searching the condition resource for a given patient will return more than just the patient's current problems. Though this is somewhat counter-intuitive to some implementers, restricting searches on Condition to only include the patient's current curated problem list excludes all the other - important - uses of the Condition resource.

Determining whether a Condition is an entry on a patient's current problem list is done by checking with the Condition resource is referenced from the correct list.

On the RESTful API, this is done using the [list search mechanism](file:///C:\temp\search.html#current):

GET [base]/AllergyIntolerance?patient=42&\_list=$current-allergies

This is a request to fetch all the allergies in the patient 42's "Currrent Problem List". Note that the server is not required to actually make a resource representation of the current allergy list available, though doing so assists clients in their audit/integrity tasks. See [List Operation "Find"](file:///C:\temp\list-operations.html#find) for further information.

In a document, current lists are determined by the code on a Composition section.

FHIR defines the following names for functional lists:

|  |  |  |  |
| --- | --- | --- | --- |
| **List** | **ResourceType** | **Description** | **Possible LOINC codes in documents / sections** |
| $current-problems | [Condition](file:///C:\temp\condition.html) | The "Currrent Problem List" - A list of current and active diagnoses as well as past diagnoses relevant to the current care of the patient | 46105-3 (Problem conditions Set) |
| $current-medications | [MedicationStatement](file:///C:\temp\medicationstatement.html)/ [MedicationRequest](file:///C:\temp\medicationrequest.html) | A list of all medications that the patient is taking. The 'current medications list' sometimes may include a mix of prescribed and over-the-counter medications - or only some of them. The list may contain a mix of [prescriptions](file:///C:\temp\medicationrequest.html) and more general [statements](file:///C:\temp\medicationstatement.html), or only one of the two. The list may also correspond to a formal reconciled medication administration schedule, but more often does not | 57828-6 (Prescription list), 10160-0 (History of medication) |
| $current-allergies | [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) | A list of known or suspected propensities to medications, foods, or environmental agents that is provided to help prevent reactions while care is occurring | 18716-1 (Allergy studies (set)), 52472-8 (Allergies and Adverse Drug Reactions), and 48765-2 (Allergies and adverse reactions Document) |
| $current-drug-allergies | [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) | A list of known or suspected propensities to medications that is provided to help prevent reactions while care is occurring. This list is a subset of the full allergies list | (same as above?) |

**Entered In Error Summary**

The entered-in-error state indicates the resource was created accidentally, and should be ignored. This state can apply to resources created by manual entry. It is usually not associated with the Clinical Workflow Process life cycle, but can be associated with the Request/Order and the Entity Availability life cycles.

This table summarizes what is expected to happen for each resource in the case that the data it contains is subsequently found to be an erroneous entry.

Note: Resources that are not listed in this table do not have any explicit documentation with regard to being entered in error.

**linked-data-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**FHIR Linked Data Module**

**Introduction**

This module describes the RDF representation for FHIR resources (FHIR/RDF) and related assets, including an OWL ontology for FHIR/RDF and a ShEx grammar to validate FHIR/RDF. *Linked Data* is structured data that is represented in an RDF format to facilitate inference and data linkage across datasets. Materials in this module are created and maintained by a collaboration between HL7 and W3C. Editor: David Booth

**Motivation for FHIR/RDF**

The purpose of defining an RDF representation of FHIR is not only to enable FHIR to be exchanged in an RDF format such as Turtle, it is also to ground the semantics of FHIR data in RDF, for use with ontologies and other RDF data. Since FHIR data is losslessly round-trippable between XML, JSON and RDF formats, any FHIR data can be used in conjunction with RDF. The semantics are the same regardless of source format.

Reasons for using RDF and ontologies with FHIR data include:

* **Shared semantics.** RDF's use of URIs as universal identifiers facilitates shared semantics across independently authored data.
* **Inference.** An ontology specifies relationships between concepts, which can be used to perform computer-based inference and formal reasoning over FHIR data. For example, using inference a query for heart valve surgeries could automatically include results that were documented as mitral valve surgeries, because a mitral valve is a kind of heart valve.
* **Data integration.** FHIR/RDF data can be integrated with other data, using RDF as a common semantic and representation layer -- including data that does not originate in an RDF format. For example, clinical trials data in XML (see https://clinicaltrials.gov/ ) can be translated to RDF and then combined with FHIR/RDF data.
* **Data validation.** An ontology provides a vocabulary of uniquely identified concepts, which facilitates data validation.
* **Error detection.** Computer-based reasoning can be used to help detect errors and inconsistencies in data and ontologies, and potential help repair them.
* **Compliance.** RDF and ontologies can be used to express compliance constraints, for example to control data projected between two privacy contexts or to encode access restrictions for queries.
* **Modularity.** RDF was designed to support modularity, such that a specialized ontology can be freely composed of a subset of concepts from a larger ontology.
* **Combining ontologies.** The FHIR ontology can be linked to other ontologies, through bridge ontologies, and used together to support ontology-enabled applications in overlapping domains.
* **Query.** SPARQL queries can be performed on FHIR/RDF data, without need for a FHIR-specific query language. SPARQL is a W3C standard query language for RDF data. It can be used uniformly to query over both FHIR data and other data -- even in the same query.

**Design of FHIR/RDF**

FHIR/RDF was designed to correspond very closely to FHIR/XML and FHIR/JSON in its look and feel, though there are some apparent differences that are necessary to ensure full-fidelity round tripping between all FHIR formats, or to accommodate FHIR's extensibility. For example, fhir:index is used to retain information about ordering within a FHIR list, and fhir:value is used to indicate the value of an element, while still allowing FHIR extensions to be attached.

**Using FHIR/RDF with Other Ontologies**

Ontologies that were designed independently almost always have some impedance mismatch when attempting to use them together, and the FHIR ontology is no exception. Many of the ontologies in the medical and life sciences domain are designed to capture facts about the world for research, such as the fact that the mitral valve is a kind of heart valve. But FHIR was designed to support the day-to-day operations of healthcare providers exchanging electronic health records (EHRs), and in this context the orientation has historically been different. When using FHIR/RDF with other ontologies this difference is likely to show up in two main ways:

* **Records versus facts.** FHIR is oriented toward recording who did what ("Dr. Jones diagnosed patient x with viral pneumonia") rather than stating absolute medical facts ("patient x has viral pneumonia").
* **Non-monotonicity.** RDF was designed to be monotonic, whereas FHIR has a few design aspects that are would be non-monotonic if they were interpreted directly in RDF. (Monotonicity means that new data cannot invalidate previous conclusions; non-monotonicity means that previous conclusions can be invalidated by new data.) For example, a modifier extension indicates that the surrounding element's meaning may be misunderstood if the modifier extension is not understood. Another example: an entered-in-error status on a FHIR resource means that the resource was created accidentally, and should be ignored.

For both of these reasons, to maintain monotonicity in RDF, FHIR/RDF should not be directly interpreted as stating facts, at least until any potentially non-monotonic elements have been removed or isolated. This could be done with a pre-processing step.

**FHIR/RDF Data Formats**

FHIR/RDF examples are provided for all FHIR resources in Turtle and JSON-LD formats.

The mime type for the Turtle format is text/turtle.

TODO: What is the status of the JSON-LD examples?

**Index**

Link to key content pages in this module:

* [FHIR/RDF Introduction](file:///C:\temp\rdf.html) See also: [Additional FHIR/RDF Introductory materials, not yet edited or merged](http://w3c.github.io/hcls-fhir-rdf/spec/)
* [FHIR/RDF formal specification (at W3C)](http://w3c.github.io/hcls-fhir-rdf/spec/)
* ShEx Validation Schema for FHIR RDF (TODO: add link)
* [FHIR Ontology](file:///C:\temp\ontology.html) See also: [Additional FHIR Ontology materials, not yet edited or merged](http://w3c.github.io/hcls-fhir-rdf/spec/ontology.html)
* [Example of AllergyIntolerance inference using FHIR/RDF with SNOMED CT. (Rough draft)](http://w3c.github.io/hcls-fhir-rdf/spec/ontology.html)

**Security and Privacy**

TODO: In this section: description of key security and privacy issues, or references to pages that deal with this. For the security / privcay module itself, this section does not exist. For other modules, the paragraph/section can end with this boilerplate: For more general considerations, see [the Security and Privacy module](file:///C:\temp\secpriv-module.html).

See also [HL7 Security and Privacy Ontology](http://wiki.hl7.org/index.php?title=Security_and_Privacy_Ontology)

**Common use Cases**

In this section: common problems in the space of the module, ways to go about solving them, or references to additional problem based linkes

**Developmental Roadmap**

In this section: what the current overall state is, what work is in train, what the goals over the next 18 months or so are

**logical.html**

**Logical Models**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

A logical model is a expression of a set of content in a FHIR ready style that helps implementers understand the content is a single package. Because of the way FHIR works, a single logical model may be actually represented during exchange by a set of resources, including a series of observations and questionnaires. Typically, profiles that define how the resources are used are derived from the logical model.

ToDo: figure out how all this happens.

**loinc.html**

**Using LOINC with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Summary**

|  |  |
| --- | --- |
| Source | LOINC is made available by the [Regenstrief Institute](https://www.regenstrief.org/) at <http://loinc.org> |
| System | The URI <http://loinc.org> identifies LOINC codes |
| Version | Where a version is used, it should be the standard LOINC version e.g. 2.48 |
| Code | The LOINC Code Identifier e.g. 21176-3. LOINC codes are not case sensitive. In addition, LOINC part codes and LOINC answer string ids can be used where appropriate |
| Display | Use either the SHORTNAME or LONG\_COMMON\_NAME field for the display |
| Inactive | codes with Property ACTIVE=DEPRECATED are considered inactive |
| Subsumption | LOINC defines the Multi-Axial Hierarchy, which is the basis for subsumption logic in LOINC |
| Filter Properties | Several properties are defined as described below |

Logically, LOINC codes and question codes correspond to [Data Elements](file:///C:\temp\dataelement.html). See [below](file:///C:\temp\main-pages.html#dataelements) for further information.

**Copyright Issues**

The terms of use for LOINC require that a [notice](file:///C:\temp\license.html#loinc) be included with any use of LOINC codes. This notice must appear in the *copyright* element of any value set that includes LOINC codes (either in the code or filter elements, or in an expansion):

<copyright value="This content LOINCÃ‚Â® is copyright Ã‚Â© 1995 Regenstrief Institute, Inc. and the LOINC Committee, and available at no cost under the license at http://loinc.org/terms-of-use"/>

Additional copyright statements may also be found in the *copyright* element. Some LOINC codes have 3rd party copyright statements. When these codes are included in a value set, they must carry their own copyright statement as well.

**Case Sensitivity**

For comparison purposes, LOINC codes, displays, and property values are not case sensitive, though implementers SHOULD maintain the correct case when using LOINC codes and property values.

**Use of LOINC PARTS**

As described in the [LOINC Manual](http://loinc.org/downloads/files/LOINCManual.pdf) section 10.2, LOINC Parts are a coded representation of a value for a dimension used to specify a LOINC Term which are assigned a non-semantic identifier with a "LP" prefix and a mod-10 check digit. Following the LOINC license, these part codes may be used in the following ways:

* In filter properties, as described below
* In [Structure Definitions](file:///C:\temp\structuredefinition-definitions.html), where the structure describes the use of a set of LOINC codes
* In a [ConceptMap](file:///C:\temp\conceptmap.html) resource, where mappings between LOINC codes and other codes are being defined

Part codes are the same LOINC system(http://loinc.org), and SHALL be represented in uppercase (e.g. LP31755-9).

**Use of LOINC Answer Lists**

LOINC also allocates Answer List and Answer String Ids for use in various forms and questionnaires. LOINC Answer String IDs are also valid LOINC codes:

<coding>

<system value="http://loinc.org"/>

<code value="LA11165-0"/>

<display value="Platelet anisocytosis"/>

</coding>

LOINC Answer List Ids are actually value set identifiers. See [below](file:///C:\temp\main-pages.html#alist) for how to use these.

**RDF**

LOIINC uses the namespace http://loinc.org/owl# as the root for LOINC concepts in the RDF space. This means that when a LOINC code is converted from the system::code pair, where the system is http://loinc.org, to the [RDF ontological form](file:///C:\temp\rdf.html), the representation is http://loinc.org/owl#[code].

**LOINC Properties**

In addition to the [standard properties](file:///C:\temp\terminology-service.html#standard-props), the following properties are defined for LOINC:

|  |  |  |
| --- | --- | --- |
| STATUS | boolean | Status of the term. For the purposes of LOINC, codes with Property ACTIVE=DEPRECATED are considered inactive |
| COMPONENT | code | First major axis-component or analyte: Analyte Name, Analyte sub-class, Challenge |
| PROPERTY | code | Second major axis-property observed: Kind of Property (also called kind of quantity) |
| TIME\_ASPCT | code | Third major axis-timing of the measurement: Time Aspect (Point or moment in time vs. time interval) |
| SYSTEM | code | Fourth major axis-type of specimen or system: System (Sample) Type |
| SCALE\_TYP | code | Fifth major axis-scale of measurement: Type of Scale |
| METHOD\_TYP | code | Sixth major axis-method of measurement: Type of Method |
| CLASS | string | An arbitrary classification of the terms for grouping related observations together |
| CONSUMER\_NAME | string | An experimental (beta) consumer friendly name for this item. The intent is to provide a test name that health care consumers will recognize; it will be similar to the names that might appear on a lab report |
| CLASSTYPE | string | 1=Laboratory class; 2=Clinical class; 3=Claims attachments; 4=Surveys |
| ORDER\_OBS | string | ?? |
| ORDER\_OBS | string | Provides users with an idea of the intended use of the term by categorizing it as an order only, observation only, or both |
| DOCUMENT\_SECTION | string | Classification of whether this LOINC code can be used a full document, a section of a document, or both |

Note that when a [$lookup](file:///C:\temp\codesystem-operations.html#lookup) operation is performed on a LOINC code, servers SHALL return the version being used (see above) in the version property. Other properties are at the discretion of the server and the client.

**LOINC Filters**

This section documents the property filters that can be used with the LOINC code system in value set composition statements.

**LOINC Property filter**

|  |  |
| --- | --- |
| Description | Allows the selection of a set of LOINC codes with a common property value (see list above) |
| Property Name | One of the names listed in the "Field Name" column in LOINC Database Structure (Appendix A of the LOINC manual) |
| Operations Allowed | = / regex |
| Values Allowed | [string value] |
| Comments | The 6 properties COMPONENT, PROPERTY, TIME\_ASPCT, SYSTEM, SCALE\_TYP, and METHOD\_TYP are most likely to be useful. |

**3rd Party Copyright**

|  |  |
| --- | --- |
| Description | Allows for the inclusion or exclusion of LOINC codes that include 3rd party copyright notices |
| Property Name | copyright |
| Operations Allowed | = |
| Values Allowed | LOINC | 3rdParty |
| Comments | LOINC = only codes with a sole copyright by Regenstrief. 3rdParty = only codes with a 3rd party copyright in addition to the one from Regenstrief. |

**Multi-Axial Hierarchy**

|  |  |
| --- | --- |
| Description | Allows for the selection of a set of codes base on their appearance in the LOINC multi-axial hierarchy |
| Property Name | parent | ancestor |
| Operations Allowed | = / in |
| Values Allowed | Part Code (or, for "in", multiple part codes separated by commas) |
| Comments | "parent" selects immediate parents only. For example, the code "44022-2" has the parent "LP52960-9". Ancestor includes parents transitively, e.g. "LP52960-9" eventually has a parent "LP31755-9", so the code "44022-2" is in the set of codes that have ancestor=LP31755-9 |

*TODO: Document Ontology*

**Implicit Value Sets**

Implicit value sets are those whose specification can be predicted based on the grammar of the underlying code system, and the known structure of the URL that refers to them. LOINC defines one set of implicit value sets: By Multi-Axial Hierarchy Entry.

If any value set resources exist with an identifier that conforms to the URL patterns specified below, the content of the resource must conform to the template provided. Profiles and other value set references are allowed to reference these value sets directly. *todo: can LOINC actually host a service that returns these?*

The value set identifier http://loinc.org/vs is a value set that contains all LOINC codes.

**LOINC Answer List**

LOINC defines a set of Answer lists, each of which contains a set of LOINC codes. LOINC answer lists are value sets. The value set identifier "http://loinc.org/vs/[id]" identifies a value set that contains a set of LOINC codes. For instance, the value set identifier http://loinc.org/vs/LL715-4 has the following definition for LOINC 2.52:

<ValueSet xmlns="http://hl7.org/fhir">

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml">

[some html that identifies that this value set

includes all LOINC codes in this answer list]

</div>

</text>

<url value="http://loinc.org/vs/LL715-4"/>

<version value="2.52"/>

<name value="Platelet morph"/>

<description value="LOINC Answer List for Platelet morph"/>

<status value="active"/>

<date value="[optional date of LOINC release]"/>

<compose>

<include>

<system value="http://loinc.org"/>

<concept>

<code value="LA11165-0"/>

<display value="Platelet anisocytosis"/>

</concept>

<concept>

<code value="LA11168-4"/>

<display value="Platelet clump"/>

</concept>

<concept>

<code value="LA11167-6"/>

<display value="Platelet large fragments"/>

</concept>

<concept>

<code value="LA11166-8"/>

<display value="Platelet satellitism"/>

</concept>

<concept>

<code value="LA11169-2"/>

<display value="Platelets.agranular"/>

</concept>

<concept>

<code value="LA11170-0"/>

<display value="Platelets.giant"/>

</concept>

<concept>

<code value="LA11172-6"/>

<display value="Platelets.large"/>

</concept>

<concept>

<code value="LA11171-8"/>

<display value="Platelets.small"/>

</concept>

</include>

</compose>

</ValueSet>

Here is an example of a LOINC Answer list used in a Questionnaire question:

<question>

<concept>

<system value="http://loinc.org"/>

<code value="11125-2"/>

<display value="Plat morph Bld"/>

</concept>

<type value="choice"/>

<options>

<reference value="http://loinc.org/vs/LL715-4"/>

</options>

</question>

**Multi-Axial Hierarchy Entry**

A value set with an identifier of "http://loinc.org/vs/[partcode]" must conform to this template, where [partcode] is a part code from the multi-axial hierarchy:

<ValueSet xmlns="http://hl7.org/fhir">

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml">

[some html that identifies that this value set

includes all LOINC codes subsumed by the identified

Multi-Axial Hierarchy Part Code]

</div>

</text>

<url value="http://loinc.org/vs/[partcode]"/>

<version value="[optional - but strongly recommended - LOINC version]"/>

<name value="LOINC Value Set from Multi-Axial Hierarchy code [partcode]"/>

<description value="All LOINC codes for [partcode or name]"/>

<status value="active"/>

<date value="[optional date of LOINC release]"/>

<compose>

<include>

<system value="http://loinc.org"/>

<filter>

<property value="ancestor"/>

<op value="="/>

<value value="[partcode]"/>

</filter>

</include>

</compose>

</ValueSet>

**LOINC and Data Elements**

LOINC codes and their properties are describing a data element that can have a value. Functionally, this overlaps with the [Data Element](file:///C:\temp\dataelement.html) resource, and, in fact, a DataElement can be created for each LOINC code based on their properties:

<DataElement xmlns="http://hl7.org/fhir">

<id value="[LOINC code]"/>

<url value="http://loinc.org/fhir/DataElement/[LOINC code]"/>

<version value="{LOINC version number]"/> <!-- though this is not required -->

<status value="[derived from STATUS]"/>

<experimental value="false"/>

<date value="[last changed date]"/>

<publisher value="http://loinc.org"/>

<title value="[display]"/>

<copyright value="This content LOINCÃ‚Â® is copyright Ã‚Â© 1995 Regenstrief Institute, Inc.

and the LOINC Committee, and available at no cost under the license at

http://loinc.org/terms-of-use"/>

<stringency value="comparable"/>

<element>

<!-- if type is Qn or OrdQn -->

<extension url="http://hl7.org/fhir/StructureDefinition/elementdefinition-allowedUnits">

<valueCodeableConcept>

<coding>

<system value="http://unitsofmeasure.org"/>

<!-- ? just example?- need a canonical extension? -->

<code value="[EXAMPLE\_UCUM\_UNITS]"/>

</coding>

</valueCodeableConcept>

</extension>

<path value="[LOINC code]"/>

<short value="[display]"/>

<definition value="[definition]"/>

<min value="0"/>

<max value="1"/>

<type>

<code value="[mapped type - see below]"/>

</type>

<!-- if this is a LOINC question with linked answers -->

<binding>

<strength value="required"/>

<valueSetUri value="http://loinc.org/vs/[linked answer list code]"/>

</binding>

</element>

</DataElement>

The canonical URL http://loinc.org/fhir/DataElement/[LOINC code] can be used to refer to Data Elements defined by LOINC such as e.g. in [Questionnaire.item.definition](file:///C:\temp\questionnaire.html).

The appropriate type is determined by mapping from the LOINC specified SCALE\_TYP for v3 to FHIR using this table:

|  |  |
| --- | --- |
| **SCALE\_TYP** | **FHIR data type** |
| Qn | Quantity |
| Ord | CodeableConcept |
| OrdQn | Quantity and CodeableConcept (e.g. both in the data element; either is allowed) |
| Nom | CodeableConcept (and string?) |
| Nar | markdown (?) |
| Multi | Attachment |
| Doc | Attachment |

**managing.html**

**Managing Resource Identity**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Each resource has a known identity, which is a URL. The identity is not stored inside the resource, but must be tracked by systems handling resources. For RESTful systems, the resource identity is the same as the URL by which it is found. When a resource is packaged in a [bundle](file:///C:\temp\bundle.html) (e.g. for[messages](file:///C:\temp\messaging.html) or [documents](file:///C:\temp\documents.html), or [exchanged in a service](file:///C:\temp\services.html)), the id is included along with the resource. When systems using FHIR resources are deployed in an implementation context where there is more than two systems exchanging resources, the way that resources are identified with URLs needs to be managed carefully.

Note that many resources also have an identifier element, and usually there can be multiple identifiers. This identifier is a logical identifier for the concept that the resource describes, and is different from the identity discussed in this section, which is the literal identity of the resource. Note that there are often multiple resource instances that describe the same concept across multiple systems, e.g. each application keeps its own copy of the patient information in a [Patient](file:///C:\temp\patient.html) resource. This can even exist within a single system, such as in the case of [patient duplicates](file:///C:\temp\patient.html#links)).

**Using Resources**

Resources are used in a variety of circumstances. Generally, these can be categorized into 3 different scenarios:

1. **Closed Trading System**: the resources are only ever exchanged between fixed systems in a tightly controlled community, such as a single EHR system with multiple modules. There is only one master server for each resource type, and resources are managed by that server. In this context, the [logical id](file:///C:\temp\resource.html#id) of a resource is sufficient to fully identify the resource
2. **Open system**: there are many peer servers, each managing a set of resources of different types. In order to identify resources, a full URL reference to the origin server is required. The grand example of an open system is the World Wide Web
3. **Partially closed, inter-linked systems**: a mixture of both closed and open systems - trading communities that are tightly managed, but have managed interactions with other closed trading systems, or with the world-wide web, or both. In fact, this combination appears to be the most likely scenario for current real-world healthcare business solutions

These combinations are why either relative (logical) or absolute references are allowed, and why a logical id is always required, in order to enable seamless exchange amongst partially closed trading systems.

**Copying Resources and Re-identification**

When resources are exchanged between systems, they may need to be re-identified (i.e. assigned a new logical resource id). When a resource is re-identified, nothing in the resource changes, but any references that point to the resource need to be updated. Whether re-identification is required or not depends on the context, as does how resource references are updated.

The normal case is that a client/receiving system accepts the server/sender's identification of a resource at face value, whether it is a relative or absolute reference. When the client/receiver wants to follow resource references, they are done using the server id (typically either by http calls or locating them in a [bundle](file:///C:\temp\bundle.html)), but other arrangements are possible. In such cases, there is no need for re-identification.

Another scenario is for a client to retrieve a resource from a server, and make its own local persistent copy. If the local resource has a life-cycle of its own (i.e. it is not just a cached resource), then it needs to have its own identity; i.e. the resource must be re-identified. The simplest case is that the client only is keeping local copies of resources from a single server. In these cases, the client can simply replace the Base URL and keep the logical id of the resource the same. In fact, if the server is using relative references, then this change doesn't involve any actual changes to the resources, it only means a re-interpretation of the references.

In some cases, however, the client may deal with multiple servers. In this case, the logical id of the resource is not guaranteed to be unique (unless all resources have a UUID for the logical id, which is allowed but not required). When the client cannot be sure that the resource identities are unique, it will have to re-identify the resources. In practice this means that the client needs to keep some kind of identity translation table, and update references to the resources it has copied locally when other resources are received.

The case of a gateway system that migrates resources from one ecosystem to another is very similar. In some limited cases, it can leave the logical id of the resources unchanged as resources are copied from one closed system to another. However in more complicated cases, it will have to modify the resource references as resources pass across the gateway.

**STU Note:** In its current form, FHIR allows various convenient implementer practices around resource identification that make integration across eco-system boundaries more difficult. In particular, either only allowing UUIDs for logical IDs or only allowing absolute references would make the boundary management problem easier but remove useful and convenient flexibility for other uses of FHIR.

Feedback is welcome [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**mapping-language.html**

**FHIR Mapping Language**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 0 | [Ballot Status](file:///C:\temp\help.html#status): [n/a yet](file:///C:\temp\history.html#pubs) |

The FHIR Specification includes a mapping language. The mapping language has a concrete syntax, defined and described in this page, and an abstract syntax, which is found in the [StructureMap](file:///C:\temp\structuremap.html) resource (and there is an [antlr grammar for the concrete syntax](file:///C:\temp\mapping.g4)). See also the [Tutorial](file:///C:\temp\mapping-tutorial.html).

The mapping language describes how one set of Directed Acyclic Graphs (an instance) is transformed to another set of directed ayclic graphs. It is not necessary for the instances to have formal declarations and/or be strongly typed - just that they have named children that themselves have properties. On the other hand, when the instances are strongly typed - specifically, when they have formal definitions that are represented as [Structure Definitions](file:///C:\temp\structuredefinition.html), the mapping language can use additional type related features.

The mapping language addresses 2 very different kinds of transformations:

* Structural changes between the source and target structures
* Differences in content and formats in string (and related) primitives contained within the structures

A map has 6 parts:

* Metadata
* Embedded [ConceptMaps](file:///C:\temp\conceptmap.html) to translate between different code systems
* References to the structures involved in the mapping
* Imports: additional Maps used by this map
* a series of groups, with a list of input variables
* A series of transformation rules in each group

**Executing Maps**

Maps are executed by a mapping engine. This takes one or more inputs of instances (directed acyclic graphs) and a map, and produces a set of outputs as specified by the map. The exact details of the form that the instances take are a matter for the map engine / application API. This language assumes that the engine can query an element in the instance for it's children, it's primitive value, and (optionally) it's type. The language also assumes that the engine has application support for the following operations:

* ValueSet validation operation
* Translation operation
* Lookup another tree of data
* Create an instance tree
* Return the correct string format to refer to a tree (input or output)

These functions consistitute a Mapping Support API that makes maps portable between different systems

Generally, it is assumed the invocation of the engine follows some pattern like this:

* The host application creates the engine, and passes it a handle to the standard services
* The host application chooses the model map resource, and asks the engine to prepare it (load, check, cache-up etc)
* The host application asks the engine to execute and provides a set of resources that match defined inputs in the map
* Any created output will be created via the standard API

Some host applications may be able to determine how to combine maps and inputs on the fly based on their metadata, and minimal configuration, while others may require manual arrangements in order to manage the map execution process.

**General Syntax Notes**

Mapping files are always plain text in unicode. Whitespace is any unicode whitespace, and the particular whitespace used is not signicant, except that unicode end of line characters terminate a comment. Comments are started by the characters "//".

The abstract model includes documentation for each item. The canonical text representation is for each item to be on it's own line, with documentation at the end of the line as a comment.

All names defined by the map language - group, rule and variable names - must be valid [ids](file:///C:\temp\datatypes.html#id) (1-64 characters, upper and lowercase letters, numbers, dashes, dots and underscores), and must start with a letter. The special boolean values 'true' and 'false' are not allowed as variable names.

**Metadata**

The first part of the mapping syntax establishes the name of the mapping:

map "[url]" = "[name]"

The letters "map" are the first non-whitespace non-comment characters in the source. This is followed by the canonical URL that identifies the map uniquely, and then a human readable name for the map.

*todo: add additional metadata?*

**Structure Definition References**

The next section of the map references the set of structure definitions that are used or produced by this map.

uses "[url]" (alias name) as [mode] // documentation

This optional section lists one or more structure definitions that the map makes use of, and indicates for each structure definition, how it is used. It may also provide an Alias - a name used for the type inside the mapping language - this may be necessary when transforming from source to target where both source and target use overlapping type names (not unusual).

Any kind of structure definition may be referenced, including data types, resources, constraints on those, and logical models.

There are 4 modes in which a structure definition may be used:

* **source**: One of more instances of this type are passed to the mapping engine when the mapping is executed, and serve as the source from which mapping is performed
* **queried**: The map may ask the (via the API) for some instances of this type. For further discussion, see [below](file:///C:\temp\main-pages.html#api)
* **target**: One or more instances of this type are passed in, and will be populated from the source material
* **produced**: The map may ask (via the API) for some instances of this type to be created. For further discussion, see [below](file:///C:\temp\main-pages.html#api)

The simplest case, which is common, is where a single structure is converted to another single structure. in this case, the map specifes one target, and one source. Such maps are easy to use automatically - the host application has content in one format, creates an empty instance of the target, and asks the mapping engine to convert.

However, many mappings are not so simple. For instance, converting from a single CDA document to FHIR typically creates a set of resources. In this case, there is a single target - a [Bundle](file:///C:\temp\bundle.html), but it is also useful to specify a set of other structure definitions that may be created as part of the bundle. Alternatively converting from one source model to another might involve looking up other information in other instances of data.

It's also possible for a map not so specify any structure definition dependencies. A map that doesn't indicate any structure definitions can still be used, but the type features of the map language can't be used, and such maps typically require special development to integrate the execution of the map into an application.

**Map Imports**

This section references additional maps that are used by this map:

imports "[url]" // documentation

Typically, maps that are imported a type based, such as a CDA --> FHIR map that makes use of a CD --> CodeableConcept map. How imported maps are actually used is discussed below.

The [url] may contain a "\*" as a wildcard character to include any matching maps that are available to the mapping engine.

**Groups**

Each Mapping source contains one or more groups of rules. Each group defines a set of related mapping rules that take the same input and output variables, that define exactly which instances are passed to the mapping, and provides names by which they may be passed when invoking the map:

group (for type) [group-name] (extends [other-group])

input [name] : [type] as [mode] // documentation

Each group has a name, which is how the mapping is invoked. The first group is special, in that this is the group invoked if there no name is provided (e.g. starting the mapping by a host application).

Each input to the group has a name. This is the name that applications use when passing the instance to the invocation engine, or that rules use when invoking the group. Inputs may have a type - and should (see the discussion above), but are not required to. Input variables also have a mode, which may be one of source or target (see above). There must be at least two input variable (source and target) - else there's nothing to map. Maps may have additional input or output inputs, where that's necessary.

Groups may extend other groups, which means that the rules in the other group also apply (typically, this is used with specialising classes in an OO context). When a group extends another group, it SHALL have the same input parameters (by name, type and mode) though their order may differ, and it may have additional parameters.

The key word for is used to indicate that this group provides a set of mappings that are intended to be used as the default way to map from source from source to target. There are 2 variants:

* for types: Use this by default when a [source] of the specified type must be converted to a [target] of the specified type, and no specific dependent rules (see below) are specified.
* for type+types: in addition, to the above use, when a [source] of the specified type must be converted to a [target] and the type of the target is not fixed, use this group

In both these cases, the group SHALL have 2 parameters, a source, and a target, in that order, and both SHALL have specified types for the inputs.

**Transform Rules**

The main portion of a map consists of a set of transform rules that describe how source content is transformed into target content. The full format for a rule looks like this:

name\_of\_rule: for src\_context.field as new\_variable where condition make tgt\_context.field as new\_variable = create([type]) then [details].

Each rule has 4 main sections:

* **Name**: The identity of the rule, for logging/debugging
* **Source Content**: One or more elements from the source that contribute to the mapping
* **Target Transform**: One or more specifications of content to create in the target model
* **Dependent Rules**: Specifies which - if any - rules or groups to apply within the scope of the rule

Rules may be applied in any order; there is no sense of sequentially applying one rule after another.

**Name**

Each rule is assigned a name. The name is used when specifying rule links, and in traces (a record generated by the conversion engine recording the transform process). Names must be unique within the context of the map.

**Source Content**

Each rule specifies one or more elements taken from the source that define variables that can be used when specifying target content, or re-used in subsequent transform rules. Multiple source elements are separated by a comma, like this:

rule\_name: for [source], [source], make ...

Each [source] contains the following items:

for context.element { : type {min..max}} {default [value]} { list-option } as variable where [FHIRPath] check [FHIRPath]

* **context**: A variable that is the context in which this rule applies. Variables are not prefixed, and must be declared elsewhere in the map ('as variable' below, or in the input source variables)
* **element**: An optional name of a child element of the context. If this is not provided, the source is the context. If this is provided, the rule will apply once for each element on the context that matches this name. If the element name contains spaces (possible in some contexts), it can be quoted using "
* **type**: A type may be specified. If a type is specified, only elements that have the specified type are selected as part of the source
* **min..max**: Specified cardinality. If a cardinality is specified, and the input document does not conform to the cardinality, the mapping engine raises an exception instead of completing the transformation
* **default**: A default value, used if there is no source value found. If there's a default value on an item that can repeat, it will only be used once
* **list-option**: by default, the rule will apply once for each occurance of the element in context. The list option can override this to specify only to apply the first, last, or not to those
* **as variable**: if an element is specified, a variable must be assigned. This variable name may be used in the target statement, and may be re-used in other dependent rules (see below). There is one special case described below (simple form) where the variable name is provided explicitly
* **where [condition]**: a fluent path (ref) expression that is evaluated to boolean on the context. If the expression returns false, the source element has no match
* **check [condition]**: a fluent path (ref) expression that is evaluated to boolean on the context. If the expression returns false, the mapping engine terminates execution with an error. Note: it's usual to have either a where or a check clause, but possible to have both. If both are provided, the where clause is evaluated first, and the check only applies if the where expression is true

If all the source elements have a match the rule applies for the permutation of the source elements (e.g. if there are 2 elements, each with 2 matches, the rule applies 4 times, one for each combination). Typically, if there is more than one source element, only one of the elements can repeat.

Once the source statement is evaluated, the engine performing the evaluation has a list of variables, each of which contains a single value for each named variable. These variables are now mapped into the target structures in the target transformation.

**Target Transform**

Each rule specifies zero or more elements to be created in the target structure. These targets can also be assigned to variables that can be used in subsequent transform rules. If no targets are specified, there are not created targets, just newly defined source variables. Multiple target elements are separated by a comma, like this:

... make [target], [target] then by...

Each [target] contains the following items:

make context.element = transform\_code(parameters...) as variable {list\_modes}

* **context**: The name of a variable that is the context in which the content will be created. Variables must be declared elsewhere in the map ('as variable' below, or in the input target variables)
* **element**: The name of a child element that is valid in the context. The created value will be placed into the named element
* **transform+parameters**: details how the content that is created is transformed from the source data. See below for possible transform codes. If no transform code is provided, then the element is autocreated. It is an error if auto-created elements are primitive types, or have more than one possible type
* **as variable**: a variable name may be assigned, which allows the created item to be re-used in the dependent rules
* **list\_modes**: control how elements that repeat are managed when the transform rule is evaluated

Context and Element are optional as a pair. If no context/element is specified, then a variable must be defined, and the created value is only available in the variable.

Each time the rule is applied, the engine determines the value from the transforms, considers the list mode, if required, and creates that specified content in the target instance. Within a given transform url, the targets are processed in order, so that a transform rule may refer to a variable defined by a prior transform rule.

The following list specifies that transforms that can be specified. Each transform takes one or more parameters:

|  |  |  |
| --- | --- | --- |
| **Name** | **parameters** | **Documentation** |
| copy | source | simply copy the source to the target as is (only allowed when the types in source and target match- typically for primitive types). In the concrete syntax, this is simply represented as the source variable |
| create | type | use the standard API to create a new instance of data. Where structure definitions have been provided, the type parameter must be a string which is a known type of a root element. Where they haven't, the application must know the name somehow |
| truncate | source, length | source must be some stringy type that has some meaningful length property |
| escape | source, format1, format2 | Change the internal escaping of a string element. Note: this is not often needed, as mostly the escaping is done on the base format |
| cast | source, type? | cast source from one type to another. target type can be left as implicit if there is one and only one target type known |
| append | source... | source is element or string - just append them all together |
| translate | source, map\_uri, output | use the [translate operation](file:///C:\temp\conceptmap-operations.html#translate). The source is some type of code or coded datatype, and the source and map\_uri are passed to the translate operation. The output determines what value from the translate operation is used for the result of the operation (code, system, display, Coding, or CodeableConcept) |
| reference | source | return a string that references the provided tree properly |
| dateOp | ?? | Perform a date operation. Parameters to be documented |
| uuid | n/a | Generate a random UUID (in lowercase). No Parameters |
| pointer | resource | Return the appropriate string to put in a Reference that refers to the resource provided as a parameter |
| evaluate | resource | Execute the supplied FHIRPath expression and use the value returned by that. The 2nd parameter - FHIRPath expression - is evaluated in the context of the first parameter, and the result used as the value. In the concrete syntax, there is a short hand for this operation, by supplying () around the parameter. In this case, there is no context for the FHIRPath expression, and it must start with a reference to one of the defined variables |
| cc | (text) or (system. Code[, display]) | Create a CodeableConcept from the parameters provided |
| c | system. Code[, display] | Create a Coding from the parameters provided |
| qty | (text) or (value, unit, [system, code]) | Create a quantity. Parameters = (text) or (value, unit, [system, code]) where text =s the natural represenation e.g. [comparator]value[space]unit |
| id | system, value[, type] | Create an identifier. where type is a code from the identifier type value set |
| cp | (value) or (system, value) | Create a contact details. If no system is provided, the system should be inferred from the content of the value |

TODO: explain how optional parameters work with transforms (append only?), document list mode

**Dependent Rules**

Once the source elements are evaluated, and any specifed targets created, the engine has a set of variables that represent source and target contexts in which further mapping may occur. The set of variables includes those provided to the group that contains the rule, and those created by the application of the rule. For some created elements that are primitive types, that's the end of the road - there's nothing more to do with them. But if either or both the source and target types are complex, there are usually additional mapping rules that need to apply to the newly created variables.

Transform rules specify what additional rules are evaluated when the rule is complete, by containing other rules:,

.. then {

.. other rules...

}

When a rule contains other rules, the variables from the containing rules are all available to the contained rules. Alternatively, a rule can nominating another group of rules from the same or an imported mapping. Each rule or group is listed by name, and then a set of parameters are provided.

.. then rule(param, param)

The parameters provided must match the parameters required by the dependenct rule, in order. In addition, the mode of the variable must match - inputs that are targets must be target variables. Note, though, that target variables can be treated as source for a group.

Groups are resolved by name by looking through all the groups in all the available maps referened by the uses (see above) statements. The name must be unique within the scope of these maps.

**Simple Form + Implicit Type Dependencies**

If no dependent rules are specified, and if the is only one source and target, and they both speecify a variable, the rule can be written in an abbreviated form:

"name" : for src.element make tgt.element

This is implicitly the same as

"name" : for src.element as vvs make create([type]) tgt.element as vvt then [typeGroup](vvs, vvt)

Where type and typeGroup are determined by the context of src.element and tgt.element:

* typeGroup is determined by looking through the available rule groups labelled as "for types" or "for type+types" and checking the types of their inputs looking for a match to the type of src.element, and [type]
* type is determined by the type of tgt.element. If tgt.element can have more than one type, then the type is inferred by looking through the available groups labelled "for type+types" for a match to the type of src.element

**Formal Grammar**

todo

**Mapping Support API**

todo

**mapping-tutorial.html**

**FHIR Mapping Language - Tutorial**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 0 | [Ballot Status](file:///C:\temp\help.html#status): [n/a yet](file:///C:\temp\history.html#pubs) |

This tutorial introduces the FHIR [mapping language](file:///C:\temp\mapping-language.html).

**Step #1: Simplest possible transform**

To start with, we're going to consider a very simple case: mapping between two structures that have the same definition, a single element with the same name and the same primitive type:

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  a : string [0..1] | TRight  a : string [0..1] |
| The left instance is transformed to the right instance by copying a to a | |

*Note that for clarity in this tutorial, all the types are prefixed with T.*

The first task to do is to set up the mapping context on a default group. All mappings are divided up into a set of groups. For now, we just set up a group named "tutorial" - the same as the name of the mapping. For this tutorial, we also declaring the source and target models, and specify that an application invokes this with a copy of the left (source) instance, and also an empty copy of the right (target) instance:

map "http://hl7.org/fhir/StructureMap/tutorial" = tutorial

uses "http://hl7.org/fhir/StructureDefinition/tutorial-left" as source

uses "http://hl7.org/fhir/StructureDefinition/tutorial-right" as target

group tutorial

input "source" : TLeft as source

input "target" : TRight as target

// rules go here

endgroup

Note that the way the input variables are set up is a choice: we choose to provide the underlying type definitions on which both source and target models are based, and we choose to specify that the invoking application most provide both the source and the target instance trees. Other options are possible; these are discussed further below. The rest of the tutorial examples use the same setup for the group.

Having set up the context, we now need to define the relationships between the source and target structures:

"rule\_a" : for source.a as a make target.a = a

This simple statement says that:

* for every source (there'll only be one)
* for any element 'a' in the source
* if there isn't any element 'a', then don't do anything
* if there is one, call it variable 'a'
* the value of property 'a' of the target will be a copy of variable a - that is, source.a

"rule\_a" is a purely arbitrary name associated with the rule that appears in logs, error messages, trace files, etc. It has no other meaning in the mapping statements.

Note that there is no types explicitly in this mapping statement, but if the underlying system has types, then the types will have to be correct. If the underlying source and target trees are strongly typed, and the mapping groups have explicit types, then a short hand form is possible:

"rule\_a" : for source.a make target.a

How this works is described [described below](file:///C:\temp\main-pages.html#step10).

**Step #2: Fields with different names**

Now consider the case where the elements have different names:

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  a1 : string [0..1] | TRight  a2 : string [0..1] |
| The left instance is transformed to the right instance by copying a1 to a2 | |

This relationship is a simple variation of the last:

"rule\_a1" : for source.a1 as b make target.a2 = b

Note that the choice of variable name is purely arbitrary. It does not need to be the same as the element name.

**Step #3: Length restriction**

Still sticking with very simple mappings, let's consider the case where there is a length restriction on the target model that is shorter than the one on the source model - in this case, 20 characters.

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  a2 : string [0..1] | TRight  a2 : string [0..1] {maxlength = 20} |
| The left instance is transformed to the right instance by copying a2 to a2, but target.a2 can only be 20 characters long | |

There are 3 different ways to express this mapping, depending on what should happen when the length of source.a is > 20 characters:

"rule\_a20a" : for source.a2 as a make target.a2 = truncate(a, 20) // just cut it off at 20 characters

"rule\_a20b" : for source.a2 as a where a2.length <= 20 make target.a2 = a // ignore it

"rule\_a20c" : for source.a2 as a check a2.length <= 20 make target.a2 = a // error if it's longer than 20 characters

Note that it is implicit here that the transformation engine is not required to expected to validate the output against that underlying structure definitions that may apply to it. An application may - and usually should - validate the outputs after the transforms, but the transform engine itself does not automatically validate the output (e.g. it does not assume that it's the final step in the process).

**Step #4: Type Conversion in Primitives**

Now for the case where there is a simple type conversion between the primitive types on the left and right, in this case from a string to an integer.

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  a21 : string [0..1] | TRight  a21 : integer [0..1] |
| The left instance is transformed to the right instance by copying a21 to a21, but a21 is converted to an integer | |

There are 3 different ways to express this mapping, depending on what should happen when a is not an integer:

"rule\_a21a" : for source.a21 as a make target.a21 = cast(a, "integer") // error if it's not an integer

"rule\_a21b" : for source.a21 as a where a.isInteger make target.a21 = cast(a, "integer") // ignore it

"rule\_a21c" : for source.a21 as a where not at1.isInteger make target.a21 = 0 // just assign it 0

More than one of these mapping rules may be present to handle all possible cases - .e.g rule\_a21b combined with rule\_a21c.

Note that the mapping language does not itself define which primitive types exist. Typically, primitive types are defined by the underlying type system for the source and target trees, and the implementation layer makes these types available to the mapping language using the FHIRPath primitive types. The mapping language uses the FHIRPath syntax for primitive constants.

**Step #5: Managing lists, part 1**

Back to the simple case where source.a22 is copied to target.a22, but in this case, a22 can repeat (in both source and target):

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  a22 : string [0..\*] | TRight  a22 : string [0..\*] |
| The left instance is transformed to the right instance by copying a22 to a22, once for each copy of a22 | |

The transform rule simply asserts that a22 maps to a22. The engine will apply the rule once for each instance of a22:

"rule\_a22" : for source.a22 as a make target.a22 = a

This will create one a22 in TRight for each a22 in TLeft.

**Step #6: Managing lists, part 2**

A more difficult case is where the source allows multiple repeats, but the target doesn't:

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  a23 : string [0..\*] | TRight  a23 : integer [0..1] |
| The left instance is transformed to the right instance by copying a23 to a23, but there can only be one copy of a23 | |

Again, there are multiple different ways to write this, depending on out desired outcome if there is more than one copy of a23:

rule\_a23a : for source.a23 as a make target.a23 = a // leave it to the transform engine

rule\_a23a : for source.a23 only\_one as a make target.a23 = a // transform engine throws an error if there is more than one

rule\_a23b : for source.a23 first as a make target.a23 = a // Only use the first one

rule\_a23b : for source.a23 last as a make target.a23 = a // Only use the last one

Leaving the outcome to the transform engine is not recommended; it may not always know whether a property is confined to a single value, and exactly what happens is unpredictable. However there are some circumstances where the appropriate action is to defer resolution, so this is allowed.

**Step #7: Simple Nesting**

Most transformations involve nested content. Let's start with a simple case, where element aa contains ab:

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  aa : [0..\*]  ab : string [1..1] | TRight  aa : [0..\*]  ab : string [1..1] |
| The left instance is transformed to the right instance by copying aa to aa, and within aa, ab to ab | |

Note that there is no specified type for the element aa. Some structure definitions (FHIR resources) do leave these elements as anonymously typed, while others explicitly type them. However since the mapping does not refer to the type, it's literal type is not important.

rule\_aa : for source.aa as s\_aa make target.aa as t\_aa then { // make aa exist

rule\_ab : for s\_aa.ab as ab make t\_aa.ab = ab // copy ab inside aa

}

This situation is handled by a pair of rules: the first rule establishes that relationship between source.aa and target.aa, and assigns 2 variable names to them. Then, the rule contains an additional set of rules (though only one in this example) to map with the context of s\_aa and t\_aa.

An alternate approach is to move the dependent rules to their own group:

rule\_aa : for source.aa as s\_aa make target.aa as t\_aa then ab\_content{s\_aa, t\_aa) // make aa exist

group ab\_content

input src as source

input tgt as target

rule\_ab : for src.ab as ab make tgt.ab = ab // copy ab inside aa

}

Note that variables are divided into source and target; source variables are read-only, and cannot have their properties changed. Variable names may be reused in different contexts - they are only valid within the group or rule that defines them, and any dependent rules or groups.

**Step #8: Translation**

A common translation pattern is to perform a translation e.g. from one set of codes to another

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  d : code [0..1] | TRight  d : code [0..1] |
| The left instance is transformed to the right instance by translating source.d from one set of codes to another | |

The key to this transformation is the [ConceptMap](file:///C:\temp\conceptmap.html) resource, which actually specifies the mapping from one set of codes to the other:

rule\_d : for source.d as d make target.d = translate(d, 'uri-of-concept-map', 'code')

This asks the mapping engine to use the [$translate](file:///C:\temp\conceptmap-operations.html#translate) operation on the terminology server to translate the code using a specified concept map, and then to put the code value of the return translation in target.d.

**Step #9: Co-dependency in translation**

Another common translation is where the target mapping for one element depends on the value of another element.

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  i : string [0..1]  m : integer [1..1] | TRight  j : [0..1]  k : [0..1] |
| How the left instance is transformed to the right instance depends on the value of m: if m < 2, then i maps to j, else it maps to k | |

This is managed using [FHIRPath](http://hl7.org/fhirpath) conditions on the mapping statements:

rule\_i1 : for source.i as i where m < 2 make target.j = i

rule\_i2 : for source.i as i where m >= 2 make target.k = i

**Step #10: Using Types**

Many/most trees are fully and strongly typed. In these cases, the mapping language can make use of the typing system to simply the mapping statements.

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  aa : TLeftInner [0..\*]    TLeftInner  ab : string [1..1] | TRight  aa : : TRightInner [0..\*]  TRightInner  ab : string [1..1] |
| The left instance is transformed to the right instance by copying aa to aa, and within aa, ab to ab | |

This is the same case as [Step 7 above](file:///C:\temp\main-pages.html#step7), but the mapping statements take advantage of the types:

rule\_aa : for source.aa make target.aa

group for types ab\_content

input src : TLeftInner as source

input tgt : TRightInner as target

rule\_ab : for src.ab make tgt.ab

}

There is 2 different things happening in this short form:

1. group for types - the "for types" indicates that this group is the default group to apply any time an element of type TLeftInner is mapped to a TRightInner
2. Both the rules take advantage of the fact that the types of both source and target are known, and compatible, and instruct the mapping execution engine to make the target appropriately

In the case of the first rule (rule\_aa), the engine finds a need to map aa to aa, and determines that it must map from TLeftInner to a TRightInner. Since a group is defined for this purpose, it creates a TRightInner in target.aa, and then applies the discovered rule as a dependency rule. Inside that rule that instructs the mapping engine to make tgt.ab from src.ab. It knows that both are primitive types, and compatible, and can apply this correctly. This short form is only applicable when there is only one source and target, when the types of both are known, and when no other dependency rules are nominated.

If the target element is polymorphic (can have more than one type), then the correct type of the target can only be inferred from the source type:

group for type+types ab\_content

input src : TLeftInner as source

input tgt : TRightInner as target

rule\_ab : for src.ab make tgt.ab

}

Not only is this group te default for (TLeftInner:TRightInner), if the engine has a TLeftInner with an unknown target type, it should create a TRightInner, and proceed as above.

It is an error if the engine locates more than one group of rules claiming to be the correct group for a type pair of a single source type.

**Step #11: Reworking Structure #1**

It's now time to start maving away from relatively simple cases to some of the harder ones to manage mappings for. The first mixes list management, and converting from a specific structure to a general structure:

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  e : string [0..\*]  f : string [1..1] | TRight  e : [0..\*]  f : string [1..1]  g : code [1..1] |
| The left instance is transformed to the right instance by adding one instance of target.e for each source.e, where the value goes into target.e.f, and the value of target.e.g is 'g1'. source.f is also transformed into the same structure, but the value of target.e.g is 'g2'. As an added complication, the value for source.f must come first | |

This leads to some more complex mapping statements:

ef\_a1: for source.e as s\_e make target.e as t\_e then {

ef\_a2: for s\_e make t\_e.f = s\_e, t\_e.g = "g1"

}

ef\_b1: for source.f as s\_f make target.e as t\_e { first } then {

ef\_b2: for s\_f make t\_e.f = s\_f, t\_e.g = "g2"

}

**Step #12: Reworking Structure #2**

The second example for reworking structure moves cardinality around the heirarchy. in this case, the source has an optional structure that contains a repeating structure, while the target puts the cardinality at the next level up:

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  az1 :[0..1]  az2 : string [1..1]  az3 : string [0..\*] | TRight  az1 :[0..\*]  az2 : string [1..1]  az3 : string [0..1] |
| The left instance is transformed to the right instance creating on target.az1 for every source.az1.az3, and then populating each az1 with the matching value of az3, and copying the value of az2 to each instance | |

The key to setting this mapping up is to create a variable context for source.az1, and then carry it down, performing the actual mappings at the next level down:

// setting up a variable for the parent

aza : for src.az1 as s\_az1 then {

// one target.az1 for each az3

azb : for s\_az1.az3 as s\_az3 make target.az1 as t\_az1 then {

// value for az2. Note that this refers to a previous context in the source

az2 : for s\_az1.az2 as az2 make t\_az1.az2 = az2

// value for az3

az3 : for s\_az3 make tgt\_az1.az3 = src\_az3

}

}

**Step #13: Multiple structures #1**

Simple mappings, such as we've dealt with so far, where the source and target structure both have the same scope, and there is only one of each, are all well and good, but there are many mappings where this is not the case. There is a set of complications when dealing with multiple instances:

* If there are multiple source inputs, how does the application know what they are? Sometimes, they are just independent inputs, but more often, the inputs are dependent on references in the source input, and therefore which source inputs are required depends on the mapping rules
* If there are multiple output instances, how are they identified as they are created, and how do the target models reference each other? Mostly, the answer is that it depends on the context; the actual identification details are not part of the mapping
* It may even be the case that the kind of output structure to produce depends on the mapping rules, so the application can't create the target structure before invoking the map

For our first example, we're going to look at creating multiple output structures from a single input structure.

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  f1 : String [0..\*]; | TRight  ptr : Resource(TRight2) [0..\*]    TRight2  f1 : String [1..1]; |
| The left instance is transformed to the right instance creating a copy of TRight2 for each f1 in the source, and then putting the value of source.f1 in TRight2.f1 | |

The key to setting this mapping up is to create a variable context for source.az1, and then carry it down, performing the actual mappings at the next level down:

f1 : for source.f1 as s\_f1 make create("TRight2") as rr, target.ptr = reference(rr) then {

f1a: for s\_f1 make rr.f2 = srcff

}

This mapping statement makes use a special known value "null" for the target context to indicate that the created element/object of type "TRight2" doesn't get added to any existing target context. Instead, it will only be available as a context in which to perform further mappings - as rule f1a does.

The mapping engine passes the create request through to the host application, which is using the mapping. It must create a valid instance of TRight, and identify it as appropriate for the technical context in which the mapping is being used. The reference transform is also passed back to the host application for it to determine how to represent the reference - but this is usally some kind of URL.

**Step #14: Multiple structures #2**

For our second example, we're going to look at the reverse: where multiple input structures create a single input structure.

|  |  |
| --- | --- |
| **Source Structure** | **Target Structure** |
| TLeft  ptr : Resource(TLeft2) [0..\*]    TLeft2  f2 : String [0..\*]; | TRight  f2 : String [1..\*]; |
| The left instance is transformed to the right instance finding each ptr reference, getting it's value foe f1, and adding this in target.f2 | |

The first task of the map is to ask the application host to find the structure identified by source.ptr, and create a variable for it

f2: [todo]

**Step #15: Grouping**

f2: [todo]

**medications-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Medications Module**

**Introduction**

This module is concerned with resources and functionality in 2 main domains:

* The ordering, dispensing, administration of medications and recording statements of medication use.
* Recording of Immunizations given (or not given), and recommendations for an individual patient at a point in time.

**Index**

|  |  |  |
| --- | --- | --- |
| * [MedicationRequest](file:///C:\temp\medicationrequest.html) * [MedicationDispense](file:///C:\temp\medicationdispense.html) * [MedicationAdministration](file:///C:\temp\medicationadministration.html) | * [MedicationStatement](file:///C:\temp\medicationstatement.html) * [Medication](file:///C:\temp\medication.html) | * [Immunization](file:///C:\temp\immunization.html) * [ImmunizationRecommendation](file:///C:\temp\immunizationrecommendation.html) |

**Medications**

|  |  |
| --- | --- |
| **Name** | **Description** |
| [MedicationRequest](file:///C:\temp\medicationrequest.html) | Represents an instruction for the administration of medication to a patient - both in the inpatient (hospital) and community setting. It can also include instructions for the dispensing, the reasons why the administration should occur and other data.  It is called an 'Request' to be consistent with other FHIR resources and the workflow pattern, but a common alias for this resource is a 'Prescription' or an 'Order'. The Order itself represents the content of the instruction and is not, by itself, actionable. The workflow process around 'fulfilling' the order is part of the [generic FHIR workflow](file:///C:\temp\workflow-module.html) (see below), with the MedicationRequest representing the contents. |
| [MedicationDispense](file:///C:\temp\medicationdispense.html) | The provision of a supply of a medication with the intention that it is subsequently consumed by a patient (usually in response to a prescription). |
| [MedicationAdministration](file:///C:\temp\medicationadministration.html) | A record of a patient actually consuming a medicine, or if it has otherwise been administered to them |
| [MedicationStatement](file:///C:\temp\medicationstatement.html) | This is a record indicating that a patient may be taking a medication now, has taken the medication in the past, or will be taking the medication in the future. The source for this information can be the patient, significant other (such as a family member or spouse), or a clinician. A common scenario where this information is captured is during the history taking process during a patient visit or stay. A medication statement is not a part of the prescribe->dispense->administer sequence, but is a report that such a sequence (or at least a part of it) did take place, resulting in a belief that the patient has received a particular medication. It may be used to construct a patients 'Current Medications' list. |
| [Medication](file:///C:\temp\medication.html) | The medication resource represents an actual medication that can be given to a patient, and referenced by the other medication resources. In many cases, this resource is not needed and the drug is indicated by a reference to the appropriate terminology and so can be represented using a codeable concept. In other cases however, it may be desired to indicate more details than the simple drug (such as the packaging, whether it is a generic medication or the active and inactive ingredients) and so the Medication resource can be used for this. |

**Immunizations**

|  |  |
| --- | --- |
| **Name** | **Description** |
| [Immunization](file:///C:\temp\immunization.html) | The Immunization resource is intended to cover the recording of current and historical administration of vaccines to patients across all healthcare disciplines in all care settings and all regions. This includes immunization of both humans and animals, but does not include the administration of non-vaccine agents, even those that may have or claim to have immunological effects. |
| [ImmunizationRecommendation](file:///C:\temp\immunizationrecommendation.html) | A patient's point-in-time immunization and recommendation (i.e. forecasting a patient's immunization eligibility according to a published schedule) with optional supporting justification |

**Common Use Cases**

* Placing a Medication Request (aka Prescription or Order). The MedicationRequest resource represents the details of what medication the prescriber intends the patient to receive and other details such as the dose, timing and route. However, while it represents that order, the actual workflow around the supply (dispensing) and administration of that medication is managed by the common [workflow functionality](file:///C:\temp\workflow-module.html), resulting in the creation of the other medication resources (MedicationDispense, MedicationAdministration) as it executes.
* Listing a patients current medications. This is represented using the [List](file:///C:\temp\list.html) resource, which allows an author to construct a 'curated' list. This has features beyond a simple collection of resources - such as the ability to state that the patient is not currently taking any medications at the time that the list is constructed, or changes that are made at a point in time (sometimes called medication reconciliation). FHIR defines a number of specific [functional lists](file:///C:\temp\lifecycle.html#lists) for this purpose, and a specific [operation](file:///C:\temp\list-operations.html#find) to retrieve it. Note that FHIR does not describe how the list should be maintained by the individual system, as that will vary widely between implementations.
* Retrieving a list of Immunizations. This can be achieved using a [simple query](file:///C:\temp\http.html#search) against the [Immunization](file:///C:\temp\immunization.html) resource type. Note that an Immunization resource can specifically state that it was not given, as well as any reactions that occurred. Note that if the reaction is considered to be an indication of an allergy or intolerance, then a separate [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) resource should be created.
* Retrieving recommendations for a specific patient at a point in time. The recommendation for immunizations to give would be represented by an ImmunizationRecommendation resource, but the determination of that recommendation is a part of [Decision Support](file:///C:\temp\clinicalreasoning-module.html), with the actual immunization schedule being represented by a generic [Plan Definition](file:///C:\temp\plandefinition.html), individual [Care Plan](file:///C:\temp\careplan.html) or some other record.
* Dispensing a medication based on a Medication Request (aka Prescription or Order). The MedicationDispense resource represents the details of the dispensing event including the actual product provided to the patient and the dosage to be administered or consumed.
* Retrieving the planned medication therapy for specific patient at a poin in time. The intended medication therapy may be represented by an individual [Care Plan](file:///C:\temp\careplan.html).

**Other resources**

There are other resources that are of particular interest in the medication domain.

* [SupplyRequest](file:///C:\temp\supplyrequest.html) and [SupplyDelivery](file:///C:\temp\supplydelivery.html) which, like MedicationRequest, are 'detail' resources used as part of [workflow](file:///C:\temp\workflow-module.html)workflow. They are concerned with the request of supplies used in the healthcare process. This includes supplies specifically used in the treatment of patients as well as supply movement within an institution (transport a set of supplies from materials management to a service unit (nurse station).

**Security and Privacy**

As with all clinical data, Medications (in particular) can be sensitive information as specific medications can indicate the presence of private information such as mental health disorders or HIV. However, withholding information about what medications a person is taking can lead to catastrophic results, and so needs to be considered very carefully. At the least, a clinician should be made aware that there is information available that they have not been given when making clinical decisions.

For more general considerations, see [the Security and Privacy module](file:///C:\temp\secpriv-module.html).

**Developmental Roadmap**

The Pharmacy workgroup has plans to improve all existing resources e.g. adding in features that support detailing our conditional orders in a structured way; evaluating requirements for supporting drug formularies.

Drug formulary work may involve updates to the Medication Resource.

**messaging.html**

**Messaging using FHIR Resources**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

FHIR Resources can be used in a traditional messaging context, much like [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) (see [detailed comparison](file:///C:\temp\comparison-v2.html)). Applications asserting conformance to this framework claim to be conformant to "FHIR messaging" (see [Conformance](file:///C:\temp\conformance-rules.html)).

In FHIR messaging, a "request message" is sent from a source application to a destination application when an event happens. Events mostly correspond to things that happen in the real world. The request message consists of a [Bundle](file:///C:\temp\bundle.html) identified by the [type](file:///C:\temp\bundle-definitions.html#Bundle.type) "message", with the first resource in the bundle being a [MessageHeader](file:///C:\temp\messageheader.html) resource. The MessageHeader resource has a code - the message event - that identifies the nature of the request message, and it also carries additional request metadata. The other resources in the bundle depend on the type of the request.

The events supported in FHIR, along with the resources that are included in them, are defined below.

The destination application processes the request and returns one or more response messages which are also a [bundle](file:///C:\temp\bundle.html) of resources identified by the [type](file:///C:\temp\bundle-definitions.html#Bundle.type) "message", with the first resource in each bundle being a [MessageHeader](file:///C:\temp\messageheader.html) resource with a response section that reports the outcome of processing the message and any additional response resources required.

* Example Request Message:
* [Example Response Message:](file:///C:\temp\message-request-link.xml.html)

**Basic Messaging Assumptions**

This specification assumes that content will be delivered from one application to another by some delivery mechanism, and then one or more responses will be returned to the source application. The exact mechanism of transfer is irrelevant to this specification, but may include file transfer, HTTP based transfer, MLLP (HL7 minimal lower layer protocol), MQ series messaging or anything else. The only requirement for the transfer layer is that requests are sent to a known location and responses are returned to the source of the request. This specification considers the source and destination applications as logical entities, and the mapping from logical source and destination to implementation specific addresses is outside the scope of this specification, though this specification does provide a direct delivery mechanism below.

The agreements around the content of the messages and the behavior of the two applications form the "contract" that describes the exchange. The contract will add regional and local agreements to the rules defined in this specification.

This specification ignores the existence of interface engines and message transfer agents that exist between the *source* and *destination*. Either they are transparent to the message/transaction content and irrelevant to this specification, or they are actively involved in manipulating the message content (in particular, the source and destination headers are often changed). If these middleware agents are modifying the message content, then they become responsible for honoring the contract that applies (including applicable profiles) in both directions.

A key aspect of a message is the impact of its content:

Some Events defined by FHIR are assigned to one of these categories, but others are not able to be categorized in advance, and the category must be determined by the content, or the context.

**Message Exchange Patterns**

Each FHIR request message has one or more response messages. There must be at least one response message so that the sender can know that the message was properly received. Multiple response messages SHALL NOT be returned for messages of consequence, and SHOULD not be returned for notifications.

In principle, source applications are not required to wait for a response to a transaction before issuing a new transaction. However in many cases, the messages in a given stream are dependent on each other, and must be sent and processed in order. In addition, some transfer methods may require sequential delivery of messages.

For this reason, a synchronous exchange pattern - where the sender sends a message, and waits on the same channel for a single response, and then sends the next message - is the easiest to understand and manage. The [$process-message](file:///C:\temp\main-pages.html#process) [operation](file:///C:\temp\operations.html) described below works in this fashion.

However synchronous message exchange does not cater for multiple response messages, which may arise when processing queries, and also imposes through-put limitations which may become relevant at high volumes. Additionally, it may not be practical or appropriate to wait for response messages. In these cases, the asynchronous message pattern [described below](file:///C:\temp\main-pages.html#async) should be used.

**MessageHeader Identifiers**

An incoming message contains two identifiers: the Bundle.id and the [MessageHeader](file:///C:\temp\messageheader.html).id. Each time a new message is created, it SHALL be assigned an identifier (MessageHeader.id) that is unique within that message stream. Note that since message streams are often merged with other streams, it is recommended that the identifier should be globally unique. This can be achieved by using a UUID or an OID. Each time a message is sent, the Bundle.id should be changed to a new value.

When a receiver receives and processes the message, it responds with a new message with a new identifier, wrapped in a bundle which also has a new id. The response message also quotes the request MessageHeader.id in MessageHeader.response.identifier so that the source system can relate the response to its request.

**Absence of Reliable Messaging**

Some of the message delivery mechanisms mentioned above are reliable delivery systems - the message is always delivered, or an appropriate error is returned to the source. However most implementations use methods which do not provide reliable messaging, and either the request or the response can get lost in transit. FHIR messaging describes a simple approach that receivers SHOULD conform to in order to handle the absence of reliable messaging that maintains predictable functionality.

If the sender of the message implements reliable messaging, it SHALL do the following when it receives no response to a message within a configured timeout period based on the value specified in the [CapabilityStatement](file:///C:\temp\capabilitystatement.html) messaging.event.category for the event associated with the message:

|  |  |
| --- | --- |
| Consequence | Resend the same message (with the same MessageHeader.id) with the same Bundle.id |
| Currency | Resend the same message (with the same MessageHeader.id) with a different Bundle.id |
| Notification | Resend the same message (with the same MessageHeader.id) with a different Bundle.id |

When a receiver implements reliable messaging, it SHALL check the incoming Bundle.id and MessageHeader.id against a cache of previously received messages. The correct action to take depends on what is received:

|  |  |
| --- | --- |
| Both the Bundle.id and MessageHeader.id have not been received | This is the normal case, and the message should be processed |
| Both envelope and message already received | The original response has been lost (failed to return to the request issuer), and the original response SHALL be resent |
| The MessageHeader.id has already been received, but the Bundle.id is new | A previously seen message has been resubmitted for processing again. The server may either reprocess the message, or reject the message |
| The Bundle.id has already been received, but the MessageHeader.id is new | This is an error - Bundle.id values should never be reused |

The duration period for caching does generally not need to be very long. At a minimum, it could be 1 minute longer than the timeout of the sending system, though it may need to be longer depending on the re-sending policies of the sending system.

Applications that implement reliable messaging declare their reliable cache period in their [Capability Statement](file:///C:\temp\capabilitystatement.html).

**Example: Consequence**

In the first example, a Clinical EHR issues an order for a particular imaging examination to be performed on a patient. This is considered to be a message of **Consequence**: multiple orders should not be created (in practice there are usually human review processes that catch multiple orders, but repeat orders create entropy in the system that is harmful). The EHR sends a message where the Bundle.id is UUID 1 (72edc4e0-6708-42ab-9734-f56721882c10), with a MessageHeader.id of UUID 2 (dad53a57-dcb4-4f18-b066-7239eb4b5229).

The EHR system never receives a response to the message; it does not know whether the request message got lost, or the imaging management systems was unable to process the request, or whether it successfully processed the message and the response was lost. In this case, the EHR system resends the message with same two identifiers.

In this case, the imaging system successfully received the message, and processed it. Because it receives the resent order after 1 minute (which is within its 15 minute cache time), and the two UUIDs 1 and 2 match a message it has already processed, it knows that it already processed the order, and simply returns the previous response. In the case of additional resent queries, the application keeps sending the original response, though it may also alert system administrators that the same original message keeps being resent, since lost messages should be a rare occurrence.

When the EHR system finally receives the message, it knows how the imaging management system responded; it can be sure because the message id from the original request is echoed in the response portion of the returned message.

**Example: Currency**

In this second example, a Clinical EHR needs to know what appointment slots are available for a particular imaging procedure. This is a message of **Currency**: available slots are ever disappearing, and ordering a slot that has become unavailable is a waste of time for the humans and systems involved. The EHR sends a message where the Bundle.id is UUID 3 (4c7f5cb2-5964-4d42-b719-e0227461818c), with a MessageHeader.id is UUID 4 (63ed7d68-b2cc-421d-ba1c-a6c7785581f2).

The EHR system never receives a response to the message; it does not know whether the request message got lost, or the imaging management systems was unable to process the request, or whether it successfully processed the message and the response was lost. In this case, the EHR system resends the message with same MessageHeader.id (UUID 4), but creates a new Bundle.id (c7c17fe4-9560-49c7-b2ae-42636476fb86).

In this case, the imaging system successfully received the message, and processed it. When it receives the resent order after 1 minute (which is within its 15 minute cache time), it sees that although the message id is the same, the Bundle.id has changed, and it reprocesses the message again, and sends a new response.

When the EHR system finally receives the message, it knows the current slot availability on the imaging management system responded.

Note that the existence of active intermediaries (or "middleware") creates the need for this protocol - the original sender matches the response to the request based on the MessageHeader.id, and so an active intermediary that choose the re-initiate a query that it previously relayed cannot change the MessageHeader.id. This protocol avoids the need for the MessageHeader.id to change, and only requires change to the Bundle.id which is never the basis for context linking outside the immediate message exchange protocol described here.

**Capability Statement**

Applications may only assert conformance to "FHIR messaging" if they publish a [Capability statement](file:///C:\temp\capabilitystatement.html) so the claim may be verified. A Capability statement lists all the message events supported (either as sender or receiver) and for each event, a profile that states which resources are bundled (sender), or are required to be bundled (receiver), and any rules about the information content of the individual resources.

**$process-message**

The simplest way to handle messages where there are also [RESTful interactions](file:///C:\temp\http.html) occurring is to use the [$process-message](file:///C:\temp\messageheader-operations.html). This operation accepts a message, processes it according to the definition of the event in the message header, and returns a one or more response messages. For example, in addition to processing the message event, a server may choose to retain all or some the resources and make them available on a RESTful interface, but is not required to do so.

When processing messages, a server may return one of several status codes:

* **200 OK**: Indicates that the message has been fully processed. If an application-level response is expected for the submitted message, that response SHALL be returned as the body of the 200 response.
* **202 Accepted**: Indicates that the receiving system has accepted custody of the message
* **204 No Content**: Indicates that the message has been fully processed and would normally have had an application-level response, but because of instructions from the sender (e.g. the [messageheader-response-request](file:///C:\temp\extension-messageheader-response-request.html) extension), no response is being provided
* **300+**: Indicates that the message was not successfully processed. The server MAY return an [OperationOutcome](file:///C:\temp\operationoutcome.html) with additional information, and SHOULD do so if the response code is 400 or greater.  
  The client SHALL interpret a 4xx response to indicate that there is no point resubmitting the unaltered message, and a 5xx response to indicate an unexpected error occurred on the part of the server, with the implication that it may be appropriate to resubmit the original message. Doing so SHOULD NOT result in a duplicate message response. Repeated failures indicate either a fatal problem with the submission or a problem with the receiving application.

The following rules apply when using $process-message:

* The operation only accepts POST transactions - any other HTTP method will result in an HTTP error
* The request content type submitted is always [a Bundle](file:///C:\temp\bundle.html) with type "message" containing a [Message Header](file:///C:\temp\messageheader.html) resource as the first resource
* The response content type returned is always [a Bundle](file:///C:\temp\bundle.html) with type "message" containing a [Message Header](file:///C:\temp\messageheader.html) resource as the first resource, or an HTTP error
* If the response is an error, the body SHOULD be an [Errors & Warning](file:///C:\temp\operationoutcome.html) resource with full details
* The mailbox may be authenticated using standard HTTP authentication methods, including OAuth

The $process-message operation can be used by any HTTP end-point that accepts FHIR messages, not just FHIR RESTful servers.

In order to ensure consistency of processing, the [logical rules regarding processing of Bundle.id and message id described above](file:///C:\temp\main-pages.html#reliable) SHALL be followed when messages are processed using this operation.

The [$process-message](file:///C:\temp\messageheader-operations.html) operation may be used synchronously, or asynchronously.

**Synchronous Operation**

Synchronous messaging is the easiest to understand; the sender sends a message to the receiver (the server), the server processes it, and then returns a response. Usually (though not always) the sender waits for the response to the current message before sending the next message. This kind of messaging exchange is the most common because it's the simplest to understand.

The following rules apply when using the $process-message operation synchronously:

* The URL (http://server/base/$process-message) has no parameters
* It is an error if the sender POSTs a message that requires multiple response messages
* Servers SHALL accept multiple concurrent message submissions and process them correctly (they are allowed to process them sequentially internally, but multiple concurrent submissions is not an error in its own right)

**Asynchronous Operation**

In Asynchronous messaging, the server acknowledges receipt of the message immediately, and responds to the sender separately. The server may respond more than once to any given message.

The following rules apply when using the $process-message operation synchronously:

* The URL has at least one parameter: http://server/base/$process-message?async=true
* The server acknowledges the message with a 200 OK with no body, or returns an HTTP error if the message cannot be processed. An [OperationOutcome](file:///C:\temp\operationoutcome.html) SHOULD be returned in such a case
* Accepting the message means that the server has understood the message enough to know where to respond
* By default, the server responds by invoking the $process-message using the sender's stated end-point in the message: POST [MessageHeader.source.endpoint]/$process-messages]
* Since the source end-point may be manipulated by message transfer engines, an alternative response address may be specified using the parameter "response-url": http://server/base/$process-message?async=true&response-url=http://server2.com/base/anything. The endpoint at the specified URL SHALL implement the signature of the $process-message operation (parameter async=true, accept a Bundle, return a 200 OK or an error)
* The server submits response messages to the appropriate end-point with the parameter async=true. There is no response message for the response messages

When a message is received, a receiver can determine from the content of the message header whether it's a new message to process, or a response to a message that has already been sent. Note that asynchronous messaging is less reliable than synchronous messaging; more can go wrong. This specification does not dictate any particular error handling protocols or responsibilities; these are left to trading partner agreements between implementers.

**Relationship between Messaging and REST**

As well as this messaging framework documented here, FHIR also defines a [RESTful API](file:///C:\temp\http.html). The messaging and RESTful frameworks are related in that both share the same set of resources on which they operate. In fact, the basic [MessageHeader](file:///C:\temp\messageheader.html) resource that the messaging framework is implemented is itself a resource that can treated in a RESTful approach.

The kinds of functionality that the RESTful API and the messaging framework offer are very similar; their primary difference is architectural in nature.

For instance, the messaging framework defines an event for notifying that a administration resource has been created or updated; the REST API offers similar services ([history](file:///C:\temp\http.html#history) and [Subscription](file:///C:\temp\subscription.html)). On the other hand, there are differences in the capabilities offered - while a patient merge can be implemented as a series of RESTful operations performed by the client that update all resources linked to the patient, when a message command to merge patient records is processed, the server will do all the work, and is also able to merge in areas not exposed on the RESTful API. The REST API, however, provides a set of basic operations on all resources that would need special definitions in the messaging framework - definitions that are not provided.

There is no expectation that RESTful systems will need to offer messaging support, or vice versa, though systems may find it useful to support both sets of functionality in order to satisfy a wider range of implementers.

As a resource that can be used with the RESTful framework, the MessageHeader resource has the normal resource end-point (/MessageHeader), which is used to manage a set of static message resources. This could be used to make an archive of past messages available. **Creating or updating MessageHeader resources in this fashion does not represent the actual occurrence of any event, nor can it trigger any logic associated with the actual event.** It is just for managing a set of message header resources.

**Asynchronous Messaging using the RESTful API**

It is possible to exchange messages using the RESTful end-point as a central point of exchange. This is not particularly efficient compared to other methods, but is useful for low-volume asynchronous exchange.

To send a message, a sender posts the message bundle to the /Bundle end-point, with a uri that identifies the receiver at [MessageHeader.destination.endpoint](file:///C:\temp\messageheader-definitions.html#MessageHeader.destination.endpoint). The RESTful server accepts the bundle, stores it as a single bundle, and indexes it on the [MessageHeader](file:///C:\temp\messageheader.html).

To receive messages, a receiver searches for all messages destined for itself, since it's last check:

GET [base]/Bundle?message.destination-uri=[rcv]&\_lastUpdated=>2015-03-01T02:00:02+01:00

The receiver works through the response, processing each message. As each message is processed, the receiver creates a response message, reversing the source and destination, and posts it back to the server.

To check for responses, the original sender searches for response messages destined for itself, since it's last check:

GET [base]/Bundle?message.destination-uri=[snd]&message.response-id:missing=false

&\_lastUpdated=>2015-03-03T06:03:522+01:00

This lightweight protocol needs ongoing administration to ensure that multiple parties do not interfere with each other by re-using the same system identifier (and against malicious attack).

**Event List**

The *message.code* element carries a [Coding](file:///C:\temp\datatypes.html#Coding) that identifies the event that the message conveys. This table lists the message event codes defined in this specification (the system value for these is "[http://hl7.org/fhir/message-events](file:///C:\temp\valueset-message-events.html)"):

The request and response details: The column values are either a resource that is included as part of the response, or an element that refers to another resource, which means that the target of these references SHALL also be in the message. In this table, the request and response columns list the focus resource for the event, along with other resources that should also be carried in the message directly (if they exist).

**STU Note:** Additional events may be defined elsewhere, though this specification does not yet define how.

Feedback is sought [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**Invoking Operations via Messages**

A message can be used to invoke an operation as defined for a RESTful interface using an operation definition. To invoke an operation using a message:

* The requester sends a message (a bundle with type = message, and a message header resource)
* The message header has an event.system of urn:ietf:rfc:3986
* The event.code is the URL from the operation definition OperationDefinition.url
* The MessageHeader.data refers to a [Parameters](file:///C:\temp\parameters.html) resource
* The parameters resource is populated appropriately as specified by the nominated operation definition

The recipient executes the operation as specified, and then:

* The receiver sends a message (a bundle with type = message, and a message header resource)
* The message header has the same event as the original message
* The MessageHeader contains a response that refers to the original request message, and a code for the outcome, with details if the operation failed
* The MessageHeader.data refers to a [Parameters](file:///C:\temp\parameters.html) resource
* The parameters resource is populated appropriately as specified for the response by the nominated operation definition
* If the operation definition specifies a single return, then this is returned as the target of the MesssageHeader.data directly

Here's an example:

<Bundle xmlns="http://hl7.org/fhir">

<id value="urn:uuid:77831928-2a35-4c08-9496-8232323bf48c"/>

<!-- normal bundle stuff -->

<entry>

<fullUrl value="urn:uuid:6080d4a7-5e05-45dc-96d5-f75329564d1f"/>

<resource>

<MessageHeader>

<id value="cac8143e-6138-4f45-b086-bb8ebf976aae">

<!-- normal message header stuff -->

<event>

<system value="urn:ietf:rfc:3986"/>

<!-- value set expansion -->

<code value="http://hl7.org/fhir/OperationDefinition/ValueSet-expand"/>

</event>

<!-- more normal message header stuff -->

<data>

<reference value="urn:uuid:00213637-dc7c-40d2-a7de-f4ef1eea5685"/>

</data>

</MessageHeader>

</resource>

</entry>

<entry>

<fullUrl value="urn:uuid:00213637-dc7c-40d2-a7de-f4ef1eea5685"/>

<resource>

<Parameters>

<parameter>

<name value="identifier"/>

<valueUri value="http://hl7.org/fhir/ValueSet/identifier-type"/>

</parameter>

</Parameters>

</resource>

</entry>

</Bundle>

Note that there's no way to anchor the execution of the operation against a URL. The only operations that can be executed in this way are defined to be executed at the System or Resource level for a particular resource.

**Invoking Search via Messages**

In the same way that a defined operation can be invoked, a regular search operation can be invoked. This also uses the [Parameters](file:///C:\temp\parameters.html) resource, with the following rules:

* The event code is "search-type" or "search-system" in the system http://hl7.org/fhir/restful-interaction
* If the event type is "search-type" there SHALL be a parameter "resourceType" with specifies the type of resource being searched
* The search parameters are converted to FHIR data types according to the following table

|  |  |
| --- | --- |
| **Search Parameter Type** | **Data Type** |
| number | integer |
| date | dateTime |
| string | string |
| token | string or Coding (split the system and code apart) |
| reference | uri |
| composite | string |
| quantity | string or Quantity (split the syntax out) |
| uri | uri |

Here's an example:

<Bundle xmlns="http://hl7.org/fhir">

<id value="urn:uuid:77831928-2a35-4c08-9496-8232323bf48c"/>

<!-- normal bundle stuff -->

<entry>

<fullUrl value="urn:uuid:c466754c-09c0-4f59-9f76-a48bd0ea27c9"/>

<resource>

<MessageHeader>

<!-- normal message header stuff -->

<event>

<system value="http://hl7.org/fhir/restful-interaction"/>

<!-- Search against Patient -->

<code value="search-type"/>

</event>

<!-- more normal message header stuff -->

<data>

<reference value="urn:uuid:59a17a19-46eb-42d9-821a-f93a0c530cac"/>

</data>

</MessageHeader>

</resource>

</entry>

<entry>

<fullUrl value="urn:uuid:59a17a19-46eb-42d9-821a-f93a0c530cac"/>

<resource>

<Parameters>

<parameter>

<name value="resourceType"/>

<valueString value="Patient"/>

</parameter>

<parameter>

<name value="gender"/>

<valueString value="m"/>

</parameter>

</Parameters>

</resource>

</entry>

</Bundle>

**metadatatypes-definitions.html**

**Metadata Types - Detailed Descriptions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| [ContactDetail](file:///C:\temp\main-pages.html#ContactDetail) | [Contributor](file:///C:\temp\main-pages.html#Contributor) | [DataRequirement](file:///C:\temp\main-pages.html#DataRequirement) | [ParameterDefinition](file:///C:\temp\main-pages.html#ParameterDefinition) |
| [RelatedArtifact](file:///C:\temp\main-pages.html#RelatedArtifact) | [TriggerDefinition](file:///C:\temp\main-pages.html#TriggerDefinition) | [UsageContext](file:///C:\temp\main-pages.html#UsageContext) |  |

**ContactDetail**

See also [Base Definition](file:///C:\temp\metadatatypes.html#ContactDetail), [Examples](file:///C:\temp\metadatatypes-examples.html#ContactDetail) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#ContactDetail).

**Contributor**

See also [Base Definition](file:///C:\temp\metadatatypes.html#Contributor), [Examples](file:///C:\temp\metadatatypes-examples.html#Contributor) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#Contributor).

**DataRequirement**

See also [Base Definition](file:///C:\temp\metadatatypes.html#DataRequirement), [Examples](file:///C:\temp\metadatatypes-examples.html#DataRequirement) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#DataRequirement).

**ParameterDefinition**

See also [Base Definition](file:///C:\temp\metadatatypes.html#ParameterDefinition), [Examples](file:///C:\temp\metadatatypes-examples.html#ParameterDefinition) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#ParameterDefinition).

**RelatedArtifact**

See also [Base Definition](file:///C:\temp\metadatatypes.html#RelatedArtifact), [Examples](file:///C:\temp\metadatatypes-examples.html#RelatedArtifact) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#RelatedArtifact).

**TriggerDefinition**

See also [Base Definition](file:///C:\temp\metadatatypes.html#TriggerDefinition), [Examples](file:///C:\temp\metadatatypes-examples.html#TriggerDefinition) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#TriggerDefinition).

**UsageContext**

See also [Base Definition](file:///C:\temp\metadatatypes.html#UsageContext), [Examples](file:///C:\temp\metadatatypes-examples.html#UsageContext) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#UsageContext).

**metadatatypes-examples.html**

**MetaData Types - Examples**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| [ContactDetail](file:///C:\temp\main-pages.html#ContactDetail) | [Contributor](file:///C:\temp\main-pages.html#Contributor) | [DataRequirement](file:///C:\temp\main-pages.html#DataRequirement) | [ParameterDefinition](file:///C:\temp\main-pages.html#ParameterDefinition) |
| [RelatedArtifact](file:///C:\temp\main-pages.html#RelatedArtifact) | [TriggerDefinition](file:///C:\temp\main-pages.html#TriggerDefinition) | [UsageContext](file:///C:\temp\main-pages.html#UsageContext) |  |

**ContactDetail**

See also [Base Definition](file:///C:\temp\metadatatypes.html#ContactDetail), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#ContactDetail) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#ContactDetail).

For ContactDetail examples, see the examples in [Library](file:///C:\temp\library-examples.html), [ServiceDefinition](file:///C:\temp\servicedefinition-examples.html), [Measure](file:///C:\temp\measure-examples.html), and [PlanDefinition](file:///C:\temp\plandefinition-examples.html).

**Contributor**

See also [Base Definition](file:///C:\temp\metadatatypes.html#Contributor), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#Contributor) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#Contributor).

For Contributor examples, see the examples in [Library](file:///C:\temp\library-examples.html), [ServiceDefinition](file:///C:\temp\servicedefinition-examples.html), [Measure](file:///C:\temp\measure-examples.html), and [PlanDefinition](file:///C:\temp\plandefinition-examples.html).

**DataRequirement**

See also [Base Definition](file:///C:\temp\metadatatypes.html#DataRequirement), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#DataRequirement) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#DataRequirement).

For DataRequirement examples, see the examples in [Library](file:///C:\temp\library-examples.html), [ServiceDefinition](file:///C:\temp\servicedefinition-examples.html), [Measure](file:///C:\temp\measure-examples.html), and [PlanDefinition](file:///C:\temp\plandefinition-examples.html).

**ParameterDefinition**

See also [Base Definition](file:///C:\temp\metadatatypes.html#ParameterDefinition), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#ParameterDefinition) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#ParameterDefinition).

For ParameterDefinition examples, see the examples in [Library](file:///C:\temp\library-examples.html), [ServiceDefinition](file:///C:\temp\servicedefinition-examples.html), [Measure](file:///C:\temp\measure-examples.html), and [PlanDefinition](file:///C:\temp\plandefinition-examples.html).

**RelatedArtifact**

See also [Base Definition](file:///C:\temp\metadatatypes.html#RelatedArtifact), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#RelatedArtifact) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#RelatedArtifact).

For RelatedArtifact examples, see the examples in [Library](file:///C:\temp\library-examples.html), [ServiceDefinition](file:///C:\temp\servicedefinition-examples.html), [Measure](file:///C:\temp\measure-examples.html), and [PlanDefinition](file:///C:\temp\plandefinition-examples.html).

**TriggerDefinition**

See also [Base Definition](file:///C:\temp\metadatatypes.html#TriggerDefinition), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#TriggerDefinition) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#TriggerDefinition).

For TriggerDefinition examples, see the examples in [ServiceDefinition](file:///C:\temp\servicedefinition-examples.html) and [PlanDefinition](file:///C:\temp\plandefinition-examples.html).

**UsageContext**

See also [Base Definition](file:///C:\temp\metadatatypes.html#UsageContext), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#UsageContext) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#UsageContext).

For UsageContext examples, see the examples in [Library](file:///C:\temp\library-examples.html), [ServiceDefinition](file:///C:\temp\servicedefinition-examples.html), [Measure](file:///C:\temp\measure-examples.html), and [PlanDefinition](file:///C:\temp\plandefinition-examples.html).

**metadatatypes-mappings.html**

**Metadata Types - Mappings**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page provides mappings for the data types. There are mappings to [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185), [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186), and (where appropriate) vCard.

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| [ContactDetail](file:///C:\temp\main-pages.html#ContactDetail) | [Contributor](file:///C:\temp\main-pages.html#Contributor) | [DataRequirement](file:///C:\temp\main-pages.html#DataRequirement) | [ParameterDefinition](file:///C:\temp\main-pages.html#ParameterDefinition) |
| [RelatedArtifact](file:///C:\temp\main-pages.html#RelatedArtifact) | [TriggerDefinition](file:///C:\temp\main-pages.html#TriggerDefinition) | [UsageContext](file:///C:\temp\main-pages.html#UsageContext) |  |

**ContactDetail**

See also [Base Definition](file:///C:\temp\metadatatypes.html#ContactDetail), [Examples](file:///C:\temp\metadatatypes-examples.html#ContactDetail) and [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#ContactDetail).

**Contributor**

See also [Base Definition](file:///C:\temp\metadatatypes.html#Contributor), [Examples](file:///C:\temp\metadatatypes-examples.html#Contributor) and [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#Contributor).

**DataRequirement**

See also [Base Definition](file:///C:\temp\metadatatypes.html#DataRequirement), [Examples](file:///C:\temp\metadatatypes-examples.html#DataRequirement) and [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#DataRequirement).

**ParameterDefinition**

See also [Base Definition](file:///C:\temp\metadatatypes.html#ParameterDefinition), [Examples](file:///C:\temp\metadatatypes-examples.html#ParameterDefinition) and [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#ParameterDefinition).

**RelatedArtifact**

See also [Base Definition](file:///C:\temp\metadatatypes.html#RelatedArtifact), [Examples](file:///C:\temp\metadatatypes-examples.html#RelatedArtifact) and [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#RelatedArtifact).

**TriggerDefinition**

See also [Base Definition](file:///C:\temp\metadatatypes.html#TriggerDefinition), [Examples](file:///C:\temp\metadatatypes-examples.html#TriggerDefinition) and [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#TriggerDefinition).

**UsageContext**

See also [Base Definition](file:///C:\temp\metadatatypes.html#UsageContext), [Examples](file:///C:\temp\metadatatypes-examples.html#UsageContext) and [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#UsageContext).

**metadatatypes.html**

**MetaData Types**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 3 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The FHIR specification defines a set of data types that are used for the resource elements. There are four categories of data types:

1. [simple / primitive types, which are single elements with a primitive value](file:///C:\temp\datatypes.html#primitive)
2. [General purpose complex types, which are re-usable clusters of elements](file:///C:\temp\datatypes.html#complex)
3. Complex data types for metadata
4. Special purpose data types: [Reference](file:///C:\temp\references.html#Reference), [Narrative](file:///C:\temp\narrative.html#Narrative), [Extension](file:///C:\temp\extensibility.html#Extension), [Meta](file:///C:\temp\resource.html#Meta), and [Dosage](file:///C:\temp\dosage.html#Dosage)

This page describes the metadata types.

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| [ContactDetail](file:///C:\temp\main-pages.html#ContactDetail) | [Contributor](file:///C:\temp\main-pages.html#Contributor) | [DataRequirement](file:///C:\temp\main-pages.html#DataRequirement) | [ParameterDefinition](file:///C:\temp\main-pages.html#ParameterDefinition) |
| [RelatedArtifact](file:///C:\temp\main-pages.html#RelatedArtifact) | [TriggerDefinition](file:///C:\temp\main-pages.html#TriggerDefinition) | [UsageContext](file:///C:\temp\main-pages.html#UsageContext) |  |

**ContactDetail**

See also [Examples](file:///C:\temp\metadatatypes-examples.html#ContactDetail), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#ContactDetail) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#ContactDetail).

The ContactDetail structure defines general contact details.

**Constraints**

ContactDetail is used in the following places:

**Interpretation of ContactDetail in different contexts**

**Contributor**

See also [Examples](file:///C:\temp\metadatatypes-examples.html#Contributor), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#Contributor) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#Contributor).

**Constraints**

Contributor is used in the following places:

**Interpretation of Contributor in different contexts**

**DataRequirement**

See also [Examples](file:///C:\temp\metadatatypes-examples.html#DataRequirement), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#DataRequirement) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#DataRequirement).

The DataRequirement structure defines a general data requirement for a knowledge asset such as a decision support rule or quality measure.

**Constraints**

A DataRequirement differs from a parameter in that it specifies the data to be provided in terms of the type of data, and specific filters on code-valued and/or date-valued attributes. Data requirements are not named because they are referenced by type within the evaluation context.

Data requirements are used by knowledge modules to communicate the set of required data to a consumer in a way that is computable (as opposed to a set of named parameters which must be integrated by hand based on the meaning of the parameter as communicated through the documentation).

DataRequirements are typically used to communicate patient-dependent information such as MedicationStatements and Encounters, whereas Parameters are typically used to communicate patient-independent information such as configuration values.

For example, consider the following CQL expression:

define "Total Colectomy Procedures":

[Procedure: "Total Colectomy Value Set"] P

where P.performedPeriod during "Measurement Period"

The criteria is looking for procedures matching the "Total Colectomy Value Set" that were performed during the "Measurement Period". In this case, "Measurement Period" is a parameter, referenced by name, whereas the reference to Procedure uses the name of the resource type, and so constitutes a data requirement of the criteria:

<dataRequirement>

<type value="Procedure"/>

<codeFilter>

<path value="code"/>

<valueSetString value="Total Colectomy Value Set"/>

</codeFilter>

<dateFilter>

<path value="performedPeriod"/>

<valuePeriod>

<start value="2016-01-01"/>

<end value="2016-12-31"/>

</valuePeriod>

</dateFilter>

</dataRequirement>

If a resource type has multiple date attributes, then it may be necessary to include multiple date criteria. For instance, in the example above the Procedure resource is using the performedPeriod date attribute. However, the Procedure resource also provides a performedDateTime (specific date and/or time when procedure was performed) attribute. Therefore, if the date criteria for the Procedure in the example was recorded using the performedDateTime attribute, the date criteria would be incomplete. To account for this additional date attribute, the example could be expanded as follows:

define "Total Colectomy Procedures":

[Procedure: "Total Colectomy Value Set"] P

where exists (P.performedPeriod during "Measurement Period")

or exists (P.performedDateTime during "Measurement Period")

The data requirement for the expanded criteria:

<dataRequirement>

<type value="Procedure"/>

<codeFilter>

<path value="code"/>

<valueSetString value="Total Colectomy Value Set"/>

</codeFilter>

<dateFilter>

<path value="performedPeriod"/>

<path value="performedDateTime"/>

</dateFilter>

</dataRequirement>

DataRequirement is used in the following places:

**ParameterDefinition**

See also [Examples](file:///C:\temp\metadatatypes-examples.html#ParameterDefinition), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#ParameterDefinition) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#ParameterDefinition).

The ParameterDefinition structure defines a parameter to a knowledge asset such as a decision support rule or quality measure.

Parameters are typically used to communicate patient-independent information such as configuration values, whereas DataRequirements are typically used to communicate patient-dependent information such as MedicationStatements and Encounters.

**Constraints**

**RelatedArtifact**

See also [Examples](file:///C:\temp\metadatatypes-examples.html#RelatedArtifact), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#RelatedArtifact) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#RelatedArtifact).

The RelatedArtifact structure defines resources related to a module such as previous and next versions of documents, documentation, citations, etc. Note that the name *resource* here is being used in a more general sense than the FHIR-specific Resource. The related resource may be a FHIR resource, or it may be another type of resource, represented using the Attachment data type.

**Constraints**

RelatedArtifact is used in the following places:

**Interpretation of RelatedArtifact in different contexts**

**TriggerDefinition**

See also [Examples](file:///C:\temp\metadatatypes-examples.html#TriggerDefinition), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#TriggerDefinition) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#TriggerDefinition).

The TriggerDefinition structure defines when a knowledge artifact is expected to be evaluated. The structure can represent three main kinds of triggering events:

* Named Event
* Scheduled Event
* Data Event

A named event is an event identified by the implementation environment. This allows any event generated within the implementation environment to be used as a trigger, but it requires pre-coordination of the names involved with the consuming environments.

A scheduled event occurs on a fixed or periodic schedule.

And finally, a data event occurs in response to some data-related event in the integrated environment such as a record being added or updated. The data-of-interest for a data event is described using a [DataRequirement](file:///C:\temp\metadatatypes.html#DataRequirement). This allows for systems to automatically invoke based on data activity occurring within the system.

**Constraints**

**UsageContext**

See also [Examples](file:///C:\temp\metadatatypes-examples.html#UsageContext), [Detailed Descriptions](file:///C:\temp\metadatatypes-definitions.html#UsageContext) and [Mappings](file:///C:\temp\metadatatypes-mappings.html#UsageContext).

The UsageContext structure defines the context of use for a module.

**Constraints**

UsageContext is used in the following places:

**Interpretation of UsageContext in different contexts**

**modulemetadata-definitions.html**

**Module Metadata Detailed Descriptions**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**ModuleMetadata**

**modulemetadata-examples.html**

**Module Metadata Examples**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

For ModuleMetadata examples, see the examples in [Library](file:///C:\temp\library-examples.html), [DecisionSupportServiceModule](file:///C:\temp\decisionsupportservicemodule-examples.html), [DecisionSupportRule](file:///C:\temp\decisionsupportrule-examples.html), [Measure](file:///C:\temp\measure-examples.html), and [OrderSet](file:///C:\temp\orderset-examples.html).

**modulemetadata-mappings.html**

**Module Metadata Mappings**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page provides mappings for Module Metadata.

**modulemetadata.html**

**Module Metadata**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The ModuleMetadata structure defines the common metadata elements used by knowledge management artifacts. This information includes descriptive and topical metadata to enable repository searches, as well as publication, governance, and evidentiary support information.

This structure contains the following general categories of elements:

* Module identity and versioning
* Description and documentation
* Publication and governance information
* Repository metadata

**Content**

**Constraints**

ModuleMetaData is used in the following places:

**Interpretation of ModuleMetadata in different contexts**

**modules-fragment - Copy.html**

[Clinical Reasoning](file:///C:\temp\clinicalreasoning-module.html)

Decision Support, Clinical Quality Measures

[Clinical](file:///C:\temp\clinicalsummary-module.html)

[Allergy](file:///C:\temp\allergyintolerance.html), [Problem](file:///C:\temp\condition.html) etc

[Diagnostics](file:///C:\temp\diagnostics-module.html)

[Observation](file:///C:\temp\observation.html), [Report](file:///C:\temp\diagnosticreport.html), [Order](file:///C:\temp\diagnosticorder.html) etc

[Medications](file:///C:\temp\medications-module.html)

[Order](file:///C:\temp\medicationrequest.html), [Dispense](file:///C:\temp\medicationdispense.html), [Administration](file:///C:\temp\medicationadministration.html), [Statement](file:///C:\temp\medicationstatement.html) etc

[Workflow](file:///C:\temp\workflow-module.html)

[Task](file:///C:\temp\task.html), [Subscription](file:///C:\temp\subscription.html) etc

[Financial](file:///C:\temp\financial-module.html)

[Claim](file:///C:\temp\claim.html), [Eligibility](file:///C:\temp\eligibility.html) etc

[Administration](file:///C:\temp\administration-module.html)

[Patient](file:///C:\temp\patient.html), [Practitioner](file:///C:\temp\practitioner.html), [Device](file:///C:\temp\device.html), [Organization](file:///C:\temp\organization.html), [Location](file:///C:\temp\location.html), [Healthcare Service](file:///C:\temp\healthcare.html)

[Ontology](file:///C:\temp\ontology-module.html)

[RDF](file:///C:\temp\rdf.html)  
todo

[Terminology](file:///C:\temp\terminology-module.html)

[CodeSystem](file:///C:\temp\codesystem.html)  
[ValueSet](file:///C:\temp\valueset.html)  
[ConceptMap](file:///C:\temp\conceptmap.html)  
[Terminology Svc](file:///C:\temp\terminology-service.html)

[Conformance](file:///C:\temp\conformance-module.html)

[StructureDefn](file:///C:\temp\structuredefinition.html)  
[CapabilityStatement](file:///C:\temp\capabilitystatement.html)  
[Profiling](file:///C:\temp\profiling.html)

[Security & Privacy](file:///C:\temp\secpriv-module.html)

[Consent](file:///C:\temp\consent.html)

[Implementer Support](file:///C:\temp\implsupport-module.html)

[Downloads](file:///C:\temp\downloads.html)  
[Testing](file:///C:\temp\testing.html)

[Foundation](file:///C:\temp\foundation-module.html)

[Base Documentation](file:///C:\temp\documentation.html), [XML](file:///C:\temp\xml.html), [JSON](file:///C:\temp\json.html), [REST API](file:///C:\temp\http.html) + [Search](file:///C:\temp\search.html), [Data Types](file:///C:\temp\datatypes.html), [Extensions](file:///C:\temp\extensibility.html)

**modules-fragment.html**

**Level 1** Basic framework on which the specification is build

[Foundation](file:///C:\temp\foundation-module.html)

[Base Documentation](file:///C:\temp\documentation.html), [XML](file:///C:\temp\xml.html), [JSON](file:///C:\temp\json.html), [REST API](file:///C:\temp\http.html) + [Search](file:///C:\temp\search.html), [Data Types](file:///C:\temp\datatypes.html), [Extensions](file:///C:\temp\extensibility.html)

**Level 2** Supporting Implementation, and binding to external specifications

[Implementer Support](file:///C:\temp\implsupport-module.html)

[Downloads](file:///C:\temp\downloads.html),  
[Common Use Cases](file:///C:\temp\usecases.html),  
[Testing](file:///C:\temp\testing.html)

[Security & Privacy](file:///C:\temp\secpriv-module.html)

[Security](file:///C:\temp\security.html),  
[Consent](file:///C:\temp\consent.html)

[Conformance](file:///C:\temp\conformance-module.html)

[StructureDefinition](file:///C:\temp\structuredefinition.html),  
[CapabilityStatement](file:///C:\temp\capabilitystatement.html),  
[Profiling](file:///C:\temp\profiling.html)

[Terminology](file:///C:\temp\terminology-module.html)

[CodeSystem](file:///C:\temp\codesystem.html),  
[ValueSet](file:///C:\temp\valueset.html),  
[ConceptMap](file:///C:\temp\conceptmap.html),  
[Terminology Svc](file:///C:\temp\terminology-service.html)

[Linked Data](file:///C:\temp\linked-data-module.html)

[RDF](file:///C:\temp\rdf.html)

**Level 3** Linking to real world concepts in the healthcare system

[Administration](file:///C:\temp\administration-module.html)

[Patient](file:///C:\temp\patient.html), [Practitioner](file:///C:\temp\practitioner.html), [Device](file:///C:\temp\device.html), [Organization](file:///C:\temp\organization.html), [Location](file:///C:\temp\location.html), [Healthcare Service](file:///C:\temp\healthcareservice.html)

**Level 4** Record-keeping and Data Exchange for the healthcare process

[Clinical](file:///C:\temp\clinicalsummary-module.html)

[Allergy](file:///C:\temp\allergyintolerance.html), [Problem](file:///C:\temp\condition.html), etc.

[Diagnostics](file:///C:\temp\diagnostics-module.html)

[Observation](file:///C:\temp\observation.html), [Report](file:///C:\temp\diagnosticreport.html), [Request](file:///C:\temp\procedurerequest.html), etc.

[Medications](file:///C:\temp\medications-module.html)

[Order](file:///C:\temp\medicationrequest.html), [Dispense](file:///C:\temp\medicationdispense.html), [Administration](file:///C:\temp\medicationadministration.html), [Statement](file:///C:\temp\medicationstatement.html), etc.

[Workflow](file:///C:\temp\workflow-module.html)

[Task](file:///C:\temp\task.html), [Subscription](file:///C:\temp\subscription.html), etc.

[Financial](file:///C:\temp\financial-module.html)

[Claim](file:///C:\temp\claim.html), [EligibilityRequest](file:///C:\temp\eligibilityrequest.html), etc.

**Level 5** Providing the ability to reason about the healthcare process

[Clinical Reasoning](file:///C:\temp\clinicalreasoning-module.html)

Decision Support, Clinical Quality Measures

**modules-list.html**

* [Foundation](file:///C:\temp\foundation-module.html): The basic definitional and exchange infrastructure on which the rest of the specification is built
* [Implementer Support](file:///C:\temp\implsupport-module.html): Services to help implementers make use of the specification
* [Security & Privacy](file:///C:\temp\secpriv-module.html): Documentation and services to create and maintain security, integrity and privacy
* [Conformance](file:///C:\temp\conformance-module.html): How to test conformance to the specification, and define implementation guides
* [Terminology](file:///C:\temp\terminology-module.html): Use and support of terminologies and related artifacts
* [Linked Data](file:///C:\temp\linked-data-module.html): How to use RDF, and how ontologies are used when defining the FHIR content
* [Administration](file:///C:\temp\administration-module.html): Basic resources for tracking patients, practitioners, organizations, devices, substances, etc
* [Clinical](file:///C:\temp\clinicalsummary-module.html): Core clinical content such as problems, allergies, and the care process (care plans, referrals) + more
* [Medications](file:///C:\temp\medications-module.html): Medication management and immunization tracking
* [Diagnostics](file:///C:\temp\diagnostics-module.html): Observations, Diagnostic reports and requests + related content
* [Workflow](file:///C:\temp\workflow-module.html): Managing the process of care, and technical artifacts to do with obligation management
* [Financial](file:///C:\temp\financial-module.html): Billing and Claiming support
* [Clinical Reasoning](file:///C:\temp\clinicalreasoning-module.html): Clinical Decision Support and Quality Measures

**modules.html**

**Getting Started with FHIR**

FHIR is a platform specification that defines a set of capabilities use across the healthcare process, in all jurisdictions, and in lots of different context. While the basics of the FHIR specification are relatively straight-forward (see the Overviews: [General](file:///C:\temp\overview.html), [Developers](file:///C:\temp\overview-dev.html), [Clinical](file:///C:\temp\overview-clinical.html), and [Architects](file:///C:\temp\overview-arch.html)), it can still be difficult to know where to start when implementing a solution based on FHIR.

This page provides some guidance to help get new implementers started on their path to successful implementation. Beyond reading the overviews (previous paragraph), where should an implementer start? Generally, an implementer needs to resolve:

* How will information be exchanged? (see [Foundation Module](file:///C:\temp\foundation-module.html#uses))
* How are terminologies being used? (see [Terminology Module](file:///C:\temp\terminology-module.html#uses))
* How will the information be secured? (see [Security and Privacy Module](file:///C:\temp\secpriv-module.html#uses))
* What information is going to be exchanged?

The remaining sections provide guidance on specific areas (Foundation, Implementer Support, Security and Privacy, Conformance, Terminology, Linked Data, Administration, Clinical, Diagnostics, Medications, Workflow, Financial and Clinical Reasoning).

**Modules**

In order to help implementers find their way around the specification and answer these questions, it is organised into a set of "modules". Each module represents a different functional area of the specification, and contains:

* **Scope and Index**:A description of the content covered by the module, and an index of the important content
* **Use cases**: guidance for common uses of the module, and how to approach them. This is a key resource for implementers familiarizing themselves with the FHIR specification
* **Security / Privacy**: information
* **Roadmap**: Where the content covered by the module is in terms of overall progress (see also, for general information: [FHIR Timelines](file:///C:\temp\timelines.html)

Broadly, the modules are organized into 3 groups:

* Infrastructure (bottom rung, and bottom row of boxes)
* Content (middle rung, and top row of boxes)
* Reasoning (top rung)

Dependencies between the modules are mainly downwards, with some horizontal dependencies. Implementers should choose the content modules to engage with based on their requirements, and should only engage with the reasoning module if they need to do clinical decision support, and/or Quality Measures.

In addition to the use case based assistance in the modules, these additional documentation pages may be useful:

* [Common Use Cases](file:///C:\temp\usecases.html): Personal Health Record, Document Sharing (XDS) and Decision Support
* [Resource Guide](file:///C:\temp\resourceguide.html): Further information about the resources and the relationship between them

Finally, one important place to look is the [registry of implementation guides](http://www.fhir.org/guides/registry), to see whether similar (or identical) requirements have been met.

**narrative-definitions.html**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**narrative-example.html**

**Example Narrative**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Plain HTML, No Styles**

**Heading 3**

**Heading 4**

**Heading 5**

**Heading 6**

Paragraph. span. [Link](file:///C:\temp\main-pages.html#link). **Bold**, br:  
*em*, *Italics*, **strong**, small, big Teletype Text, small, *Definition term*, q, *var*. All provided by HL7, for FHIR (*cite*).

Paragraph in a blockquote, with an hr after it:

Paragraph in a div (Link Target)

* Unordered List Item

1. Ordered List Item

DT Item

DD Item

Some Pre Text

with a line break

Table:

| Table Caption | | |
| --- | --- | --- |
| **Head Cell 1** | **Head Cell 2** | **Head Cell 3** |
| Foot Cell 1 | Foot Cell 2 | Foot Cell 3 |
| Body Cell 1 | Body Cell 2 | Body Cell 3 |

Code Block Sample Block

**External Styles**

**Text:**

Example Text: bold, italics, underline and strikethrough

This paragraph is left aligned. The content should be laid out aligned at the left of the screen. The content should be laid out aligned at the left of the screen. The content should be laid out aligned at the left of the screen. The content should be laid out aligned at the left of the screen. The content should be laid out aligned at the left of the screen.

This paragraph is right aligned. The content should be laid out aligned at the right of the screen. The content should be laid out aligned at the right of the screen. The content should be laid out aligned at the right of the screen. The content should be laid out aligned at the right of the screen. The content should be laid out aligned at the right of the screen.

This paragraph is center aligned. The content should be laid out aligned at the center of the screen. The content should be laid out aligned at the center of the screen. The content should be laid out aligned at the center of the screen. The content should be laid out aligned at the center of the screen. The content should be laid out aligned at the center of the screen.

This paragraph is justified. The content should be laid out aligned at both the left and right of the screen. The content should be laid out aligned at both the left and right of the screen. The content should be laid out aligned at both the left and right of the screen. The content should be laid out aligned at both the left and right of the screen. The content should be laid out aligned at both the left and right of the screen.

**Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Border Left | Border Right | Border Top | Border Bottom |

**List:**

1. arabic (Item 1)
2. Item Two
3. little-roman (Item 1)
4. Item Two
5. big-roman (Item 1)
6. Item Two
7. little-alpha (Item 1)
8. Item Two
9. big-alpha (Item 1)
10. Item Two

* disc (Item 1)
* Item Two
* unlist(Item 1)
* Item Two
* circle (Item 1)
* Item Two
* square (Item 1)
* Item Two

**Internal Styles**

Example Text: **bold**, italics, underline and ~~strikethrough~~. Font-Family Serif and Sans Serif, Font-size 50% 80% 150%, Font-Color Navy Maroon Brown, Background-color Aqua Silver Pink.

Whitespace Control:

Normal Whitespace Test, long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long sentence

No-Wrap Whitespace Test, long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long sentence

Pre Whitespace

Test, long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long sentence

Pre-Line Whitespace Test, long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long sentence

Pre-Wrap Whitespace Test, long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long long sentence

This paragraph is left aligned. The content should be laid out aligned at the left of the screen. The content should be laid out aligned at the left of the screen. The content should be laid out aligned at the left of the screen. The content should be laid out aligned at the left of the screen. The content should be laid out aligned at the left of the screen.

This paragraph is right aligned. The content should be laid out aligned at the right of the screen. The content should be laid out aligned at the right of the screen. The content should be laid out aligned at the right of the screen. The content should be laid out aligned at the right of the screen. The content should be laid out aligned at the right of the screen.

This paragraph is center aligned. The content should be laid out aligned at the center of the screen. The content should be laid out aligned at the center of the screen. The content should be laid out aligned at the center of the screen. The content should be laid out aligned at the center of the screen. The content should be laid out aligned at the center of the screen.

This paragraph is justified. The content should be laid out aligned at both the left and right of the screen. The content should be laid out aligned at both the left and right of the screen. The content should be laid out aligned at both the left and right of the screen. The content should be laid out aligned at both the left and right of the screen. The content should be laid out aligned at both the left and right of the screen.

1. armenian (Item 1)
2. Item Two
3. cjk-ideographic (Item 1)
4. Item Two
5. decimal (Item 1)
6. Item Two
7. decimal-leading-zero (Item 1)
8. Item Two
9. georgian (Item 1)
10. Item Two
11. hebrew (Item 1)
12. Item Two
13. hiragana (Item 1)
14. Item Two
15. hiragana-iroha (Item 1)
16. Item Two
17. inherit (Item 1)
18. Item Two
19. katakana (Item 1)
20. Item Two
21. katakana-iroha (Item 1)
22. Item Two
23. lower-alpha (Item 1)
24. Item Two
25. lower-greek (Item 1)
26. Item Two
27. lower-latin (Item 1)
28. Item Two
29. lower-roman (Item 1)
30. Item Two
31. none (Item 1)
32. Item Two
33. upper-alpha (Item 1)
34. Item Two
35. upper-latin (Item 1)
36. Item Two
37. upper-roman (Item 1)
38. Item Two
39. upper-roman (Item 1)
40. Item Two

**narrative.html**

**Narrative**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 3 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Any resource that is a [DomainResource](file:///C:\temp\domainresource.html) (almost all types of resource) may include a human-readable narrative that contains a summary of the resource, and may be used to represent the content of the resource to a human.

If narrative is present with a status other than 'empty', it SHALL reflect all content needed for a human to understand the essential clinical and business information for the resource. I.e. It must be safe to render only the narrative of the resource without displaying any of the resource's discrete/encoded information. Resource definitions and/or profiles on resources may define what content should be represented in the narrative to ensure clinical safety.

The narrative for a resource is allowed to contain additional information that is not in the structured data, including human-edited content. Such additional information SHALL be in the scope of the definition of the resource, though it is common for the narrative to include additional descriptional information extracted from other referenced resources when describing references. Narrative for a resource should include summary information about any referenced resources that would be required for a consumer of the resource to be able to understand the key, essential information about a resource without retrieving any additional resources.

For example, the narrative for a MedicationRequest might include brief summary information about the referenced patient, prescriber and medication. Some resources (e.g. List, Composition) may provide specific rules about what content must (or must not) be included in the resource narrative. Consideration should be given to the fact that referenced resources may be updated without updating referencing resources, so the proportion of content of a referenced resource included in a referencing resource should be limited. Systems MAY choose how narrative is generated, including how much de-referencing to perform, but SHALL NOT assume that the resource is rendered in any particular context when generating narrative, since resources will be used in multiple contexts.

Resources SHOULD always contain narrative to support human-consumption as a fallback. Structured data SHOULD NOT generally contain information of importance to human readers that is omitted from the narrative. Creators of FHIR resources should not assume that systems will render (or that humans will see) data that is not in the narrative. However, in strictly managed trading systems where all systems share a common data model and additional text is unnecessary or even a clinical safety risk, the narrative may be omitted. Implementers should give careful consideration before doing this, as it will mean that such resources can only be understood in the limited trading environment. Closed trading partner environments are very likely to open up during the lifetime of the resources they define. Also, many workflow steps involving finding and aggregating resources are much more difficult or tedious if the resources involved do not have their own text.

In some cases, a resource may only have text with little or no additional discrete data (as long as all minOccurs=1 elements are satisfied). This may be necessary for data from legacy systems where information is captured as a "text blob" or where text is additionally entered raw or narrated and encoded in formation is added later.

Note that [contained](file:///C:\temp\references.html#contained) Resources SHALL NOT have a narrative of their own.

The narrative is an XHTML fragment with a flag to indicate its relationship to the data:

The contents of the *div* element are an XHTML fragment that SHALL contain only the basic HTML formatting elements described in chapters 7-11 (except section 4 of chapter 9) and 15 of the HTML 4.0 standard, <a> elements (either name or href), images and internally contained style attributes. The XHTML content SHALL NOT contain a head, a body element, external stylesheet references, deprecated elements, scripts, forms, base/link/xlink, frames, iframes, objects or event related attributes (e.g. onClick). This is to ensure that the content of the narrative is contained within the resource and that there is no active content. Such content would introduce [security issues](file:///C:\temp\security.html#narrative) and potentially safety issues with regard to extracting text from the XHTML.

The div element SHALL have some non-whitespace content (text or an image).

<narrative>

<div xmlns="http://www.w3.org/1999/xhtml">This is a simple

example with only plain text</div>

</narrative>

<narrative>

<div xmlns="http://www.w3.org/1999/xhtml">

<p>

This is an <i>example</i> with some <b>xhtml</b> formatting.

</p>

</div>

</narrative>

The inner portion of the div content is often used for the *innerHTML* property in a browser. In order to simplify this kind of processing, when the narrative is represented in JSON, it SHALL be encoded so that the characters between the first '>' and the last '<' delimiters is the content of the <div> element; e.g.

"div": "<div>text</div>"

is legal, but this is not:

"div": "<?xml ...><div>text</div>"

Note that the XHTML is contained in general XML so there is no support for HTML entities like &nbsp; or &copy; etc. Unicode characters SHALL be used instead. Unicode &#160; substitutes for &nbsp;.

The narrative content SHOULD be in the [language of the resource](file:///C:\temp\resource-definitions.html#Resource.language), but there is no reason to expect that HTML type tooling would understand the resource [language](file:///C:\temp\resource.html#content) element. For this reason, a lang attribute on the <div> SHOULD also be used (see [the note in the HTML 5 specification about use of language](http://www.w3.org/html/wg/drafts/html/master/dom.html#the-lang-and-xml:lang-attributes)).

**Image References**

Image source data (the src attribute) may refer to an image found in the resource (as a contained [Media](file:///C:\temp\media.html) or [Binary](file:///C:\temp\binary.html) resource) by its *id*:

<Patient xmlns="http://hl7.org/fhir">

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml">

<p>... <img src="#pic1"/>. ....</p>

</div>

</text>

<contained>

<Binary><id value="pic1"/><contentType value="image/gif"/><content value="MEKH....SD/Z"/></Binary>

</contained>

</Patient>

References between the narrative and the resource data (in either direction) are mediated by the XML id/idref attributes. in JSON, the property "id" is used which is equivalent to the XML attribute "id".

The id attribute SHALL have a unique value *within the resource* with regard to any other id attributes: the uniqueness and resolution scope of these id references is within the resource that contains them. Contained resources are included in the id uniqueness scope of the resource that contains them.

If multiple resources are combined into a single combined document, such as a [Bundle](file:///C:\temp\bundle.html), duplicate values of the id attribute may occur between resources. This SHALL be managed by applications reading the resources.

Since images that are not contained in the resource cannot be guaranteed to be available when the resource is presented to a user, the source for any images that are an essential part of the narrative SHOULD always be embedded as a [data: url](https://tools.ietf.org/html/rfc2397), in an attachment or a contained resource.

**Styling the XHTML**

The XHTML fragment in the narrative may be styled using cascading stylesheets with either external or internal styles. External styles are applied using the *class* and *id* attributes on the XHTML elements and internal styles are applied using a *style* attribute on the XHTML elements directly.

In order to minimise manageability and security issues, authoring systems cannot specify the CSS stylesheet to use directly. Instead, the application that displays the resource provides the stylesheets. This means that the rendering system chooses what styles can be used, but the authoring system must use them in advance. Authoring systems can use these classes, which SHALL be supported by all rendering systems:

|  |  |  |
| --- | --- | --- |
| bold | Bold | { font-weight: bold } |
| italics | Italics Text | { font-style: italic } |
| underline | Underlined Text | { text-decoration: underline } |
| strikethrough | Strikethrough Text | { text-decoration: line-through } |
| left | Left Aligned | { text-align : left } |
| right | Right Aligned | { text-align : right } |
| center | Center Aligned | { text-align : center } |
| justify | Justified | { text-align : justify } |
| border-left | Border on the left | { border-left: 1px solid grey } |
| border-right | Border on the right | { border-right: 1px solid grey } |
| border-top | Border on the top | { border-top: 1px solid grey } |
| border-bottom | Border on the bottom | { border-bottom: 1px solid grey } |
| arabic | List is ordered using Arabic numerals: 1, 2, 3 | { list-style-type: decimal } |
| little-roman | List is ordered using little Roman numerals: i, ii, iii | { list-style-type: lower-roman } |
| big-roman | List is ordered using big Roman numerals: I, II, III | { list-style-type: upper-roman } |
| little-alpha | List is ordered using little alpha characters: a, b, c | { list-style-type: lower-alpha } |
| big-alpha | List is ordered using big alpha characters: A, B, C | { list-style-type: upper-alpha } |
| disc | List bullets are simple solid discs | { list-style-type: disc } |
| circle | List bullets are hollow discs | { list-style-type : circle } |
| square | List bullets are solid squares | { list-style-type: square } |
| unlist | List with no bullets | { list-style-type: none } |

Note: for testing purposes, there is an [example resource](file:///C:\temp\basic-example-narrative.xml.html) that includes all of these styles. It is also available [as XHTML](file:///C:\temp\narrative-example.html) and a [standard stylesheet](file:///C:\temp\fhir-runtime.css) that includes all of these styles. Use of styles not on this list will require an arrangement between producing and consuming systems.

Authoring systems may refer to additional classes, but cannot rely on the fact that they will be supported. If the additional classes are critical for safe rendering, trading partner agreements will be required.

Authoring systems may also use internal styles using the *style* attribute. This has the advantage of not depending on external interpretation, but also has the side effect of making content more difficult to manage when rendering, so applications should use this approach with care.

Authoring systems may fix the following styling aspects of the content:

* bold, italic, underline, strikethrough
* font color, family and size
* background color, text alignment
* whitespace interpretation
* ordered list number format (since it may be referred to in text)

These style properties are specified in-line using the style attribute. Rendering systems SHOULD respect any of these rendering styles when they are specified in the style attribute, although appropriate interpretation is allowed in certain contexts (e.g. a low-contrast display for dark rooms or a high-contrast display for the visually impaired may adjust colors accordingly).

Note that rendering systems are allowed to ignore or override any of the internal or external styles described above, but SHOULD be careful to ensure that this is only done in the context of well maintained trading partner agreements, as altering the presentation of the text may create clinical safety issues.

Authors MAY specify additional styles and style properties as specified in the CSS specification, but these are extensions to this specification and renderers are not required to heed them. It SHOULD be safe to view the narrative without these additional styling features available.

Note that there are additional rules around styling for [documents](file:///C:\temp\documents.html#css) presentation.

**Clinical Safety Concerns**

Health care records are often associated with legislative and business requirements for very long retention times (up to a century), and extreme risk aversion with regards to inconsistent display across a variety of devices. Although the narrative is allowed to use the standard XHTML and CSS features as described above, implementations are encouraged to show restraint when using the features available. Even when trading partner arrangements limit the current requirements made on a system, experience shows that these trading arrangements will likely broaden over time.

In particular:

* complex layered layouts requiring careful testing of the match between the XHTML div and span elements and styles, and those that include nested tables (possibly with images) as well make rendering consistency difficult, and implementations SHOULD avoid these
* The use of styles e.g. bolding, italics and color; SHOULD NOT be used as the sole way to convey meaning or semantics, but should be used in conjunction with other data elements to ensure consistent, long term interoperability.

**ncimeta.html**

**Using the NCI Metathesaurus with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Summary**

|  |  |
| --- | --- |
| Source | NCI Metathesaurus [the NCI Center for Biomedical Informatics and Information Technology (CBIIT)](http://cbiit.nci.nih.gov) |
| System | The URI <http://ncimeta.nci.nih.gov> identifies the NCI Metathesaurus |
| Version | There is no version or versioning associated with the NCI metathesaurus |
| Code | The Concept Unique Identifier (CUI) is used for the code value for a Metathesaurus concept |
| Display | The name should be used as the display for English usage (e.g. "*Aerosol Dose Form*" for CUI C1112870) |
| Inactive | Todo: Describe how it is determined which concepts are inactive |
| Subsumption | No Subsumption relationships are defined for the NCI Metathesaurus |
| Filter Properties | None are described yet |

**Version Issues**

There are no staged releases of the NCI metathesaurus, so there is no versioning policy.

**Copyright/License Issues**

The NCI metathesaurus is in the public domain, so there are no copyright notices needed in value sets that refer to NCI metathesaurus concepts, and there are no licensing requirements limiting use of NCI metathesaurus concepts in instances or systems.

**MCI Metathesaurus MySQL Database**

Like [RxNorm](file:///C:\temp\rxnorm.html) the RRF files that are the distributed source of the NCI Metathesaurus can be used to populate a MySQL database that contains the data. This page provides SQL statements that describe how to implement the features of the NCI Metathesaurus correctly against this database. They are provided only for implementer convenience, and do not imply that any particular approach must be used in implementations. (Note: for consistency, the RxNorm table and column names are used and the CUIs are 1 character longer, so the scripts must be updated).

For example, the correct display name for a CUI is

select STR from rxnconso where RXCUI = :code and SAB = 'RXNORM' and TTY <> 'SY'

.

**NCI metathesaurus Filter Properties**

This section documents the property filters that can be used with the RxNorm code system in value set composition statements.

The base SQL statement for returning a list of CUIS that conform to these filters is:

select RXCUI from RXNCONSO where SAB = 'RXNORM' and TTY <> 'SY'

**Semantic Type**

|  |  |
| --- | --- |
| Description | Allows to choose a set of CUIs based on their Semantic Type |
| Property Name | STY |
| Operations Allowed | = / in |
| Values Allowed | [column:]value |
| Comments | If not column is specified, the default column is TUI |
| SQL | and RXCUI in (select RXCUI from RXNSTY where [:column] = :value) |

**Source**

|  |  |
| --- | --- |
| Description | Allows for selection of the set of concepts that have mappings to a particular rxnorm source |
| Property Name | SAB |
| Operations Allowed | = / in |
| Values Allowed | Values from the SAB table (e.g. select RSAB from RXNsab) |
| SQL | and RXCUI in (select RXCUI from RXNconso where SAB = :value) |

**Term Type**

|  |  |
| --- | --- |
| Description | Allows for selection of a concept based on its designated type |
| Property Name | TTY |
| Operations Allowed | = / in |
| Values Allowed | TTY values from the RxNorm Concept table (e.g. select distinct TTY from rxnconso) |
| SQL | and TTY = :value |

**Relationship**

|  |  |
| --- | --- |
| Description | Allows for selection of a concept based on its relationships |
| Property Name | [REL] |
| Operations Allowed | = / in |
| Values Allowed | CUI:[RXCUI] or AUI:[RXAUI] must be a valid CUI or AUI. Note that a CUI does not need to have a SAB=RXNORM entry to be used here |
| Comments | [REL] (:rel) is one of AQ, CHD, PAR, QB, RB, RN, RO, RQ, SIB or SY |
| SQL | for CUI:  and (RXCUI in (select RXCUI from rxnconso where RXCUI in (select RXCUI1 from rxnrel where REL = :rel and RXCUI2 = :value))  for AUI:  and (RXCUI in (select RXCUI from rxnconso where RXAUI in (select RXAUI1 from rxnrel where REL = :rel and RXAUI2 = :value)) |

**Relationship Type**

|  |  |
| --- | --- |
| Description | Allows for selection of a concept based on the type of its relationships |
| Property Name | [RELA] |
| Operations Allowed | = / in |
| Values Allowed | CUI:[RXCUI] or AUI:[RXAUI] must be a valid CUI or AUI. Note that a CUI does not need to have a SAB=RXNORM entry to be used here |
| Comments | [RELA] (:rela) is one of the relationship types listed in the NCI file "Relationships\_Help\_Page.txt" - the current list (nearly 1000 types) is at the end of the page |
| SQL | for CUI:  and (RXCUI in (select RXCUI from rxnconso where RXCUI in (select RXCUI1 from rxnrel where RELA = :rel and RXCUI2 = :value))  for AUI:  and (RXCUI in (select RXCUI from rxnconso where RXAUI in (select RXAUI1 from rxnrel where RELA = :rel and RXAUI2 = :value)) |

**Implicit Value Sets**

This section needs investigation

**Current NCI Metathesaurus relationship types**

* 3\_UTR\_of
* 5\_UTR\_of
* Abnormal\_Cell\_Affected\_By\_Chemical\_Or\_Drug
* Abnormality\_Associated\_With\_Allele
* Abstract\_of
* access\_device\_used\_by
* access\_of
* action\_of
* active\_ingredient\_of
* active\_metabolites\_of
* Activity\_Of\_Allele
* adheres\_to
* adjacent\_to
* afferent\_to
* agent\_in
* alias\_of
* Allele\_Absent\_From\_Wild-type\_Chromosomal\_Location
* Allele\_Has\_Abnormality
* Allele\_Has\_Activity
* Allele\_In\_Chromosomal\_Location
* Allele\_of
* Allele\_Plays\_Altered\_Role\_In\_Process
* Allele\_Plays\_Role\_In\_Metabolism\_Of\_Chemical\_Or\_Drug
* allelic\_variant\_of
* Amino\_Acid\_Variant\_of
* analyzed\_by
* analyzes
* Anatomic\_Structure\_Has\_Location
* Anatomic\_Structure\_Is\_Physical\_Part\_Of
* anatomical\_site
* Anatomy\_Originated\_From\_Biological\_Process
* Aneuploidy\_Addition\_of
* Aneuploidy\_Deletion\_of
* anterior\_to
* application\_of
* Arm\_Location\_of
* Arm\_of
* arterial\_supply\_of
* articulates\_with
* associated\_disease
* associated\_finding\_of
* associated\_genetic\_condition
* associated\_morphology\_of
* associated\_procedure\_of
* associated\_with
* Associated\_With\_Malfunction\_Of\_Gene\_Product
* attaches\_to
* attributed\_constitutional\_part\_of
* attributed\_continuous\_with
* attributed\_part\_of
* attributed\_regional\_part\_of
* Author\_of
* Band\_Location\_of
* Band\_of
* bearer\_of
* Biological\_Process\_Has\_Associated\_Location
* Biological\_Process\_Has\_Initiator\_Chemical\_Or\_Drug
* Biological\_Process\_Has\_Initiator\_Process
* Biological\_Process\_Has\_Result\_Anatomy
* Biological\_Process\_Has\_Result\_Biological\_Process
* Biological\_Process\_Has\_Result\_Chemical\_Or\_Drug
* Biological\_Process\_Involves\_Chemical\_Or\_Drug
* Biological\_Process\_Involves\_Gene\_Product
* Biological\_Process\_Is\_Part\_Of\_Process
* Biological\_Process\_Results\_From\_Biological\_Process
* Biomarker\_Type\_Includes\_Gene
* Biomarker\_Type\_Includes\_Gene\_Product
* blood\_supply\_of
* bounded\_by
* bounds
* branch\_of
* branch\_part\_of
* causative\_agent\_of
* cause\_of
* cell\_connecting\_part\_of
* cell\_shape\_of
* cell\_surface\_specialization\_of
* Cell\_Type\_Is\_Associated\_With\_EO\_Disease
* Cell\_Type\_Or\_Tissue\_Affected\_By\_Chemical\_Or\_Drug
* Centromere\_of
* CH3\_Status\_of
* Chemical\_Or\_Drug\_Affects\_Abnormal\_Cell
* Chemical\_Or\_Drug\_Affects\_Cell\_Type\_Or\_Tissue
* Chemical\_Or\_Drug\_Affects\_Gene\_Product
* Chemical\_Or\_Drug\_Has\_Mechanism\_Of\_Action
* Chemical\_Or\_Drug\_Has\_Physiologic\_Effect
* Chemical\_Or\_Drug\_Initiates\_Biological\_Process
* Chemical\_Or\_Drug\_Is\_Metabolized\_By\_Enzyme
* Chemical\_Or\_Drug\_Is\_Product\_Of\_Biological\_Process
* Chemical\_Or\_Drug\_Metabolism\_Is\_Associated\_With\_Allele
* Chemical\_Or\_Drug\_Plays\_Role\_In\_Biological\_Process
* chemical\_structure\_of
* Chemotherapy\_Regimen\_Has\_Component
* Chromosomal\_Location\_of
* Chromosomal\_Location\_Of\_Allele
* Chromosomal\_Location\_of\_Wild-type\_Gene
* Chromosomal\_Structural\_Variant
* Chromosome\_Involved\_In\_Cytogenetic\_Abnormality
* Chromosome\_Mapped\_To\_Disease
* class\_code\_classified\_by
* classified\_as
* classifies
* classifies\_class\_code
* clinical\_course\_of
* common\_name\_of
* Completely\_Excised\_Anatomy\_Has\_Procedure
* Completely\_Excised\_Anatomy\_May\_Have\_Procedure
* Complex\_Has\_Physical\_Part
* component\_of
* Concept\_In\_Subset
* conceptual\_part\_of
* conjugate\_component\_of
* consider
* consider\_from
* consists\_of
* Constituent\_Amino\_Acid\_of
* Constituent\_Element\_of
* Constituent\_Protein\_of
* Constituent\_Variant\_of
* constitutes
* constitutional\_part\_of
* contained\_in
* contains
* context\_binding\_of
* continuation\_branch\_of
* continuous\_with
* continuous\_with\_distally
* continuous\_with\_proximally
* contraindicated\_with\_disease
* contraindicating\_class\_of
* contraindicating\_mechanism\_of\_action\_of
* contraindicating\_physiologic\_effect\_of
* Cytogenetic\_Abnormality\_Involves\_Chromosome
* Data\_Element\_Of
* definitional\_manifestation\_of
* degree\_of
* Deleted\_Region\_End\_Band
* Deleted\_Region\_Start\_Band
* denoted\_by
* denotes
* dependent\_of
* derivatized\_to
* derives\_from
* determines\_parameter\_for
* determines\_property
* develops\_from
* device\_used\_by
* diagnosed\_by
* diagnoses
* direct\_device\_of
* direct\_morphology\_of
* direct\_procedure\_site\_of
* direct\_substance\_of
* Disease\_Excludes\_Abnormal\_Cell
* Disease\_Excludes\_Cytogenetic\_Abnormality
* Disease\_Excludes\_Finding
* Disease\_Excludes\_Molecular\_Abnormality
* Disease\_Excludes\_Normal\_Cell\_Origin
* Disease\_Excludes\_Normal\_Tissue\_Origin
* Disease\_Excludes\_Primary\_Anatomic\_Site
* Disease\_Has\_Abnormal\_Cell
* Disease\_Has\_Accepted\_Treatment\_With\_Regimen
* Disease\_Has\_Associated\_Anatomic\_Site
* Disease\_Has\_Associated\_Disease
* Disease\_Has\_Associated\_Gene
* Disease\_Has\_Cytogenetic\_Abnormality
* Disease\_Has\_Finding
* Disease\_Has\_Metastatic\_Anatomic\_Site
* Disease\_Has\_Molecular\_Abnormality
* Disease\_Has\_Normal\_Cell\_Origin
* Disease\_Has\_Normal\_Tissue\_Origin
* Disease\_Has\_Primary\_Anatomic\_Site
* Disease\_Is\_Grade
* Disease\_Is\_Marked\_By\_Gene
* Disease\_Is\_Stage
* Disease\_Mapped\_To\_Chromosome
* Disease\_Mapped\_To\_Gene
* Disease\_May\_Have\_Abnormal\_Cell
* Disease\_May\_Have\_Associated\_Disease
* Disease\_May\_Have\_Cytogenetic\_Abnormality
* Disease\_May\_Have\_Finding
* Disease\_May\_Have\_Molecular\_Abnormality
* Disease\_May\_Have\_Normal\_Cell\_Origin
* Disease\_Pathogenesis\_Involves\_Gene
* disease\_with\_contraindication
* distal\_to
* DOI\_of
* dose\_form\_of
* doseformgroup\_of
* drains\_into
* drug\_contraindicated\_for
* due\_to
* Duplicated\_Region\_End\_Band
* Duplicated\_Region\_Start\_Band
* effect\_may\_be\_inhibited\_by
* Effect\_of
* efferent\_to
* encapsulated\_component\_of
* Encoded\_by
* Encodes
* Endogenous\_Product\_Related\_To
* energy\_used\_by
* entrapment\_site\_of
* entrapped\_component\_of
* entry\_version\_of
* Enzyme\_Metabolizes\_Chemical\_Or\_Drug
* EO\_Anatomy\_Is\_Associated\_With\_EO\_Disease
* EO\_Disease\_Has\_Associated\_Cell\_Type
* EO\_Disease\_Has\_Associated\_EO\_Anatomy
* EO\_Disease\_Has\_Property\_Or\_Attribute
* EO\_Disease\_Maps\_To\_Human\_Disease
* epithelial\_cell\_shape\_of
* evaluation\_of
* Excised\_Anatomy\_Has\_Procedure
* Excised\_Anatomy\_May\_Have\_Procedure
* exhibited\_by
* exhibits
* Exon\_of
* expanded\_form\_of
* external\_to
* fascicular\_architecture\_of
* Feature\_of
* finding\_context\_of
* finding\_informer\_of
* finding\_method\_of
* finding\_site\_of
* focus\_of
* form\_of
* function\_of
* Gene\_Associated\_With\_Disease
* Gene\_Encodes\_Gene\_Product
* Gene\_Found\_In\_Organism
* Gene\_Has\_Abnormality
* Gene\_Has\_Physical\_Location
* Gene\_In\_Chromosomal\_Location
* Gene\_Involved\_In\_Molecular\_Abnormality
* Gene\_Involved\_In\_Pathogenesis\_Of\_Disease
* Gene\_Is\_Biomarker\_Of
* Gene\_Is\_Biomarker\_Type
* Gene\_Is\_Element\_In\_Pathway
* Gene\_Location\_of
* Gene\_Mapped\_To\_Disease
* Gene\_Mutant\_Encodes\_Gene\_Product\_Sequence\_Variation
* Gene\_of
* Gene\_Plays\_Role\_In\_Process
* Gene\_Product\_Affected\_By\_Chemical\_Or\_Drug
* Gene\_Product\_Encoded\_By\_Gene
* Gene\_Product\_Expressed\_In\_Tissue
* Gene\_Product\_Has\_Abnormality
* Gene\_Product\_Has\_Associated\_Anatomy
* Gene\_Product\_Has\_Biochemical\_Function
* Gene\_Product\_Has\_Chemical\_Classification
* Gene\_Product\_Has\_Organism\_Source
* Gene\_Product\_Has\_Structural\_Domain\_Or\_Motif
* Gene\_Product\_Is\_Biomarker\_Of
* Gene\_Product\_Is\_Biomarker\_Type
* Gene\_Product\_Is\_Element\_In\_Pathway
* Gene\_Product\_Is\_Physical\_Part\_Of
* Gene\_Product\_Malfunction\_Associated\_With\_Disease
* Gene\_Product\_Plays\_Role\_In\_Biological\_Process
* Gene\_Product\_Sequence\_Variation\_Encoded\_By\_Gene\_Mutant
* Genomic\_Mutation\_Of
* germ\_origin\_of
* gives\_rise\_to
* Has\_3\_UTR
* Has\_5\_UTR
* has\_Abstract
* has\_access
* has\_action
* has\_active\_ingredient
* has\_active\_metabolites
* has\_additive
* has\_adherent
* has\_affiliation
* has\_agent
* has\_alias
* Has\_Allele
* has\_allelic\_variant
* Has\_Amino\_Acid\_Variant
* Has\_Aneuploidy\_Addition
* Has\_Aneuploidy\_Deletion
* has\_application
* Has\_Arm
* Has\_Arm\_Location
* has\_arterial\_supply
* has\_associated\_finding
* has\_associated\_morphology
* has\_associated\_procedure
* has\_atmospheric\_component
* has\_attributed\_constitutional\_part
* has\_attributed\_part
* has\_attributed\_regional\_part
* has\_Author
* Has\_Band
* Has\_Band\_Location
* has\_been\_treated
* has\_bioassay\_data
* has\_bioassays
* has\_biomaterial\_characteristics
* has\_blood\_supply
* has\_branch
* has\_branch\_part
* has\_cancer\_site
* has\_category
* has\_causative\_agent
* Has\_CDRH\_Parent
* has\_cell\_connecting\_part
* has\_cell\_shape
* has\_cell\_surface\_specialization
* Has\_Centromere
* Has\_CH3\_Status
* has\_chemical\_structure
* has\_chromosomal\_aberration\_classification
* Has\_Chromosomal\_Location
* has\_citation
* has\_clinical\_course
* has\_clinical\_finding
* has\_clinical\_record
* has\_clinical\_treatment
* has\_common\_name
* has\_component
* has\_component\_part
* has\_compound
* has\_conceptual\_part
* has\_conjugated\_component\_part
* Has\_Constituent\_Amino\_Acid
* Has\_Constituent\_Element
* Has\_Constituent\_Protein
* Has\_Constituent\_Variant
* has\_constitutional\_part
* has\_context\_binding
* has\_continuation\_branch
* has\_contraindicated\_drug
* has\_contraindicating\_class
* has\_contraindicating\_mechanism\_of\_action
* has\_contraindicating\_physiologic\_effect
* has\_cubic\_volume
* Has\_Data\_Element
* has\_database
* has\_database\_entry\_type
* has\_datum\_value
* has\_definitional\_manifestation
* has\_degree
* has\_dependent
* has\_diameter
* has\_direct\_device
* has\_direct\_morphology
* has\_direct\_procedure\_site
* has\_direct\_substance
* has\_disease\_location
* has\_disease\_staging
* has\_disease\_state
* has\_DOI
* has\_donor
* has\_dose\_form
* has\_doseformgroup
* Has\_Effect
* has\_encapsulated\_component\_part
* has\_endpoint\_of\_measurement
* has\_entrapment\_site
* has\_entrapped\_component\_part
* has\_entry\_version
* has\_epithelial\_cell\_shape
* has\_evaluation
* Has\_Exon
* has\_expanded\_form
* has\_experiment\_design
* has\_experiment\_design\_type
* has\_experiment\_factors
* has\_factor\_value
* has\_factor\_value\_ontology\_entry
* has\_family\_member
* has\_family\_relationship
* has\_fascicular\_architecture
* Has\_Feature
* has\_feature\_shape
* has\_fiducials
* has\_finding\_context
* has\_finding\_informer
* has\_finding\_method
* has\_finding\_site
* has\_focus
* has\_form
* Has\_Free\_Acid\_Or\_Base\_Form
* has\_function
* Has\_Gene
* Has\_Gene\_Location
* Has\_Gene\_Product\_Element
* Has\_Genomic\_Mutation
* has\_germ\_origin
* has\_hardware
* has\_height
* has\_host
* has\_host\_part
* has\_identification\_type
* has\_image\_format
* has\_indicator
* has\_indirect\_device
* has\_indirect\_morphology
* has\_indirect\_procedure\_site
* has\_individual
* has\_individual\_genetic\_characteristics
* has\_ingredient
* has\_ingredients
* has\_inherent\_3d\_shape
* has\_inheritance\_type
* has\_initial\_time\_point
* has\_innervation\_source
* has\_input\_participant
* has\_insertion
* has\_integral\_part
* has\_intent
* has\_interpretation
* Has\_Intron
* has\_Journal\_Name
* has\_laterality
* has\_length
* has\_location
* has\_lymphatic\_drainage
* has\_MAGE\_description
* has\_manifestation
* has\_manufacturer
* has\_mapping\_qualifier
* has\_mass
* Has\_Maternal\_Uniparental\_Disomy
* has\_maximum\_measurement
* has\_measure
* has\_measurement
* has\_measurement\_method
* has\_measurement\_type
* has\_mechanism\_of\_action
* has\_member
* has\_method
* Has\_Mode\_of\_Inheritance
* has\_multi\_level\_category
* has\_muscle\_attachment
* has\_muscle\_insertion
* has\_muscle\_origin
* has\_nerve\_supply
* Has\_NICHD\_Parent
* has\_node\_value
* has\_node\_value\_type
* has\_nodes
* Has\_Nucleotide\_Repeat
* Has\_Nucleotide\_Variant
* has\_nutrient\_component
* has\_occurrence
* has\_organism\_part
* has\_orientation
* has\_origin
* has\_output\_participant
* has\_owner
* has\_owning\_affiliate
* has\_owning\_section
* has\_owning\_subsection
* has\_parent\_organization
* has\_part
* has\_part\_modified
* has\_participant
* Has\_Paternal\_Uniparental\_Disomy
* has\_pathological\_process
* has\_performer
* has\_permuted\_term
* has\_pharmacokinetics
* Has\_Physical\_Part\_Of\_Anatomic\_Structure
* has\_physical\_state
* has\_physiologic\_effect
* has\_precise\_ingredient
* has\_primary\_segmental\_supply
* has\_print\_name
* has\_prior\_disease\_state
* has\_priority
* has\_procedure\_context
* has\_procedure\_device
* has\_procedure\_morphology
* has\_procedure\_site
* has\_product\_component
* has\_property
* has\_property\_set
* has\_protocol
* has\_providers
* has\_Publication\_Year
* has\_PubMedID
* has\_quality
* has\_quantified\_form
* has\_reason\_for\_deprecation
* has\_recipient\_category
* has\_regional\_part
* has\_result
* has\_revision\_status
* has\_role
* has\_route\_of\_administration
* Has\_RT\_Product
* Has\_Salt\_Form
* has\_scale
* has\_scale\_type
* has\_secondary\_segmental\_supply
* has\_segment
* has\_segmental\_composition
* has\_segmental\_supply
* has\_severity
* has\_shape
* has\_single\_level\_category
* has\_software
* has\_sort\_version
* has\_species
* has\_specimen
* has\_specimen\_procedure
* has\_specimen\_source\_identity
* has\_specimen\_source\_morphology
* has\_specimen\_source\_topography
* has\_specimen\_substance
* has\_subject\_relationship\_context
* Has\_Subset
* has\_supported\_concept\_property
* has\_supported\_concept\_relationship
* has\_surgical\_approach
* has\_systemic\_part
* Has\_Target
* Has\_Telomere
* has\_temperature\_condition
* has\_temporal\_context
* has\_test\_result
* has\_test\_type
* has\_therapeutic\_class
* has\_time\_period
* has\_tradename
* Has\_Transcript
* has\_treatment
* has\_tributary
* has\_type
* has\_unit
* has\_unit\_of\_measure
* has\_units
* has\_URI
* has\_venous\_drainage
* has\_version
* Human\_Disease\_Maps\_To\_EO\_Disease
* Human\_Sex\_Determinant
* icd\_dagger
* identification\_type\_of
* Imaged\_Anatomy\_Has\_Procedure
* included\_in
* includes
* indicator\_of
* indirect\_device\_of
* indirect\_morphology\_of
* indirect\_procedure\_site\_of
* induced\_by
* induces
* inferior\_to
* ingredient\_of
* ingredients\_of
* inherence\_for
* inherent\_3d\_shape\_of
* inheres\_in
* inheritance\_type\_of
* innervates
* insertion\_of
* instrument\_used\_by
* intent\_of
* internal\_to
* interpretation\_of
* interprets
* Intron\_of
* INV\_Chromosomal\_Structural\_Variant
* INV\_Deleted\_Region\_End\_Band
* INV\_Deleted\_Region\_Start\_Band
* INV\_Duplicated\_Region\_End\_Band
* INV\_Duplicated\_Region\_Start\_Band
* INV\_Human\_Sex\_Determinant
* INV\_Inverted\_Region\_End\_Band
* INV\_Inverted\_Region\_End\_Exon
* INV\_Inverted\_Region\_End\_Gene
* INV\_Inverted\_Region\_End\_UTR
* INV\_Inverted\_Region\_Start\_Band
* INV\_Inverted\_Region\_Start\_Exon
* INV\_Inverted\_Region\_Start\_Gene
* INV\_Inverted\_Region\_Start\_Intron
* INV\_Involves
* INV\_Isochromosome\_Origin
* INV\_Karyotype\_Class
* INV\_Source\_Band
* INV\_Source\_Exon
* INV\_Source\_Gene
* INV\_Source\_Intron
* INV\_Target\_Band
* INV\_Target\_Exon
* INV\_Target\_Gene
* INV\_Target\_Intron
* inverse\_has\_additive
* inverse\_has\_affiliation
* inverse\_has\_atmospheric\_component
* inverse\_has\_been\_treated
* inverse\_has\_bioassay\_data
* inverse\_has\_bioassays
* inverse\_has\_biomaterial\_characteristics
* inverse\_has\_cancer\_site
* inverse\_has\_category
* inverse\_has\_chromosomal\_aberration\_classification
* inverse\_has\_citation
* inverse\_has\_clinical\_finding
* inverse\_has\_clinical\_record
* inverse\_has\_clinical\_treatment
* inverse\_has\_compound
* inverse\_has\_cubic\_volume
* inverse\_has\_database
* inverse\_has\_database\_entry\_type
* inverse\_has\_diameter
* inverse\_has\_disease\_location
* inverse\_has\_disease\_staging
* inverse\_has\_disease\_state
* inverse\_has\_donor
* inverse\_has\_experiment\_design
* inverse\_has\_experiment\_design\_type
* inverse\_has\_experiment\_factors
* inverse\_has\_factor\_value
* inverse\_has\_factor\_value\_ontology\_entry
* inverse\_has\_family\_member
* inverse\_has\_family\_relationship
* inverse\_has\_feature\_shape
* inverse\_has\_fiducials
* inverse\_has\_hardware
* inverse\_has\_height
* inverse\_has\_host
* inverse\_has\_host\_part
* inverse\_has\_image\_format
* inverse\_has\_individual
* inverse\_has\_individual\_genetic\_characteristics
* inverse\_has\_initial\_time\_point
* inverse\_has\_length
* inverse\_has\_MAGE\_description
* inverse\_has\_manufacturer
* inverse\_has\_mass
* inverse\_has\_maximum\_measurement
* inverse\_has\_measurement\_type
* inverse\_has\_node\_value
* inverse\_has\_node\_value\_type
* inverse\_has\_nodes
* inverse\_has\_nutrient\_component
* inverse\_has\_organism\_part
* inverse\_has\_owner
* inverse\_has\_parent\_organization
* inverse\_has\_part\_modified
* inverse\_has\_performer
* inverse\_has\_prior\_disease\_state
* inverse\_has\_property\_set
* inverse\_has\_protocol
* inverse\_has\_providers
* inverse\_has\_reason\_for\_deprecation
* inverse\_has\_software
* inverse\_has\_species
* inverse\_has\_test\_result
* inverse\_has\_test\_type
* inverse\_has\_treatment
* inverse\_has\_type
* inverse\_has\_units
* inverse\_has\_URI
* inverse\_isa
* inverse\_was\_tested\_for
* Inverted\_Region\_End\_Band
* Inverted\_Region\_End\_Exon
* Inverted\_Region\_End\_Gene
* Inverted\_Region\_End\_UTR
* Inverted\_Region\_Start\_Band
* Inverted\_Region\_Start\_Exon
* Inverted\_Region\_Start\_Gene
* Inverted\_Region\_Start\_Intron
* Involves
* Is\_Abnormal\_Cell\_Of\_Disease
* Is\_Abnormality\_Of\_Gene
* Is\_Abnormality\_Of\_Gene\_Product
* is\_anatomical\_site\_of
* Is\_Associated\_Anatomic\_Site\_Of
* Is\_Associated\_Anatomy\_Of\_Gene\_Product
* Is\_Associated\_Disease\_Of
* Is\_Biochemical\_Function\_Of\_Gene\_Product
* is\_borne\_by
* Is\_Chemical\_Classification\_Of\_Gene\_Product
* Is\_Chromosomal\_Location\_Of\_Gene
* Is\_Component\_Of\_Chemotherapy\_Regimen
* Is\_Cytogenetic\_Abnormality\_Of\_Disease
* is\_datum\_of
* Is\_Finding\_Of\_Disease
* Is\_Grade\_Of\_Disease
* is\_integral\_part\_of
* is\_interpreted\_by
* Is\_Location\_Of
* Is\_Location\_Of\_Anatomic\_Structure
* Is\_Location\_Of\_Biological\_Process
* Is\_Marked\_By\_Gene\_Product
* is\_measurement\_endpoint\_of
* Is\_Mechanism\_Of\_Action\_Of\_Chemical\_Or\_Drug
* Is\_Metastatic\_Anatomic\_Site\_Of\_Disease
* Is\_Molecular\_Abnormality\_Of\_Disease
* Is\_Normal\_Cell\_Origin\_Of\_Disease
* Is\_Normal\_Tissue\_Origin\_Of\_Disease
* Is\_Not\_Abnormal\_Cell\_Of\_Disease
* Is\_Not\_Cytogenetic\_Abnormality\_Of\_Disease
* Is\_Not\_Finding\_Of\_Disease
* Is\_Not\_Molecular\_Abnormality\_Of\_Disease
* Is\_Not\_Normal\_Cell\_Origin\_Of\_Disease
* Is\_Not\_Normal\_Tissue\_Origin\_Of\_Disease
* Is\_Not\_Primary\_Anatomic\_Site\_Of\_Disease
* Is\_Organism\_Source\_Of\_Gene\_Product
* Is\_Physical\_Location\_Of\_Gene
* Is\_Physiologic\_Effect\_Of\_Chemical\_Or\_Drug
* Is\_Primary\_Anatomic\_Site\_Of\_Disease
* Is\_Property\_Or\_Attribute\_Of\_EO\_Disease
* Is\_Qualified\_By
* is\_realized\_in
* Is\_Related\_To\_Endogenous\_Product
* Is\_Stage\_Of\_Disease
* Is\_Structural\_Domain\_Or\_Motif\_Of\_Gene\_Product
* Is\_Target\_Of\_Agent
* is\_temperature\_of
* is\_time\_period\_of
* isa
* Isochromosome\_Origin
* Journal\_Name\_of
* Karyotype\_Class
* Kind\_Is\_Domain\_Of
* Kind\_Is\_Range\_Of
* larger\_than
* laterality\_of
* location\_of
* lymphatic\_drainage\_of
* manifestation\_of
* mapped\_from
* mapped\_to
* mapping\_qualifier\_of
* Maternal\_Uniparental\_Disomy\_of
* May\_Be\_Abnormal\_Cell\_Of\_Disease
* May\_Be\_Associated\_Disease\_Of\_Disease
* may\_be\_caused\_by
* May\_Be\_Cytogenetic\_Abnormality\_Of\_Disease
* may\_be\_diagnosed\_by
* May\_Be\_Finding\_Of\_Disease
* May\_Be\_Molecular\_Abnormality\_Of\_Disease
* May\_Be\_Normal\_Cell\_Origin\_Of\_Disease
* may\_be\_prevented\_by
* may\_be\_qualified\_by
* may\_be\_treated\_by
* may\_cause
* may\_diagnose
* may\_inhibit\_effect\_of
* may\_prevent
* may\_qualify
* may\_treat
* measure\_of
* measured\_by
* measurement\_method\_of
* measurement\_of
* measures
* mechanism\_of\_action\_of
* member\_of
* metabolic\_site\_of
* method\_of
* modality\_is\_related\_to
* Mode\_of\_Inheritance
* Molecular\_Abnormality\_Involves\_Gene
* mth\_british\_form\_of
* mth\_expanded\_form\_of
* mth\_has\_british\_form
* mth\_has\_expanded\_form
* mth\_has\_plain\_text\_form
* mth\_has\_xml\_form
* mth\_plain\_text\_form\_of
* mth\_xml\_form\_of
* muscle\_attachment\_of
* muscle\_insertion\_of
* muscle\_origin\_of
* Negative\_Protein\_Expression
* Negatively\_Expressed\_By
* negatively\_regulated\_by
* negatively\_regulates
* nerve\_supply\_of
* Nucleotide\_Repeat\_of
* Nucleotide\_Variant\_of
* occurs\_after
* occurs\_before
* occurs\_in
* Organism\_Has\_Gene
* orientation\_of
* origin\_of
* owning\_affiliate\_of
* owning\_section\_of
* owning\_subsection\_of
* parameter\_determined\_from
* Parent\_Is\_CDRH
* Parent\_Is\_NICHD
* part\_component\_of
* part\_of
* Partially\_Excised\_Anatomy\_Has\_Procedure
* Partially\_Excised\_Anatomy\_May\_Have\_Procedure
* participates\_in
* Paternal\_Uniparental\_Disomy\_of
* pathological\_process\_of
* Pathway\_Has\_Gene\_Element
* permuted\_term\_of
* pharmacokinetics\_of
* physical\_state\_of
* physiologic\_effect\_of
* Positive\_Protein\_Expression
* Positively\_Expressed\_By
* positively\_regulated\_by
* positively\_regulates
* posterior\_to
* precise\_ingredient\_of
* primary\_segmental\_supply\_of
* print\_name\_of
* priority\_of
* procedure\_context\_of
* procedure\_device\_of
* Procedure\_Has\_Completely\_Excised\_Anatomy
* Procedure\_Has\_Excised\_Anatomy
* Procedure\_Has\_Imaged\_Anatomy
* Procedure\_Has\_Partially\_Excised\_Anatomy
* Procedure\_Has\_Target\_Anatomy
* Procedure\_May\_Have\_Completely\_Excised\_Anatomy
* Procedure\_May\_Have\_Excised\_Anatomy
* Procedure\_May\_Have\_Partially\_Excised\_Anatomy
* procedure\_morphology\_of
* procedure\_site\_of
* Process\_Altered\_By\_Allele
* Process\_Includes\_Biological\_Process
* Process\_Initiates\_Biological\_Process
* Process\_Involves\_Gene
* product\_component\_of
* projects\_from
* projects\_to
* projects\_towards
* property\_determined\_from
* property\_of
* proximal\_to
* Publication\_Year\_of
* PubMedID\_of
* Qualifier\_Applies\_To
* quality\_of
* quantified\_form\_of
* reagent\_used\_in
* realizes
* receives\_attachment
* receives\_drainage\_from
* receives\_input\_from
* receives\_projection\_from
* recipient\_category\_of
* reformulated\_to
* reformulation\_of
* Regimen\_Has\_Accepted\_Use\_For\_Disease
* regional\_part\_of
* regulated\_by
* regulates
* related\_modality
* related\_to
* replaced\_by
* replaces
* responsive\_to\_stimulus
* result\_of
* result\_of\_regulation
* results\_in
* revision\_status\_of
* Role\_Has\_Domain
* Role\_Has\_Parent
* Role\_Has\_Range
* Role\_Is\_Parent\_Of
* role\_of
* route\_of\_administration\_of
* RT\_Product\_of
* scale\_of
* scale\_type\_of
* secondary\_segmental\_supply\_of
* see
* see\_from
* Segment\_of
* segmental\_composition\_of
* segmental\_supply\_of
* sends\_output\_to
* severity\_of
* shape\_of
* sib\_in\_branch\_of
* sib\_in\_isa
* sib\_in\_part\_of
* sib\_in\_tributary\_of
* site\_of\_metabolism
* smaller\_than
* sort\_version\_of
* Source\_Band
* Source\_Exon
* Source\_Gene
* Source\_Intron
* specifies\_value
* specimen\_of
* specimen\_procedure\_of
* specimen\_source\_identity\_of
* specimen\_source\_morphology\_of
* specimen\_source\_topography\_of
* specimen\_substance\_of
* stimulus\_causes\_response
* subject\_relationship\_context\_of
* Subset\_Includes\_Concept
* Subset\_of
* substance\_used\_by
* Subsumed\_By
* Subsumes
* superior\_to
* supported\_concept\_property\_in
* supported\_concept\_relationship\_in
* surgical\_approach\_of
* surrounded\_by
* surrounds
* systemic\_part\_of
* Target\_Anatomy\_Has\_Procedure
* Target\_Band
* Target\_Exon
* Target\_Gene
* Target\_Intron
* technique\_used\_for
* Telomere\_of
* temporal\_context\_of
* therapeutic\_class\_of
* Tissue\_Is\_Expression\_Site\_Of\_Gene\_Product
* tradename\_of
* Transcript\_of
* treated\_by
* treats
* tributary\_of
* unit\_of
* unit\_of\_measurement\_of
* use
* used\_by
* used\_for
* uses
* uses\_access\_device
* uses\_device
* uses\_energy
* uses\_instrument
* uses\_reagent
* uses\_substance
* uses\_technique
* value\_specified\_at
* venous\_drainage\_of
* version\_of
* was\_tested\_for

**ndc.html**

**Using NDC and NHRIC Codes with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The National Drug Codes (NDC) and National Health Related Items Code (NHRIC) codes are codes issued by the FDA for tracking drugs and devices. Note that the NHRIC codes are being replaced by the [UDI system](http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/UniqueDeviceIdentification/).

**Summary**

|  |  |
| --- | --- |
| Source | [National Drug Code Directory](http://www.fda.gov/Drugs/InformationOnDrugs/ucm142438.htm) and the [NHRIC Labeler Codes](http://www.fda.gov/ForIndustry/DataStandards/StructuredProductLabeling/ucm191017.htm) |
| System | The URI to identify NDC/NHRIC codes is http://hl7.org/fhir/sid/ndc |
| Version | Use YYYMMDD for the date of publication, but see note below |
| Code | The 10 digit NDC code, with "-" included. Note that different NDC codes have different positions for the "-": 1234-5678-90, 12345-6789-0, or 12345-678-90. The "-" must be correct for each NDC code |
| Display | Use the PACKAGEDESCRIPTION column value from the TSV or Excel distribution file |
| Inactive | Todo: Describe how it is determined which concepts are inactive |
| Subsumption | No Subsumption relationships are defined for the NDC codes |
| Filter Properties | None are described yet |

**Version Issues**

The FDA published list of NDC codes for [finished drug products](http://www.fda.gov/Drugs/InformationOnDrugs/ucm142438.htm) is updated daily. Use the format YYYYMMDD to refer to a particular distribution. Note that while only valid NDC codes appear in the distribution file, there are other NDC codes that organizations have assigned but not yet reported to FDA. Therefore the full set of NDCs that exists in the marketplace is unknown and cannot be versioned completely.

**Copyright/License Issues**

NDC codes have no copyright acknnowledgment or license requirements.

**NDF-RT Filter Properties**

No need for filters identified yet.

**Implicit Value Sets**

No need for implicit value sets identified yet.

**ndfrt.html**

**Using NDF-RT (the National Drug File - Reference Terminology) with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Summary**

|  |  |
| --- | --- |
| Source | the National Drug File - Reference Terminology - prepared by [Veterans Health Administration](http://www.va.gov/health/), and distributed as part of [UMLS](http://www.nlm.nih.gov/research/umls/) by the [NLM](http://www.nlm.nih.gov/) ([direct link](http://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/NDFRT/)) |
| System | The URI to identify NDF-RT is not resolved. As a temporary arrangement, the URL "http://hl7.org/fhir/ndfrt" is to be used |
| Version | A version is not needed. (Use the date of the UMLS release for the version of NDF-RT if a version is desired.) |
| Code | The NUI is used for the code value for an NDF-RT concept |
| Display | ?? |
| Inactive | Todo: Describe how it is determined which concepts are inactive |
| Subsumption | Subsumption testing is based in the Is-a relationship defined by NDFRT |
| Filter Properties | None are described yet |

*This URL is temporary while the NDF-RT and FHIR teams discuss the long term arrangements*. Further documentation can be [found in evs](http://evs.nci.nih.gov/ftp1/NDF-RT/NDF-RT%20Documentation.pdf).

**Version Issues**

NDF-RT is released as part of UMLS. Therefore each successive release has the date of the UMLS release as its version.

**Copyright/License Issues**

NDF-RT has no copyright acknowledgement required. However, users must adhere to the UMLS license.

**NDF-RT Filter Properties**

This section documents the property filters that can be used with the SNOMED CT code system in value set composition statements.

**By Subsumption**

|  |  |
| --- | --- |
| Description | Select a set of concepts based on subsumption testing |
| Property Name | concept |
| Operations Allowed | is-a |
| Values Allowed | NUI |
| Comments | Includes all concepts that have a transitive is-a relationship with the concept Id provided in the value as an NUI (including the concept itself) |

Others yet to be done.

**Implicit Value Sets**

Yet to be done.

**newfooter.html**

Ã‚Â© HL7.org 2011+. FHIR STU3 Candidate (v-) generated on . [QA Page](file:///C:\temp\qa.html)   
Links: | [Version History](file:///C:\temp\history.html) | [Table of Contents](file:///C:\temp\toc.html) | [Credits](file:///C:\temp\credits.html) | [Compare to DSTU2](http://services.w3.org/htmldiff?doc1=http%3A%2F%2Fhl7.org%2Ffhir%2FDSTU2%2F&doc2=) |  | [Propose a change](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemAdd&tracker_id=677)

**newheader.html**

**newnavbar.html**

[FHIR](file:///C:\temp\index.html)

* [Home](file:///C:\temp\index.html)
* [Getting Started](file:///C:\temp\modules.html)
* [Documentation](file:///C:\temp\documentation.html)
* [Resources](file:///C:\temp\resourcelist.html)
* [Profiles](file:///C:\temp\profilelist.html)
* [Extensions](file:///C:\temp\extensibility-registry.html)
* [Operations](file:///C:\temp\operationslist.html)
* [Services](file:///C:\temp\servicelist.html)

**ns.html**

**Namespaces defined by FHIR**

**ontology.html**

**OWL Ontology**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page and the RDF forms are particularly prone to change. The page is not part of the current ballot, and so at the most it can be a draft page in STU 3. Comments on this and the page content are welcome.

**Why an ontology**

**Overview of the ontology**

**Using the ontology**

**op-example-request.html**

**Operation Request Example**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

POST [base]/ValueSet/$expand HTTP/1.1

Content-Type: application/fhir+xml

<!--

This is an example of a operation request for a value set

expansion where the value set is submitted on the fly

-->

<Parameters xmlns="http://hl7.org/fhir">

<parameter>

<name value="filter"/>

<valueString name="abdo"/>

</parameter>

<parameter>

<name value="valueset"/>

<resource>

<ValueSet>

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml"><!-- Snipped for brevity --></div>

</text>

<identifier value="http://hl7.org/fhir/ValueSet/body-site"/>

<name value="SNOMED CT Body Structures"/>

<publisher value="FHIR Project team"/>

<telecom>

<system value="url"/>

<value value="http://hl7.org/fhir"/>

</telecom>

<description value="This value set includes all the &quot;Clinical finding&quot; SNOMED CT codes (i.e. codes

with an is-a relationship with 91723000: Anatomical structure)"/>

<status value="draft"/>

<compose>

<include><!-- all the descendants of clinical finding, not include itself -->

<system value="http://snomed.info/sct"/>

<filter><!-- todo: work this over. what this means is any concepts where they have is-a with 91723000.

how should this be done? -->

<property value="concept"/>

<op value="is-a"/>

<value value="91723000"/>

</filter>

</include>

</compose>

</ValueSet>

</resource>

</parameter>

</Parameters>

**operations.html**

**Extended Operations on the RESTful API**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The [RESTful API](file:///C:\temp\http.html) defines a set of common interactions (read, update, search, etc.) performed on a repository of typed resources. These interactions follow the RESTful paradigm of managing state by **C**reate/**R**ead/**U**pdate/**D**elete actions on a set of identified resources. While this approach solves many use cases, there is some specific functionality that can be met more efficiently using an RPC-like paradigm, where named operations are performed with inputs and outputs (**E**xecute). Operations are used (a) where the server needs to play an active role in formulating the content of the response, not merely return existing information, or (b)where the intended purpose is to cause side effects such as the modification of existing resources, or creation of new resources. This specification describes a lightweight operation framework that seamlessly extends the RESTful API.

Operations have the following general properties:

* Each operation has a name
* Each operation has a list of 'in' and 'out' parameters
* Parameters are either resources, data types or search parameters
* Operations are subject to the same security constraints and requirements as the RESTful API
* The URIs for the operation end-points are based on the existing RESTful API address scheme
* Operations may make use of the existing repository of resources in their definitions
* Operations may be performed on a specific resource, a resource type, or a whole system

**Executing an Operation**

Operations are executed using a URL derived from the FHIR endpoint, where the name of the operations is prefixed by a "dollar sign" ('$') character. For example:

POST http://fhir.someserver.org/fhir/Patient/1/$everything

When an operation is idempotent, and the parameters are all primitive data types with no extensions (as is the case with the example above), it may be invoked using GET as well.

Operations can be invoked on four types of FHIR endpoints:

* The "base" FHIR service endpoint (e.g. http://fhir.someserver.org/fhir): These are operations that operate on the full scale of the server. For example, "return me all extensions known by this server"
* A Resource type (e.g. http://fhir.someserver.org/fhir/Patient): These are operations that operate across all instances of a given resource type
* A Resource instance (e.g. http://fhir.someserver.org/fhir/Patient/1): These are operations that involve only a single instance of a Resource, like the $everything operation above does
* A specific version of a resource instance (http://fhir.someserver.org/fhir/Patient/1/\_history/4): These operations involve only a specific version of a single instance of a FHIR Resource and exists only to allow manipulation of profile and tag metadata of past versions

The body of the invocation contains a special infrastructure resource called [Parameters](file:///C:\temp\parameters.html), which represents a collection of named parameters as <key,value> pairs, where the value may be any primitive or complex datatype or even a full Resource. It may also include strings formatted as search parameter types.

Upon completion, the operation returns another Parameters resource, containing one or more output parameters. This means that a FHIR operation can take a set of zero or more parameters *in* and return a set of zero or more result parameters *out*. Both the body of the POST and the returned result are always a Resource.

Some Operations with primitive input types and a single Resource output parameter named 'return' can be invoked using a GET directly, with parameters as HTTP URL parameters. In this case, the response is simply the resource that is the return value, with no Parameters resource. These kinds of usage are discussed further below.

**FHIR defined Operations**

See [the list of defined operations](file:///C:\temp\operationslist.html).

**Implementation Defined Operations**

Implementations are able to define their own operations in addition to those defined here. Name clashes between operations defined by different implementers can be resolved by the use of the server's [Capability Statement](file:///C:\temp\capabilitystatement.html).

Also, the definition of these or additional run time operations does not prevent the use of other kinds of operations that are not dependent on and/or not integrated with the RESTful API, provided that their addressing scheme does not clash with the scheme defined here.

**Defining an Operation**

Each Operation is defined by:

* A context for the Operation - *system*, *resource type*, or *resource instance*
* A name for the Operation
* A list of parameters along with their definitions

For each parameter, the following information is needed:

* Name - the name of the parameter. For implementer convenience, the name should be a valid token (see below)
* Use - In | Out | Both
* Type - a data type or a Resource type
* Search Type - for string search parameters, what kind of search parameter they are (& and what kind of modifiers they have)
* Profile - a [StructureDefinition](file:///C:\temp\structuredefinition.html) that applies additional restrictions about the resource
* Documentation - a description of the parameter's use
* (Optional) Search Type - if the type is a string, and the parameter is being used like a search parameter, which kind of search type applies

Parameters may be nested into multi-part parameters. Each part has the same information as a parameter, except for use, which is taken from the parameter it is part of.

The resource [Operation Definition](file:///C:\temp\operationdefinition.html) is used to provide a computable definition of the Operation.

**Extending an Operation**

Implementations are able to extend an operation by defining new named parameters. Implementations can publish their own extended definitions using the [Operation Definition](file:///C:\temp\operationdefinition.html) resource, and this variant definition can use OperationDefinition.base to refer to the underlying definition.

Note that the FHIR specification will never define any parameter names starting with "x-".

**Executing an Operation Synchronously**

Operations are typically executed synchronously: a client sends a request to a server that includes the operation's *in* parameters and the server replies with the operation's *out* parameters.

The URL for an operation end-point depends on its context:

* system: the URL is [base]/$[name]
* resource type: the URL is [base]/[type]/$[name]
* resource instance: the URL is [base]/[type]/[id]/$[name]

**Operation Request**

An operation is generally invoked by performing an HTTP POST to the operation's end-point. The submitted content is the special [Parameters](file:///C:\temp\parameters.html) format (the "in" parameters) - a list of named parameters. For an example, see [the value set expansion request example](file:///C:\temp\op-example-request.html). Note that when parameters have a search type, the search modifiers are available, and are used on the parameter name in the Parameters resource (e.g. "code:in").

Note that the same arrangement as for the RESTful interface applies with respect to [content types](file:///C:\temp\http.html#mime-type).

If all the parameters for the operation are [primitive types](file:///C:\temp\datatypes.html#primitive), and the operation is *idempotent* (see [HTTP specification definition of idempotent](http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html)), the operation may be invoked by performing an HTTP GET operation where all of the values of the parameters are appended to the URL in the search portion of the URL (e.g. after the '?' character). Servers SHALL support this method of invocation. E.g.

GET [base]/ValueSet/$expand?url=http://hl7.org/fhir/ValueSet/body-sit&filter=abdo

When using the HTTP GET operation, if there is a repeating parameter for the extended operation the values for that parameter are repeated by repeating the named parameter. E.g. Observation $stats statistic parameter

GET [base]/Observation/$stats?subject=Patient/123&code=55284-4&system=http://loinc.org&duration=1&statistic=average&statistic=min&statistic=max&statistic=count

If the only parameter to a specific invocation of the operation is a resource, then the operation can also be executed by a POST with that resource as the body of the request.

Servers MAY choose to support submission of the parameters represented in [multi-part/form-data](https://www.ietf.org/rfc/rfc2388.txt) format as well, which can be useful when testing an operation using HTML forms.

**Operation Response**

If an operation succeeds, an HTTP Status success code is returned. This will usually be a 2xx code, though it may also be a 303 See Other. Other kinds of 3xx codes should be understood to indicate that the operation did not proceed, and the client will need to re-issue the operation if it can perform the redirection (e.g. may get redirected to an authentication step). User agents should note that servers may issue redirects, etc. to authenticate the client in response to an operation request. An HTTP status code of 4xx or 5xx indicates an error, and an [OperationOutcome](file:///C:\temp\operationoutcome.html) SHOULD be returned with details.

In general, an operation response uses the same [Parameters](file:///C:\temp\parameters.html) format whether there is only one or there are multiple named *out* parameters.

If there is only one *out* parameter, which is a Resource with the parameter name "return" then the parameter format is not used, and the response is simply the resource itself.

The resources that are returned by the operation may be retained and made available in the resource repository on the operation server. In that case, the server will provide the identity of the resource in the returned resources. When resources that are not persisted are returned in the response, they will have no id property.

**Executing an Operation Asynchronously**

**STU Note:** there is presently no mechanism to execute operations asynchronously in a RESTful manner. However, the [messaging page](file:///C:\temp\messaging.html#operations) describes a way to execute operations asynchronously using messages.

Provide feedback/discussion [here](http://wiki.hl7.org/index.php?title=FHIR_Asynchronous_Exchange).

**operationslist.html**

**Defined RESTful API Operations**

The [RESTful API](file:///C:\temp\http.html) defines a set of common interactions (read, update, search, etc.) performed on a repository of typed resources. For further information concerning how operations are defined and invoked, see [Extended Operations on the RESTful API](file:///C:\temp\operations.html).

This is a full list of the operations defined by this specification:

Notes:

* The special operations on the meta element also operate on previous versions of a resource (/\_history/). They are the only operations that can manipulate versions other than the "current" version.
* Implementation Guides can define additional operations

**overview-arch.html**

**FHIR Overview - Architects**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

At its core, FHIR contains two primary components:

1. Resources - a collection of information models that define the data elements, constraints and relationships for the Ã¢â‚¬Å“business objectsÃ¢â‚¬Â most relevant to healthcare. From a model-driven architecture perspective, FHIR resources are notionally equivalent to a physical model implemented in XML or JSON. See [the formal definition](file:///C:\temp\resource.html).
2. APIs Ã¢â‚¬â€œ a collection of well-defined interfaces for interoperating between two applications. Although not required, the FHIR specification targets RESTful interfaces for API implementation. See [details on FHIR RESTful interfaces](file:///C:\temp\http.html).

In the healthcare domain, the set of Ã¢â‚¬Å“business objectsÃ¢â‚¬Â is not universally defined, but there is a notional and ongoing evolutionary, consensus-based process for standardizing on a core set of common business objects including things like Ã¢â‚¬Å“a patientÃ¢â‚¬Â, Ã¢â‚¬Å“a procedureÃ¢â‚¬Â, Ã¢â‚¬Å“an observationÃ¢â‚¬Â, Ã¢â‚¬Å“an orderÃ¢â‚¬Â, etc. (see [a list of defined resources](file:///C:\temp\resourcelist.html)). The FHIR specification provides a framework for defining these healthcare business objects (Ã¢â‚¬Å“resourcesÃ¢â‚¬Â), for relating them together in a compositional manner, for implementing them in a computable form, and for sharing them across well-defined interfaces. The framework contains a verifiable and testable syntax, a set of rules and constraints, methods and interface signatures for Ã¢â‚¬Å“FHIR-awareÃ¢â‚¬Â APIs, and specifications for the implementation of a server capable of requesting and delivering FHIR business objects.

From an operational perspective, HL7Ã¢â‚¬â„¢s internal standards development and governance processes determine what constitutes a resource and which resources exist. In addition, the FHIR specification also provides a mechanism for contextualizing resources for specific needs within specific bounds (see [Profiling Resources](file:///C:\temp\profiling.html)).

**Architecture Frameworks and FHIR Alignment**

FHIR resources fit firmly within the information architecture domain and the FHIR APIs for data exchange address aspects of application architecture.

|  |  |
| --- | --- |
| From a [TOGAF perspective](http://www.opengroup.org/subjectareas/enterprise/togaf), FHIR address aspects architecture views related information model definition and data exchange in the Information Systems Architectures portion of the TOGAF Architecture Development Method. |  |
| With regards to the [Zachman Framework](https://www.zachman.com), FHIR fits within the What and the How dimensions of the Architect, Engineer and Technician Perspectives |  |
| When considering the [HL7 Services Aware Interoperability Framework (SAIF)](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=3), FHIR resources and RESTful APIs represent the Ã¢â‚¬Å“Physical ModelsÃ¢â‚¬Â and Ã¢â‚¬Å“Interface ImplementationsÃ¢â‚¬Â within the Platform Specific Specifications layer of the Information Models and the Behavioral Models Viewpoints, respectively. |  |

**FHIR and Architectural Principles**

FHIRÃ¢â‚¬â„¢s primary purpose is to address interoperability with well-structured, expressive data models and simple, efficient data exchange mechanisms. In addition, FHIR aligns to the following architectural principles:

1. Reuse and Composability Ã¢â‚¬â€œ FHIR resources are designed with the 80/20 rule in mind Ã¢â‚¬â€œ focus on the 20% of requirements that satisfy 80% of the interoperability needs. To this end, resources are designed to meet the general or common data requirements of many use cases to avoid the proliferation of numerous, overlapping and redundant resources. Extension and customizations exist (see [FHIR Profiles](file:///C:\temp\profiling.html)) to allow common, somewhat generic resources to be adopted and adapted as needed for specific use case requirements. In addition, FHIR resources are highly composable in that resources commonly refer to other resources. This further promotes reuse and allows for complex structures to be built from more atomic resources.
2. Scalability Ã¢â‚¬â€œ Aligning FHIR APIs to the REST architectural style ensure that all transactions are stateless which reduces memory usage, eliminates the needs for Ã¢â‚¬Å“stickyÃ¢â‚¬Â sessions within a server farm and therefore supports horizontal scalability.
3. Performance Ã¢â‚¬â€œ FHIR resources are lean and optimized for exchange across the network. This has the potential to improve performance in complex transactions across multiple systems connected via a shared and finite network.
4. Usability Ã¢â‚¬â€œ FHIR resources are understood by technical experts and non-technical people alike. Even if the details of XML or JSON syntax are not understood, non-technical people can view these in any browser or text reader and understand the contents within them.
5. Data Fidelity Ã¢â‚¬â€œ FHIR is strongly typed and has mechanisms built in for clinical terminology linkage and validation. In addition, XML and JSON documents can be validated syntactically as well as against a defined set of business rules. This promotes high data fidelity and goes a long way towards using FHIR to achieve semantic interoperability.
6. Implementability Ã¢â‚¬â€œ One of the driving forces for FHIR is the need to create a standard with high adoption across disparate developer communities. FHIR is easily understood and readily implemented using industry standards and common mark-up and data exchange technologies.

There are additional architecture principles related to consistency, granularity, referential integrity, and others that are not as well established or proven. See the section below on Outstanding Issues for details.

**FHIR Decomposition**

As discussed, FHIRÃ¢â‚¬â„¢s principle components are resources and RESTful APIs. However, there is more to the FHIR specification including the components depicted below.

NOTE: The term Ã¢â‚¬Å“componentÃ¢â‚¬Â is used loosely to mean a part of something and does not intend to carry the specific meanings for this term provided by rigorous ontologies, modeling frameworks, or other architectural and organizational constructs. Diagrammatically, the components below are depicted below as UML classes. This is done purely to take advantage of the semantics afforded using this notation. FHIR is neither objected oriented in its modelling approach nor are the component that make up the FHIR specification UML classes or objects in the formal sense. Likewise, the UML packages shown below are notional and used for organizational purposes only.

As shown in the diagram below, it is convenient to think of the FHIR specification as having components that address the following:

* Information Model Ã¢â‚¬â€œ the components of FHIR related to the creation of FHIR resources
* Constraints Ã¢â‚¬â€œ the components of FHIR addressing constraints and validity
* Terminology Ã¢â‚¬â€œ the components of FHIR related to clinical terminologies and ontologies
* Usage Ã¢â‚¬â€œ the component of FHIR addressing the use of FHIR in a run-time capacity

Component definitions:

* FHIR Information Model
  + Base Clases: [Element](file:///C:\temp\element.html) and [Resource](file:///C:\temp\resource.html)
  + Definitions for Base Classes: [ElementDefinition](file:///C:\temp\elementdefinition.html) and [StructureDefinition](file:///C:\temp\structuredefinition.html)
  + [Data Types](file:///C:\temp\datatypes.html)
* FHIR Conformance Model
  + [Capability Statement](file:///C:\temp\capabilitystatement.html)
  + [Profile](file:///C:\temp\profiling.html)
* FHIR Terminology
  + [Code System](file:///C:\temp\codesystem.html)
  + [Value Set](file:///C:\temp\valueset.html)
* FHIR Usage
  + [REST API](file:///C:\temp\http.html)

**Creating Resources Ã¢â‚¬â€œ Architectural Considerations**

The following list provides general guidelines that apply when FHIR resources are defined. Most of these items are not enforced programmatically requiring human due diligence and governance to ensure adherence.

* Resources should have a clear boundary; one that matches one or more logical transaction scopes
* Resources should differ from each other in meaning, not just in usage (e.g., each different way to use a lab report should not result in a different resource)
* Resources need to have a natural identity
* Most resources should be very common and used in many different business transactions
* Resources should not be specific or detailed enough to preclude support for a wide range of business transactions
* Resources should be mutually exclusive [this is a very important consideration that helps to reduce redundancy and ambiguity]
* Resources should use other resources, but they should be more than just compositions of other resources; each resource should introduce novel content
* Resources should be organized into a logical framework based on the commonality of the resource and what it links to (see resource framework below)
* Resources should be large enough to provide meaningful context; resources that contain only a few attributes are likely too small to provide meaningful business value
* Resources should reflect general usage:
* if most systems treat something as a single concept, that suggests a single resource; if most systems treat something as a distinct concepts, then that suggests multiple resources
* if two different uses of a "resource" would result in wildly different interpretations of what constitutes "core" then that suggests two resources might be appropriate.
* There is a bias towards fewer resources rather than more

**Organizing FHIR Resources**

It is impractical to model the entirety of health data in a single information model. Every modeling initiative in healthcare from HL7 version 2 message specifications to FHIR resources decomposes the healthcare domain into smaller, more manageable sub-domains or information model snippets. With FHIR, each resource is essentially a snippet of the larger healthcare information domain.

When breaking down the healthcare information model into smaller chunks (or resources for FHIR), it is important to have a framework and set of guidelines to promote consistency and integrity within the resource structures and in the way resources reference each other. The framework shown below includes health information model sub-categories organized into layers based on their degree of commonness. The layers and categories are useful for identifying which parts of healthcare information are the most common and therefore need to be the most consistently defined and tightly governed. The categories at the top layers are the most common and contain the FHIR resources that support the largest number of common healthcare transactions.

Descriptions of the layers in the framework:

1. **Foundation Reaources**: Foundation resources are the most rudimentary, foundational resources. They are often used for infrastructural tasks. Although not prohibited, they are not always referenced by other resources.
2. **Base Resources**: Layer two consists of base resources. These are often the leaf nodes of a resource graph. In other words, they are often referenced by other resources, but don't typically reference other resources themselves. These resources are typically the most commonly used, and therefore require the highest degree of consistency and architectural rigor. Governance is greatest for resources in layers one and two.
3. **Clinical Resources**: Layer 3 includes the resources that are clinical in nature but are also very common across many use cases. This includes resources for clinical observations, clinical treatment, care provision, and medications. These resources can be used by themselves, but typically build on the resources in later two. For example, an observation resource will reference the patient resource from layer two. These resources are also frequently contextualized when they are referenced by resources in layers three, four and five.
4. **Financial Resources**: Layer four is dedicated to financial resources. Logically, financial resources build on clinical and base resources. For example, a billing resource will reference clinical events and activities as well as base resources like a patient.
5. **Specialized Resources**: In layer five, we find more specialized resources for less common use cases. These resources almost always reference resources in lower layers. Given that FHIR places priority on satisfying the most common use cases, there are fewer resources in this layer.
6. **Resource Contextualization**: Layer 6 does not contain resources. However, it does extend the composition framework made up by the first five layers of resources. Layer 6 includes profiles and graphs. Profiles are used to extend, constrain, or otherwise contextualize resources for a given purpose. Graphs are compositions of resources, or webs of resource, that contain attributes of their own.

The complete set of FHIR resources organized against this framework is found on the [Resources page](file:///C:\temp\resourcelist.html).

The framework serves three primary purposes:

1. Organize resource for navigation and identification
2. Classify resources into categories based on common sense groupings or patterns describing expected structures and/or behaviors amongst resources in the same category
3. Disseminate resources across layers to stratify relative common-ness with the most common resources in the top layers

Purposes 2 and 3 set the foundation for future architectural rigor and resource governance to optimize consistency, integrity and predictability of new or refined resources in the future. The actual rules and patterns will be defined and refined in future FHIR releases. However, one general guideline to state now is that resources generally reference resources in the same layer or higher. In other words, a layer 4 resource will typically only reference resources in layers 4, 3, 2 or 1. There is nothing prohibiting a layer 4 resource from referencing a layer 5 resource, but this is not as common. Given this guideline, it is possible to identify the resources that are likely to be most common across use cases and therefore demand the highest degree of consistency and governance. Further, the framework helps identify the areas where creating new FHIR resources is the highest priority. It is generally a higher priority to create FHIR resources in the higher layers (layers 1, 2 and 3) than it is to create FHIR resources in the lower layers (layers 4 and 5) because the higher layer resources will provide the greatest value across the largest number of use cases and stakeholders. This is not to say that the business transactions needed for the higher layers are not important, itÃ¢â‚¬â„¢s just that they are not as common across the whole healthcare space.

The 6th layer of the framework are not actually resources. Profiles and Clusters are extensions of resources or resource compositions that continue the progression through the FHIR Composition Framework. The provide additional contextualization required to satisfy certain use cases.

There are several benefits expected from aligning the creation of FHIR resources to this framework, including::

* Organization and manageability of health domains - the framework provides a basis for decomposition and modularity
* Identifying commonality - the framework teases out the common areas from the less common areas
* FHIR resources prioritization - the framework provides a structure for determining priorities and delegating work
* Tiered governance levels - the framework separates the areas needing the most stringent and universal governance from those that require more context-specific governance

The framework is further elaborated in the [FHIR Resource Considerations page](http://wiki.hl7.org/index.php?title=FHIR_Resource_Considerations).

Another useful tool for visualizing how FHIR resources are organized relative to each other can be found using the Resource Reference Visualization tool on [clinFHIR](https://fhirblog.com/2016/06/14/resource-reference-visualization-in-clinfhir/).

**FHIR Implementation Ã¢â‚¬â€œ High-Level Design Considerations**

**FHIR Servers and the FHIR APIs**

A FHIR REST server is any software that implements the FHIR APIs and uses FHIR resources to exchange data. The diagram below describes the FHIR interface definitions. The methods are classified as:

* iServeInstance Ã¢â‚¬â€œ methods that perform Get, Put or Delete operations on a resource
* iServeType Ã¢â‚¬â€œ methods that get type information or metadata about resources
* iServeSystem Ã¢â‚¬â€œ methods that expose or enable system behaviors.

Additional details on the FHIR APIs can be found at [the FHIR RESTful Api](file:///C:\temp\http.html) and the [Operations Framework](file:///C:\temp\operations.html).

**FHIR Transactions**

As mentioned, FHIR resources are optimized for stateless transactions with RESTful APIs. Although this is not the only way FHIR resources can be used, these types or transactions are the only ones with defined interfaces and behaviors in the FHIR specification.

FHIR transactions follow a simple request and response transaction pattern. The request and response can be for be for a single payload or can operate as batch. The payload or a request and response consists of a header and the content of interest. See diagram below for details.

**Security**

*(section to be filled out)* (but see [Security](file:///C:\temp\security.html) in the meantime).

Example Use Cases Using FHIR

For illustrative purposes, the following diagram depicts a simple use case of a patient accessing their personal health record (portal) enabled by an underlying electronic medical record (EMR) system. The EMR plays the role of the FHIR server in this example.

The pre-conditions for this use case are:

* the EMR implements the necessary FHIR APIs
* the EMR implements the necessary authentication and authorization mechanisms
* the patient is successfully authenticated and authorized to access FHIR resources

The basic flow of the use case is that the patient registers (if required), logs in, enters search criteria to identify a patient or patients of interest (the patient is most like themselves in this use case), retrieves clinical documents for the patient and retrieves clinical resources for the patient. The use cases utilizes the GET methods on the iServeInstance interface and works with the following types of FHIR resources:

* The Patient resource
* One or more document resource(s)
* One or more clinical resource(s)

Although this example use case is very simple, more complex transactions using a combination of GETs, PUTs and DELETEs against resources and metadata can be envisioned. However, the exact details of these use cases including which methods are used, the orchestration of methods and the specific resources involved are outside the scope of the FHIR specification.

**Outstanding Issues**

* **Resource Consistency and Granularity** Ã¢â‚¬â€œ there is nothing intrinsically prohibiting one resource from duplicating the same information as another resource. Further, there is nothing prohibiting resources with the same information from defining and modeling the data elements differently. HL7 has a number of processes to ensure that resources are consistently designed, but the question is when to be consistent within the specification, and when to be consistent with the real world practices of healthcare - these are sometimes in conflict with each other. Resource granularity is a related potential problem as there are variations in the size, complexity and comprehensiveness of the existing resources.   
    
  Further, the degree to which the FHIR specification can impose consistency is limited to how much agreement can be gained across various communities. While the [Implementers Safety Check List](file:///C:\temp\safety.html) and the Considerations for [FHIR Resource Considerations](http://wiki.hl7.org/index.php?title=FHIR_Resource_Considerations) provide guidance and promote consistency, rules for achieving complete consistency of both content and granularity amongst resources are neither completely defined nor completely enforced. Considering that FHIR is still a new and emerging standard, an over-abundance of constraint and rigor has been avoided to maximize initial adoption. Further, there is a natural tension between consistency and an architectural virtue and the practicalities of supporting the real practice of health care. Considering that FHIR ultimately is a reflection of the health business processes it supports, FHIR will always carry forward some of the data discrepancies, inconsistencies and gaps that are present in the practice of healthcare across different organizations and practitioners. Nonetheless, the issues of resource consistency and granularity is a topic that gets considerable ongoing discussion, and may change as FHIR approaches a final normative standard and as FHIR adoption approaches a level where more control is warranted, or a more information/process consistency emerges in the existing healthcare systems.
* **Resource References** Ã¢â‚¬â€œ there are currently a lack of strict rules for what resources should be referenced by other resources and under what circumstance. There is potential for ambiguity, duplication, inaccurate and/or conflicting information communicated by a resource graph (a collection of linked resources). Imagine the scenario where Resource Type A (e.g., procedure) references Resource Type B (e.g., encounter) and Resource Type C (e.g., patient), and Resource Type B (e.g., encounter) also references Resource Type C (e.g., patient). In this scenario, is a reference to Resource A to Resource C meant to provide the same information as the reference from Resource B to Resource C? If so, is this duplication of information problematic? Note that this is not unique to FHIR - it is an innate property of information systems. If an actual instances of A, and the B that a references, reference different instances of Resource C (e.g. the procedure references patient X and an encounter for patient Y), how does the system know that the references are intentionally different versus an error or data anomaly? The problem is that there is limited ability to describe the intent of the reference which leads to the possibility of ambiguity and error. The [Linkage resource](file:///C:\temp\linkage.html) can be used to help with this problem, but additional capabilities may be considered in the future to allow systems to address referential integrity.
* **Conditional Semantics** Ã¢â‚¬â€œ Currently, the constraints for element definitions including things like data types, value sets, optionality and cardinality are defined at design time with limited consideration for variable run-time semantics. Imagine the scenario where the value of Data Element Y (e.g., Ã¢â‚¬Å“intolerance typeÃ¢â‚¬Â) is constrained differently depending on the value of Data Element X (e.g., Ã¢â‚¬Å“causative agentÃ¢â‚¬Â) in a given instance of a resource. For example, if the instance of an Intolerance Resource has the Ã¢â‚¬Å“intolerance typeÃ¢â‚¬Â data element populated with Ã¢â‚¬Å“food intoleranceÃ¢â‚¬Â, then Ã¢â‚¬Å“causative agentÃ¢â‚¬Â should be constrained to only valid values for this value set (e.g., valid foods instead of medications or environmental agents). Tools for addressing deep semantic consistency in this regard are only gradually developing.
* **Business Rule Enforcement and Validation** Ã¢â‚¬â€œ As governance increases and more resource rules are defined, it may be advantageous to have a resource validation tool that checks for things like resource consistency, duplication, referential integrity, circular or non-sensical references, and other defined and approved validation rules. Once rules are agreed to, this level of automation can help address the other issues outlined above. These kind of faciliities are planned for the future.

**overview-clinical.html**

**FHIR Overview - Clinicians**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

FHIR (*Fast Health Interoperability Resources*) is designed to enable the exchange of healthcare-related information. This includes clinical data as well as healthcare-related administrative, public health and research data. It covers both human and veterinary medicine and is intended to be useable world-wide in a wide variety of contexts, including in-patient, ambulatory care, acute care, long-term care, community care, allied health, etc.

The FHIR specification is targeted to individuals and organizations developing software and architecting interoperable solutions that will be using FHIR. The FHIR specification does not attempt to define good or best clinical practices, nor does it provide guidance on user interfaces or workflows. Guidance in these areas may be useful, but it is outside of FHIR's scope.

Because of FHIR's focus on implementation, many aspects of the specification deal with the technical underpinnings of the exchange of clinical information between electronic systems. This section provides an introduction to what FHIR provides and tries to highlight those portions of the specification that are likely to be of most interest to the clinical community while skipping over some of the technical minutiae of interoperability. However, clinical readers are welcome to explore some of the more technical areas if they find them of interest.

**Resources**

From a clinical perspective, the most important parts of the FHIR specification to understand are the Resources. Think of Resources as paper "forms" reflecting different types of clinical and administrative information that can be captured and shared. The FHIR specification defines a generic "form template" for each type of clinical information - so one for allergies, one for prescriptions, one for referrals, etc.

FHIR data consists of repositories containing completed "forms" (resource instances). The resource instances describe patient-related information (such as demographics, health conditions and procedures) as well as administrative information (such as practitioners, organizations and locations). Some resources are infrastructure components used to support the technical exchange of information by describing what systems are able to do, defining allowed sets of codes, etc. FHIR repositories might be elecronic health record (EHR) systems, pharmacy systems, hospital information systems (HISs), etc. Some systems, such as clinical decision support engines, may expose FHIR interfaces even though they don't actually store any patient or administrative information themselves.

Each Resource defines a small amount of highly-focused data. A single resource doesn't say very much, but a collection of Resources taken together creates a useful clinical record. Information systems map the actions that a user takes (look up patient records, make a note in their history, etc.) to operations on the relevant resources.

**Extensibility and Profiling**

The paper forms (Resources) in FHIR are somewhat generic. They have to be useable in different countries and by different types of clinicians in different contexts (human care, veterinary care, public health, research, etc.). Recognizing that a *one size fits all* approach is not appropriate in the healthcare space, FHIR provides the ability to adjust the forms (Resources) to be able to handle the needs of different implementation spaces by defining"extensions" as well as enforcing constraints. For example, a "prescription" form might have extension elements added to support tracking of restricted medications while also constraining the codes that can be used to communicate types of drugs to a particular national standard. Forms are designed in such a way that these changes can be made without changing how systems pass forms around enabling any system to consume completed forms even if they have additional elements added whether or not those additional elements are used by the receiving system.

To keep the base forms that everyone uses from being overly complex, FHIR has a rule that, in most cases, a resource will only include data elements if there's an expectation that most implementations will use that particular data element. That doesn't mean the data must always exist. For example, most systems in the world are capable of tracking "deceased date" for patient, even though that element will be blank for many patient records. On the other hand, not as many systems track hair color, so hair color would be omitted from the base form and those systems that need it (perhaps in a specialty clinical or research setting) can use a FHIR extension to capture it if needed.

To keep the number of resources reasonable, some of them are fairly broad. For example, the [Observation](file:///C:\temp\observation.html) resource is used for vital signs, lab results, psychological assessments and a variety of other things. To support setting rules for more narrow areas (e.g. "*What should I send if I want to share a blood pressure?*"), FHIR allows the creation of Profiles. There will be a great deal of clinical work involved in forming consensus around how different types of detailed clinical information should be captured and shared in particular settings. Tooling to support the creation of profiles directly by clinicians is part of the plan for FHIR, but is still in the very early stages.

**Narrative**

FHIR is intended to support sharing data in a computable manner. I.e. The information shared should be useable for computer-mediated processes such as decision support, rules triggering, trend analysis, etc. However, not every system is the same and not all systems are able to recognize all discrete data. Also, there is still considerable value in data exchange in circumstances where not all or even none of that data is captured in a discrete manner. For this reason, FHIR resources support sharing not only discrete data for computation, but also a human-readable view so that the humans on each end of a healthcare information exchange can still get a full picture of what's going on.

Narrative is expected to exist for most resource instances, although it can be omitted in a few limited circumstances. In some cases the narrative will be generated from discrete information. For example the narrative for a patient might look like this:

|  |
| --- |
| **Peter James Chalmers (OFFICIAL), Jim** **identifier**: MRN = 12345 (USUAL) **telecom**: ph: (03) 5555 6473(WORK) **gender**: MALE **birthDate**: Dec 25, 1974 **deceased**: false **address**: 534 Erewhon St PleasantVille Vic 3999 (HOME) |

In other cases, the narrative might be free-form text commentary entered directly by a practitioner such as referral letters, pathology reports, etc. Certain parts of the narrative content could also later be exposed as discrete data.

**Interfaces**

In addition to defining the "forms" for data exchange (Resources), FHIR also defines a set of interfaces by which systems actually share that information. There are four primary mechanisms or "paradigms" of exchange supported by FHIR: via a REST interface, by exchanging Documents, by sending and receiving Messages and by exposing and invoking Services.

**REST**

REST is the simplest exchange mechanism. Continuing with the "form" metaphor, a RESTful server can be thought of as a room full of filing cabinets. Within the room is a cabinet for each "type" of form (or Resource) it supports. The cabinet contains folders where each folder has a unique number and represents one particular real-world thing: one Patient, one Encounter one Medication, etc. Each folder (which represents a single Resource *instance*) contains multiple pieces of paper, with each piece of paper representing a specific "version" of that instance. Every time someone updates a record, a new piece of paper is added to the top of the file folder. To see the history of a resource, you simply have to flip through the pieces of paper in the folder.

Note that a typical medical record is generally a big "folder-of-folders" with many different types of 'forms' or 'reports' gathered together. This is convenient for someone who wants to review the whole record, but inconvenient for someone updating bits of it. There's always contention for access to it to update the right part. In the computer application, the record will be decomposed to its smallest components for management purposes, and a computer will (or should), assemble the correct bits as required, by following references that exist from one piece of information and the next.

Now picture a clerk at the front door of that room. You can pass the clerk a requisition to have them do something with the information in those file cabinets. The "clerk" and the set of requisition forms on the clerk's counter make up the FHIR restful API. With that API, you can do the following:

* *search*: Have the clerk search through the folders for one(s) that meet a set of search criteria and give you a copy of the top piece of paper in each relevant folder
* *read*: Get a copy of the top piece of paper (which is the most current version) from a specific folder in one of the cabinets
* *create*: Add a new folder to the appropriate cabinet (with a new number)
* *update*: Add a new page (version) to the contents of a specific folder
* *delete*: Remove a folder from the cabinet (or more accurately, virtually remove it by putting a sticker on it saying "do not open")
* *history*: Look at all the pages in a single folder, in a particular file cabinet or possibly in the whole room. Such a broad request would be used primarily for administrative purposes rather than clinical purposes
* *transaction*: Give the clerk a multiple pieces of paper to place in distinct folders at the same time
* *operation*: Ask the clerk to perform an action or procedure on papers from one or more of the folders - for example, averaging numbers across patients, producing a summary record, or perform a complex search just by ticking a box on a requisition saying "do that one"

EHRs and other systems may present a more sophisticated interface to their end users, but behind the scenes they're all making these same types of requests to the file clerk.

**Documents**

Documents are a familiar mechanism for sharing information in the healthcare space. They are useful whenever there's a desire to guide how a consumer of information will navigate it and there's a need to have a "frozen" view of information that can be reliably retrieved even years in the future. Examples of document-like things in healthcare include discharge summaries and lab reports.

In FHIR, there's a special resource called [Composition](file:///C:\temp\composition.html) that acts as the "cover page" for a document. It identifies the title, author date, relevant patient and the table of contents. A FHIR document can be thought of as a set of sheets (Resource *instances*) stacked together with a title page on top that's stapled together. That stapled collection can then be stored or passed around, conveying a complete set of information at once.

**Messaging**

Much healthcare information exchange happens using a messaging paradigm. In messaging, a set of information is sent from one system to another, typically triggered by an event in the sender system. For example, a patient being admitted, a lab test being ordered, a drug being administered, the clock striking 12:00 or someone pressing a button. The message serves to notify the receiver that the event occurred as well as provide details about any existing data that was modified or new data that was created. Typically receiving a message means there's an expectation that the receiving system will "do something" in response.

A message might request that a lab order be fulfilled or notify a system that two patient records have been merged or that a patient has been transferred from one bed to another. A message is similar to a document in that it collects resources together, however for a message, the "cover page" is a [MessageHeader](file:///C:\temp\messageheader.html) that acts as a requisition. And rather than using a staple, the resources are joined together with a paper-clip and there's no expectation that the receiving system will store the contents of the message exactly as received, if at all.

**Services**

Services can be thought of as a light-weight way of doing messaging. Rather than a full cover page, a small sticky note is attached to the front of a resource. And sometimes rather than sending a full piece of paper, the relevant pieces are cut out and sent as fragments. The response to a requisition is a similarly paper-clipped bundle of resource instances. Services are likely to be used for things like decision support. E.g. "*Is there a problem with prescribing medication X for patient Y?*" or "*What's the recommended care plan for a patient with conditions A, B and C?*"

**Approaching the specification**

A FHIR-based system's capabilities are defined by what the Resources can say and from a clinical perspective, these things define the clinical record:

* the kinds of Resources that are defined
* their data contents, and rules about the data such as what terminology codes are supported and/or required
* how resources reference to each other
* how you can search for information

This information can all be found in the resource definition pages. The resources most likely to be of interest can be found on the following modules:

* [Clinical Summary & Care provision](file:///C:\temp\clinicalsummary-module.html)
* [Diagnostics](file:///C:\temp\diagnostics-module.html)
* [Medications](file:///C:\temp\medications-module.html)
* [Administrative](file:///C:\temp\administration-module.html)

Instructions on how to interpret the information found on the resource pages can be found [here](file:///C:\temp\formats.html). The *Logical table* or the *UML* views are likely to be easiest to understand. Don't forget to look at the examples tab for an idea of what kind of information can be expressed. Seeing how elements are used to convey real data is often more useful than just looking at definitions. Also, look at the Profiles tab to see examples of how different resources can be constrained for use in particular contexts.

Clinician and other domain expertise and feedback is always welcome as we continue refining the FHIR specification. At the top of each resource page is a link to the home page for the work group responsible for that particular resource. If you have feedback on resource design, consider getting involved.

**overview-dev.html**

**FHIR Overview - Developers**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

FHIR (*Fast Health Interoperability Resources*) is designed to enable information exchange to support the provision of healthcare in a wide variety of settings. The specification builds on and adapts modern, widely used RESTful practices to enable the provision of integrated healthcare across a wide range of teams and organizations.

The intended scope of FHIR is broad, covering human and veterinary, clinical care, public health, clinical trials, administration and financial aspects. The standard is intended for global use and in a wide variety of architectures and scenarios.

**Framework**

FHIR is based on "Resources" which are the common building blocks for all exchanges. Resources are an instance-level representation of some kind of healthcare entity. All resources have the following features in [common](file:///C:\temp\resource.html):

* A URL that identifies the resource
* Common metadata
* A [human-readable XHTML summary](file:///C:\temp\narrative.html)
* A set of defined data elements - a different set for each type of resource
* An [extensibility framework](file:///C:\temp\extensibility.html) to support variation in healthcare

Resource instances are represented as either [XML](file:///C:\temp\xml.html), [JSON](file:///C:\temp\json.html) or [RDF](file:///C:\temp\rdf.html) and there are currently different [resource types defined](file:///C:\temp\resourcelist.html) in the FHIR specification.

This specification describes a set of resources - that is, a set of resource types that describe the set of resource instances that can actually be exchanged. The term 'Resource' is sometimes used without clarifying whether it specifically refers to types or instances - the context of sue makes this clear.

**Example Resource Instance**

This is an example of how a [patient](file:///C:\temp\patient.html) is represented as a FHIR object in [JSON](file:///C:\temp\json.html). An [XML encoding](file:///C:\temp\xml.html) is also defined in the specification.

{

"resourceType": "Patient",

"id" : "23434",

"meta" : {

"versionId" : "12",

"lastUpdated" : "2014-08-18T15:43:30Z"

}

"text": {

"status": "generated",

"div": "<!-- Snipped for Brevity -->"

},

"extension": [

{

"url": "http://example.org/consent#trials",

"valueCode": "renal"

}

],

"identifier": [

{

"use": "usual",

"label": "MRN",

"system": "http://www.goodhealth.org/identifiers/mrn",

"value": "123456"

}

],

"name": [

{

"family": [

"Levin"

],

"given": [

"Henry"

],

"suffix": [

"The 7th"

]

}

],

"gender": {

"text": "Male"

},

"birthDate": "1932-09-24",

"active": true

}

Each instance of a resource consists of:

* **resourceType** (line 2) - Required: FHIR defines many different types of resources. See [the full index](file:///C:\temp\resourcelist.html)
* **id** (line 3) - The id of this resource. Always present when a resource is exchanged, except during the create operation (below)
* **meta** (lines 4 - 7) - Usually Present: [Common use/context data to all resources](file:///C:\temp\resource.html#meta) and managed by the infrastructure. Missing if there is no metadata
* **text** (lines 8 - 11) - Recommended: XHTML that provides a [human readable representation](file:///C:\temp\narrative.html) for the resource
* **extension** (lines 12 - 17) - Optional: [Extensions](file:///C:\temp\extensibility.html) defined by the extensibility framework
* **data** (lines 18 - 43) - Optional: data elements - a different set, defined for each type of resource

Note that although this specification always shows the JSON properties in the order that they are defined, many JSON libraries order properties by other criteria.

**Interactions**

For manipulation of resources, FHIR provides a [REST API](file:///C:\temp\http.html) with a rich but simple set of interactions:

* [Create](file:///C:\temp\http.html#create) = POST https://example.com/path/{resourceType}
* [Read](file:///C:\temp\http.html#read) = GET https://example.com/path/{resourceType}/{id}
* [Update](file:///C:\temp\http.html#update) = PUT https://example.com/path/{resourceType}/{id}
* [Delete](file:///C:\temp\http.html#delete) = DELETE https://example.com/path/{resourceType}/{id}
* [Search](file:///C:\temp\http.html#search) = GET https://example.com/path/{resourceType}?search parameters...
* [History](file:///C:\temp\http.html#history) = GET https://example.com/path/{resourceType}/{id}/\_history
* [Transaction](file:///C:\temp\http.html#transaction) = POST https://example.com/path/ *(POST a transaction bundle to the system)*
* [Operation](file:///C:\temp\operations.html) = GET https://example.com/path/{resourceType}/{id}/${opname}

The FHIR specification describes other kinds of exchanges beyond this simple RESTful API, including exchange of groups of resources as [Documents](file:///C:\temp\documents.html), as [Messages](file:///C:\temp\messaging.html), and by using various types of [Services](file:///C:\temp\services.html).

**Managing Variability**

There is a wide variation between different geo-political jurisdictions and segments of the healthcare industry, and no central authority to impose common business practices. Because of this, the FHIR specification defines an [extension framework](file:///C:\temp\extensibility.html) and defines [a framework for managing variability](file:///C:\temp\profiling.html).

Another key aspect of the variability encountered in healthcare is that the same information may be represented differently and with different levels of detail, granularity and nesting by various parties across the system. For example, in some cases a blood pressure measurement may be just a simple observation, a vital sign measure, while in other cases can be a rich set of highly defined data that includes things like controlled vocabularies for posture, exercise, etc. The [resource types](file:///C:\temp\resourcelist.html) defined in this specification focus on the general, common use cases. Richer and more specific content can be supported and standardized by [defining "profiles"](file:///C:\temp\profiling.html) on the base resource types.

**Managing Versions**

FHIR Servers do not have to support versioning, though they are strongly encouraged to do so. There are three different levels of versioning support for FHIR servers:

* Versioning and .meta.version are not supported (and usually, .meta.lastUpdated is not supported either)
* Versioning and the VersionId meta-property are supported, but a history of old versions is not kept
* Versioning and the VersionId meta-property are supported, and a history of old versions is available

In addition, servers may require that [version](file:///C:\temp\http.html#concurrency) aware updates are used, to prevent over-writing changes, but this is not described on this page.

**Creating a resource**

To [create a resource](file:///C:\temp\http.html#create), send an HTTP POST request to the resource type's respective end point.

POST https://example.com/path/{resourceType}

In the example below we see the creation of a Patient.

POST {some base path}/Patient HTTP/1.1

Authorization: Bearer 37CC0B0E-C15B-4578-9AC1-D83DCED2B2F9

Accept: application/fhir+json

Content-Type: application/fhir+json

Content-Length: 1198

{

"resourceType": "Patient",

...(properties)

}

Submit a new patient to the server, and ask it to store the patient with an id of its own choice.

Notes:

* **/Patient** (line 1) - the manager for all patients - use the name of the type of resource
* **Authorization** (line 2) - see [Security for FHIR](file:///C:\temp\security.html)
* **Accept, Content-Type** (lines 3-4) - the content type for all FHIR resources as represented in JSON (or application/fhir+xml for the XML version). FHIR resources are always represented in UTF-8
* **id** - The client does not need to provide an id for a resource that is being created - the server will assign one. If the client assigns one, the server will overwrite it
* Resource Content, lines 8+ - There's no meta property at this point. The rest of the resource is the same content as shown above

**Create Response**

A response will contain an HTTP code 201 to indicate that the Resource has been created successfully. A location header indicates where the resource can be fetched in subsequent requests. The server may choose to return an [OperationOutcome](file:///C:\temp\operationoutcome.html) resource, but is not required to do so.

HTTP/1.1 201 Created

Content-Length: 161

Content-Type: application/fhir+json

Date: Mon, 18 Aug 2014 01:43:30 GMT

ETag: "1"

Location: http://example.com/Patient/347

{

"resourceType": "OperationOutcome",

"text": {

"status": "generated",

"div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">The operation was successful</div>"

}

}

Notes:

* **HTTP/1.1 201** (line 1) - the operation was successful. Note that use of [HTTP v 1.1](https://www.w3.org/Protocols/rfc2616/rfc2616.html) is strongly recommended but not required
* **ETag** (line 5) - used in the [version aware update](file:///C:\temp\http.html#update) pattern (if the server supports versioning)
* **Location** (line 6) - the id the server assigned to the resource. The id in the URL must match the id in the resource when the resource is subsequently returned
* **operationOutcome** (line 9) - OperationOutcome resources in this context have no id or meta element (they have no managed identity)

**Error response**

For a variety of reasons, servers may need to return an error. Clients should be alert to authentication related responses, but FHIR content related errors should be returned using an appropriate HTTP status code, with an [OperationOutcome](file:///C:\temp\operationoutcome.html) resource to provide additional information. Here is an example of a server rejecting a resource because of server defined business rules:

HTTP/1.1 422 Unprocessable Entity

Content-Length: 161

Content-Type: application/fhir+json

Date: Mon, 18 Aug 2014 01:43:30 GMT

{

"resourceType": "OperationOutcome",

"text": {

"status": "generated",

"div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">MRN conflict

- the MRN 123456 is already assigned to a different patient</div>"

},

}

Notes:

* The server can return additional structured information using the details of the [OperationOutcome](file:///C:\temp\operationoutcome.html)

**Read Request**

[Reading a resource](file:///C:\temp\http.html#read) is done by sending HTTP GET requests to the desired Resource Type end-point.

GET https://example.com/path/{resourceType}/{id}

Here's an example.

GET /Patient/347?\_format=xml HTTP/1.1

Host: example.com

Accept: application/fhir+xml

Cache-Control: no-cache

Notes:

* **347** (line 1) - The id of the resource that is being fetched
* **\_format=xml** (line 1) - this is another method for clients to indicate the desired response format, in addition to using the accept header, and is useful for clients that don't have access to the HTTP Headers (e.g. XSLT transforms) (see [Mime Types](file:///C:\temp\http.html#mimetypes))
* **cache control** (line 4) - Concurrency control is important, though FHIR itself says nothing about it - see <http://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html> or <https://www.mnot.net/cache_docs/>

**Read Response**

The response to a GET contains the Resource.

HTTP/1.1 200 OK

Content-Length: 729

Content-Type: application/fhir+xml

Last-Modified: Sun, 17 Aug 2014 15:43:30 GMT

ETag: "1"

<?xml version="1.0" encoding="UTF-8"?>

<Patient xmlns="http://hl7.org/fhir">

<id value="347"/>

<meta>

<versionId value="1"/>

<lastUpdated value="2014-08-17T15:43:30Z"/>

</meta>

<!-- content as shown above for patient -->

</Patient>

Notes:

* **id** (line 9) - The id of the resource. This must match the id in the read request
* **versionId** (line 11) - The current version id of the resource (if the server supports versioning). Best practice is that this value matches the ETag (see [version aware update](file:///C:\temp\http.html#update)), but clients must never assume this
* Note that servers are not required to support versioning, but are strongly encouraged to do so
* **lastUpdated** (line 12) - if present, this must match the value in the HTTP header

**Search Request**

In addition to getting single known resources it's possible to find a collection of resources by [searching the resource type end-point](file:///C:\temp\http.html#search) with a [set of criteria](file:///C:\temp\search.html) describing the set of resources that should be retrieved, and their order. The general pattern is:

GET https://example.com/path/{resourceType}?criteria

The criteria is a set of HTTP parameters that specify which resources to return. The search operation

https://example.com/base/MedicationRequest?patient=347

returns all the prescriptions for the patient created above.

**Search Response**

The response to a search request is a [Bundle](file:///C:\temp\bundle.html): a list of matching resources with some metadata:

HTTP/1.1 200 OK

Content-Length: 14523

Content-Type: application/fhir+xml

Last-Modified: Sun, 17 Aug 2014 15:49:30 GMT

{

"resourceType": "Bundle",

"type": "searchset",

"id" : "eceb4882-5c7e-4ca4-af62-995dfb8cef01"

"meta" : {

"lastUpdated" : "2014-08-19T15:49:30Z"

},

"total": "3",

"link": [

{

"relation" : "next",

"url" : "https://example.com/base/MedicationRequest?patient=347&searchId=ff15fd40-ff71-4b48-b366-09c706bed9d0&page=2"

}, {

"relation" : "self",

"url" : "https://example.com/base/MedicationRequest?patient=347"

}

],

"entry": [

{

"resource" : {

"resourceType": "MedicationRequest",

"id" : "3123",

"meta" : {

"versionId" : "1",

"lastUpdated" : "2014-08-16T05:31:17Z"

},

... content of resource ...

},

},

... 2 additional resources ....

]

}

Notes:

* **resourceType/type** (line 7/8) - the result of a search is always a bundle of type "searchset"
* **id** (line 9) - An identifier assigned to this particular bundle. The server should assign a unique id to this bundle that it will not be re-used
* **meta.lastUpdated** (line 11) - (if the server supports versioning) This should match the HTTP header, and should be the date the search was executed, or more recent, depending on how the [server handles ongoing updates](file:///C:\temp\search.html#currency). The lastUpdated data SHALL be the same or more recent than the most recent resource in the results
* **total** (line 13) - The total number of matches in the search results. Not the number of matches in this particular bundle, which may be a [paged view into the results](file:///C:\temp\http.html#search)
* **link** (line 14) - A set of named links that give related contexts to this bundle. Names defined in this specification: [first](file:///C:\temp\http.html#search), [prev](file:///C:\temp\http.html#search), [next](file:///C:\temp\http.html#search), [last](file:///C:\temp\http.html#search), [self](file:///C:\temp\http.html#search)
* **entry** (line 23) - Actual resources in this set of results
* **entry.resource.id** (line 27) - Note that in some bundles, the entry.resource.idmust be [unique in the bundle](file:///C:\temp\bundle.html#bundle-unique)
* The search operation is also able to [return additional related resources](file:///C:\temp\search.html#include) as well

**Update Request**

The client sends the server a new version of the resource to replace the existing version - it PUTs it to the location of the existing resource:

PUT https://example.com/path/{resourceType}/{id}

Note that there does not need to be a resource already existing at {id} - the server may elect to automatically create the resource at the specified address. Here is an example of updating a patient:

PUT /Patient/347 HTTP/1.1

Host: example.com

Content-Type: application/fhir+json

Content-Length: 1435

Accept: application/fhir+json

If-Match: 1

{

"resourceType": "Patient",

"id" : "347",

"meta" : {

"versionId" : "1",

"lastUpdated" : "2014-08-18T15:43:30Z"

},

...

}

Notes:

* **resourceType** (line 1) - "Patient" in the URL must match the resource type in the resource (line 9)
* **resource id** (line 1, "347") - This must match the id in the resource (line 10)
* **If-Match** (line 6) - if this is provided, it must match the value in meta.versionId (line 12), and the server must check the version integrity, or return 412 if it doesn't support versions
* **meta.lastUpdated** (line 13) - This value is ignored, and will be updated by the server (mostly, but not always, if the server does not support versioning)
* **resource content** (line 14) - Not shown here, the same as Patient above

**Update Response**

The response to an update request has metadata / status, and optionally an OperationOutcome:

HTTP/1.1 200 OK

Content-Length: 161

Content-Type: application/fhir+json

Date: Mon, 18 Aug 2014 01:43:30 GMT

ETag: "2"

Location: https://example.com/Patient/347/\_history/2

{

"resourceType": "OperationOutcome",

"text": {

"status": "generated",

"div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">The operation was successful</div>"

}

}

Notes:

* **ETag** (line 5) - This is the versionId of the new version, and is also found in the location header (line 6) (if the server supports versioning)

**Base Resource Content**

Here is an example that shows all the information found in all resources, fully populated:

{

"resourceType" : "X",

"id" : "12",

"meta" : {

"versionId" : "12",

"lastUpdated" : "2014-08-18T15:43:30Z",

"profile" : ["http://example-consortium.org/fhir/profile/patient"],

"security" : [{

"system" : "http://hl7.org/fhir/v3/ActCode",

"code" : "EMP"

}],

"tag" : [{

"system" : "http://example.com/codes/workflow",

"code" : "needs-review"

}]

},

"implicitRules" : "http://example-consortium.org/fhir/ehr-plugins",

"language" : "X"

}

Implementers notes:

* **resourceType** (line 2) - always found in every resource. In XML, this is the name of the root element for the resource
* **id** (line 3) - defined when the resource is created, and never changed. Only missing when the resource is first created
* **meta.versionId** (line 5) - changes each time any resource contents change (except for the last 3 elements in meta - profile, security and tag)
* **meta.lastUpdated** (line 6) - Changes when the versionId changes. Systems that don't support versions usually don't track lastUpdated either
* **meta.profile** (line 7) - An assertion that the content conforms to a profile. See [Extending and Restricting Resources](file:///C:\temp\profiling.html#resources) for further discussion. Can be changed as profiles and value sets change or the system rechecks conformance
* **meta.security** (lines 8 - 11) - [Security labels](file:///C:\temp\security-labels.html) applied to this resource. These tags connect resources in specific ways to the overall security policy and infrastructure. Security tags can be updated when the resource changes, or whenever the security sub-system chooses to
* **meta.tag** (lines 12 - 16) - [Tags](file:///C:\temp\resource.html#Meta) applied to this resource. Tags are used to relate resources to process and workflow. Applications are not required to consider the tags when interpreting the meaning of a resource. Tags can be [updated](file:///C:\temp\http.html#tags) without changing the resource version
* **implicitRules** (line 17) - indicates that there is a [custom agreement](file:///C:\temp\profiling.html#agreement) about how the resources are used that must be understood in order to safely process the resource. Use of this is discouraged because it restricts sharing, but sometimes necessary
* **language** (line 18) - The [base language of the resource](file:///C:\temp\narrative.html#language). The resource is allowed to have content from other languages; this is just the base, but should be the main language of the resource

The base properties of all resources are defined on the resource types [Resource](file:///C:\temp\resource.html) and [DomainResource](file:///C:\temp\domainresource.html).

Want more information?

* [Getting Started](file:///C:\temp\modules.html)
* [Resource Index](file:///C:\temp\resourcelist.html)
* [Resource Guide](file:///C:\temp\resourceguide.html)
* [Documentation Index](file:///C:\temp\documentation.html)
* [Support Links](http://wiki.hl7.org/index.php?title=FHIR_Support_Page)

**overview.html**

**FHIR Overview**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Welcome to the FHIR (Fast Healthcare Interoperability Resources) Specification, which is a standard for exchanging healthcare information electronically. This page provides an overview of the standard, and serves as a road map for first-time readers of the specification to help find your way around FHIR quickly.

**Background**

Healthcare records are increasingly becoming digitized. As patients move around the healthcare ecosystem, their electronic health records must be available, discoverable, and understandable. Further, to support automated clinical decision support and other machine-based processing, the data must also be structured and standardized. (See [Coming digital challenges in healthcare](file:///C:\temp\change.html))

[HL7](http://hl7.org) has been addressing these challenges by producing healthcare data exchange and information modeling standards for over 20 years. FHIR is a new specification based on emerging industry approaches, but informed by years of lessons around requirements, successes and challenges gained through defining and implementing [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185), [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) and the RIM, and [CDA](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7). FHIR can be used as a stand-alone data exchange standard, but can and will also be used in partnership with existing widely used standards. (See [Comparing FHIR to other HL7 standards](file:///C:\temp\comparison.html))

FHIR aims to simplify implementation without sacrificing information integrity. It leverages existing logical and theoretical models to provide a consistent, easy to implement, and rigorous mechanism for exchanging data between healthcare applications. FHIR has built-in mechanisms for traceability to the HL7 RIM and other important content models. This ensures alignment to HL7's previously defined patterns and best practices without requiring the implementer to have intimate knowledge of the RIM or any [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) derivations. (See [Comparing FHIR to other HL7 standards](file:///C:\temp\comparison.html))

**Components**

The basic building block in FHIR is a [Resource](file:///C:\temp\resource.html). All exchangeable content is defined as a resource. Resources all share the following set of characteristics:

* A common way to [define](file:///C:\temp\resource.html) and [represent](file:///C:\temp\formats.html) them, building them from [data types](file:///C:\temp\datatypes.html) that define common reusable patterns of elements
* A common set of [metadata](file:///C:\temp\resource.html#metadata)
* A [human readable](file:///C:\temp\narrative.html) part

**Approach**

**Approach to information modeling**

The philosophy behind FHIR is to build a base set of resources that, either by themselves or when combined, satisfy the majority of common use cases. FHIR resources aim to define the information contents and structure for the core information set that is shared by most implementations. There is a [built-in extension mechanism](file:///C:\temp\extensibility.html) to cover the remaining content as needed.

FHIR modeling uses a composition approach. In comparison, [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) modeling is based on "model by constraint" (see [Comparing FHIR to other HL7 standards](file:///C:\temp\comparison.html)). With FHIR, specific use cases are usually implemented by combining resources together through the use of [resource references](file:///C:\temp\references.html). Although a single resource might be useful by itself for a given use case, it is more common that resources will be combined and tailored to meet use case specific requirements. Two special kinds of resources are used to describe how resources are defined and used:

* [Capability Statement](file:///C:\temp\capabilitystatement.html) - describes the interfaces that an implementation exposes for exchange of data
* [StructureDefinition](file:///C:\temp\structuredefinition.html) - provide additional rules that serve to constrain the optionality, cardinality, terminology bindings, data types and extensions defined in the resources used by the implementation

**The Specification**

Broadly, the FHIR specification is broken up into a [set of modules](file:///C:\temp\modules.html#modules):

Resources have a wide range of uses, from pure clinical content such as [care plans](file:///C:\temp\careplan.html) and [diagnostic reports](file:///C:\temp\diagnosticreport.html) through to pure infrastructure such as [Message Header](file:///C:\temp\messageheader.html) and [Capability Statements](file:///C:\temp\capabilitystatement.html). They all share common technical characteristics (see below for a more formal definition), but they are used in totally different fashions. Note that you do not have to use REST to make use of resources.

**Where to Start**

The best place to start is to quickly read the [Resources](file:///C:\temp\resourcelist.html) list to get a sense of what resources exist and then look at the [Patient resource](file:///C:\temp\patient.html) definition to see what resource definitions look like, and then read these background pages:

* [Resource Definitions](file:///C:\temp\resource.html) - basic background to how resources are defined
* About Resources: the [Narrative](file:///C:\temp\narrative.html) they all contain, and how [Resources refer to each other](file:///C:\temp\references.html)
* [Formats](file:///C:\temp\formats.html): [XML](file:///C:\temp\xml.html) and [JSON](file:///C:\temp\json.html)
* About [Extensibility](file:///C:\temp\extensibility.html): a key way that the specification is kept simple
* If you are coming to FHIR with a background in another HL7 standard ([HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185), [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) or [CDA](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7)), [The Relationship between FHIR and other HL7 Standards](file:///C:\temp\comparison.html) may also be useful

**Header Tabs**

These header tabs found through-out the specification are important, and many readers miss them:

Resources and the [data types](file:///C:\temp\datatypes.html) that they use are presented in a concise easy to read XML-like format, but they also have detailed descriptions of their contents. In addition, most resources are mapped to several different formats, including [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185), the [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) RIM, [CDA](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7), DICOM, and others. Also, all resources come with at least one example (sometimes many more) and, where appropriate, with profiles that describe their use in specific circumstances. Finally, some resources include notes that help implementers understand the design rationale underlying them.

**Finding Additional Information and Providing Feedback**

While intended to be consumable by a variety of audiences, the FHIR specification is targeted to the implementation community - those who will actually write the software that uses the specification. To help meet the needs of the implementation community, the editors have strived to keep the specification concise to reduce the amount of reading that must be done before writing useful code. For this reason, information that is not essential to the implementation process, such as considered alternatives, points of contention, future plans, etc. have been excluded from this specification. As well, it is likely that from time-to-time, implementers will encounter situations where the specification is unclear or incomplete. Finally, there may be circumstances where the specification is broken or where a change could allow it to better meet implementer needs.

HL7 has therefore provided a number of mechanisms through which additional information about FHIR can be sought and maintained and through which support and requests for change can be made.

**Comments**

Each page includes a "comments" section at the bottom of the page where questions and discussion about that particular portion of the FHIR specification can occur. The comments will be actively monitored by FHIR editors and HL7 work groups, so questions should be answered in a fairly timely fashion. This content will occasionally be curated to ensure ongoing relevance, particularly if the specification is subsequently updated to eliminate confusion that may have spawned an initial comment.

**The FHIR Wiki**

The FHIR project team also maintains a [wiki](http://wiki.hl7.org/index.php?title=FHIR) where development processes, methodology and design decisions are documented. Implementers and others can also contribute to the wiki to provide additional guidance and supplemental information not found in the specification. Note that wiki content is not authoritative and is not relevant for determining conformance to the FHIR specification. As well, some wiki content may not be up to date with the most recent version of the FHIR specification.

A wiki page exists for each page in the FHIR specification. These pages will generally be used to capture background rationale, decision points and other information not relevant to implementers. Additional pages defined include [FHIR methodology](http://wiki.hl7.org/index.php?title=FHIR_Guide_to_Designing_Resources), use of the [FHIR design tools](http://wiki.hl7.org/index.php?title=FHIR_Guide_to_Authoring_Resources), etc. To explore the FHIR wiki, you can start at the [root page](http://wiki.hl7.org/index.php?title=FHIR)

**Formal Change requests**

Formal requests for change can be submitted [here](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemAdd&tracker_id=677). (There's a link at the bottom of each page as well.) These will be reviewed by the appropriate work group and a decision made on their incorporation into the specification, including which release (if any) they will be part of.

**Additional Information sources/Engagement Mechanisms**

In addition to the above mechanisms, HL7 provides a Stack Overflow tag, list servers and online chat system to provide various levels of implementer support and engagement. Instructions for accessing these other mechanisms (and instructions for how best to make use of them) can be found at the [Support Links](http://wiki.hl7.org/index.php?title=FHIR_Support_Page) (and also [here](http://wiki.hl7.org/index.php?title=FHIR#More_help_and_Asking_Questions)).

**page.html**

**profile.html**

See here: [this link](file:///C:\temp\structuredefinition.html).

**profilelist.html**

**Profiles defined as part of FHIR**

This specification is a common platform standard that must be [adapted to particular use cases](file:///C:\temp\profiling.html). Some particular use cases are common or important enough to be described as a part of the specification itself. These are published as a "Profile" - a group of [Structure Definitions](file:///C:\temp\structuredefinition.html) (Constraints or Extensions), [Value Sets](file:///C:\temp\valueset.html), and examples that are all defined with a common purpose. Additional profiles may be registered on the HL7 FHIR registry at <http://hl7.org/fhir/registry>

**profiling-examples.html**

**Slicing and Discriminator Examples**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Slicing Patient Contact Details**

One common use of slicing is to describe different constraints on different kinds of patient contact details. In this example, as Patient.telecom is defined as: ContactPoint [0..\*] where the ContactPoint has system, value and use.

Consider the case where the profile should say:

* There must be a single home phone number
* The patient may have a work phone number
* The patient may have an email address
* No other types of contact are allowed

An example of a patient resource that meets these rules:

<Patient>

... snip ...

<telecom>

<system value="phone" />

<use value="home" />

<value value="5551234567" />

</telecom>

<telecom>

<system value="email" />

<value value="someone@acme.org" />

</telecom>

... snip ...

</Patient>

To do this, the profile that implements these rules needs to do the following:

* On the base Patient.telecom element: define that slicing is discriminated by system and use, and that the slices are closed ("No other types of contact are allowed"). Order is left unfixed
* Then define 3 slices:
  1. home phone: fixed values for system and use, and cardinality 1..1
  2. work phone: fixed values for system and use, and cardinality 0..1
  3. email: fixed value for system, cardinality 0..0 for use, and cardinality 0..1

In a StructureDefinition, this will look like:

<!-- setting up the slicing -->

<element>

<path value="Patient.telecom"/>

<slicing>

<discriminator value="system"/>

<discriminator value="use"/>

<rules value="closed"/>

</slicing>

<!-- net cardinality rules -->

<min value="1"/>

<max value="3"/>

</element>

<!-- first slice: home phone -->

<element>

<path value="Patient.telecom"/>

<name value="HomePhone"/> <!-- mandatory - gives the slice a name -->

<min value="1"/>

<max value="1"/>

</element>

<element>

<path value="Patient.telecom.system"/>

<min value="1"/>

<fixedCode value="phone"/>

</element>

<element>

<path value="Patient.telecom.value"/>

<min value="1"/>

</element>

<element>

<path value="Patient.telecom.use"/>

<min value="1"/>

<fixedCode value="home"/>

</element>

<!-- second slice: work phone -->

<element>

<path value="Patient.telecom"/>

<name value="WorkPhone"/> <!-- mandatory - gives the slice a name -->

<min value="0"/>

<max value="1"/>

</element>

<element>

<path value="Patient.telecom.system"/>

<min value="1"/>

<fixedCode value="phone"/>

</element>

<element>

<path value="Patient.telecom.value"/>

<min value="1"/>

</element>

<element>

<path value="Patient.telecom.use"/>

<min value="1"/>

<fixedCode value="work"/>

</element>

<!-- third slice: email -->

<element>

<path value="Patient.telecom"/>

<name value="Email"/> <!-- mandatory - gives the slice a name -->

<min value="0"/>

<max value="1"/>

</element>

<element>

<path value="Patient.telecom.system"/>

<min value="1"/>

<fixedCode value="email"/>

</element>

<element>

<path value="Patient.telecom.value"/>

<min value="1"/>

</element>

<element>

<path value="Patient.telecom.use"/>

<min value="0"/>

</element>

Note: lots of definition detail has been left out, and only the parts relevant to the pattern are shown. Also, providing a fixed value makes the minimum cardinality irrelevant, but it is shown here for completeness.

This table illustrates the relationship between the instance and the ElementDefinitions:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Path** | **Name** | **Min** | **Max** | **Fixed** |
| <Patient> | Patient |  |  |  |  |
|  | Patient.telecom |  | 1 | 3 |  |
| <telecom> | Patient.telecom | HomePhone | 1 | 1 |  |
| <system value="phone" /> | Patient.telecom.system |  | 1 | 1 | phone |
| <value value="5551234567" /> | Patient.telecom.value |  | 1 | 1 |  |
| <use value="home" /> | Patient.telecom.use |  | 1 | 1 | home |
| </telecom> |  |  |  |  |  |
|  | Patient.telecom | WorkPhone | 0 | 1 |  |
| <telecom> | Patient.telecom | Email | 0 | 1 |  |
| <system value="email" /> | Patient.telecom.system |  | 1 | 1 | email |
| <value value="someone@acme.org" /> | Patient.telecom.value |  | 1 | 1 |  |
| </telecom> | Patient.telecom.use |  |  |  |  |

**Blood Pressure Example**

Another use of Slicing is for Blood Pressure Measurements, where the profile says:

* There must be two components
* The first has LOINC code 8480-6, and a quantity
* The second has LOINC code 8462-4, and a quantity
* Other components are allowed (posture, etc., but not profiled in the base blood pressure profile)

An example of an Observation resource that meets these rules:

<Observation>

... snip ...

<component>

<code>

<coding>

<system value="http://loinc.org" />

<code value="8480-6" />

<display value="Systolic blood pressure" />

</coding>

</code>

<valueQuantity>

<value value="120" />

<unit value="mmHg" />

<system value="http://unitsofmeasure.org" />

<code value="mm[Hg]" />

</valueQuantity>

</component>

<component>

<code>

<coding>

<system value="http://loinc.org" />

<code value="8462-4" />

<display value="Diastolic blood pressure" />

</coding>

</code>

<valueQuantity>

<value value="80" />

<unit value="mmHg" />

<system value="http://unitsofmeasure.org" />

<code value="mm[Hg]" />

</valueQuantity>

</component>

</Patient>

To do this, the profile that implements these rules needs to do the following:

* On the base Observation.component element: define that slicing is discriminated by code. Order is left unfixed, and rules left open.
* Then define 2 slices:
  1. systolic: fixed values for code, cardinality 1..1, value is a Quantity
  2. diastolic: fixed values for code, cardinality 1..1, value is a Quantity

In a StructureDefinition, this will look like:

<!-- setting up the slicing -->

<element>

<path value="Observation.component"/>

<slicing>

<discriminator value="code"/>

</slicing>

<!-- net cardinality rules -->

<min value="2"/>

<max value="\*"/>

</element>

<!-- first slice: systolic -->

<element>

<path value="Observation.component"/>

<name value="systolic"/> <!-- mandatory - gives the slice a name -->

<min value="1"/>

<max value="1"/>

</element>

<element>

<path value="Observation.component.code"/>

<min value="1"/>

<fixedCodeableConcept>

<coding>

<system value="http://loinc.org" />

<code value="8480-6" />

<display value="Systolic blood pressure" />

</coding>

</fixedCodeableConcept>

</element>

<element>

<path value="Observation.component.valueQuantity"/>

<min value="1"/>

</element>

<!-- second slice: diastolic -->

<element>

<path value="Observation.component"/>

<name value="diastolic"/> <!-- mandatory - gives the slice a name -->

<min value="1"/>

<max value="1"/>

</element>

<element>

<path value="Observation.component.code"/>

<min value="1"/>

<fixedCodeableConcept>

<coding>

<system value="http://loinc.org" />

<code value="8462-4" />

<display value="Diastolic blood pressure" />

</coding>

</fixedCodeableConcept>

</element>

<element>

<path value="Observation.component.valueQuantity"/>

<min value="1"/>

</element>

Note: lots of definition detail has been left out, and only the parts relevant to the pattern are shown. E.g. A real blood pressure profile would fix unit, a overall Observation code etc.

**Extensions**

For another example, consider slicing extensions. The base extension on every element is defined as a list (0..\*) of extensions, and each extension has a url that identifies it, and a value. Consider an example where a profile defines that for a particular element (named Patient), there are two extensions, with URLs http://acme.com/a and http://acme.com/b. In addition, the profile allows other extensions to be used.

Technically, the profile achieves this by "slicing" the extension list, into two slices, and saying that the slicing is "open" - that there can be other slices introduced. Here are the relevant parts of the Profile on patient:

<StructureDefinition xmlns="http://hl7.org/fhir">

<!-- snip -->

<baseType value="Patient" />

<baseDefinition value="http://hl7.org.fhir/StructureDefinition/Patient" />

<derivation value="constraint" />

<snapshot>

<element>

<path value="Patient"/>

<!-- snip -->

</element>

<element>

<path value="Patient.extension"/>

<!-- this first element defines the slicing, and carries the base definition forward -->

<slicing>

<discriminator value="url"/> <!-- Extensions are always discriminated by URL -->

<ordered value="false"/> <!-- we don't care what order they appear in -->

<rules value="open"/> <!-- other extensions can be used -->

</slicing>

<!-- -- snip definition -->

</element>

<!-- first extension -->

<element>

<path value="Patient.extension"/>

<!-- snip most of definition -->

<type>

<code value="Extension"/>

<!-- the profile for an extension is a reference to the extension definition itself -

this implies a profile, and happens to fix the @url value to the desired URL -->

<profile value="http://acme.com/a"/>

</type>

</element>

<!-- second extension -->

<element>

<path value="Patient.extension"/>

<!-- snip most of definition -->

<type>

<code value="Extension"/>

<!-- the profile for an extension is a reference to the extension definition itself -

this implies a profile, and happens to fix the @url value to the desired URL -->

<profile value="http://acme.com/b"/>

</type>

</element>

<!-- snip rest of profile -->

</snapshot>

</StructureDefinition>

Here's a patient example that conforms to this profile:

<Patient xmlns="http://hl7.org/fhir">

<!-- two extensions, the order doesn't matter -->

<extension url="http://acme.com/b">

<!-- this has the right url, and so matches the second slice -->

<!-- snip whatever value extension would have -->

</extension>

<extension url="http://acme.com/a">

<!-- this has the right url, and so matches the first slice -->

<!-- snip whatever value extension would have -->

</extension>

<!-- the rest of patient -->

</Patient>

**Diagnostic Report & Observation**

In this example, a profile on an diagnostic report says that it must have 4 observations, each with a different LOINC code (e.g. a classic lab panel). In this case (taken from the [Example Lipid Profile](file:///C:\temp\lipid-report.html)), the structure that applies to DiagnosticReport will say that there are 4 slices on DiagnosticReport.result, each conforming to a different structure, which are also contained in the same profile. Each of those structures will constrain the LOINC code in the observation.

<!-- first structure, the DiagnosticReport -->

<StructureDefinition xmlns="http://hl7.org/fhir">

<!-- snip -->

<url value="http://acme.org/fhir/StructureDefinition/lipid-report"/>

<name value="LipidProfile"/>

<baseType value="DiagnosticReport"/>

<baseDefinition value="http://hl7.org/fhir/StructureDefinition/DiagnosticReport"/>

<derivation value="constraint" />

<!-- snip -->

<snapshot>

<!-- snip elements -->

<element>

<!-- first definition for result -->

<path value="DiagnosticReport.result"/>

<slicing>

<!-- this is sliced by the code value of the target of the reference -->

<discriminator value="reference.code"/>

<!-- have to be in the specified order -->

<ordered value="true"/>

<!-- this profile says, no other observations allowed -->

<rules value="closed"/>

</slicing>

<!-- snip definition -->

</element>

<!-- first slice: Cholesterol -->

<element>

<path value="DiagnosticReport.result"/>

<name value="Cholesterol"/>

<!-- snip definition parts -->

<type>

<code value="Reference"/>

<!-- this element must conform to the "Cholesterol" structure -->

<profile value="http://acme.org/fhir/StructureDefinition/Cholesterol"/>

</type>

</element>

<!-- next 3 slices all the same, but different names for profile -->

<element>

<path value="DiagnosticReport.result"/>

<name value="Triglyceride"/>

<!-- snip definition parts -->

<type>

<code value="Reference"/>

<!-- this element must conform to the "Triglyceride" structure -->

<profile value="http://acme.org/fhir/StructureDefinition/Triglyceride"/>

</type>

</element>

<element>

<path value="DiagnosticReport.result"/>

<name value="LDLCholesterol"/>

<!-- snip definition parts -->

<type>

<code value="Reference"/>

<!-- this element must conform to the "LDLCholesterol" structure -->

<profile value="http://acme.org/fhir/StructureDefinition/LDLCholesterol"/>

</type>

</element>

<element>

<path value="DiagnosticReport.result"/>

<name value="HDLCholesterol"/>

<!-- snip definition parts -->

<type>

<code value="Reference"/>

<!-- this element must conform to the "HDLCholesterol" structure -->

<profile value="http://acme.org/fhir/StructureDefinition/HDLCholesterol"/>

</type>

</element>

<!-- snip elements -->

</snapshot>

</StructureDefinition>

<!-- now, the second structure, for the Cholesterol profile -->

<StructureDefinition>

<url value="http://acme.org/fhir/StructureDefinition/Cholesterol"/>

<name value="Cholesterol"/>

<baseType value="Observation"/>

<baseDefinition value="http://hl7.org/fhir/StructureDefinition/Observation"/>

<derivation value="constraint" />

<snapshot>

<!-- snip elements -->

<element>

<!-- this the element definition for name. Because of the

slicing / discriminator rules in the LipidReport profile

that references it, it is required to fix the value of

the name element -->

<path value="Observation.code"/>

<definition>

<!-- there are actually 3 ways to fix a CodeableConcept

to a single fixed value. Here, we used the simplest one -->

<valueCodeableConcept>

<!-- just fix the value to the right code -->

<coding>

<system value="http://loinc.org"/>

<code value="35200-5"/>

<display value="Cholesterol"/>

</coding>

</valueCodeableConcept>

</definition>

</element>

<!-- snip elements -->

</snapshot>

</StructureDefinition>

<!-- Triglyceride profile -->

<StructureDefinition>

<url value="http://acme.org/fhir/StructureDefinition/Triglyceride"/>

<name value="Triglyceride"/>

<baseType value="Observation"/>

<baseDefinition value="http://hl7.org/fhir/StructureDefinition/Observation"/>

<derivation value="constraint" />

<snapshot>

<!-- snip elements -->

<element>

<!-- this the element definition for name. Because of the

slicing / discriminator rules in the LipidReport profile

that references it, it is required to fix the value of

the name element -->

<path value="Observation.code"/>

<definition>

<!-- there's actually 3 ways to fix a CodeableConcept

to a single fixed value. Here, we used the simplest one -->

<valueCodeableConcept>

<!-- just fix the value to the right code -->

<coding>

<system value="http://loinc.org"/>

<code value="35217-9"/>

<display value="Triglyceride"/>

</coding>

</valueCodeableConcept>

</definition>

</element>

<!-- snip elements -->

</snapshot>

</StructureDefinition>

<!-- LDLCholesterol profile -->

<StructureDefinition>

<url value="http://acme.org/fhir/StructureDefinition/LDLCholesterol"/>

<name value="LDLCholesterol"/>

<baseType value="Observation"/>

<baseDefinition value="http://hl7.org/fhir/StructureDefinition/Observation"/>

<derivation value="constraint" />

<snapshot>

<!-- snip elements -->

<element>

<!-- this the element definition for name. Because of the

slicing / discriminator rules in the LipidReport profile

that references it, it is required to fix the value of

the name element -->

<path value="Observation.code"/>

<definition>

<!-- because of the way that LDL cholesterol measurements works

(well, in this context- it varies), there's 2 different LOINC

codes for either measured or calculated. So here, we bind to

a value set -->

<binding>

<conformance value="required"/> <!-- must be required if this is a discriminator -->

<!-- snip the actual value set reference, but it refers to a value

set with two LOINC codes, one for each kind of LDL, which in

this case are LOINC codes 18262-6 and 13457-7 -->

</binding>

</definition>

</element>

<!-- snip elements -->

</snapshot>

</StructureDefinition>

<!-- HDLCholesterol profile -->

<StructureDefinition>

<url value="http://acme.org/fhir/StructureDefinition/HDLCholesterol"/>

<name value="HDLCholesterol"/>

<baseType value="Observation"/>

<baseDefinition value="http://hl7.org/fhir/StructureDefinition/Observation"/>

<derivation value="constraint" />

<snapshot>

<!-- snip elements -->

<element>

<!-- this the element definition for name. Because of the

slicing / discriminator rules in the LipidReport profile

that references it, it is required to fix the value of

the name element -->

<path value="Observation.code"/>

<definition>

<!-- there's actually 3 ways to fix a CodeableConcept

to a single fixed value. Here, we used the simplest one -->

<valueCodeableConcept>

<!-- just fix the value to the right code -->

<coding>

<system value="http://loinc.org"/>

<code value="2085-9"/>

<display value="LDL Cholesterol"/>

</coding>

</valueCodeableConcept>

</definition>

</element>

<!-- snip elements -->

</snapshot>

</StructureDefinition>

Here is an instance that meets the rules for this profile:

<!-- first, the diagnostic report -->

<DiagnosticReport xmlns="http://hl7.org/fhir">

<!-- snip -->

<!-- here's the set of results. We don't know what

slices they are or anything until we go off, find

the references, and look in them -->

<result>

<reference value="Observation/cholesterol"/>

</result>

<result>

<reference value="Observation/triglyceride"/>

</result>

<result>

<reference value="Observation/ldlcholesterol"/>

</result>

<result>

<reference value="Observation/hdlcholesterol"/>

</result>

<!-- snip -->

</DiagnosticReport>

<!-- Observation, id = cholesterol -->

<Observation xmlns="http://hl7.org/fhir">

<!-- the observation starts with the name, as specified

by the profile for the first slice -->

<code>

<coding>

<system value="http://loinc.org"/>

<code value="35200-5"/>

<display value="Cholesterol"/>

</coding>

</code>

<!-- snip -->

</Observation>

<!-- Observation, id = triglyceride -->

<!-- this code matches the second slice. good -->

<Observation xmlns="http://hl7.org/fhir">

<code>

<coding>

<system value="http://loinc.org"/>

<code value="35217-9"/>

<display value="Triglyceride"/>

</coding>

</code>

<!-- snip -->

</Observation>

<!-- Observation, id = hdlcholesterol -->

<!-- this code matches the fourth slice. good -->

<Observation xmlns="http://hl7.org/fhir">

<code>

<coding>

<system value="http://loinc.org"/>

<code value="2085-9"/>

<display value="HDL Cholesterol"/>

</coding>

</code>

<!-- snip -->

</Observation>

<!-- Observation, id = ldlcholesterol -->

<!-- this code matches the third slice. good -->

<Observation id="ldlcholesterol">

<code>

<coding>

<system value="http://loinc.org"/>

<code value="13457-7"/>

<display value="LDL Chol. (Calc)"/>

</coding>

</code>

<!-- snip -->

</Observation>

Note that this version isn't valid, because the slices are not in the correct order:

<!-- first, the diagnostic report -->

<DiagnosticReport xmlns="http://hl7.org/fhir">

<!-- snip -->

<!-- here's the set of results. We don't know what

slices they are or anything until we go off, find

the references, and look in them -->

<result>

<reference value="Observation/cholesterol"/>

</result>

<result>

<reference value="Observation/triglyceride"/>

</result>

<result>

<reference value="Observation/hdlcholesterol"/>

</result>

<result>

<reference value="Observation/ldlcholesterol"/>

</result>

<!-- snip -->

</DiagnosticReport>

**Composition Sections**

Most uses of Composition involve conformance to a profile that specifies which sections will exist, and what their contents will be. This is yet another example of slicing. A typical document content profile might specify a section structure something like this:

* Reason for visit Narrative, LOINC Code 29299-5
* Medications, LOINC Code 46057-6
  + Prescribed Medications, LOINC Code 66149-6
  + OTC medications, 66150-4 (optional)
* Vital Signs, LOINC Code 8716-3

Real profiles will contain lots of detail about the sections, but these are omitted here in the interests of clarity.

An example of a Composition that meets these rules:

<Composition>

... snip ...

<section>

<code>

<coding>

<system value="http://loinc.org" />

<use value="29299-5" />

<value value="Reason for visit Narrative" />

</coding>

</code>

... snip ...

</section>

<section>

<code>

<coding>

<system value="http://loinc.org" />

<use value="46057-6" />

<value value="Medications section" />

</coding>

</code>

... snip ...

<section>

<code>

<coding>

<system value="http://loinc.org" />

<use value="66149-6" />

<value value="Prescribed medications" />

</coding>

</code>

... snip ...

</section>

<section>

<code>

<coding>

<system value="http://loinc.org" />

<use value="66150-4" />

<value value="Over the counter medications" />

</coding>

</code>

... snip ...

</section>

</section>

<section>

<code>

<coding>

<system value="http://loinc.org" />

<use value="8716-3" />

<value value="Vital signs" />

</coding>

</code>

... snip ...

</section>

</Composition>

To do this, the profile that implements these rules needs to do the following:

* On the base Composition.section element: define that slicing is discriminated by code, and that the slices are closed and ordered
* Then define 3 slices on Composition.section:
  1. reason-for-visit: fixed code, cardinality 1..1
  2. medication: fixed code, cardinality 1..1
  3. vital-signs: fixed code, cardinality 1..1
* Then, in the medication slice, slice the Composition.section.section: define that slicing is discriminated by code, and that the slices are closed and ordered
* Then define 2 slices on the Composition.section.section in medication:
  1. prescribed: fixed code, cardinality 1..1
  2. otc: fixed code, cardinality 0..1

In a StructureDefinition, this will look like:

<!-- setting up the slicing on Composition.section -->

<element>

<path value="Composition.section"/>

<slicing>

<discriminator value="code"/>

<ordered value="true"/>

<rules value="closed"/>

</slicing>

<!-- net cardinality rules -->

<min value="3"/>

<max value="3"/>

</element>

<!-- first slice: reason for visit -->

<element>

<path value="Composition.section"/>

<name value="reason-for-visit"/>

<min value="1"/>

<max value="1"/>

</element>

<element>

<path value="Composition.section.code"/>

<min value="1"/>

<fixedCodeableConcept>

<coding>

<system value="http://loinc.org" />

<use value="29299-5" />

<value value="Reason for visit Narrative" />

</coding>

</fixedCodeableConcept>

</element>

<!-- second slice: medications -->

<element>

<path value="Composition.section"/>

<name value="medications"/>

<min value="1"/>

<max value="1"/>

</element>

<element>

<path value="Composition.section.code"/>

<min value="1"/>

<fixedCodeableConcept>

<coding>

<system value="http://loinc.org" />

<use value="46057-6" />

<value value="Medications section" />

</coding>

</fixedCodeableConcept>

</element>

<!-- setting up the inner slicing on medication Composition.section.section -->

<element>

<path value="Composition.section.section"/>

<slicing>

<discriminator value="code"/>

<ordered value="true"/>

<rules value="closed"/>

</slicing>

<!-- net cardinality rules -->

<min value="1"/>

<max value="2"/>

</element>

<!-- first inner slice: prescribed medications -->

<element>

<path value="Composition.section.section"/>

<name value="prescribed"/>

<min value="1"/>

<max value="1"/>

</element>

<element>

<path value="Composition.section.section.code"/>

<min value="1"/>

<fixedCodeableConcept>

<coding>

<system value="http://loinc.org" />

<use value="66149-6" />

<value value="Prescribed medications" />

</coding>

</fixedCodeableConcept>

</element>

<!-- second inner slice: over the counter medications -->

<element>

<path value="Composition.section.section"/>

<name value="otc"/>

<min value="0"/>

<max value="1"/>

</element>

<element>

<path value="Composition.section.section.code"/>

<min value="1"/>

<fixedCodeableConcept>

<coding>

<system value="http://loinc.org" />

<use value="66150-4" />

<value value="Over the counter medications" />

</coding>

</fixedCodeableConcept>

</element>

<!-- third slice: Vital Signs -->

<element>

<path value="Composition.section"/>

<name value="vital-signs"/>

<min value="1"/>

<max value="1"/>

</element>

<element>

<path value="Composition.section.code"/>

<min value="1"/>

<fixedCodeableConcept>

<coding>

<system value="http://loinc.org" />

<use value="8716-3" />

<value value="Vital signs" />

</coding>

</fixedCodeableConcept>

</element>

**profiling.html**

**Profiling FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The base FHIR specification (this specification) describes a set of base resources, frameworks and APIs that are used in many different contexts in healthcare. However there is wide variability between jurisdictions and across the healthcare ecosystem around practices, requirements, regulations, education and what actions are feasible and/or beneficial.

For this reason, the FHIR specification is a "platform specification" - it creates a common platform or foundation on which a variety of different solutions are implemented. As a consequence, this specification usually requires further adaptation to particular contexts of use. Typically, these adaptations specify:

* Rules about which resource elements are or are not used, and what additional elements are added that are not part of the base specification
* Rules about which API features are used, and how
* Rules about which terminologies are used in particular elements
* Descriptions of how the Resource elements and API features map to local requirements and/or implementations

Note that because of the nature of the healthcare ecosystem, there may be multiple overlapping sets of adaptations - by healthcare domain, by country, by institution, and/or by vendor/implementation.

**Glossary**

FHIR defines a cascade of artifacts for this purpose:

|  |  |  |
| --- | --- | --- |
| **Artifact** | **Description** | [**DAF**](http://hl7.org/fhir/daf)**example** |
| Implementation Guide (IG) | A coherent and bounded set of adaptations that are published as a single unit. Validation occurs within the context of the Implementation Guide | [DAF IG](http://hl7.org/fhir/daf) |
| Package | A group of related adaptations that are published as a group within an Implementation Guide | [DAF Medication Usage](http://hl7.org/fhir/daf/daf-medicationusage.html) |
| Conformance Resource | A single resource in a package that makes rules about how an implementation works. These are described below | [DAF Prescription](http://hl7.org/fhir/daf/medicationorder-daf.html) |

The term "profile" is a general one that is used about either a "package" or an "item". "Profiling" is a general term that describes the process of creating an implementation guide, or any of the conformance resources found in one.

**Conformance Resources**

Typically, Implementation Guides both restrict and extend APIs, resources and terminologies. FHIR provides a set of resources that can be used to represent and share the decisions that have been made, and allows implementers to build useful services from them. These resources are known as the conformance resources. These conformance resources allow implementers to:

* Indicate that [some API calls](file:///C:\temp\http.html) are not used for a particular situation, and provide additional details about how API calls are used ([Capability Statement](file:///C:\temp\capabilitystatement.html))
* Add additional [operations](file:///C:\temp\operations.html) or [search parameters](file:///C:\temp\search.html) not in the base specification (using the [OperationDefinition](file:///C:\temp\operationdefinition.html) resource or the [SearchParameter](file:///C:\temp\searchparameter.html) Resource
* Define how a particular structure (Resource, Extension or Data Type) is used ([StructureDefinition](file:///C:\temp\structuredefinition.html) Resource):
  + Describe how existing elements in resources are used
  + Identify existing elements that are not used
  + Define extensions that can be used in resources or data types
* Mix custom and standard terminologies and choose which codes from these to use for a particular coded element([Value Set](file:///C:\temp\valueset.html) and StructureDefinition Resources)
* Map between local and standard terminologies or content models ([Concept Map](file:///C:\temp\conceptmap.html) Resource)
* Register system namespaces for identifiers and terminologies ([NamingSystem](file:///C:\temp\namingsystem.html) Resource)
* Describe and register in a Data Dictionary specific Data Elements that are used across systems ([DataElement](file:///C:\temp\dataelement.html) Resource)

These resources need to be used as discussed below, and also following the basic concepts for extension that are described in ["Extensibility"](file:///C:\temp\extensibility.html). For implementer convenience, the specification itself publishes its base definitions using these same resources.

**Two uses of Profiles**

The [CapabilityStatement](file:///C:\temp\capabilitystatement.html) resource describes two different uses for profiles on resources: Resource Profiles and System Profiles. Resource Profiles are specified using the *CapabilityStatement.rest.resource.profile* element and System Profiles are specified using the *CapabilityStatement.profile*element.

**CapabilityStatement.rest.resource.profile**

These profiles describe the general features that are supported by the system for each kind of resource. Typically, this is the superset of all the different use-cases implemented by the system. This is a resource-level perspective of a system's functionality.

**CapabilityStatement.profile**

These profiles describe the information handled/produced by the system on a per use case basis. Some examples of the uses for these kind of profiles:

* A Laboratory service producing a set of different reports - general chemistry, blood count, etc. Typical labs would support several hundred different reports
* A care manager which handles a set of different types of care plans and associated clinical resources
* A medications formulary that handles several different levels of sophistication in its medication representations

Typically, these profiles are a series of variations on the same set of resources - different use cases leading to handling the resources that represent them differently. These usecases described above all pertain to system that produce and publish data, but the same concept applies to systems that consume data. For instance:

* An expert service that provides analysis on several different sets of data conforming to a particular pattern - tests x,y and z with particular codes and units

For producer and a consumer systems to exchange data successfully based on one of these system supported profiles, it's not enough to know that the systems happen to have system profiles that overlap for the use case of interest; the consumer must be able to filter the total set of resources made available by the producer system and deal only with the ones relevant to the use case.

As an example consider a laboratory system generating thousands of reports a day. 1% of those reports are a particular endocrine report that a decision support system knows how to process. Both systems declare that they support the particular endocrine report profile, but how does the expert system actually find the endocrine reports that it knows how to process?

One possible option is for the expert system to receive every single report coming from the lab system, check whether it conforms to the profile or not, and then decide whether to process it. Checking whether a resource conforms to a particular profile or not is a straight forward operation (one option is to use the [provided tools for this](file:///C:\temp\downloads.html)), but this is very inefficient way - the expert system has to receive and process 100 times many resources as it uses. To help a consumer find the correct set of reports for a use-case, a producer of resources also SHALL, for any profile declared in Conformance.profile:

1. [Mark resources](file:///C:\temp\resource.html#meta) with profile assertions documenting the profile(s) they conform to (this enables indexing by the profile)
2. (if a server) support searching by the [\_profile parameter](file:///C:\temp\search.html#profile) for the declared profiles

Beyond these requirements, a producer of resources SHOULD ensure that any resource instance that would reasonably be expected to conform to the declared profiles SHOULD be published in this form.

**STU Note:** there are many uninvestigated issues associated with this use of profiles. HL7 is actively seeking feedback from users who experiment in this area, and users should be prepared for changes to features and obligations in this area in the future.

Feedback [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**Extending and Restricting the API**

A CapabilityStatement resource lists the REST interactions (read, update, search, etc.) that a server provides or that a client uses, along with some supporting information for each. It can also be used to define a set of desired behavior (e.g. as part of a specification or a Request for Proposal). The only interaction that servers are required to support is the [capabilities](file:///C:\temp\http.html#capabilities) interaction itself - to retrieve the server's CapabilityStatement. Beyond that, servers and clients support and use whichever API calls are relevant to their use case.

In addition to the operations that FHIR provides, servers may provide additional operations that are not part of the FHIR specification. Implementers can safely do this by appending a custom operation name prefixed with '$' to an existing FHIR URL, as the [Operations framework](file:///C:\temp\operations.html) does. The Conformance resource supports defining what OperationDefinitions make use of particular names on an end-point. If services are defined that are not declared using OperationDefinition, it may be appropriate to use longer names, reducing the chance of collision (and confusion) with services declared by other interfaces. The base specification will never define operation names with a "." in them, so implementers are recommended to use some appropriate prefix for their names (such as "ihe.someService") to reduce the likelihood of name conflicts.

Implementations are encouraged, but not required, to define operations using the standard FHIR operations framework - that is, to declare the operations using the OperationDefinition resource, but some operations may involve formats that can't be described that way.

Implementations are also able to extend the FHIR API using additional content types. For instance, it might be useful to [read](file:///C:\temp\http.html#read) or [update](file:///C:\temp\http.html#update) the appointment resources using a vCard based format. vCard defines its own mime type, and these additional mime types can safely be used in addition to those defined in this specification.

**Extending and Restricting Resources**

Extending and restricting resources (collectively known as 'profiling a resource') is done with a "StructureDefinition" resource, which is a statement of rules about how the elements in a resource are used, and where extensions are used in a resource.

**Changing Cardinality**

One key function of profiles is to change the cardinality of an element. A profile can restrict the cardinality of an element within the limits of the base structure it is constraining. This table summarizes what types of restrictions are allowed:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| derived (across) base (down) | 0..0 (Not used) | 0..1 (optional) | 0..n (optional, many) | 1..1 (required) | 1..n (at least 1) |
| 0..1 | yes | yes | no | yes | no |
| 0..\* | yes | yes | yes | yes | yes |
| 1..1 | no | no | no | yes | no |
| 1..\* | no | no | no | yes | yes |

When a profile is constraining another profile where there are more cardinality options (e.g. low is not just 0 or 1, and high is not just 1 or \*), the same principles still apply: the constraining profile can only allow what the base profile allows.

**Limitations of Use**

What Structure Definitions can do when they are constraining existing resources and datatypes is limited in some respects:

* Profiles cannot break the rules established in the base specification (e.g. cardinality as described above)
* Profiles cannot specify default values or meanings for elements
* Profiles cannot give more specific names to elements
* It must be safe to process a resource without knowing the profile

The consequence of this is that if a profile mandates extended behavior that cannot be ignored, it must also mandate the use of a [modifier extension](file:///C:\temp\extensibility.html#modifiers). Another way of saying this is that knowledge must be explicit in the instance, not implicit in the profile.

As an example, if a profile wished to describe that a [Procedure](file:///C:\temp\procedure.html) resource was being negated (e.g. asserting that it never happened), it could not simply say in the profile itself that this is what the resource means; instead, the profile must say that the resource must have an extension that represents this knowledge.

There is a facility to mark resources to indicate that they can only be safely understood by a process that is aware of and understands a set of published rules. For more information, see [Restricted Understanding of Resources](file:///C:\temp\resource.html#implicitRules).

**Using Structure Definitions**

A "constraint" Structure Definition specifies a set of restrictions on the content of a FHIR resource or data type, or an additional set of constraints on an existing profile. A given structure definition is identified by its canonical URL, which SHOULD be the URL at which it is published. The following kinds of statements can be made about how an element is used, using a series of [Element Definitions](file:///C:\temp\elementdefinition.html):

* Restricting the cardinality of the element; e.g. the base might allow 0..\*, and a particular application might support 1..2
* Ruling out use of an element by setting its maximum cardinality to 0
* Restricting the contents of an element to a single fixed value
* Making additional constraints on the content of nested elements within the resource (expressed as XPath statements)
* Restricting the types for an element that allows multiple types
* Requiring a typed element or the target of a resource reference to conform to another structure profile (declared in the same profile, or elsewhere)
* Specifying a binding to a different terminology value set (see below)
* Providing refined definitions, comments/usage notes and examples for the elements defined in a Resource to reflect the usage of the element within the context of the Profile
* Providing more specific or additional mappings (e.g. to [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) or [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186)) for the resource when used in a particular context
* Declaring that one or more elements in the structure must be 'supported' (see below)

Any changed definitions SHALL be restrictions that are consistent with the rules defined in the resource in the FHIR Specification from which the profile is derived. Note that some of these restrictions can be enforced by tooling (and are by the FHIR tooling), but others (e.g. alignment of changes to descriptive text) cannot be automatically enforced.

Note that structure definitions cannot 'remove' mappings and constraints that are defined in the base structure, but for purposes of clarity, they can refrain from repeating them.

A structure definition contains a linear list of [element definitions](file:///C:\temp\elementdefinition.html). The inherent nested structure of the elements is derived from the *path* value of each element. For instance, a sequence of the element paths like this:

* Root
* Root.childA
* Root.childA.grandchild1
* Root.childB

defines the following structure:

<Root>

<childA>

<grandChild1/>

</childA>

<childB/>

</Root>

or its JSON equivalent. The structure is coherent - children are never implied, and the path statements are always in order. The element list is a linear list rather than being explicitly nested because element definitions are frequently re-used in multiple places within a single definition, and this re-use is easier with a flat structure.

**Differential vs Snapshot**

Structure Definitions may contain a differential statement, a snapshot statement or both.

Differential statements describe only the differences that they make relative to another structure definition (which is most often the base FHIR resource or data type). For example, a profile may make a single element mandatory (cardinality 1..1). In the case of a differential structure, it will contain a single element with the path of the element being made mandatory, and a cardinality statement. Nothing else is stated - all the rest of the structural information is implied (note: this means that a differential profile can be sparse and only mention the elements that are changed, without having to list the full structure. This rule includes the root element - it is not needed in a sparse differential).

Note that a differential can choose not to constrain elements. Doing so means that the profile will be more flexible in terms of compatibility with other profiles, but will require more work to support from implementing systems. Alternatively, a profile can constrain all optional elements to be not present (max cardinality = 0) - this closes the content, which makes implementation easier, but also reduces its usefulness.

In order to properly understand a differential structure, it must be applied to the structure definition on which it is based. In order to save tools from needing to support this operation (which is computationally intensive - and impossible if the base structure is not available), a StructureDefinition can also carry a "snapshot" - a fully calculated form of the structure that is not dependent on any other structure. The FHIR project provides tools for the common platforms that can populate a snapshot from a differential (note that the tools generate complete verbose snapshots; they do not support suppressing mappings or constraints).

StructureDefinitions can contain both a differential and a snapshot view. In fact, this is the most useful form - the differential form serves the authoring process, while the snapshot serves the implementation tooling. StructureDefinition resources used in operational systems should always have the snapshot view populated.

**Slicing**

One common feature of constraining Structure Definitions is to take an element that may occur more than once (e.g. in a list), and split the list into a series of sublists, each with different restrictions on the elements in the sublist with associated additional meaning. In FHIR, this operation is known as "Slicing" a list. It is common to "slice" a list into sub-lists each containing just one element, effectively putting constraints on each element in the list. This technique can also be used on elements that do not repeat, but that have a choice of data types.

Here is an example to illustrate the process:

In this example, the base structure definition for the resource [Observation](file:///C:\temp\observation.html) defines the "component" element which contains a nested code and a value for observations that have multiple values. A classic example of this kind of observation is a blood pressure measurement - it contains 2 values, one for systolic, and one for diastolic ([example](file:///C:\temp\observation-example-bloodpressure.html)).

This diagram shows the conceptual process of 'slicing' the component list into systolic and diastolic slices (note: to avoid clutter, the "name" attribute of Observation is shown as just a code not a full CodeableConcept).

The structure definition for Blood Pressure splits the component list into two sublists of one element each: a systolic element, and a diastolic element. Each of these elements has a fixed value for the code element (a fixed LOINC code for the name), and both have a value of type Quantity. This process is known as "slicing" and the Systolic and Diastolic elements are called "slices".

Note that when the resource is exchanged, the serialization format that is exchanged is not altered by the constraining definition. This means that the item profile names defined in the structure definition ("systolic", etc. in this example) are never exchanged. A resource instance looks like this:

<Observation>

...

<component>

<code {LOINC="8480-6"}/>

<value ...>

</component>

<component>

<code {LOINC="8462-4"}/>

<value ...>

</component>

</Observation>

In order to determine that the first related item corresponds to "Systolic" in the structure definition, so that it can determine to which additional constraints for a sub-list the item conforms, the system checks the values of the elements. In this case, the "code" element in the target resource can be used to determine which slice that target refers to. This element is called the "discriminator".

**Discriminator**

In the general case, systems processing resources using a structure definition that slices a list can determine the slice corresponding to an item in the list by checking whether the item's content meets the rules specified for the slice. This would require a processor to be able to check all the rules applied in the slice and to do so speculatively in a depth-first fashion. Both of these requirements are inappropriately difficult for an operational system, and particularly for generated code (e.g. software that is automatically produced based on the StructureDefinition). Thus, to provide a better way to distinguish slices, a sliced element can designate a field or set of fields that act as a "discriminator" - they are used to tell the slices apart.

When a discriminator is provided, the composite of the values of the elements designated in the discriminator is unique and distinct for each possible slice and applications can easily determine which slice an item in a list is. The intention is that this can be done in generated code, e.g. using a switch/case statement.

When a constraining structure designates one or more discriminators, it SHALL ensure that the possible values for each slice are different and non-overlapping, so that the slices can easily be distinguished.

Each discriminator is a pair of values: a type that indicates how the field is processed when evaluating the discriminator, and a [FHIRPath](http://hl7.org/fhirpath) expression that identifies the element in which the discriminator is found. There are five different processing types for discriminators:

The FHIRPath statement that allows for the selection of the element on which the discriminator is based is a restricted FHIRPath statement that is allowed to include:

* Element selections (e.g. FHIRPath statements without "()" such as component.value)
* The function extension(url) to allow selection of a particular extension
* The function resolve() to allow slicing to across resource boundaries

Each slice must use the [element definition](file:///C:\temp\elementdefinition.html) for the element in the descriminator(s) to ensure that the slices are clearly differentiated (by assigning a fixed value, a specific type, or a profile, depending on the discriminator type. If the type is 'value', then the element definition must use either[ElementDefinition.fixed[x]](file:///C:\temp\elementdefinition-definitions.html#ElementDefinition.fixed_x_)), or if the element has a terminology binding, a required binding with a [Value Set](file:///C:\temp\valueset.html#required) that enumerates the list of possible codes in the value set ("extensional definition").

It is the composite (combined) values of the discriminators that are unique, not each discriminator alone. For example, a slice on a list of items that are references to other resources could designate fields from different resources, where each resource only has one of the designated elements, as long as they are distinct across slices.

A structure definition is not required to designate any discriminator at all for a slice, but those that don't identify discriminators are describing content that is very difficult to process, and so this is discouraged.

Within a structure definition, a slice is defined using multiple *element* entries that share a *path* but have distinct *name*s. These entries together form a "slice group" that is:

1. **Initiated by a "slicing entry"** That is, the first *element* in a slice group must contain a *slicing* property that defines the *discriminator* for all members of the group. It also contains the unconstrained definition of the element that is sliced, potentially including children of the unconstrained element, if there are any
2. **Mutually exclusive**. This means that each *element* in a slice group SHALL describe a distinct set of values for the group's *discriminators*. Because of this constraint, an element in a resource **instance** will never match more than one *element* in a given slice group. If no discriminators are named, it SHOULD still be possible to differentiate the slices based on their properties, though it may be substantially harder to do so.
3. **Serialized as a group**. The entries in a slice group must be **adjacent** in a serialized structure definition, **or**, if there are any intervening elements, those elements must be "compatible with" the group. Concretely, this means that any intervening elements must have a *path* that starts with the slice groups's *path*. For example, an *element* with a *path* of *Observation.name.extension* would be compatible with (and thus, would not "break up") a slice group whose path was *Observation.name*

Some examples of descriminators:

|  |  |  |  |
| --- | --- | --- | --- |
| **Context** | **Discriminator Type** | **Discriminator Path** | **Interpretation** |
| List.entry | value | item.reference.resolve().name | Entries are differentiated by the name element on the target resource - probably an observation, which could be determined by other information in the profile |
| List.entry | type | item.reference.resolve() | Entries are differentiated by the type of the target element that the reference points to |
| List.entry | profile | item.reference.resolve() | Entries are differentiated by a profile tag on the target of the reference, as specified by a structure definition (todo: how to do that?) |
| List.entry | value | item.extension("http://acme.org/extensions/test").code | Entries are differentiated by the value of the code element in the extension with the designated URL |
| List.entry.extension | value | url | Extensions are differentiated by the value of their url property (usually how extensions are sliced) |
| List.entry | type, value | item.reference.resolve(), item.reference.resolve().code | Extensions are differentiated by the combination of the type of the referenced resource, and, if it has one, the code element of that resource. This would be appropriate for where a List might be composed of a Condition, and set of observations, each differentiated by its name - the condition has no name, so that is evaluated as a null in the discriminator set |
| Observation.value[x] | type | $this | Different constraints (e.g. "must support", usage notes, vocabulary bindings, etc.) are asserted for different supported types for the multi-typed element Observation.value[x] |

Note that discriminator types of type and profile can also be used where a repeating element contains a resource directly (e.g. [DomainResource.contained](file:///C:\temp\domainresource-definitions.html#DomainResource.contained), [Bundle.entry](file:///C:\temp\bundle-definitions.html#Bundle.entry), [Parameters.parameter.resource](file:///C:\temp\parameters-definitions.html#Parameters.parameter.resource)).

The [examples of slicing and discriminators](file:///C:\temp\profiling-examples.html) show exactly how this and other typical uses of slicing are represented in profiles.

Note that extensions are always sliced by the url element, though they may be resliced on additional elements where required.

**Default Slice**

There is a special slice, called the default slice. This allows a profile to describe a set if specific slices, and then make a set of rules that apply to all the remaining content that is not in one of the defined slices. Some rules about the default slice:

* It is identified because the name of the slice is @default. The sliceName '@default' is reserved and cannot be used in any other context
* Default slices are only allowed when the slicing rule = closed
* Default slices must not fix the value of the discriminator elements
* Default slices can be re-sliced in dependent profiles

One use of a default slice would be the case where the profile slices an identifier element to require a set of known identifiers, where the type element is prohibited (since they are known identifiers) but requires type on all other identifiers if any are present. In this case, the default slice makes no rules about the identiifer.system (which is the slicing discriminator), but fixes the cardinality of type to 1..1 in the @default slice.

**Re-profiling and Re-slicing**

Profiles can be based on other profiles, and apply further constraints to those already specified. This is a useful technique, but implementers should be wary of over-use - humans have understanding the implications of deep stacks of constraining profiles.

When a profile constrains another profile, it can make additional constrainta, including extending the discriminator, adding new slices (if the slices are not already closed), and slicing inside the existing slices.

The rules for changing the slicing are as follows:

* Rule = open can be changed to rule = closed, and unordered can be changed to ordered)
* If a discriminator for an element is declared in a parent profile, child profiles referencing that element must either declare the same discriminator or must declare a new discriminator that includes the parent discriminator content. I.e. additional discriminator paths may be added, but none of the existing paths can be removed.

It's sometimes necessary to slice data that has already been sliced in the base profile - that is, create new slices within the existing slices. This is called "Re-slicing". The rules for re-slicing are as follows:

When you slice, you define a name for each new slice. The name has to be unique across the set of slices in the profile. So if profile A defines an element X with cardinality 0..\*, and profile B is derived from profile A, then profile B can either:

1. make a constraint on X with no name - in which case the profile is adding constraints to all slices of X; or
2. make a constraint on X with a name - in which case the profile is describing a specific slice on X, and the constraints only apply to that slice; or
3. it can do both

Then, profile C derives from profile B. Profile C can do the following:

1. make a constraint on X with no name - in which case the profile is constraining all appearances of X; or
2. make a constraint on X with a different name from that used in profile B - in which case the profile is describing a specific new slice on X, and the constraints only apply to that slice; or
3. make a constraint on X with the same name as that used in profile B - in which case the profile is making new constraints on the slice defined in profile B; or
4. some combination of the above options

Note: it is possible for Profile C to make rules that are incompatible with profile B, in which case there is no set of instances that can be valid against profile C

In addition to the above, there are times when Profile C will need to further slice a slice defined in B. In this case, there's a need to reference both the name of the original slice from Profile B as well as to define a name for the slice defined within Profile C. This is done by separating the names using "/". For example, if Profile B defines the slice "example", and profile C defines the slice "example/example1", then this is deemed to be "example1" slice of the example slice. This process can continue indefinitely by separating each layer of slicing names with the "/" character. This pattern applies to @default too: @default/@default.

**Extension Definitions**

An extension definition defines the URL that identifies the extension and which is used to refer to the extension definition when it is used in a resource.

The extension definition also defines the context where the extension can be used (usually a particular path or a data type) and then defines the extension element using the same details used to profile the structural elements that are part of resources. This means that a single extension can be defined once and used on different Resource and/or datatypes, e.g. one would only have to define an extension for "hair color" once, and then specify that it can be used on both Patient and Practitioner.

For further discussion of defining and using extensions, along with some examples, see [Extensibility](file:///C:\temp\extensibility.html).

**Using Extensions in Profiles**

Once defined, an extension can be used in an instance of a resource without any Profile declaring that it can, should or must be, but Profiles can be used to describe how an extension is used.

To actually prescribe the use of an extension in an instance, the extension list on the resource needs to be sliced. This is shown in [the extensibility examples](file:///C:\temp\extensibility-examples.html#sliceextensions)

Note that the minimum cardinality of an extension SHALL be a valid restriction on the minimum cardinality in the definition of the extension. if the minimum cardinality of the extension is 1 when it is defined, it can only be mandatory when it is added to a profile. This is not recommended - the minimum cardinality of an extension should usually be 0.

**Binding Definitions**

Coded elements have bindings that link from the element to a definition of the set of possible codes the element may contain. The binding identifies the definition of the set of possible codes and controls how tightly the set of the possible codes is interpreted.

The set of possible codes is either a formal reference to a [ValueSet](file:///C:\temp\valueset.html) resource, which may be version specific, or a general reference to some web content that defines a set of codes. The second is most appropriate where a set of values is defined by some external standard (such as mime types). Alternatively, where the binding is incomplete (e.g. under development) just a text description of the possible codes can be provided.

Bindings have a property that defines how the strongly implementations are required to use the set of codes. See [Binding Strength](file:///C:\temp\terminologies.html#strength).

**Mixing Custom and Standard Terminologies**

[CodeSystem](file:///C:\temp\codesystem.html) resources can be used to carry definitions of local codes ([Example](file:///C:\temp\codesystem-example.html)) and [ValueSets](file:///C:\temp\valueset.html) can mix a combination of local codes and standard codes (e.g. LOINC, SNOMED), or just to choose a particular set of standard codes (examples: LOINC, SNOMED, RxNorm). Profiles can bind to these value sets instead of the ones defined in the base specification, following these rules:

|  |  |
| --- | --- |
| [**Binding Strength**](file:///C:\temp\terminologies.html#binding)**in base specification** | **Customization Rules in Profiles** |
| required | The value set can only contain codes contained in the value set specified by the FHIR specification |
| extensible | The value set can contain codes not found in the base value set. These additional codes SHOULD NOT have the same meaning as existing codes in the base value set |
| preferred | The value set can contain whatever is appropriate for local use |
| example | The value set can contain whatever is appropriate for local use |

Note that local codes are not as interoperable as standard published code systems (e.g. LOINC, SNOMED CT, so it is preferable to use standard code systems.

**Changing Binding Strength in Profiles**

A profile can change the terminology binding of an element - both strength and value set - within the limits of the base structure it is constraining. This table summarizes the changes that can be made to the binding strength:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| derived (across) base (down) | required | extensible | preferred | example |
| required | yes | no | no | no |
| extensible | yes | yes | no | no |
| preferred | yes | yes | yes | no |
| example | yes | yes | yes | yes |

Note that a constraining profile may leave the binding strength the same and change the value set instead. Whatever the constraining profile does, it cannot make codes valid that are invalid in the base structure/profile.

**Must Support**

One property that can be declared on profiles that is not declared on the resource or data type definitions is "Must Support". This is a boolean property. If true, it means that systems claiming to conform to a given profile must "support" the element. This is distinct from cardinality. It is possible to have an element with a minimum cardinality of "0", but still expect systems to support the element.

The meaning of "support" is not defined by the base FHIR specification, but can be set to true in a profile. When a profile does this, it SHALL also make clear exactly what kind of "support" is required. Examples might include:

* The system must be able to store and retrieve the element
* The system must display the element to the user and/or allow the user to capture the element via the UI
* The element must appear in an output report
* The element must be taken into account when performing decision support, calculations or other processing
* etc.

The specific meaning of "Must Support" for the purposes of a particular profile SHALL be described in the element.definition, the general StructureDefinition.description or in other documentation for the implementation guide the profile is part of.

If creating a profile based on another profile, Must Support can be changed from false to true, but cannot be changed from true to false. Note that an element that has the property IsModifier is not necessarily a "key" element (e.g. one of the important elements to make use of the resource), nor is it automatically mustSupport - however both of these things are more likely to be true for IsModifier elements than for other elements.

**Search Criteria**

The final thing implementations can do is to define search criteria in addition to those defined in the specification itself. Search criteria fall into one of four categories:

1. Enabling search on core elements that don't have standard search criteria defined (e.g. searching Observation by normal range)
2. Enabling search on elements that already have a standard search criteria defined, but with custom matching rules; e.g. a sounds-like search on Practitioner name
3. Enabling search on a particular extension
4. Enabling search that doesn't correspond to a single element but rather a combination of elements or computation on an element; e.g. searching for patients by age

Additional Search Parameters can be defined using the [SearchParameter](file:///C:\temp\searchparameter.html) resource.

**Presenting Profiles**

When this specification describes a profile, the profile is presented in 5 different forms:

1. Text Summary
2. Differential Table
3. Snapshot Table
4. XML Template
5. JSON Template

**Text Summary**

This presents a short summary of the profile - a combination of the author's summary, and some automatically generated summary content

**Differential Table**

This is a view of the differential statement ([see above](file:///C:\temp\main-pages.html#snapshot)). For context, additional information not in the differential is also show partially transparent

**Snapshot Table**

This is a view of the snapshot produced by the profile ([see above](file:///C:\temp\main-pages.html#snapshot)). The information is a comprehensive view of what the profile means

**XML Template**

Not one yet

**JSON Template**

Not one yet

**Supporting Multiple Profiles**

Applications may be required to support more than one profile at a time. A typical example might be an EHR application that is required to support a general purpose data sharing profile (such as [DAF](http://hl7.org/fhir/daf)), and also must support specific profiles for decision support using the same interface.

The impact of supporting two sets of profiles depends on whether resources are being created or consumed. When an application is creating content, it has to create content that conforms to both sets of profiles - that is, the intersection of the profiles. When an application is consuming information, then it must be able to consume content that conforms to either set of profiles - that is, the union of the profiles.

Since applications generally consume and produce resources at the same time, conforming to more than one profile may not be possible, unless the profiles are designed to make statements at different levels - and the case above is one such case, where one profile is focused on data access, provenance, and availability, the other profile is focused on clinical content.

Accordingly, profiles can relate to each other in four different ways. Each profile can be thought of in terms of the set of instances that conform to the profile:

1. non-overlapping: there no instances that conform to profiles A & B (technically, the intersection of profiles A & B is an empty set)
2. partly overlapping: some instances conform to both A & B, but others only conform to A or B
3. one set contained in the other: all resources that conform to A conform to B, but only some of the ones that conform to B conform to A (or vice versa)
4. identical sets: all resources that conform to A also conform to B and any resources that don't conform to B, don't conform to A

Profiles can be compared to determine their compatibility. One such comparison can be found (no - todo: bring this into the build) between [DAF](http://hl7.org/fhir/daf) and [QICore](http://hl7.org/fhir/qicore). Note that this comparison is generated by tooling under ongoing development, and is purely draft content to demonstrate the idea of profile comparison.

**pushpull.html**

**Managing Push and Pull**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The primary focus of FHIR is exchanging healthcare data between systems. Whenever one system has information that another needs or wants, one question that automatically arises is whether the source system will push the data, or whether the destination system will pull the data.

|  |  |
| --- | --- |
| Push | As new information becomes available, or whenever it deems appropriate, the source system sends the information to the destination. It expects the destination system to maintain and index the information on receipt. The source system must trust the destination system to manage access/security appropriately |
| Pull | The source system maintains and indexes data. When the destination system wants or needs the information, it must retrieve it form the source system. Either the source or the destination (or both) manages access/security |

Note that it is possible to mix these modes by adding an interface repository into the mix. For example, to connect a push-based source with a pull-based destination, simply have the source push information at the repository, and the destination can pull data from it as required. If the source is pull based, the interface repository can query it for new data on a regular basis, and when it finds any, it can pull it into the repository and/or push it to the destination.

Which combination of push, pull and interface repositories is appropriate for any given context depends on a host of factors around deployment architecture, network and business topology, security policies, etc. However, it is often constrained by the capabilities of the various systems and the standards used in the exchange.

The FHIR specification supports and enables both push and pull:

* **REST**: The REST API can be used in either fashion - this is discussed in more depth below
* **Messages**: FHIR defines message events for both push and pull
* **Services**: Services may be defined to support either model
* **Documents**: This question doesn't apply, because documents are bundles that are exchanged using either REST, messages, or services

**Push and Pull on a RESTful interface**

* Push: The source is a client, and when new data is available, it uses [create](file:///C:\temp\http.html#create), [update](file:///C:\temp\http.html#update), and [transaction](file:///C:\temp\http.html#transaction) to push data to the destination, which is a server
* Pull: The source is a server, and the destination is a client. When the client wants to use the data, it uses a combination of [search](file:///C:\temp\http.html#search) and [read](file:///C:\temp\http.html#read) so the users can find the data they need
* Push/Pull: The source is a server, and the destination is a client. On a regular basis, the destination queries the server for its [history](file:///C:\temp\http.html#history), and replicates the changes in its own copy of the database

**qa.html**

**FHIR QA Report**

[See also the Work Group List Page](file:///C:\temp\wglist.html)

**r2maps.html**

**Transforms between DSTU 2 and STU 3**

In addition to a [list of all differences between DSTU 2 and STU 3](file:///C:\temp\diff.html), this specification also provides transforms between DSTU and STU 3 for most resources that were in DSTU 2. These transforms are expressed using the [FHIR Mapping Language](file:///C:\temp\mapping-language.html), and serve as both formal documentation of the relationship between the designs in R2 and R3, and can also be used to convert between the formats automatically.

The following table summarizes the state of the R2:R3 maps for each resource type.

Note the following:

* Transforms are tested using the following method:
  1. For each example resource in the R2 release of the FHIR specification that has a transform to R3
  2. apply the transform to R3
  3. validate against the R3 specification
  4. apply the reverse transform back to R2
  5. compare the outcome with the original resource - ideally, it should be identical
* All transforms should execute - that is, they should be able to take any R2 resource, represent the content in R3, and then generate an R2 representation for the same content. Note, though, that in a few cases, the differences between R2 and R3 designs are so significant that not all examples can be successfully transformed at all.
* For some resources, the differences in design between R2 and R3 are sufficiently great that the R2 examples cannot reproduce the same output after conversion to R3 and then back to R2. For example, the R3 resource may have a different set of status codes with imperfect maps
* For some resources, the information available in the R2 examples is not enough to generate valid R3 resources, and so the resources generate R3 validation errors. (e.g. R2 resources missing an element that is required in R3)
* For all these reasons, the maps provided in the specification are not authoritative; they are provided to help implementers who need to convert between R2 and R3 forma, but additional revision may be required in an implementation specific context
* The Bundle tests include the large distribution files (all valuesets, structure definitions, etc) and so generate large numbers of validation errors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resource** | **# Tests** | **% Execute OK** | **% RoundTrip Ok** | **% R3 Valid** | **R3 Error Count** |

**rdf.html**

**Resource Description Framework (RDF) Representation**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 0 | [Ballot Status](file:///C:\temp\help.html#status): [Draft](file:///C:\temp\history.html#pubs) |

|  |  |
| --- | --- |
| http://w3c-domains.com/img/w3c-96.png | This page and the RDF forms are jointly maintained by the HL7 FHIR project and the [W3C Semantic Web Health Care and Life Sciences Interest Group](https://www.w3.org/blog/hcls/). |

FHIR resources can be represented as an [RDF graph](http://www.w3.org/RDF/) serialised in the [Turtle format](https://www.w3.org/TR/turtle/), or as [JSON-LD](http://json-ld.org). The RDF format is defined to assist the process of bridging between operational data exchange and formal knowledge processing systems. While the RDF form offers a fully functional representation of FHIR resources, it has different operational characteristics to the [JSON](file:///C:\temp\json.html) and [XML](file:///C:\temp\xml.html) representations, and would be implemented for different reasons. Systems focused on operational exchange of data would not generally use choose to use RDF.

This page describes:

* [Representing FHIR resources in Turtle](file:///C:\temp\main-pages.html#instance)
* [The FHIR definitions as Turtle](file:///C:\temp\main-pages.html#class)
* [How to exchange RDF using the RESTful API](file:///C:\temp\main-pages.html#api)
* [Experimental JSON-LD format](file:///C:\temp\main-pages.html#jsonld)

**Turtle Format for Resources**

A FHIR resource is represented by a series of RDF triples. The Turtle representation for a resource is described using this format:

Turtle Template

[ a fhir:Observation; fhir:nodeRole fhir:treeRoot;

# from Resource: id; meta; implicitRules; and language

# from DomainResource: text; contained; extension; and modifierExtension

fhir:Obervation.identifier [ [Identifier](file:///C:\temp\main-pages.html) ]; # 0..\* Unique Id for this particular observation

fhir:Obervation.status [ fhir:value "<code>" ]; # R! registered | preliminary | final | amended +

fhir:Obervation.code [ [CodeableConcept](file:///C:\temp\main-pages.html) ]; # 1..1 R! Type of observation (code / type)

fhir:Obervation.subject [ fhir:reference [ [Patient](file:///C:\temp\main-pages.html)|[Group](file:///C:\temp\main-pages.html)|[Device](file:///C:\temp\main-pages.html)|[Location](file:///C:\temp\main-pages.html) ] ]; # 0..1 Who and/or what this is about

fhir:Obervation.encounter [ fhir:reference [ [Encounter](file:///C:\temp\main-pages.html) ] ]; # 0..1 Healthcare event during which this observation is made

# effective[x]: 0..1 Clinically relevant time/time-period for observation. One of these 2:

fhir:Obervation.effectiveDateTime [ fhir:value "<dateTime>" ];

fhir:Obervation.effectivePeriod [ [Period](file:///C:\temp\main-pages.html) ];

]

Using this format:

* To build a valid Turtle instance of a resource, replace the contents of the property values with valid content as described by the type rules and content description found in the property value for each element
* Relationship names are case-sensitive (though duplicates that differ only in case are never defined)
* Relationships can appear in any order
* Content within a resource is always represented with anonymous nodes - only resources can be identified
* Nodes are never empty. If an element is present in the resource, it SHALL have properties as defined for its type, or 1 or more [extensions](file:///C:\temp\extensibility.html)
* The **R!** denotes that a relationship is mandatory - it must be present (or in an array, at least one item must be present)
* Note that this specification produces turtle that is nicely formatted and well laid out, but this is not required nor expected
* The MIME-type for this format is text/turtle. Other MIME types: application/json-ld for the json-ld variant, and text/shex for the SHEX (RDF schemas)

**Representing resources**

Each resource is represented as a set of RDF triples represented using the Turtle syntax. When a resource has a persistent identity (e.g. it can be found at a particular URL - usually a FHIR RESTful server), then that URL is it's identity. Resources with no persistent identity (e.g. bundles from search results) have the the identity of the root document - "<>" in turtle syntax.

Some resources can contain other resources. Given that the relationships can appear in any order, it cannot be assumed that the first encountered element represents the resource of interest that is being represented by the set of turtle statements. The focal resource - where to start when parsing - is the resource with the relationship fhir:nodeRole to fhir:treeRoot. if there is more than one node labelled as a 'treeRoot' in a set of turtle statements, it cannot be determined how to parse them as a single resource.

**Representing Repeating Elements**

Elements that can repeat are represented with a relationship

fhir:index [n]

where [n] is a zero-based integer offset (ie.g the first element has an index of 0). Lists are never sparse; it is an error if there are missing items in the repeat sequence.

Note: this measn that the rdf:list structure is not used.

**Representing Primitive Elements**

Primitive elements - elements with a primitive type - are represented as regular nodes so that the elements extensions can be represented. The actual value of the primitive type is represented using the fhir:value predicate:

fhir:value "[value]"^^xs:type

The value has two parts: a literal string that contains the value, and, if applicable, one of the following schema types:

* boolean
* integer
* decimal
* base64Binary
* dateTime
* date
* gYear
* gYearMonth
* time

The choice is made based on the types as specified for the [primitive type](file:///C:\temp\datatypes.html#primitive). Note that the correct schema type for a date/dateTime must be determined by inspecting the value of the date for precision.

The fhir:value property can never be empty. Either the relationship is absent, or it is present with at least one character of content. XHTML is represented as an escaped xs:string.

**Representing References**

A [Reference](file:///C:\temp\references.html#Reference) element is representing using the same rules as above:

fhir:Observation.subjectReference [

fhir:Reference.reference [ fhir:value "Patient/example" ];

fhir:Reference.display [ fhir:value "Example Patient" ];

];

This allows faithful round tripping of the resource between the Turtle format and the JSON and XML formats. However it's very useful for an RDF processor if the RDF graph links to the target of the reference directly. This can be represented using the fhir:link property:

fhir:Observation.subjectReference [

fhir:link <http://hl7.org/fhir/Patient/example>;

fhir:Reference.reference [ fhir:value "Patient/example" ]

];

The correct value for the fhir:link relationship must be determined by resolving the [rules for resolving references](file:///C:\temp\references.html) for the various reference types to a literal URL that refers to the correct content in the local RDF context.

The fhir:link relationship can be added automatically as part of generating the resource representation, or it can be injected by a post-processor that knows how to convert the raw references into RDF-suitable references.

**Representing Concepts**

The same logic applies to the [Coding](file:///C:\temp\datatypes.html#Coding) data type. These are represented directly in turtle by serialising their properties as specified above:

fhir:Observation.code [

fhir:CodeableConcept.coding [

fhir:index 0;

fhir:Coding.system [ fhir:value "http://loinc.org" ];

fhir:Coding.code [ fhir:value "29463-7" ];

fhir:Coding.display [ fhir:value "Body Weight" ]

];

fhir:CodeableConcept.coding [

fhir:index 1;

fhir:Coding.system [ fhir:value "http://snomed.info/sct" ];

fhir:Coding.code [ fhir:value "27113001" ];

fhir:Coding.display [ fhir:value "Body weight" ]

]

];

For reasoners using the RDF graph, it's very useful to make the implicit concept references in these Codings explicit using a rdf:type assertion ("a" in turtle):

fhir:Observation.code [

fhir:CodeableConcept.coding [

fhir:index 0;

a loinc:29463-7;

fhir:Coding.system [ fhir:value "http://loinc.org" ];

fhir:Coding.code [ fhir:value "29463-7" ];

fhir:Coding.display [ fhir:value "Body Weight" ]

];

fhir:CodeableConcept.coding [

fhir:index 2;

a sct:27113001;

fhir:Coding.system [ fhir:value "http://snomed.info/sct" ];

fhir:Coding.code [ fhir:value "27113001" ];

fhir:Coding.display [ fhir:value "Body weight" ]

]

];

These rdf:type assertions can be made by any agent that knows how to convert from the code system to the correct ontological representation on the RDF context. Note that a few code systems have standard ontological representations, but many don't. Again, these assertions can be made by the serialiser, or injected by a post-processor.

**Schema**

FHIR uses ShEx for representing the turtle schema. See [fhir.shex](file:///C:\temp\fhir.shex) for definitions.

**Experimental JSON-LD format**

As a by-product of the RDF format, a JSON-LD format is defined to represent exactly the same information as the Turtle format, but using the JSON-LD syntax. Some specific notes about the JSON-LD format:

* There is a single FHIR context: [url]/fhir.jsonld, where [url] is the URL of the applicable version (e.g. http://hl7.org/fhir/STU3). Implementers should use a version specific reference, not the current version (http://hl7.org/fhir) since the contents of this change as new full releases of FHIR are published)
* Each resource has @type on every resource, since the resources are inherently polymorphic at the format level. In addition, the root node explicitly carries the property "role" : "fhir:treeRoot"
* When parsing the JSON-LD format, applications should not use the literal JSON directly, but use the json-ld processed form. Any json-ld that results in the same content post-processing is considered conformant.
* The JSON-LD format is derived from the RDF format, and processing the JSON-LD will result in the same set of triples as the turtle format, though there may be minor differences in the types of the nodes depending on the RDF library used.
* For booleans and integers, JSON boolean and numbers are used. However for decimal values, a JSON string must be used.

**STU Note:** The definition of the JSON-LD format means that the FHIR specification defines two different JSON formats, each with a different purpose. It seems unlikely that normal implementers will prefer the JSON-LD format over the [other JSON format](file:///C:\temp\json.html), since that is more dense, but there are other implementers who would prefer the the JSON-LD format. On the other hand, defining two different JSON formats seems like a recipe for confusion. Feedback from implementers is welcome.

**RDF Representation of FHIR**

In addition to the basic representation of FHIR resources in turtle format, a Turtle representation of the FHIR infrastructure and definitions is also published, for the following purposes:

* Providing the class definitions to support RDF based representation of resource instances
* Supporting knowledge based analysis of the FHIR specification itself
* Providing knowledge of use at run-time for converting between FHIR and other content models
* Supporting reasoning across the information/terminology model boundary

The RDF definitions are published as a series of turtle files: [RIM](file:///C:\temp\rim.ttl) and [FHIR](file:///C:\temp\fhir.ttl). Note: these are out of sync with the serialization above; this will be fixed in the lead up to the Montreal connectathon.

**Using RDF with the REST API**

TODO

**redirect.html**

See here: [this link](file:///C:\temp\site\index.html).

**references-definitions.html**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**references.html**

**Resource References**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 3 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Many of the defined elements in a resource are references to other resources. Using these references, the resources combine to build a web of information about healthcare.

Resources contain two types of references:

* **Internal "contained" references** - references to other resources packaged inside the source resource
* **External references** - references to resources found elsewhere

References are always defined and represented in one particular direction - from one resource (source) to another (target). References are either provided as a literal URL, which may either be absolute or relative, or as a logical identifier. Resolving the references is discussed below.

The corresponding reverse relationship from the target to the source exists in a logical sense, but is not represented explicitly in the target resource. For external references, navigating these reverse relationships requires some external infrastructure to track the relationship between resources (the [REST API](file:///C:\temp\http.html) provides one such infrastructure by providing the ability to [search](file:///C:\temp\http.html#search) the reverse relationship by naming search parameters for the references, and by providing support for [reverse includes](file:///C:\temp\search.html#revinclude)).

Because resources are processed independently, relationships are not considered to be transitive. For example, if a [Condition](file:///C:\temp\condition.html) resource references a particular [Patient](file:///C:\temp\patient.html) as its subject, and references a [Procedure](file:///C:\temp\procedure.html) resource as its cause, there is no automatic rule or implication that the procedure has the same patient for its subject. Instead, the subject of the procedure must be established directly in the Procedure resource itself. Another way to state this is that the context of the subject is not "inherited", nor does it "conduct" along the relationship to procedure. The only exception to this is the case of contained resources (see below). Note that in practice, the relationships need to describe a logical and coherent record, and in the case of the Condition and Procedure described here, they would usually be required to have the same patient for their subjects, and profiles may make rules about this.

In a resource, references are represented with a reference (literal reference), an identifier (logical reference), and a display (text description of target).

**Constraints**

At least one of reference, identifier and display SHALL be present (unless an extension is provided).

**Literal References**

The reference is the key element - resources are identified and addressed by their URL. It contains a url that is either

* an absolute URL
* a relative URL, which is relative to the [Service Base URL](file:///C:\temp\http.html#root), or, in a bundle, the Bundle.entry.fullUrl (see [Resolving References in Bundles](file:///C:\temp\bundle.html#references))
* an internal fragment reference (see "Contained Resources" below)

Notes:

* Using absolute URLs provides a stable, scalable approach suitable for a cloud/web context, while using relative/logical references provides a flexible approach suitable for use when trading across closed ecosystem boundaries. (see ["Managing Resource Identity"](file:///C:\temp\managing.html) for further discussion)
* Absolute URLs do not need to point to a [FHIR RESTful server](file:///C:\temp\http.html), though this is the preferred approach. Whether or not the reference is to a FHIR RESTful server, the reference SHALL point to a Resource as defined by this specification.   
  Note: This regex is true if the reference to a resource is consistent with a FHIR API:
* ((http|https)://([A-Za-z0-9\\\.\:\%\$]\/)\*)?()\/(\/\_history\/)?

However conformance with this regex is no guarantee that the end-point is a FHIR server

* URLs are always considered to be case-sensitive
* References SHALL be a reference to an actual FHIR resource, and SHALL be resolveable (allowing for access control, temporary unavailability, etc). Resolution can be either by retrieval from the URL, or, where applicable by resource type, by treating an absolute reference as a canonical URL ([see below](file:///C:\temp\main-pages.html#canonical)) and looking it up in a local registry/repository

A relative reference to the [Patient](file:///C:\temp\patient.html) "034AB16" in an element named context on a FHIR RESTful server:

<patient>

<reference value="Patient/034AB16" />

</patient>

An absolute reference to a [Structure Definition](file:///C:\temp\structuredefinition.html) in an element named profile:

<profile>

<reference value="http://fhir.hl7.org/svc/StructureDefinition/c8973a22-2b5b-4e76-9c66-00639c99e61b" />

</profile>

**Logical References**

In many contexts where FHIR is used, applications building a resource may know an identifier for the target of the reference, but there is no way for the application to convert this to a literal reference that directly references an actual resource. This situation may arise for several reasons:

* There is no server exposing any such resource. This is often the case with national identifiers (e.g. US SSN or NPI), and such identifiers are widely used
* The server that exposes the resource is not available to the source application, so it has no way to resolve an identifier to a reference
* The application is not in a RESTful environment - it is creating a message or a document

In this cases, the source application may provide the identifier as a logical reference to the entity that the target resource would describe.

A logical reference to the [Patient](file:///C:\temp\patient.html) with an SSN of 000111111:

<patient>

<identifier>

<system value="http://hl7.org/fhir/sid/us-ssn" />

<value value="000111111" />

</identifier>

</patient>

There is no requirement that a Reference.identifier point to something that is actually exposed as a FHIR instance, but it SHALL point to a business concept that would be expected to be exposed as a FHIR instance, and that instance would need to be of a FHIR resource type allowed by the reference.

When processing a resource, an application may be able to use the identifier directly, on the grounds that all it needs is the identifier, or it may be able to resolve the identifier directly. Alternatively, it may be able to use a server to resolve the logical reference to a literal reference to a resource.

Irrespective of how the resolution occurs, any system processing a logical reference will only be able to resolve the identifier to a reference if it understands the business context in which the identifier is used. Sometimes this is global (e.g. a national identifier) but often it is not.

For this reason, none of the useful mechanisms described for working with references (e.g. [chaining](file:///C:\temp\search.html#chaining), [includes](file:///C:\temp\search.html#include)) are possible, nor should servers be expected to be able to automatically resolve the reference. Servers may accept an identifier based reference untouched, resolve it, and/or reject it - see [CapabilityStatement.rest.resource.referencePolicy](file:///C:\temp\capabilitystatement-definitions.html#CapabilityStatement.rest.resource.referencePolicy).

When both an identifier and a literal reference are provided, the literal reference is preferred. Applications processing the resource are allowed - but not required - to check that the identifier matches the literal reference, if they understand how to resolve the logical reference.

Applications converting a logical reference to a literal reference may choose to leave the logical reference present, or remove it.

**Reference Description**

Irrespective of whether a literal and/or logical reference is provided, or neither, the display element may be used to provide a very short description of the target resource.

<custodian>

<reference value="Organization/123" />

<display value="HL7, Inc" />

</custodian>

This text can be used by any application that cannot resolve the reference to fill out the text portion of a hyperlink referring to the target resource, for instance. It can also save time fetching a target resource, and determining how to convert it to a very short textual description.

In general, the display, if populated, does not have identical content to the Resource.text of the referenced resource. The purpose is to identify what's being referenced, not to more fully describe it.

**Canonical URLs**

Many resource types have a defined element "url" which is the canonical URL that always identifies the resource. These include all the conformance and knowledge resources (most of the resources not found in the [Patient Compartment](file:///C:\temp\compartmentdefinition-patient.html)).

The canonical URL remains the same when the resource is copied from server to server, while the logical id of the resource - it's local identifier - usually changes as the resource is copied. The canonical URL serves as a stable logical identifier for the conformance artifact, and **is the preferred way to reference a conformance or knowledge resource**. The canonical URL is also the location where the master copy of the artifact is found.

References to these resource may use the Reference type described above, but they can also be referenced using a [uri](file:///C:\temp\datatypes.html#uri).

When the type of the canonical reference is a uri, the URL may include a version, in order be precise about which version of the resource is being referred to. To do this, append the version to the canonical url with a '|' like this:

<valueSetUri value="http://hl7.org/fhir/StructureDefinition/my-profile|0.8"/>

This is a version specific reference to a profile. Note that this is the StructureDefinition.version not the StructureDefinition.meta.versionId. Searching for this on a FHIR server would look like this:

GET fhir/ValueSet?url=http://hl7.org/fhir/ValueSet/clinical-findings&version=0.8

Note that if a canonical reference does not have a version, and the server finds multiple versions for the value set, the system using the reference should pick the latest version of the target resource and use that. Servers SHOULD support version specific searching for canonical URLs but automatically detecting the presence of a |[version] and performing the appropriate search.

Systems resolving references to resources that might have cannonical URLs SHOULD first try to resolve the reference using the cannonical URL, and then fall back to direct resolution using the URL as a literal reference if a local version of the canonical resource cannot be found. This approach is safe because the local version cannot be a different artifact than the master copy, though implementations will need to make appropriate arrangements regarding the currency of their local copy of the artifact.

**Contained Resources**

In some circumstances, the content referred to in the resource reference does not have an independent existence apart from the resource that contains it - it cannot be identified independently, and nor can it have its own independent transaction scope. Typically, such circumstances arise where resources are being assembled by a secondary user of the source data, such as a middleware engine. If the data available when the resource is constructed does not include record keys or absolute identification information, then a properly identified resource cannot be assembled, and even if an arbitrary identification was associated with it, the resource could never be the subject of a transaction outside the context of the resource that refers to it.

For example, consider a situation where an interface engine is creating a [Condition](file:///C:\temp\condition.html) record on a patient from an [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) message, and the only information about the primary surgeon is her first name and last name (REL-7.2 & REL-7.3). In the absence of a controlled practitioner directory, this is not enough information to create an identified [Practitioner](file:///C:\temp\practitioner.html) resource since more than one practitioner might have the same name.

In these circumstances, the resource is placed directly in-line in the resource. **This SHOULD NOT be done when the content can be identified properly, as once the identification is lost, it is extremely difficult (and context dependent) to restore it again.**

An example of a contained resource:

<Condition xmlns="http://hl7.org/fhir">

<contained>

<Practitioner>

<id value="p1"/>

<name>

<family value="Person"/>

<given value="Patricia"/>

</name>

</Practitioner>

</contained>

<!-- other attributes -->

<asserter>

<reference value="#p1" />

</asserter>

<!-- other attributes -->

</Condition>

The same example in JSON:

{

"resourceType" : "Condition",

"contained": [

{

"resourceType" : "Practitioner",

"id" : "p1",

"name" : [{

"family" : "Person",

"given" : ["Patricia"]

}]

}],

"asserter" : {

"reference" : "#p1"

}

}

Design Note: Contained resources are still a reference rather than being inlined directly into the element that is the reference (e.g. "custodian" above) to ensure that a single approach to resolving resource references can be used. Though direct containment would seem simpler, it would still be necessary to support internal references where the same contained resource is referenced more than once. In the end, all that it would achieve is creating additional options in the syntax. For users using XPath to process the resource, the following XPath fragment resolves the internal reference:

ancestor::f:\*[not(parent::f:\*)]/f:contained/\*[@id=substring-after(**current()**/f:reference/@value, '#')]

Some notes about use and interpretation of contained resources:

* The contained element SHALL NOT have extensions on it (though contained resources can still contain extensions).
* The contained resource can be put in any resource that inherits from DomainResource. The contained element is then located at the beginnning of the resource after any text narrative and before any extension.
* Contained resources share the same internal id resolution space as the parent resource (for id attributes, see below).
* When resolving references, references are resolved by looking through the 'container' resource - the one that contains the other resources. Since there are no nested contained resources, there is only one container resource.
* References to contained resources are never resolved outside the container resource. Specifically, resolution stops at the elements Bundle.entry.resource and Parameters.parameter.resource, but not at DomainResource.contained.
* Contained resources SHALL NOT contain additional contained resources.
* Contained resources SHALL NOT contain any narrative.
* A contained resource SHALL only be included in a resource if something in that resource (potentially another contained resource) has a reference to it.

**STU Note:** It might be necessary to allow for the reverse (that is, for a contained resource to reference the container). However this would cause many difficult consequences on the API. Feedback on this issue is welcome [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

Resources that are contained inline do not "inherit" context from their parent resource. For instance, if the parent resource contains a "subject", and the contained resource also has a "subject" element defined, there is no implication that the contained resource has the same subject as the parent resource.

Resources can only be contained in other resources if there is a reference from the resource to the contained resource. This is intended to ensure that the meaning of the contained resource is clear, and that there is no confusion as to its significance.

**STU Note:** There are some identified use cases where it would be useful to include resources that refer to the contained resource rather than the container referring to the contained resource, but this has a series of structural ramifications for the API. Whether these can be resolved is an open issue for investigation during the period of trial use.

Feedback is welcome [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**resource-definitions.html**

**Resource Definitions - Detailed Descriptions**

**Resource**

**Reference**

**resourceguide.html**

**Guide to Resources**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [Draft](file:///C:\temp\history.html#pubs) |

The FHIR specification defines a set of resources, and an infrastructure for handling resources. In order to use FHIR to create solutions for integration requirements, implementers must map their problems to resources and their content.

The resources are classified into 6 sections:

1. [**Clinical**](file:///C:\temp\main-pages.html#clinical): The content of a clinical record
2. [**Identification**](file:///C:\temp\main-pages.html#identification): Supporting entities involved in the care process
3. [**Workflow**](file:///C:\temp\main-pages.html#workflow): Manage the healthcare process
4. [**Financial**](file:///C:\temp\main-pages.html#financial): Resources that support the billing and payment parts of FHIR
5. [**Conformance**](file:///C:\temp\main-pages.html#conformance): Resources use to manage specification, development and testing of FHIR solutions
6. [**Infrastructure**](file:///C:\temp\main-pages.html#infrastructure): General functionality, and resources for internal FHIR requirements

This page describes the resources and their functional intent in more detail to assist implementers to understand their purpose and scope, and their supporting classifications.

Where to find common concepts in this specification:

|  |  |  |
| --- | --- | --- |
| **Concept** | **Example** | **Where to find** |
| **Clinical Findings** | | |
| Laboratory Results | Blood panels such as CBC with Differential, Liver Panel, etc. | [DiagnosticReport](file:///C:\temp\diagnosticreport.html) with [Observations](file:///C:\temp\observation.html) |
| Imaging Study Findings | CT Scans, MRI, Plain Radiographs, Ultrasounds) | [DiagnosticReport](file:///C:\temp\diagnosticreport.html) (some with [Observations](file:///C:\temp\observation.html)) |
| Diagnostic Test Results | EKG, pulmonary function test, EEG | [Observations](file:///C:\temp\observation.html) (and maybe a [DiagnosticReport](file:///C:\temp\diagnosticreport.html)) |
| Vital Signs | Temperature, Blood Pressure, Heart Rate, Respiratory Rate | [Observation](file:///C:\temp\observation.html) |
| Other Physical Exam Findings | Auscultation findings | [Observation](file:///C:\temp\observation.html) |
| Pulmonary Artery Catheter readings | Pulmonary artery pressure | [Observation](file:///C:\temp\observation.html) |
| **Patient Problems, Allergies and Adverse Events** | | |
| Allergy | Food or drug allergies | [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) |
| Clinical Diagnosis | Diabetes, Congestive Heart Failure | [Condition](file:///C:\temp\condition.html) |
| Adverse Event | Adverse reaction to an agent, falls, adverse surgical events, hospital infections | [AdverseEvent](file:///C:\temp\adverseevent.html) |
| **Patient History** | | |
| Chief Complaint | Cough, Pain, Fever, Fatigue | [Condition](file:///C:\temp\condition.html) |
| Past Surgical History | Appendectomy, Hernia repair | [Procedure](file:///C:\temp\procedure.html) |
| Past Medical History | Diabetes, Congestive heart failure | [Condition](file:///C:\temp\condition.html) |
| MAR (Medication Administration Record) | Warfarin 5mg PO administered on 12/10/2013 at 3pm | [MedicationAdministration](file:///C:\temp\medicationadministration.html) |
| Home Meds | Warfarin 5mg, 30 day supply, dispensed on 12/01/2013 | [MedicationStatement](file:///C:\temp\medicationstatement.html) |
| Social History | Sexual behavior, Smoking status, Alcohol intake, Illicit drug use | [Observation](file:///C:\temp\observation.html) |
| Family History | Mother has diabetes | [FamilyMemberHistory](file:///C:\temp\familymemberhistory.html) |
| Signs & Symptoms | from a review of systems- Pain, Fever | [Condition](file:///C:\temp\condition.html) |
| **Suggested Physician Orders** | | |
| Proposal for a laboratory test | A blood panel, a stool analysis | [ProcedureRequest](file:///C:\temp\procedurerequest.html) |
| Proposal for an imaging procedure | CT Scan, MRI, X-Rays | [ProcedureRequest](file:///C:\temp\procedurerequest.html) |
| Proposed Diet Order | An oral diet order | [NutritionOrder](file:///C:\temp\nutritionorder.html) |
| Proposed respiratory order | Oxygen delivery | Not done yet |
| Proposed Medications | Aspirin, Lisinopril | [MedicationRequest](file:///C:\temp\medicationrequest.html) |
| Proposed Supply | Wheel Chair, Food Tray | [SupplyRequest](file:///C:\temp\supplyrequest.html) |
| **Interdisciplinary Care Planning** | | |
| Patient Goal | Reduce risk of falls, lose weight | [Goal](file:///C:\temp\goal.html) (as part of a [CarePlan](file:///C:\temp\careplan.html)) |
| Intervention | Patient assessments | [ProcedureRequest](file:///C:\temp\procedurerequest.html) |

In addition, to the information on this page, see also [Common Use Cases](file:///C:\temp\usecases.html).

**Resource Classification**

In addition to the overall classification above, Resources and the data elements in them are tapped into an overall ontology that provides consistent meaning across the resources.

One of the most fundamental aspects of a resource is tense - where it fits into the time line. Some resources are records of past events, some are plans for the future, etc. This table summarizes the categories:

|  |  |
| --- | --- |
| **Type** | **Description** |
| Past | A record of an event that has happened |
| Present | A record that serves an exchange in real time |
| Future | A record that describes a future intent |
| Ongoing | A record that is maintained over time that tracks the state of a patient |
| Unrelated | A record that doesn't relate to time (e.g. a fixed entity) |

Note that even future records end up in the past after the event they are concerned with passes.

**Clinical**

**General**

AllergyIntolerance, Condition, and Procedure are common resources that appear throughout the patient record - summaries of the patient status or history. FamilyMemberHistory tracks significant health issues for relatives of the patient, since these are significant risk factors for the patient. These resources may be found in patient, episode, discharge summaries, and consultation records.

ClinicalImpression (aka ClinicalAssessment) records the core of a clinical consultation/assessment/impression. When a single text note is inserted directly in the patient record, this is typically a ClinicalImpression.summary. A RiskAssessment is an assessment - usually, but not necessarily, by a decision support engine of the likely outcomes of some course of action for a patient.

A DetectedIssue is an active alert that there is a clinical issue with/between one or more active or proposed clinical actions for a patient. One way it may be used is as part of an error message in response to an attempted operation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) | [Ongoing](file:///C:\temp\main-pages.html#ongoing) | Act | This has a [Current List](file:///C:\temp\lifecycle.html#current) |
| [Condition](file:///C:\temp\condition.html) | [Ongoing](file:///C:\temp\main-pages.html#ongoing) | Act | This has a [Current List](file:///C:\temp\lifecycle.html#current) |
| [Procedure](file:///C:\temp\procedure.html) | [Past](file:///C:\temp\main-pages.html#past) | Act | [Clinical Workflow](file:///C:\temp\lifecycle.html#clinical) |
| [FamilyMemberHistory](file:///C:\temp\familymemberhistory.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [ClinicalImpression](file:///C:\temp\clinicalimpression.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [RiskAssessment](file:///C:\temp\riskassessment.html) | [Present](file:///C:\temp\main-pages.html#present) | Act | [Clinical Workflow](file:///C:\temp\lifecycle.html#clinical) |
| [DetectedIssue](file:///C:\temp\detectedissue.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |

**Care Provision**

These resources support planning of care provision - care and treatment plans.

CarePlan and Goal are the primary structures around which future planning for the patient is based. These are used to define care and treatement planned for a patient in the future, and may be updated and adjusted as ongoing care is provided.

ReferralRequest is the basis for a proposed/requested transfer of care from one clinician to another. Typically, these are associated with a transfer of records, and a physical relocation of the patient, but neither of these are required. A ProcedureRequest records a request for a procedure to be carried out.

NutritionOrder is a request to supply a diet, formula feeding (enteral) or oral nutritional supplement to a patient and VisionPrescription contains the details of visual aids requested for a patient.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [CarePlan](file:///C:\temp\careplan.html) | [Ongoing](file:///C:\temp\main-pages.html#ongoing) | Act |  |
| [Goal](file:///C:\temp\goal.html) | [Future](file:///C:\temp\main-pages.html#future) | Act | [Clinical Workflow](file:///C:\temp\lifecycle.html#clinical) |
| [ReferralRequest](file:///C:\temp\referralrequest.html) | [Future](file:///C:\temp\main-pages.html#future) | Act | [Clinical Workflow](file:///C:\temp\lifecycle.html#clinical) |
| [ProcedureRequest](file:///C:\temp\procedurerequest.html) | [Future](file:///C:\temp\main-pages.html#future) | Act | [Clinical Workflow](file:///C:\temp\lifecycle.html#clinical) |
| [NutritionOrder](file:///C:\temp\nutritionorder.html) | [Future](file:///C:\temp\main-pages.html#future)/[Ongoing](file:///C:\temp\main-pages.html#ongoing) | Act |  |
| [VisionPrescription](file:///C:\temp\visionprescription.html) | [Future](file:///C:\temp\main-pages.html#future) | Act |  |

**Medication Management**

Supports the medication and immunization processes. Some points of note:

* Prescription = a MedicationRequest
* In some records, medication orders, administration, and statements may not be well differentiated. Generally, use MedicationStatement if records are unclear
* A medication chart (or variants) will use multiple different types of medication resources
* When prescribing is done by an external system (e.g. most ambulatory e-prescribing), the Medication resource is not used. It is generally used to represent information in formularies and to describe custom formulations

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Medication](file:///C:\temp\medication.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role |  |
| [Medication](file:///C:\temp\medication.html) | [Medication](file:///C:\temp\main-pages.html#medication) | Act | [Clinical Workflow](file:///C:\temp\lifecycle.html#clinical) |
| [MedicationDispense](file:///C:\temp\medicationdispense.html) | [Past](file:///C:\temp\main-pages.html#past) | Act | [Clinical Workflow](file:///C:\temp\lifecycle.html#clinical) |
| [MedicationRequest](file:///C:\temp\medicationrequest.html) | [Future](file:///C:\temp\main-pages.html#future) | Act | [Request/Order](file:///C:\temp\lifecycle.html#order). This has a [current List](file:///C:\temp\lifecycle.html#current) |
| [MedicationStatement](file:///C:\temp\medicationstatement.html) | [Ongoing](file:///C:\temp\main-pages.html#ongoing) / [Past](file:///C:\temp\main-pages.html#past) | Act | [Entity Availability Workflow](file:///C:\temp\lifecycle.html#entity). This has a [Current List](file:///C:\temp\lifecycle.html#current) |
| [Immunization](file:///C:\temp\immunization.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [ImmunizationRecommendation](file:///C:\temp\immunizationrecommendation.html) | [Future](file:///C:\temp\main-pages.html#future) | Act |  |

**Diagnostics**

Resources concerned with observing the patient, and the diagnostic service process built around this. The Observation resource is a widely used general purpose tool. Typical uses include recording the following kinds of data:

* Laboratory Results
* Vital Signs (including Blood Pressure)
* Physical Examinations
* Social history (e.g. Smoking History)

and many more things beside. Note that 'everything is an observation', but implementations should not use Observation where [Condition](file:///C:\temp\condition.html) is more appropriate.

ProcedureRequest and DiagnosticReport support the overall diagnostic process. A DiagnosticReport connects the process of 'reporting', linking process identifiers, visual presentations of a report, and conclusions/interpretations with the Observations. ProcedureRequests capture the request from an authorized clinician to initiate the diagnostic process. Specimen and BodySite are used to record the details of specimen collection although BodySite can also be used with other resources (e.g. Operation location details on a Procedure).

The resources ImagingManifest and ImagingStudy are provided to make links to diagnostic images available within the clinical record, where the actual images will be provided by dedicated systems (the [Media](file:///C:\temp\media.html) resource is intended for sharing actual images directly, which is not generally suitable for high resolution, high volume diagnostic images).

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Observation](file:///C:\temp\observation.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [DiagnosticReport](file:///C:\temp\diagnosticreport.html) | [Past](file:///C:\temp\main-pages.html#past) | Act | [Entity Availability Workflow](file:///C:\temp\lifecycle.html#entity) |
| [ProcedureRequest](file:///C:\temp\procedurerequest.html) | [Future](file:///C:\temp\main-pages.html#future) | Act | [Request/Order](file:///C:\temp\lifecycle.html#order) |
| [Specimen](file:///C:\temp\specimen.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role | [Entity Availability Workflow](file:///C:\temp\lifecycle.html#entity) |
| [BodySite](file:///C:\temp\bodysite.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role |  |
| [ImagingManifest](file:///C:\temp\imagingmanifest.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [ImagingStudy](file:///C:\temp\imagingstudy.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |

**Administrative Resources**

These resources provide support for identifying the various entities involved in healthcare: people, organizations, substances, devices, etc.

**Individuals**

The Patient resource represents the recipient of healthcare. This concept is not always referred to as a 'patient' - other words such as client, customer, resident, etc. are commonly used, but in this specification, the recipient of healthcare is simply known as the patient. The patient resource includes basic demographics and next of kin information.

The Practitioner resource is used to identify and describe any person (or even animal) involved in the provision of care, including doctors, nurses, clerical staff, specialist support staff, etc.

RelatedPerson is used to describe individuals who are related to the patient who are involved in the healthcare process, beyond merely being next of kin - they may sign documents, administer medications, be the source of information, or simply provide care by chance at an accident.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Patient](file:///C:\temp\patient.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role | [Entity Availability Workflow](file:///C:\temp\lifecycle.html#entity) |
| [Practitioner](file:///C:\temp\practitioner.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role |  |
| [RelatedPerson](file:///C:\temp\relatedperson.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role |  |

**Groups**

In most provision of healthcare, some Organization is responsible and accountable for the care, and references appear throughout the clinical record. The organization is a legal entity that may operate at multiple locations. A HealthcareService represents the services offered by one or more organizations at a single location.

A Group is used to describe one or more individuals - usually people, though other kinds of entities are allowed. Typically, these are used with aggregate reporting, though sometimes care is provided to a set of individuals at once (some counseling procedures, public health procedures, herds of cattle, etc.).

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Organization](file:///C:\temp\organization.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role |  |
| [HealthcareService](file:///C:\temp\healthcareservice.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role |  |
| [Group](file:///C:\temp\group.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Entity |  |

**Entities**

Location - for where things happen. Locations are either by instance (e.g. this particular room at a given address or GPS coordinates) or by kind (e.g. the back of some ambulance). The same applies to Substance, for some kind of material (e.g. chemical) used in healthcare - it may refer to an identified instance (e.g. bottle) or a kind of chemical.

The Person resource is used to track associations between different patient records across the provision of healthcare (e.g. "master patient indexes"). Person resources are not referred directly from other resources; their use is for dedicated person registries.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Location](file:///C:\temp\location.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role | [Entity Availability Workflow](file:///C:\temp\lifecycle.html#entity) |
| [Substance](file:///C:\temp\substance.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Entity |  |
| [Person](file:///C:\temp\person.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Entity |  |

**Device**

Device is used to track non-consumable manufactured materials through the healthcare process. It includes implants, large instruments, non-medical things (including software), and containers. Devices - and therefore the device resource - have many purposes, including stock control, locating devices, and tracking implants.

DeviceComponent and DeviceMetric are used to report the status and characteristics of the kinds of medical devices that produce a stream of one or more data points ([Observations](file:///C:\temp\observation.html)), usually about a patient.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Device](file:///C:\temp\device.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role |  |
| [DeviceComponent](file:///C:\temp\devicecomponent.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role |  |
| [DeviceMetric](file:///C:\temp\devicemetric.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role |  |

**Research**

ResearchStudy is used to capture information related to the planning and execution of clinical and pre-clinical studies. It covers both human and veterinary studies and safety, efficacy, comparative effectiveness and retrospective trials.

ResearchSubject capture the information around the participation of a specific individual ([Patient](file:///C:\temp\patient.html)) in a specific ResearchStudy. This information includes their time of participation, associated study arm, status within the study, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [ResearchStudy](file:///C:\temp\researchstudy.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Act |  |
| [ResearchSubject](file:///C:\temp\researchsubject.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Role |  |

**Work Flow**

**Encounters**

Encounter tracks an interaction between a patient and healthcare provider(s). Specifically, this resource tracks the administrative details of the interaction, not the clinical details (though these usual link to it). Aliases for this include 'admission', 'consultation', and sometimes 'appointment'. Encounters may be nested - e.g. when the patient goes to physiotherapy as part of a hospital admission.

An EpisodeOfCare is an association between a patient and a care provider that may last over several encounters. The purpose of EpisodeOfCare is track a provider (person or organization) that has an interest in the ongoing care of the patient.

A Communication is a record of a formal communication with a patient or another party with an interest in their care - a letter, email, phone call etc.

A Flag is warning maintained in the system that alerts care providers to potential issues they should be aware of when providing care to a patient, or at a location.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Encounter](file:///C:\temp\encounter.html) | [Future](file:///C:\temp\main-pages.html#future) / [Ongoing](file:///C:\temp\main-pages.html#ongoing) / [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [EpisodeOfCare](file:///C:\temp\episodeofcare.html) |  | Act |  |
| [Communication](file:///C:\temp\communication.html) | [Past](file:///C:\temp\main-pages.html#past) | Act | [Clinical Workflow](file:///C:\temp\lifecycle.html#clinical) |
| [Flag](file:///C:\temp\flag.html) | [Ongoing](file:///C:\temp\main-pages.html#ongoing) | Act | [Entity Availability Workflow](file:///C:\temp\lifecycle.html#entity) |

**Scheduling**

These resources are concerned with planning times and locations for future delivery of care:

* A Schedule describes a set of times in the future that a service or individual may be available
* A Slot is an instance of a time in a schedule that may be reserved
* An Appointment is proposed or confirmed future event, with a list of participants who may or may not have confirmed their attendance
* An AppointmentResponse is an individual's acceptance or rejection of an appointment

The scheduling systems is closely aligned with the iCal system so that systems may leverage web bookings while maintaining integrity with the rest of the clinical record.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Appointment](file:///C:\temp\appointment.html) | [Future](file:///C:\temp\main-pages.html#future) | Act |  |
| [AppointmentResponse](file:///C:\temp\appointmentresponse.html) | [Future](file:///C:\temp\main-pages.html#future) | Act |  |
| [Schedule](file:///C:\temp\schedule.html) | [Future](file:///C:\temp\main-pages.html#future) |  |  |
| [Slot](file:///C:\temp\slot.html) | [Future](file:///C:\temp\main-pages.html#future) | Act |  |

**Order Management**

Todo: these resources are all draft. Bringing them to maturity and testing their support for real-world scenarios is a major ongoing focus of the FHIR project.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Task](file:///C:\temp\task.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [CommunicationRequest](file:///C:\temp\communicationrequest.html) | [Future](file:///C:\temp\main-pages.html#future) | Act | [Request/Order](file:///C:\temp\lifecycle.html#order) |
| [DeviceRequest](file:///C:\temp\devicerequest.html) | [Present](file:///C:\temp\main-pages.html#present) | Act | [Request/Order](file:///C:\temp\lifecycle.html#order) |
| [DeviceUseStatement](file:///C:\temp\deviceusestatement.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [ProcessRequest](file:///C:\temp\processrequest.html) | [Present](file:///C:\temp\main-pages.html#present) | Act | [Request/Order](file:///C:\temp\lifecycle.html#order) |
| [ProcessResponse](file:///C:\temp\processresponse.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [SupplyDelivery](file:///C:\temp\supplydelivery.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [SupplyRequest](file:///C:\temp\supplyrequest.html) | [Future](file:///C:\temp\main-pages.html#future) | Act |  |

**Infrastructure**

These resources provide generally useful functionality, and/or are referenced directly from the base FHIR framework ([RESTful API](file:///C:\temp\http.html), [messaging](file:///C:\temp\messaging.html), [documents](file:///C:\temp\documents.html)).

**Information Management**

Questionnaire and QuestionnaireResponse are for general purpose data collection from a patient or other information source. They represent the set of questions to be answered, and an individual's responses to those questions. A questionnaire can contain any kind of data, including the data represented in all other resources. It is anticipated that the data will be processed into other kinds of resources for further usage and exchange.

The Provenance resource is used to make statements about where a resource came from - that is, who, what, why, when and where for its creation. This may be used in assessing the integrity, reliability and usefulness of a resource or the data in it. Note that many resources contain some limited amount of this information directly, because it is deemed essential to using/finding/filtering/understanding the resource directly. Whenever information is found in both a resource, and in the provenance statements for the resource, it is expected to be the same.

An AuditEvent is an observation by a system that a record was altered (created, updated, deleted) or otherwise accessed (queried, transferred, copied), with audit information about the event. Typically, the provenance record is created before the resource is altered, by the (sub)system causing the change to the resource, while the audit record is created after the change by the (sub)system responsible for storing it.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Questionnaire](file:///C:\temp\questionnaire.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [QuestionnaireResponse](file:///C:\temp\questionnaireresponse.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [Provenance](file:///C:\temp\provenance.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [AuditEvent](file:///C:\temp\auditevent.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |

**Documents**

In clinical practice, there are use cases for both exchanging information in a highly granular form, and for exchanging it in packages known as 'documents'. This specification provides two different kinds of support for creating and exchanging documents:

* Composition allows for the construction of a detailed [document package](file:///C:\temp\documents.html) using other resources for all content of the document
* DocumentReference and DocumentManifest are used for referring to and/or exchanging external documents ([CDA](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7), PDF, other kinds of records, etc.) either as single documents or groups of documents respectively

The List resource is used to link a set of other kinds of resources into a single list. This might be done for many reasons - maintaining a list of work to do, of patients of interest, or of a set of current problems for a patient.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Composition](file:///C:\temp\composition.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Unrelated |  |
| [DocumentManifest](file:///C:\temp\documentmanifest.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Document | [Entity Availability Workflow](file:///C:\temp\lifecycle.html#entity) |
| [DocumentReference](file:///C:\temp\documentreference.html) |  | Document | [Entity Availability Workflow](file:///C:\temp\lifecycle.html#entity) |
| [List](file:///C:\temp\list.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Act |  |

**Structures**

Basic support structure Resources with general usefulness:

* Media stores an image, video, or sound recording, with metadata that can be used to link it into the rest of the healthcare record as represented in resources
* Binary is a container for content of any type, to include content that is in some format other than a FHIR resource
* Bundle acts as an envelope for a set of other resources as they are gathered together for exchange. Bundles may be long lived (documents) or very short lived (search results)
* Basic is a general container that can be used to represent data for which a specific resource doesn't yet exist (general extension point)

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Media](file:///C:\temp\media.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [Binary](file:///C:\temp\binary.html) | [Unrelated](file:///C:\temp\main-pages.html) | Entity |  |
| [Bundle](file:///C:\temp\bundle.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Entity |  |
| [Basic](file:///C:\temp\basic.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | InfrastructureRoot |  |

**Exchange**

These resources support the exchange process directly:

* MessageHeader is the header for a message - identifies its context, sender and receiver, and content
* OperationOutcome is returned to indicate the detailed outcome of a particular FHIR operation or interaction - success, or failure
* Parameters is input or output to [Operations](file:///C:\temp\operations.html)
* Subscription allows one system to subscribe to events on another system

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [MessageHeader](file:///C:\temp\messageheader.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [OperationOutcome](file:///C:\temp\operationoutcome.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [Parameters](file:///C:\temp\parameters.html) | [Present](file:///C:\temp\main-pages.html#present) | InfrastructureRoot |  |
| [Subscription](file:///C:\temp\subscription.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |

**Conformance**

This specification is a base platform specification that defines a set of general capabilities. Specific systems build on top of that to create interoperable solutions by [making a set of conformance rules](file:///C:\temp\profiling.html). These conformance rules are supported by a this set of "conformance" resources.

**Terminology**

A ValueSet defines a set of codes from one or more [code systems](file:///C:\temp\terminologies.html) that are used for validation, lookup etc. A ConceptMap maps value sets between different code systems.

A NamingSystem resource represents an external provider of codes or identifiers (e.g. an external code system such as SNOMED CT, or an identification system such as an institution MRN). The naming system resource exists to share the identification and registration of these systems, to foster consistency in their identification.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [ValueSet](file:///C:\temp\valueset.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) |  |  |
| [ConceptMap](file:///C:\temp\conceptmap.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) |  |  |
| [NamingSystem](file:///C:\temp\namingsystem.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) |  |  |

**Content**

StructureDefinition and DataElement both describe a set of data elements that can represent data for the purposes of collection, analysis and exchange. They have slightly different purposes:

* StructureDefinition is used specifically to describe the structures used by or with this specification
* DataElement describes more general data items that are represented and exchanged in many formats and/or contexts

StructureMap supports a general purpose mapping language that can be used to transfer data from one structure to another.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [StructureDefinition](file:///C:\temp\structuredefinition.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) |  |  |
| [StructureMap](file:///C:\temp\structuremap.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Unrelated |  |
| [DataElement](file:///C:\temp\dataelement.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) | Entity |  |

**Behavior**

A CapabilityStatement is a statement of a set of system capabilities for use as system discovery, or conformance expectations.

An OperationDefinition describes an operation that can be executed on a server, and a SearchParameter defines a search parameter that can be used on a server. CompartmentDefinition provides information about how resources are associated on a server. MessageDefintion describes the behavior associated with receiving a particular message.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [CapabilityStatement](file:///C:\temp\capabilitystatement.html) | [Unrelated](file:///C:\temp\main-pages.html) |  |  |
| [OperationDefinition](file:///C:\temp\operationdefinition.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) |  |  |
| [SearchParameter](file:///C:\temp\searchparameter.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) |  |  |
| [CompartmentDefinition](file:///C:\temp\compartmentdefinition.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) |  |  |

**Miscellaneous**

ImplementationGuide is used to gather all the other resources into a single package for publication, and also to define the bounds of an interoperability solution.

A TestScript specifies a series of operations and outcomes that a system is expected to meet. TestScripts are for development and acceptance testing, but may also be considered for use with production systems, if the tests are carefully constructed and vetted.

A TestReport contains the test execution results of a TestScript against a series of operations and outcomes that a system is expected to meet. TestReports provide a static recording of TestScript results for subsequent retrieval and review..

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [ImplementationGuide](file:///C:\temp\implementationguide.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) |  |  |
| [TestScript](file:///C:\temp\testscript.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) |  |  |
| [TestReport](file:///C:\temp\testreport.html) | [Unrelated](file:///C:\temp\main-pages.html#unrelated) |  |  |

**Financial**

These resources are all in draft status and more details will be provided here in due course.

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Category** | **RIM** | **Lifecycle** |
| [Coverage](file:///C:\temp\coverage.html) | [Ongoing](file:///C:\temp\main-pages.html#ongoing) | Act |  |
| [EligibilityRequest](file:///C:\temp\eligibilityrequest.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [EligibilityResponse](file:///C:\temp\eligibilityresponse.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [EnrollmentRequest](file:///C:\temp\enrollmentrequest.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [EnrollmentResponse](file:///C:\temp\enrollmentresponse.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [Claim](file:///C:\temp\claim.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [ClaimResponse](file:///C:\temp\claimresponse.html) | [Present](file:///C:\temp\main-pages.html#present) | Act |  |
| [PaymentNotice](file:///C:\temp\paymentnotice.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [PaymentReconciliation](file:///C:\temp\paymentreconciliation.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |
| [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html) | [Past](file:///C:\temp\main-pages.html#past) | Act |  |

**resourcelist-examples.html**

**Resource Examples**

TODO: fill this page out.

**resourcelist.html**

**Resource Index**

* [Categorized](file:///C:\temp\main-pages.html#tabs-1)
* [Alphabetical](file:///C:\temp\main-pages.html#tabs-2)

This page is provided to help find resources quickly. There is also a more [detailed classification, ontology, and description](file:///C:\temp\resourceguide.html). For background to the layout on the layers in this page, see the [Architect's Overview](file:///C:\temp\overview-arch.html#framework).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Foundation | Terminology | | Conformance | Security | | Documents | | Other |
| * [CodeSystem](file:///C:\temp\codesystem.html) * [ValueSet](file:///C:\temp\valueset.html) * [ConceptMap](file:///C:\temp\conceptmap.html) * [ExpansionProfile](file:///C:\temp\expansionprofile.html) * [NamingSystem](file:///C:\temp\namingsystem.html) | | * [StructureDefinition](file:///C:\temp\structuredefinition.html) * [StructureMap](file:///C:\temp\structuremap.html) * [DataElement](file:///C:\temp\dataelement.html) * [CapabilityStatement](file:///C:\temp\capabilitystatement.html) * [MessageDefinition](file:///C:\temp\messagedefinition.html) * [OperationDefinition](file:///C:\temp\operationdefinition.html) * [SearchParameter](file:///C:\temp\searchparameter.html) * [CompartmentDefinition](file:///C:\temp\compartmentdefinition.html) * [ImplementationGuide](file:///C:\temp\implementationguide.html) | * [Provenance](file:///C:\temp\provenance.html) * [AuditEvent](file:///C:\temp\auditevent.html) * [Consent](file:///C:\temp\consent.html) | | * [Composition](file:///C:\temp\composition.html) * [DocumentManifest](file:///C:\temp\documentmanifest.html) * [DocumentReference](file:///C:\temp\documentreference.html) | | * [Basic](file:///C:\temp\basic.html) * [Binary](file:///C:\temp\binary.html) * [Bundle](file:///C:\temp\bundle.html) * [MessageHeader](file:///C:\temp\messageheader.html) * [OperationOutcome](file:///C:\temp\operationoutcome.html) * [Parameters](file:///C:\temp\parameters.html) * [Subscription](file:///C:\temp\subscription.html) * [Linkage](file:///C:\temp\linkage.html) |
|  | | | | | | | | |
| Base | Individuals | | Entities | Workflow | | Management | |  |
| * [Patient](file:///C:\temp\patient.html) * [Practitioner](file:///C:\temp\practitioner.html) * [PractitionerRole](file:///C:\temp\practitionerrole.html) * [RelatedPerson](file:///C:\temp\relatedperson.html) * [Person](file:///C:\temp\person.html) * [Group](file:///C:\temp\group.html) | | * [Organization](file:///C:\temp\organization.html) * [HealthcareService](file:///C:\temp\healthcareservice.html) * [Endpoint](file:///C:\temp\endpoint.html) * [Location](file:///C:\temp\location.html) * [Substance](file:///C:\temp\substance.html) * [Device](file:///C:\temp\device.html) * [DeviceComponent](file:///C:\temp\devicecomponent.html) * [DeviceMetric](file:///C:\temp\devicemetric.html) | * [Task](file:///C:\temp\task.html) * [Appointment](file:///C:\temp\appointment.html) * [AppointmentResponse](file:///C:\temp\appointmentresponse.html) * [Schedule](file:///C:\temp\schedule.html) * [Slot](file:///C:\temp\slot.html) * [ProcessRequest](file:///C:\temp\processrequest.html) * [ProcessResponse](file:///C:\temp\processresponse.html) | | * [Encounter](file:///C:\temp\encounter.html) * [EpisodeOfCare](file:///C:\temp\episodeofcare.html) * [Flag](file:///C:\temp\flag.html) * [List](file:///C:\temp\list.html) * [Library](file:///C:\temp\library.html) | |  |
|  | | | | | | | | |
| Clinical | Summary | | Diagnostics | Medications | | Care Provision | | Request & Response |
| * [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) * [AdverseEvent](file:///C:\temp\adverseevent.html) * [Condition](file:///C:\temp\condition.html) (Problem) * [Procedure](file:///C:\temp\procedure.html) * [FamilyMemberHistory](file:///C:\temp\familymemberhistory.html) * [ClinicalImpression](file:///C:\temp\clinicalimpression.html) * [DetectedIssue](file:///C:\temp\detectedissue.html) | | * [Observation](file:///C:\temp\observation.html) * [DiagnosticReport](file:///C:\temp\diagnosticreport.html) * [Sequence](file:///C:\temp\sequence.html) * [Specimen](file:///C:\temp\specimen.html) * [BodySite](file:///C:\temp\bodysite.html) * [ImagingStudy](file:///C:\temp\imagingstudy.html) * [ImagingManifest](file:///C:\temp\imagingmanifest.html) * [QuestionnaireResponse](file:///C:\temp\questionnaireresponse.html) | * [Medication](file:///C:\temp\medication.html) * [MedicationRequest](file:///C:\temp\medicationrequest.html) * [MedicationAdministration](file:///C:\temp\medicationadministration.html) * [MedicationDispense](file:///C:\temp\medicationdispense.html) * [MedicationStatement](file:///C:\temp\medicationstatement.html) * [Immunization](file:///C:\temp\immunization.html) * [ImmunizationRecommendation](file:///C:\temp\immunizationrecommendation.html) | | * [CarePlan](file:///C:\temp\careplan.html) * [CareTeam](file:///C:\temp\careteam.html) * [Goal](file:///C:\temp\goal.html) * [ReferralRequest](file:///C:\temp\referralrequest.html) * [RequestGroup](file:///C:\temp\requestgroup.html) * [ProcedureRequest](file:///C:\temp\procedurerequest.html) * [RiskAssessment](file:///C:\temp\riskassessment.html) * [VisionPrescription](file:///C:\temp\visionprescription.html) * [NutritionOrder](file:///C:\temp\nutritionorder.html) | | * [DeviceRequest](file:///C:\temp\devicerequest.html) * [DeviceUseStatement](file:///C:\temp\deviceusestatement.html) * [SupplyRequest](file:///C:\temp\supplyrequest.html) * [SupplyDelivery](file:///C:\temp\supplydelivery.html) * [Communication](file:///C:\temp\communication.html) * [CommunicationRequest](file:///C:\temp\communicationrequest.html) |
|  | | | | | | | | |
| Financial | Support | | Billing | Payment | | General | |  |
| * [Coverage](file:///C:\temp\coverage.html) * [EligibilityRequest](file:///C:\temp\eligibilityrequest.html) * [EligibilityResponse](file:///C:\temp\eligibilityresponse.html) * [EnrollmentRequest](file:///C:\temp\enrollmentrequest.html) * [EnrollmentResponse](file:///C:\temp\enrollmentresponse.html) | | * [Claim](file:///C:\temp\claim.html) * [ClaimResponse](file:///C:\temp\claimresponse.html) | * [PaymentNotice](file:///C:\temp\paymentnotice.html) * [PaymentReconciliation](file:///C:\temp\paymentreconciliation.html) | | * [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html) * [Contract](file:///C:\temp\contract.html) * [Account](file:///C:\temp\account.html) * [ChargeItem](file:///C:\temp\chargeitem.html) | |  |
|  | | | | | | | | |
| Specialized | Public Health & Research | | Definitional Artifacts | Clinical Decision Support | | Quality Reporting | | Testing |
| * [ResearchStudy](file:///C:\temp\researchstudy.html) * [ResearchSubject](file:///C:\temp\researchsubject.html) | | * [Questionnaire](file:///C:\temp\questionnaire.html) * [ActivityDefinition](file:///C:\temp\activitydefinition.html) * [ServiceDefinition](file:///C:\temp\servicedefinition.html) * [PlanDefinition](file:///C:\temp\plandefinition.html) | * [GuidanceResponse](file:///C:\temp\guidanceresponse.html) | | * [Measure](file:///C:\temp\measure.html) * [MeasureReport](file:///C:\temp\measurereport.html) | | * [TestScript](file:///C:\temp\testscript.html) * [TestReport](file:///C:\temp\testreport.html) |
|  | | | | | | | | |
| **Alphabetical** | | | | | | | | |
| **A-D:**   * [Account](file:///C:\temp\account.html) * [ActivityDefinition](file:///C:\temp\activitydefinition.html) * [AllergyIntolerance](file:///C:\temp\allergyintolerance.html) * [AdverseEvent](file:///C:\temp\adverseevent.html) * [Appointment](file:///C:\temp\appointment.html) * [AppointmentResponse](file:///C:\temp\appointmentresponse.html) * [AuditEvent](file:///C:\temp\auditevent.html) * [Basic](file:///C:\temp\basic.html) * [Binary](file:///C:\temp\binary.html) * [BodySite](file:///C:\temp\bodysite.html) * [Bundle](file:///C:\temp\bundle.html) * [CapabilityStatement](file:///C:\temp\capabilitystatement.html) * [CarePlan](file:///C:\temp\careplan.html) * [CareTeam](file:///C:\temp\careteam.html) * [ChargeItem](file:///C:\temp\chargeitem.html) * [Claim](file:///C:\temp\claim.html) * [ClaimResponse](file:///C:\temp\claimresponse.html) * [ClinicalImpression](file:///C:\temp\clinicalimpression.html) * [CodeSystem](file:///C:\temp\codesystem.html) * [Communication](file:///C:\temp\communication.html) * [CommunicationRequest](file:///C:\temp\communicationrequest.html) * [CompartmentDefinition](file:///C:\temp\compartmentdefinition.html) * [Composition](file:///C:\temp\composition.html) * [ConceptMap](file:///C:\temp\conceptmap.html) * [Condition](file:///C:\temp\condition.html) (aka Problem) * [Consent](file:///C:\temp\consent.html) * [Contract](file:///C:\temp\contract.html) * [Coverage](file:///C:\temp\coverage.html) * [DataElement](file:///C:\temp\dataelement.html) | | **D-I:**   * [DetectedIssue](file:///C:\temp\detectedissue.html) * [Device](file:///C:\temp\device.html) * [DeviceComponent](file:///C:\temp\devicecomponent.html) * [DeviceMetric](file:///C:\temp\devicemetric.html) * [DeviceRequest](file:///C:\temp\devicerequest.html) * [DeviceUseStatement](file:///C:\temp\deviceusestatement.html) * [DiagnosticReport](file:///C:\temp\diagnosticreport.html) * [DocumentManifest](file:///C:\temp\documentmanifest.html) * [DocumentReference](file:///C:\temp\documentreference.html) * [EligibilityRequest](file:///C:\temp\eligibilityrequest.html) * [EligibilityResponse](file:///C:\temp\eligibilityresponse.html) * [Encounter](file:///C:\temp\encounter.html) * [Endpoint](file:///C:\temp\endpoint.html) * [EnrollmentRequest](file:///C:\temp\enrollmentrequest.html) * [EnrollmentResponse](file:///C:\temp\enrollmentresponse.html) * [EpisodeOfCare](file:///C:\temp\episodeofcare.html) * [ExpansionProfile](file:///C:\temp\expansionprofile.html) * [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html) * [FamilyMemberHistory](file:///C:\temp\familymemberhistory.html) * [Flag](file:///C:\temp\flag.html) * [Goal](file:///C:\temp\goal.html) * [Group](file:///C:\temp\group.html) * [GuidanceResponse](file:///C:\temp\guidanceresponse.html) * [HealthcareService](file:///C:\temp\healthcareservice.html) * [ImagingManifest](file:///C:\temp\imagingmanifest.html) * [ImagingStudy](file:///C:\temp\imagingstudy.html) | | | **I-P:**   * [Immunization](file:///C:\temp\immunization.html) * [ImmunizationRecommendation](file:///C:\temp\immunizationrecommendation.html) * [ImplementationGuide](file:///C:\temp\implementationguide.html) * [Library](file:///C:\temp\library.html) * [Linkage](file:///C:\temp\linkage.html) * [List](file:///C:\temp\list.html) * [Location](file:///C:\temp\location.html) * [Measure](file:///C:\temp\measure.html) * [MeasureReport](file:///C:\temp\measurereport.html) * [Media](file:///C:\temp\media.html) * [Medication](file:///C:\temp\medication.html) * [MedicationAdministration](file:///C:\temp\medicationadministration.html) * [MedicationDispense](file:///C:\temp\medicationdispense.html) * [MedicationRequest](file:///C:\temp\medicationrequest.html) * [MedicationStatement](file:///C:\temp\medicationstatement.html) * [MessageDefinition](file:///C:\temp\messagedefinition.html) * [MessageHeader](file:///C:\temp\messageheader.html) * [NamingSystem](file:///C:\temp\namingsystem.html) * [NutritionOrder](file:///C:\temp\nutritionorder.html) * [Observation](file:///C:\temp\observation.html) * [OperationDefinition](file:///C:\temp\operationdefinition.html) * [OperationOutcome](file:///C:\temp\operationoutcome.html) * [Organization](file:///C:\temp\organization.html) * [Parameters](file:///C:\temp\parameters.html) * [Patient](file:///C:\temp\patient.html) * [PaymentNotice](file:///C:\temp\paymentnotice.html) * [PaymentReconciliation](file:///C:\temp\paymentreconciliation.html) * [Person](file:///C:\temp\person.html) * [PlanDefinition](file:///C:\temp\plandefinition.html) | | **P-Z:**   * [Practitioner](file:///C:\temp\practitioner.html) * [PractitionerRole](file:///C:\temp\practitionerrole.html) * [Procedure](file:///C:\temp\procedure.html) * [ProcedureRequest](file:///C:\temp\procedurerequest.html) * [ProcessRequest](file:///C:\temp\processrequest.html) * [ProcessResponse](file:///C:\temp\processresponse.html) * [Provenance](file:///C:\temp\provenance.html) * [Questionnaire](file:///C:\temp\questionnaire.html) * [QuestionnaireResponse](file:///C:\temp\questionnaireresponse.html) * [ReferralRequest](file:///C:\temp\referralrequest.html) * [RelatedPerson](file:///C:\temp\relatedperson.html) * [RequestGroup](file:///C:\temp\requestgroup.html) * [ResearchStudy](file:///C:\temp\researchstudy.html) * [ResearchSubject](file:///C:\temp\researchsubject.html) * [RiskAssessment](file:///C:\temp\riskassessment.html) * [Schedule](file:///C:\temp\schedule.html) * [SearchParameter](file:///C:\temp\searchparameter.html) * [Sequence](file:///C:\temp\sequence.html) * [ServiceDefinition](file:///C:\temp\servicedefinition.html) * [Slot](file:///C:\temp\slot.html) * [Specimen](file:///C:\temp\specimen.html) * [StructureDefinition](file:///C:\temp\structuredefinition.html) * [StructureMap](file:///C:\temp\structuremap.html) * [Subscription](file:///C:\temp\subscription.html) * [Substance](file:///C:\temp\substance.html) * [SupplyDelivery](file:///C:\temp\supplydelivery.html) * [SupplyRequest](file:///C:\temp\supplyrequest.html) * [Task](file:///C:\temp\task.html) * [TestScript](file:///C:\temp\testscript.html) * [TestReport](file:///C:\temp\testreport.html) * [ValueSet](file:///C:\temp\valueset.html) * [VisionPrescription](file:///C:\temp\visionprescription.html) | |

Additional Resources will be added in the future. A list of hypothesized resources can be found on the [HL7 wiki](http://wiki.hl7.org/index.php?title=FHIR_Resource_Types). Feel free to add any you think are missing or engage with one of the [HL7 Work Groups](http://www.hl7.org/Special/committees/index.cfm) to submit a [proposal](http://wiki.hl7.org/index.php?title=Category:FHIR_Resource_Proposal) to define a resource of particular interest.

**resources-definitions.html**

**Resources - Detailed Descriptions**

**Resource**

**resources-examples.html**

**Resource Format Examples**

The specification includes many example resources. Most of the examples can be found in the examples section for each defined resource. In addition, there are some general examples of sets of resources:

* [Integrated Examples](file:///C:\temp\integrated-examples.html)

**resources.html**

**rxnorm.html**

**Using RxNorm with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Summary**

|  |  |
| --- | --- |
| Source | RxNorm is made available by the [US National Library of Medicine](http://www.nlm.nih.gov/) at <http://www.nlm.nih.gov/research/umls/rxnorm> |
| System | The URI <http://www.nlm.nih.gov/research/umls/rxnorm> identifies the RxNorm code system |
| Version | Where a version is used, it should be the date of release, encoded as in the download files, e.g. "07092014" |
| Code | The code value for an RxNorm code is a Concept Identifier (CUI), and only CUIs for which there is an SAB=RXNORM |
| Display | The correct display for a CUI is the string description for it associated with the source RXNORM. Display values are not case sensitive |
| Inactive | Todo: Describe how it is determined which concepts are inactive |
| Subsumption | No Subsumption relationships are defined by RxNorm |
| Filter Properties | Several properties are defined as described below |

**RxNorm MySQL Database**

The RxNorm scripts are able to populate a MySQL database that contains the data from RxNorm. This page provides SQL statements that describe how to implement the features of the RxNorm terminology correctly against this database. These are provided for implementer convenience, and do not imply that any particular approach be used in implementations.

For example, the correct display name for a CUI is 'Select STR from rxnconso where RXCUI = :code and SAB = 'RXNORM' and TTY <> 'SY'.

**Copyright/License Issues**

Using RxNorm codes of type SAB=RXNORM as this specification describes does not require a UMLS license. Access to the full set of RxNorm definitions, and/or additional use of other RxNorm structures and information requires a UMLS license. The use of RxNorm in this specification is pursuant to HL7's status as a licensee of the NLM UMLS. HL7's license does not convey the right to use RxNorm to any users of this specification; implementers must acquire a license to use RxNorm in their own right.

**RxNorm Filter Properties**

This section documents the property filters that can be used with the RxNorm code system in value set composition statements.

The base SQL statement for returning a list of CUIs that conform to these filters is:

Select RXCUI from rxnconso where SAB = 'RXNORM' and TTY <> 'SY'

**Semantic Type**

|  |  |
| --- | --- |
| Description | Allows for the selection of a set of CUIs based on their Semantic Type |
| Property Name | STY |
| Operations Allowed | = / in |
| Values Allowed | [column:]value |
| Comments | If not column is specified, the default column is TUI |
| SQL | and RXCUI in (select RXCUI from rxnsty where [:column] = :value) |

**Source**

|  |  |
| --- | --- |
| Description | Allows for the selection of a set of concepts that have mappings to a particular RxNorm concept |
| Property Name | SAB |
| Operations Allowed | = / in |
| Values Allowed | Values from RxNorm SAB table (e.g. select RSAB from rxnsab) |
| SQL | and RXCUI in (select RXCUI from rxnconso where SAB = :value) |

**Term Type**

|  |  |
| --- | --- |
| Description | Allows for the selection of a concept based on its designated type |
| Property Name | TTY |
| Operations Allowed | = / in |
| Values Allowed | TTY values from the RxNorm Concept table (e.g. select distinct TTY from rxnconso) |
| SQL | and TTY = :value |

**Relationship**

|  |  |
| --- | --- |
| Description | Allows for the selection of a concept based on its relationships |
| Property Name | [REL] |
| Operations Allowed | = / in |
| Values Allowed | CUI:[RXCUI] or AUI:[RXAUI] must be a valid CUI or AUI. Note that a CUI does not need to have an SAB=RXNORM entry to be used here |
| Comments | [REL] (:rel) is one of SY, SIB, RN, PAR, CHD, RB or RO |
| SQL | for CUI:  and (RXCUI in (select RXCUI from rxnconso where RXCUI in (select RXCUI1 from rxnrel where REL = :rel and RXCUI2 = :value))  for AUI:  and (RXCUI in (select RXCUI from rxnconso where RXAUI in (select RXAUI1 from rxnrel where REL = :rel and RXAUI2 = :value)) |

**Relationship Type**

|  |  |
| --- | --- |
| Description | Allows for the selection of a concept based on the type of its relationships |
| Property Name | [RELA] |
| Operations Allowed | = / in |
| Values Allowed | CUI:[RXCUI] or AUI:[RXAUI] must be a valid CUI or AUI. Note that a CUI does not need to have an SAB=RXNORM entry to be used here |
| Comments | [RELA] (:rela) is one of the relationship types defined in [RxNorm Appendix 1](https://www.nlm.nih.gov/research/umls/rxnorm/docs/2016/appendix1.html) (the "RELA" column) |
| SQL | for CUI:  and (RXCUI in (select RXCUI from rxnconso where RXCUI in (select RXCUI1 from rxnrel where RELA = :rel and RXCUI2 = :value))  for AUI:  and (RXCUI in (select RXCUI from rxnconso where RXAUI in (select RXAUI1 from rxnrel where RELA = :rel and RXAUI2 = :value)) |

**RxNorm Properties**

In addition to the [standard properties](file:///C:\temp\terminology-service.html#standard-props), the following properties are defined for RxNorm:

(Yet to be done).

**Implicit Value Sets**

Implicit value sets are those whose specification can be predicted based on the grammar of the underlying code system, and the known structure of the URL that refers to them. At the time of this publication, RxNorm does not define implicit value sets.

The identifier http://www.nlm.nih.gov/research/umls/rxnorm/vs represents a value set that contains all RxNorm CUIs.

**safety.html**

**Clinical Safety**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This specification defines data elements, resources, formats, methods and APIs for exchanging healthcare data between different participants in the healthcare process. As such, Clinical Safety is a key concern with regard to the specification and it's many and various implementations.

This page needs further development.

**Implementer's Safety Check List**

FHIR is as simple to implement as we know how to make it. However, due to the nature of healthcare, and healthcare processes, and cultural concerns, there are a number of features in FHIR that implementers are obliged to consider in order to implement safe systems.

This section is a check list to help implementers be sure that they've considered all the parts of FHIR that impact on their system design with regard to safety.

1.  Production exchange of patient or other sensitive data will always use some form of [encryption on the wire](file:///C:\temp\security.html#http)
2.  For each resource that my system handles, I've reviewed the [Modifier elements](file:///C:\temp\conformance-rules.html#isModifier)
3.  My system checks for [modifierExtension](file:///C:\temp\extensibility.html#modifierExtension) elements
4.  My system supports [elements labelled as "must-support"](file:///C:\temp\conformance-rules.html#mustSupport) in the [profiles](file:///C:\temp\profiling.html) that apply to my system
5.  For each resource that my system handles, my system handles the full [Life cycle](file:///C:\temp\lifecycle.html) (status codes, currency issues, and erroneous entry status)
6.  My system can [render narratives properly](file:///C:\temp\narrative.html#css) (where they are used)
7.  My system has documented how [distributed resource identification](file:///C:\temp\managing.html#distributed) works in its relevant contexts of use, and where (and why) [contained](file:///C:\temp\references.html#contained) resources are used
8.  My system manages lists of [current resources](file:///C:\temp\lifecycle.html#current) correctly
9.  My system makes the right [Provenance](file:///C:\temp\provenance.html) statements and [AuditEvent](file:///C:\temp\auditevent.html) logs, and uses the right [security labels](file:///C:\temp\security-labels.html#core) where appropriate
10.  My system checks that the right [Patient consent](file:///C:\temp\consent.html) has been granted (where applicable)
11.  When other systems [return http errors from the RESTful API](file:///C:\temp\http.html#summary) and [Operations](file:///C:\temp\operations.html) (perhaps using [Operation Outcome](file:///C:\temp\operationoutcome.html)), my system checks for them and handles them appropriately
12.  My system publishes a [Capability Statement](file:///C:\temp\capabilitystatement.html) with [StructureDefinitions](file:///C:\temp\structuredefinition.html), [ValueSets](file:///C:\temp\valueset.html), and [OperationDefinitions](file:///C:\temp\operationdefinition.html), etc., so other implementers know how the system functions
13.  My system caters for [parameters that have missing values](file:///C:\temp\search.html#missing) when doing search operations
14.  My system ensures that system clocks are synchronised using a protocol like NTP or SNTP, or my server is robust against clients that have the wrong clock set
15.  My system ensures checks for patient links (and/or merges) and handles data that is linked to patients accordingly

Obviously this list is only a small part of the overall safety check list for an application, which will have checks regarding jurisdictionally mandated policies, internal integrity, etc.

In addition, server developers should check these specific additional checks for client convenience:

1.  Server: CORS ([cross-origin resource sharing](http://enable-cors.org/)) is enabled (many clients are javascript apps running in a browser)
2.  JSON is supported (many clients are javascript apps running in a browser; XML is inconvenient at best)
3.  JSON is returned correctly when errors happen (clients often don't handle HTML errors well)
4.  the \_format header is supported correctly

**sc.html**

**FHIR Status Codes Grid**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This is an internal QA page. The intent of this table is to help reviewers compare the usage of status codes across resources: where the status code means the same thing, it should have the same code. The columns have no formal definition; their intent is to be inferred from the codes listed in the column.

Note: This page is not proscriptive; it's descriptive and comparative, to assist with review. The FHIR methodology says that if two domains use different terms for the same concept (e.g. 'withdrawn' and 'cancelled') then the domain concept name should be used. Conversely, where there is no explicit reason for the codes to differ based on domain usage, they should not.

Note: **Bold** indicates that the element is marked as a modifier.

**search.html**

**Search**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Searching for resources is fundamental to the mechanics of FHIR. Search operations traverse through an existing set of resources filtering by parameters supplied to the search operation. The text below describes the FHIR search framework, starting with simple cases moving to the more complex. Implementers need only implement the amount of complexity that they require for their implementations.

**Summary Table**

|  |  |  |
| --- | --- | --- |
| **Search Parameter Types** | **Parameters for all resources** | **Search result parameters** |
|  | [\_id](file:///C:\temp\main-pages.html#id) [\_lastUpdated](file:///C:\temp\main-pages.html#lastUpdated) [\_tag](file:///C:\temp\main-pages.html#tag) [\_profile](file:///C:\temp\main-pages.html#profile) [\_security](file:///C:\temp\main-pages.html#security) [\_text](file:///C:\temp\main-pages.html#text) [\_content](file:///C:\temp\main-pages.html#content) [\_list](file:///C:\temp\main-pages.html#list) [\_has](file:///C:\temp\main-pages.html#has) [\_type](file:///C:\temp\main-pages.html#_type) [\_query](file:///C:\temp\main-pages.html#query) | [\_sort](file:///C:\temp\main-pages.html#sort) [\_count](file:///C:\temp\main-pages.html#count) [\_include](file:///C:\temp\main-pages.html#include) [\_revinclude](file:///C:\temp\main-pages.html#revinclude) [\_summary](file:///C:\temp\main-pages.html#summary) [\_elements](file:///C:\temp\main-pages.html#elements) [\_contained](file:///C:\temp\main-pages.html#contained) [\_containedType](file:///C:\temp\main-pages.html#containedType) |

In addition, there is a special search parameter called [\_filter](file:///C:\temp\main-pages.html#filter) that allows for an alternative method of searching.

Also, there is a single page that lists [all the search parameters](file:///C:\temp\searchparameter-registry.html).

**Introduction**

In the simplest case, a search is executed by performing a GET operation in the RESTful framework:

GET [base]/[type]?name=value&...{&\_format=[mime-type]}}

For this RESTful search (see [definition in RESTful API](file:///C:\temp\http.html#search)), the parameters are a series of name=[value] pairs encoded in the URL or as an application/x-www-form-urlencoded submission for a POST:

POST [base]/[type]/\_search{?[parameters]{&\_format=[mime-type]}}

The server determines which of the set of resources it serves meet the specific criteria, and returns the results in the HTTP response as a [bundle](file:///C:\temp\bundle.html) which includes the resources that are the results of the search. Note that the \_format parameter works for search [like for other interactions](file:///C:\temp\http.html#mime-type).

[Search operations](file:///C:\temp\http.html#search) are executed in one of three defined contexts that control which set of resources are being searched:

* A specified resource type: GET [base]/[type]?parameter(s)
* A specified [compartment](file:///C:\temp\compartmentdefinition.html), perhaps with a specified resource type in that compartment: GET [base]/Patient/[id]/[type]?parameter(s)
* All resource types: GET [base]?parameter(s) (parameters common to all types only)

Search operations can also be implemented in [the messaging framework](file:///C:\temp\messaging.html#search).

The server determines which of their resources meet the criteria contained in the search parameters as described below. However the server has the prerogative to return additional search results if it believes them to be relevant. Note: There is a special search for the most relevant context in which the search set is indeterminate: [Patient MPI Search](file:///C:\temp\patient.html#mpi).

Search using GET may include sensitive information in the search parameters. Therefore secure communications and endpoint management are recommended, see [Security Communications](file:///C:\temp\security.html#http)

**Handling Errors**

If a server is unable to execute a search request, it may return an error. A HTTP status code of 403 signifies that the server refused to perform the search, while other 4xx and 5xx codes signify that some sort of error has occurred. When the search fails, a server SHOULD return an [OperationOutcome](file:///C:\temp\operationoutcome.html) detailing the cause of the failure. Note: An empty search result is not a failure.

In some cases, parameters may cause an error. For instance:

* A parameter may refer to a non-existent resource e.g. GET [base]/Observation?subject=101, where "101" does not exist
* A parameter may refer to an unknown code e.g. GET [base]/Observation?code=loinc|1234-1, where the LOINC code "1234-1" is not known to the server
* A parameter may refer to a time that is out of scope e.g. GET [base]/Condition?onset=le1995, where the system only has data going back to 2001
* A parameter may use an illegal or unacceptable modifier e.g. GET [base]/Condition?onset:text=1995, where the modifier cannot be processed by the server
* A data time parameter may have incorrect format e.g. GET [base]/Condition?onset=23%20May%202009
* A parameter may be unknown or unsupported (see below)

Where the content of the parameter is syntactically incorrect, servers SHOULD return an error. However where the issue is a logical condition (e.g. unknown subject or code), the server SHOULD process the search, including processing the parameter - with the result of returning an empty search set, since the parameter cannot be satisfied.

In such cases, the search process MAY include an [OperationOutcome](file:///C:\temp\operationoutcome.html) in the search set that contains additional hints and warnings about the search process. This is included in the search results as an entry with [search mode](file:///C:\temp\bundle-definitions.html#Bundle.entry.search.mode) = [outcome](file:///C:\temp\valueset-search-entry-mode.html). Clients can use this information to improve future searches.

**Unknown and unsupported parameters**

Servers may receive parameters from the client that they do not recognise, or may receive parameters they recognise but do not support (either in general, or for a specific search). In general, servers SHOULD ignore unknown or unsupported parameters for the following reasons:

* Various HTTP stacks and proxies may add parameters that aren't under the control of the client
* The client can determine what parameters the server used by examing the self link in the return (see [below](file:///C:\temp\main-pages.html#conformance))

Clients can specify how the server should behave, by using the prefer header

* Prefer: handling=strict: Client requests that the server return an error for any unknown or unsupported parameter
* Prefer: handling=lenient: Client requests that the server ignore any unknown or unsupported parameter

Servers SHOULD honor the client's request, but are not required to do so.

**Standard Parameters**

**Parameters for all resources**

The following parameters apply [to all resources](file:///C:\temp\resource.html#search): . In addition, the search parameter [\_text](file:///C:\temp\main-pages.html#text) and [\_filter](file:///C:\temp\main-pages.html#filter), (documented below) also applies to all resources (as do the search result parameters).

The search parameter \_id refers to the logical id of the resource, and can be used when the search context specifies a resource type:

GET [base]/Patient?\_id=23

This search finds the patient resource with the given id (there can only be one resource for a given id). Functionally, this is equivalent to a [simple read operation](file:///C:\temp\http.html#read):

GET [base]/Patient/23

However, the search with parameter \_id returns a bundle with the requested resource, instead of just the resource itself. Additional parameters can be added which may provide additional functionality on top of this base read equivalence (e.g. [\_include](file:///C:\temp\main-pages.html#include)). Note that although the \_id parameter has a type of token, because servers SHALL use exact match with it, there is no system for the \_id parameter.

The search parameter \_lastUpdated can be used to select resources based on the last time they were changed:

GET [base]/Observation?\_lastUpdated=gt2010-10-01

This search finds any observations changed since 1-Oct 2010. When this search parameter is used, applications should consider synchronization approaches ([RESTful history](file:///C:\temp\http.html#history) or the [Subscription resource](file:///C:\temp\subscription.html)).

The search parameters [\_tag](file:///C:\temp\resource.html#simple-tags), [\_profile](file:///C:\temp\resource.html#profile-tags) and [\_security](file:///C:\temp\resource.html#security-labels) parameters search on the equivalent elements in the [meta element](file:///C:\temp\resource.html#meta). For example,

GET [base]/Condition?\_tag=http://acme.org/codes|needs-review

searches for all Condition resources with the tag:

{

"system" : "http://acme.org/codes",

"code" : "needs-review"

}

In the same manner:

GET [base]/DiagnosticReport?\_profile=http://hl7.org/fhir/StructureDefinition/lipid

GET [base]/DiagnosticReport?\_profile=Profile/lipid

restricts the search to only DiagnosticReport resources that are tagged as conforming to a particular profile. The second reference is relative, and refers a local profile on the same server.

[\_tag](file:///C:\temp\resource.html#simple-tags), [\_profile](file:///C:\temp\resource.html#profile-tags) and [\_security](file:///C:\temp\resource.html#security-labels) parameters are all token types (see [below](file:///C:\temp\main-pages.html#token)).

**Parameters for each resource**

In addition to the \_id parameter which exists for all resources, each FHIR resource type defines its own set of search parameters with their names, types, and meanings. These search parameters are on the same page as the resource definitions, and are also published as part of the standard Capability statement ([XML](file:///C:\temp\capabilitystatement-base.xml.html) or [JSON](file:///C:\temp\capabilitystatement-base.json.html)).

In general, the defined search parameters correspond to a single element in the resource, but this is not required, and some search parameters refer to the same type of element in multiple places, or refer to derived values.

Some search parameters defined by resources are associated with more than one path in a resource. This means that the search parameter matches if any of the paths contain matching content. If a path matches, the whole resource is returned in the search results. The client may have to examine the resource to determine which path contains the match.

Servers are not required to implement any of the standard search parameters (except for the \_id parameter described above), and may define their own parameters.

**Search Parameter Types**

Each search parameter is defined by a type that specifies how the search parameter behaves. These are the defined parameter types:

The search parameters can also append "modifiers" that control their behavior. The kinds of modifiers that available is dependent on the type of parameter being modified.

**Modifiers**

Parameters are defined per resource. Parameter names may specify a modifier as a suffix. The modifiers are separated from the parameter name by a colon. Modifiers are:

* For all parameters (except combination): :missing; e.g. gender:missing=true (or false). Searching for gender:missing=true will return all the resources that don't have a value for the gender parameter (which usually equates to not having the relevant element in the resource). Searching for gender:missing=false will return all the resources that have a value for the gender parameter.
* For string: :exact (the match needs to be exact, no partial matches, case sensitive and accent-sensitive), or :contains (case insensitive and accent-insensitive, partial match at start or end), instead of the default behavior (case insensitive and accent-insensitive, partial matches at the end of the string).
* For token: :text (the match does a partial searches on the text portion of a CodeableConcept or the display portion of a Coding), instead of the default search which uses codes. Other defined modifiers are :in, :below, :above and :not-in which are described below.
* For reference: :[type] where [type] is the name of a type of resource
* For uri: :below, :above indicate that instead of an exact match, either the search term left matches the value, or vice-versa.

Server SHALL reject any search request that contains is suffixed by a modifier that the server does **not** support for that parameter. For example, if the server supports the name search param, but not the :exact modifier on the name, it should reject a search with the parameter name:exact=Bill, using an HTTP 400 error with an [OperationOutcome](file:///C:\temp\operationoutcome.html) with a [clear error message](file:///C:\temp\operationoutcome-example-searchfail.html).

**Prefixes**

For the ordered parameter types of [number](file:///C:\temp\main-pages.html#number), [date](file:///C:\temp\main-pages.html#date), and [quantity](file:///C:\temp\main-pages.html#quantity), a prefix to the parameter value may be used to control the nature of the matching. To avoid URL escaping and visual confusion, the following prefixes are used:

|  |  |  |
| --- | --- | --- |
| eq | the value for the parameter in the resource is equal to the provided value | the range of the search value fully contains the range of the target value |
| ne | the value for the parameter in the resource is not equal to the provided value | the range of the search value does not fully contain the range of the target value |
| gt | the value for the parameter in the resource is greater than the provided value | the range above the search value intersects (i.e. overlaps) with the range of the target value |
| lt | the value for the parameter in the resource is less than the provided value | the range below the search value intersects (i.e. overlaps) with the range of the target value |
| ge | the value for the parameter in the resource is greater or equal to the provided value | the range above the search value intersects (i.e. overlaps) with the range of the target value, or the range of the search value fully contains the range of the target value |
| le | the value for the parameter in the resource is less or equal to the provided value | the range below the search value intersects (i.e. overlaps) with the range of the target value or the range of the search value fully contains the range of the target value |
| sa | the value for the parameter in the resource starts after the provided value | the range of the search value does not overlap with the range of the target value, and the range above the search value contains the range of the target value |
| eb | the value for the parameter in the resource ends before the provided value | the range of the search value does overlap not with the range of the target value, and the range below the search value contains the range of the target value |
| ap | the value for the parameter in the resource is approximately the same to the provided value. Note that the recommended value for the approximation is 10% of the stated value (or for a date, 10% of the gap between now and the date), but systems may choose other values where appropriate | the range of the search value overlaps with the range of the target value |

If no prefix is present, the prefix eq is assumed. Note that the way search parameters operate is not the same as the way the operations on two numbers work in a mathematical sense. sa (starts-after) and eb (ends-before) are not used with integer values.

For each prefix above, two interpretations are provided - the simple intent of the prefix and the interpretation of the parameter when applied to ranges. The range interpretation is provided for decimals and dates. Searches are always performed on values that are implicitly or explicitly a range. For instance, the number 2.0 has an implicit range of 1.95 to 2.05, and the date 2015-08-12 has an implicit range of the all the time during that day. If the target value is a [Range](file:///C:\temp\datatypes.html#range), a [Period](file:///C:\temp\datatypes.html#period), or a [Timing](file:///C:\temp\datatypes.html#timing), then the target is explicitly a range. Three ranges are identified:

|  |  |  |
| --- | --- | --- |
| range of the value | The limits implied by the precision of the value | The number 2.0 has a range of 1.95 to 2.05 The date 2015-08-12 has a range from 00:00 to 00:00 exclusive |
| range below the value | Up to the specified value | The range below 2.0 includes any value less or equal to <2.00000000000000000000 The range before 2015-08-12T05:23:45 includes any time up to 2015-08-12T05:23:45.000000000000000 |
| range above the value | The specified value and up | The range above 2.0 includes any value greater or equal to <2.00000000000000000000 The range after 2015-08-12T05:23:45 includes any time after 2015-08-12T05:23:45.000000000000000 |

The proper use of these ranges is discussed further below.

**number**

Sarching on a simple numerical value in a resource. Examples:

|  |  |
| --- | --- |
| [parameter]=100 | Values that equal 100, to 3 significant figures precision, so range [99.5 ... 100.5) |
| [parameter]=100.00 | Values that equal 100, to 5 significant figures precision, so range [99.995 ... 100.005). Whole numbers also equal 100.00, but not 100.01 |
| [parameter]=lt100 | Values that are less than 100 |
| [parameter]=le100 | Values that are less or equal to 100 |
| [parameter]=gt100 | Values that are greater than 100 |
| [parameter]=ge100 | Values that are greater or equal to 100 |
| [parameter]=ne100 | Values that are not equal to 100 |

Note: Uncertainty does not factor in evaluations. The precision of the numbers is considered arbitrarily high. (The way search parameters operate in resources is not the same as whether two numbers are equal to each other in a mathematical sense).

Here are some example searches:

|  |  |
| --- | --- |
| **Search** | **Description** |
| GET [base]/Encounter?length=gt20 | Search for all the encounters longer than 20 days |
| GET [base]/ImmunizationRecommendation?deo-number=2 | Search for any immunization recommendation recommending a second dose |

**date**

A date parameter searches on a date/time or period. As is usual for date/time related functionality, while the concepts are relatively straight-forward, there are a number of subtleties involved in ensuring consistent behavior.

The date parameter format is yyyy-mm-ddThh:mm:ss[Z|(+|-)hh:mm] (the standard XML format).

Technically, this is any of the [date](file:///C:\temp\datatypes.html#date), [dateTime](file:///C:\temp\datatypes.html#dateTime), and [instant](file:///C:\temp\datatypes.html#instant) data types; e.g. Any degree of precision can be provided, but it SHALL be populated from the left (e.g. can't specify a month without a year), except that the minutes SHALL be present if an hour is present, and you SHOULD provide a time zone if the time part is present. Note: Time can consist of hours and minutes with no seconds, unlike the XML Schema dateTime type. Some user agents may escape the : characters in the URL, and servers SHALL handle this correctly.

Date parameters may be used with the following data types:

|  |  |
| --- | --- |
| [date](file:///C:\temp\datatypes.html#date) | The range of the value is the day, month, or year as specified |
| [dateTime](file:///C:\temp\datatypes.html#dateTime) | The range of the value as defined above; e.g. For example, the date 2013-01-10 specifies all the time from 00:00 on 10-Jan 2013 to immediately before 00:00 on 11-Jan 2013 |
| [instant](file:///C:\temp\datatypes.html#instant) | An instant is considered a fixed point in time with an interval smaller than the precision of the system, i.e. an interval with an effective width of 0 |
| [Period](file:///C:\temp\datatypes.html#Period) | Explicit, though the upper or lower bound may not actually be specified in resources. |
| [Timing](file:///C:\temp\datatypes.html#Timing) | the specified scheduling details are ignored and only the outer limits matter. For instance, a schedule that specifies every second day between 31-Jan 2013 and 24-Mar 2013 includes 1-Feb 2013, even though that is on an odd day that is not specified by the period. This is to keep the server load processing queries reasonable. |

Implicitly, a missing lower boundary is "less than" any actual date. A missing upper boundary is "greater than" any actual date. The use of the prefixes:

|  |  |
| --- | --- |
| [parameter]=eq2013-01-14 | * 2013-01-14T00:00 matches (obviously) * 2013-01-14T10:00 matches * 2013-01-15T00:00 does not match - it's not in the range |
| [parameter]=ne2013-01-14 | * 2013-01-15T00:00 matches - it's not in the range * 2013-01-14T00:00 does not match - it's in the range * 2013-01-14T10:00 does not match - it's in the range |
| [parameter]=lt2013-01-14T10:00 | * 2013-01-14 matches, because it includes the part of 14-Jan 2013 before 10am |
| [parameter]=gt2013-01-14T10:00 | * 2013-01-14 matches, because it includes the part of 14-Jan 2013 after 10am |
| [parameter]=ge2013-03-14 | * "from 21-Jan 2013 onwards" is included because that period may include times after 14-Mar 2013 |
| [parameter]=le2013-03-14 | * "from 21-Jan 2013 onwards" is included because that period may include times before 14-Mar 2013 |
| [parameter]=sa2013-03-14 | * "from 15-Mar 2013 onwards" is included because that period starts after 14-Mar 2013 * "from 21-Jan 2013 onwards" is not included because that period starts before 14-Mar 2013 * "before and including 21-Jan 2013" is not included because that period starts (and ends) before 14-Mar 2013 |
| [parameter]=eb2013-03-14 | * "from 15-Mar 2013 onwards" is not included because that period starts after 14-Mar 2013 * "from 21-Jan 2013 onwards" is not included because that period starts before 14-Mar 2013, but does not end before it * "before and including 21-Jan 2013" is not included because that period ends before 14-Mar 2013 |
| [parameter]=ap2013-03-14 | * 14-Mar 2013 is included - as it exactly matches * 21-Jan 2013 is not included because that is near 14-Mar 2013 * 15-Jun 2015 is not included - as it is not near 14-Mar 2013. Note that the exact value here is at the discretion of the system |

Other notes:

* When the date parameter is not fully specified, matches against it are based on the behavior of intervals, where:
  + Dates with only the year specified are equivalent to an interval that starts at the first instant of January 1st to the last instant of December 31st, e.g. 2000 is equivalent to an interval of [2000-01-01T00:00, 2000-12-31T23:59].
  + Dates with the year and month are equivalent to an interval that starts at the first instant of the first day of the month and ends on the last instant of the last day of the month, e.g. 2000-04 is equivalent to an interval of [2000-04-01T00:00, 2000-04-30T23:59].
* Where possible, the system should correct for timezones when performing queries. Dates do not have time zones, and time zones should not be considered. Where both search parameters and resource element date times do not have time zones, the servers local time zone should be assumed.

To search for all the procedures in a patient compartment that occurred over a 2 year period:

GET [base]/Patient/23/Procedure?date=ge2010-01-01&date=le2011-12-31

**string**

For a simple string search, a string parameter serves as the input for a case- and accent-insensitive search against sequences of characters. By default, a field matches a string query if the value of the field equals or starts with the supplied parameter value, after both have been normalized by case and accent. The :contains modifier returns results that include the supplied parameter value anywhere within the field being searched. The :exact modifier returns results that match the entire supplied parameter, including casing and accents.

Examples:

|  |  |
| --- | --- |
| [base]/Patient?given=eve | Any patients with a name containing a given part with "eve" at the start of the name. This would include patients with the given name "Eve", "Evelyn". |
| [base]/Patient?given:contains=eve | Any patients with a name with a given part containing "eve" at any position. This would include patients with the given name "Eve", "Evelyn", and also "Severine". |
| [base]/Patient?given:exact=Eve | Any patients with a name with a given part that is exactly "Eve". Note: This would not include patients with the given name "eve" or "EVE". |

An additional modifier :text can be used to specify a search with advanced text handling (see [below](file:///C:\temp\main-pages.html#text)) though only a few servers are expected to offer this facility.

When a string search parameter refers to the types [HumanName](file:///C:\temp\datatypes.html#HumanName) and [Address](file:///C:\temp\datatypes.html#Address), the search covers the elements of type string, and does not cover elements such as use and period. For robust search, servers should search the parts of a family name independently. E.g. Searching either Carreno or Quinones should match a family name of "CarreÃƒÂ±o QuiÃƒÂ±ones". HL7 affiliates may make more specific recommendations about how search should work in their specific culture.

It is at the discretion of the server whether to pre-process names, addresses, and contact details to remove separator characters prior to matching in order to ensure more consistent behavior. For example, a server might remove all spaces and - characters from phone numbers. What is most appropriate varies depending on culture and context. A server may also use a free-text style searches on this property to achieve the best results When searching whole names and addresses (not parts), servers may also use flexible match or a free-text style searches on names to achieve the best results.

**uri**

The uri parameter refers to a URI ([RFC 3986](https://tools.ietf.org/html/rfc3986)) element. Matches are precise (e.g. case, accent, and escape) sensitive, and the entire URI must match. The modifier :above or :below can be used to indicate that partial matching is used. For example:

GET [base]/ValueSet?url=http://acme.org/fhir/ValueSet/123

GET [base]/ValueSet?url=http://acme.org/fhir/

GET [base]/ValueSet?url=urn:oid:1.2.3.4.5

The first line is a request to find any value set with the exact url "http://acme.org/fhir/ValueSet/123". The second line performs a search that will return any value sets that have a URL that starts with "http://acme.org/fhir/". The converse - the search for any value set above a given specific URL. This approach may be useful for searching name systems, but it is generally less useful than the :below search. The third line shows an example of searching by an OID. Note that the :above and :below modifiers only apply to URLs, and not URNS such as OIDs.

Note that for [Canonical URLs](file:///C:\temp\references.html#canonical), servers SHOULD support searching by Canonical URL, and SHOULD support automatically detecting a |[version] portion as part of the Canonical URL, and interpreting that portion as a search on the version.

**token**

A token type is a parameter that provides an exact match search, either on a string of characters, potentially scoped by a URI. It is mostly used against a code or identifier data type where the value may have a URI that scopes its meaning, where the search is performed against the pair from a Coding or an Identifier. Tokens are also used against other fields where exact matches are required - uris, booleans, and [ContactPoints](file:///C:\temp\datatypes.html#ContactPoint). In these cases, the URI portion is not used.

For tokens, matches are literal (e.g. not based on [subsumption](file:///C:\temp\codesystem.html#subsumption) or other code system features), but not case sensitive. To use subsumption based logic, use the modifiers below, or list all the codes in the heirarchy. The syntax for the value is one of the following:

* **[parameter]=[code]**: the value of [code] matches a Coding.code or Identifier.value irrespective of the value of the system property
* **[parameter]=[system]|[code]**: the value of [code] matches a Coding.code or Identifier.value, and the value of [system] matches the system property of the Identifier or Coding
* **[parameter]=|[code]**: the value of [code] matches a Coding.code or Identifier.value, and the Coding/Identifier has no system property
* **[parameter]=[system]|**: any element where the value of [system] matches the system property of the Identifier or Coding

Note: The namespace URI and code both must be [escaped](file:///C:\temp\main-pages.html#escaping) correctly. If a system is not applicable (e.g. an element of type [uri](file:///C:\temp\datatypes.html#uri), then just the form [parameter]=[code] is used.

Token search parameters are used for the following data types:

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **URI** | **Code** | **Comments** |
| [Coding](file:///C:\temp\datatypes.html#Coding) | Coding.system | Coding.code |  |
| [CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept) | CodeableConcept.coding.system | CodeableConcept.coding.code | Matches against any coding in the CodeableConcept |
| [Identifier](file:///C:\temp\datatypes.html#Identifier) | Identifier.system | Identifier.value |  |
| [ContactPoint](file:///C:\temp\datatypes.html#Identifier) | ContactPoint.use | ContactPoint.value | The use is prepended by [http://hl7.org/fhir/contact-point-system](file:///C:\temp\valueset-contact-point-system.html)/ |
| [code](file:///C:\temp\datatypes.html#code) | (implicit) | code | the system is defined in the value set (though it's not usually needed) |
| [boolean](file:///C:\temp\datatypes.html#boolean) | (implicit) | boolean | The implicit system is [http://hl7.org/fhir/special-values](file:///C:\temp\valueset-special-values.html) |
| [string](file:///C:\temp\datatypes.html#string) | n/a | string | Token is sometimes used for string to indicate that exact matching is the correct default search stategy |

Note: The use of token search parameters for boolean fields: the boolean values "true" and "false" are also represented as formal codes in the [Special Values](file:///C:\temp\valueset-special-values.html) code system, which is useful when boolean values need to be represented in a [Coding](file:///C:\temp\datatypes.html#coding) data type. The namespace for these codes is http://hl7.org/fhir/special-values, though there is usually no reason to use this, as a simple true or false is sufficient.

**Modifiers:**

|  |  |
| --- | --- |
| **Modifier** | **Use** |
| :text | The search parameter is processed as a string that searches text associated with the code/value - either *CodeableConcept.text*, *Coding.display*, or *Identifier.type.text*. |
| :not | Reverse the code matching described in the paragraph above. Note that this includes resources that have no value for the parameter - e.g. ?gender:not=male includes all patients that do have have gender = male, including patients that do not have a gender at all |
| :above | The search parameter is a concept with the form [system]|[code], and the search parameter tests whether the coding in a resource [subsumes](file:///C:\temp\codesystem.html#subsumption) the specified search code. For example, the search concept has an is-a relationship with the coding in the resource, and this includes the coding itself. |
| :below | the search parameter is a concept with the form [system]|[code], and the search parameter tests whether the coding in a resource is subsumed by the specified search code. For example, the coding in the resource has an is-a relationship with the search concept, and this includes the coding itself. |
| :in | The search parameter is a URI (relative or absolute) that identifies a value set, and the search parameter tests whether the coding is in the specified [value set](file:///C:\temp\valueset.html). The reference may be literal (to an address where the value set can be found) or logical (a reference to ValueSet.url). If the server can treat the reference as a literal URL, it does, else it tries to match known logical ValueSet.url values. |
| :not-in | The search parameter is a URI (relative or absolute) that identifies a value set, and the search parameter tests whether the coding is not in the specified value set. |

Most servers will only process value sets that are already known/registered/supported internally. However, servers can elect to accept any valid reference to a value set. Servers may elect to consider concept mappings when testing for subsumption relationships.

Example searches:

|  |  |
| --- | --- |
| **Search** | **Description** |
| GET [base]/Patient?identifier=http://acme.org/patient|2345 | Search for all the patients with an identifier with key = "2345" in the system "http://acme.org/patient" |
| GET [base]/Patient?gender=male | Search for any patient with a gender that has the code "male" |
| GET [base]/Patient?gender:not=male | Search for any patient with a gender that does not have the code "male" |
| GET [base]/Patient?active=true | Search for any patients that are active |
| GET [base]/Condition?code=http://acme.org/conditions/codes|ha125 | Search for any condition with a code "ha125" in the code system "http://acme.org/conditions/codes" |
| GET [base]/Condition?code=ha125 | Search for any condition with a code "ha125". Note that there is not often any useful overlap in literal symbols between code systems, so the previous example is generally preferred |
| GET [base]/Condition?code:text=headache | Search for any Condition with a code that has a text "headache" associated with it (either in the text, or a display) |
| GET [base]/Condition?code:in=http%3A%2F%2Fsnomed.info%2Fsct%3Ffhir\_vs%3Disa%2F126851005 | Search for any condition in the SNOMED CT value set "http://snomed.info/sct?fhir\_vs=isa/126851005" that includes all descendants of "Neoplasm of liver" |
| GET [base]/Condition?code:below=126851005 | Search for any condition that is subsumed by the SNOMED CT Code "Neoplasm of liver". Note: This is the same outcome as the previous search |
| GET [base]/Condition?code:in=http://acme.org/fhir/ValueSet/cardiac-conditions | Search for any condition that is in the institutions list of cardiac conditions |

**quantity**

A quantity parameter searches on the [Quantity](file:///C:\temp\datatypes.html#Quantity) data type. The syntax for the value follows the form:

* **[parameter]=[prefix][number]|[system]|[code]** matches a quantity with the given unit

The prefix is optional, and is as described [above](file:///C:\temp\main-pages.html#prefix), both regarding how precision and comparator/range operators are interpreted. Example searches:

|  |  |
| --- | --- |
| **Search** | **Description** |
| GET [base]/Observation?value=5.4|http://unitsofmeasure.org|mg | Search for all the observations with a value of 5.4 mg where mg is understood as a UCUM unit (system/code) |
| GET [base]/Observation?value=5.4||mg | Search for all the observations with a value of 5.4 mg where the unit - either the code or the stated human unit (unit) are "mg" |
| GET [base]/Observation?value=le5.4|http://unitsofmeasure.org|mg | Search for all the observations where the value of is less than 5.4 mg where mg is understood as a UCUM unit |
| GET [base]/Observation?value=ap5.4|http://unitsofmeasure.org|mg | Search for all the observations where the value of is about 5.4 mg where mg is understood as a UCUM unit |

The search processor may choose to perform a search based on [canonical units](file:///C:\temp\datatypes.html#quantity) (e.g. any value where the units can be converted to a value in mg in the case above).

**reference**

A reference parameter refers to [references between resources](file:///C:\temp\references.html). For example, find all Conditions where the subject reference is a particular patient, where the patient is selected by name or identifier. The interpretation of a *reference* parameter is either:

* **[parameter]=[id]** the logical [id] of a resource using a local reference (i.e. a relative reference)
* **[parameter]=[type]/[id]** the logical [id] of a resource of a specified type using a local reference (i.e. a relative reference), for when the reference can point to different types of resources (e.g. [Observation.subject](file:///C:\temp\observation-definitions.html#Observation.subject))
* **[parameter]=[url]** where the [url] is an absolute URL - a reference to a resource by its absolute location

Note: A relative reference resolveing to the same value as a specified absolute URL, or vice versa, qualifies as a match. For example, if the search parameter value is Patient/123, then this will find references like this:

<patient>

<reference value="Patient/123"/>

</patient>

If the server base address is http://example.org/fhir, then the full URL for that reference is http://example.org/fhir/Patient/123, which means that the search term also matches patient references like this:

<patient>

<reference value="http://example.org/fhir/Patient/123"/>

</patient>

In addition, searching for reference=http://example.org/fhir/Patient/123 will also match both references.

Some references are allowed to point to more than one type of resource; e.g. subject : Reference(Patient|Group|Device|..). In these cases, multiple resources may have the same logical identifier. Servers SHOULD reject a search where the logical id refers to more than one matching resource across different types. In order to allow the client to perform a search in these situations the type is specified explicitly:

GET [base]/Observation?subject=Patient/23

This searches for any observations where the subject refers to the patient resource with the logical identifier "23". A modifier is also defined to to allow the client to be explicit about the intended type:

GET [base]/Observation?subject:Patient=23

This has the same effect as the previous search. The modifier becomes useful when used with chaining as explained in the next section. Note: The [type] modifier can't be used with a reference to a resource found on another server, since the server would not usually know what type that resource has. However, since these are absolute references, there can be no ambiguity about the type.

In some cases, search parameters are defined with an implicitly limited scope. For example, [Observation](file:///C:\temp\observation.html) has an element subject, which is a reference to one of a number of types. This has a matching search parameter subject, which refers to any of the possible types. In addition to this, there is another search parameter patient, which also refers to Observation.subject, but is limited to only include references of type [Patient](file:///C:\temp\patient.html). When using the patient search parameter, there is no need to specify ":Patient" as a modifier, or "Patient/" in the search value, as this must always be true.

**Chained parameters**

In order to save a client from performing a series of search operations, reference parameters may be "chained" by appending them with a period (.) followed by the name of a search parameter defined for the target resource. This can be done recursively, following a logical path through a graph of related resources, separated by .. For instance, given that the resource [DiagnosticReport](file:///C:\temp\diagnosticreport.html) has a search parameter named *subject*, which is usually a reference to a [Patient](file:///C:\temp\patient.html) resource, and the Patient resource includes a parameter *name* which searches on patient name, then the search

GET [base]/DiagnosticReport?subject.name=peter

is a request to return all the lab reports that have a subject whose name includes "peter". Because the Diagnostic Report subject can be one of a set of different resources, it's necessary to limit the search to a particular type:

GET [base]/DiagnosticReport?subject:Patient.name=peter

This request returns all the lab reports that have a subject which is a patient, whose name includes "peter".

Advanced Search Note: Where a chained parameter searches a resource reference that may have more than one type of resource as its target, the parameter chain may end up referring to search parameters with the same name on more than one kind of resource at once. Servers SHOULD reject a search where the logical id refers to more than one matching resource across different types. For example, the client has to specify the type explicitly using the syntax in the second example above.

**Reverse Chaining**

The \_has parameter provides limited support for reverse chaining - that is, selecting resources based on the properties of resources that refer to them (instead of chaining, above, where resources can be selected based on the properties of resources that they refer to). Here is an example of the \_has parameter:

GET [base]/Patient?\_has:Observation:patient:code=1234-5

This requests the server to return Patient resources, where the patient resource is referred to by at least one Observation where the observation has a code of 1234, and where the Observation refers to the patient resource in the patient search parameter.

"Or" searches are allowed (e.g. GET [base]/Patient?\_has:Observation:subject:code=123,456), and multiple \_has parameters are allowed (e.g. GET [base]/Patient?\_has:Observation:subject:code=123&\_has:Observation:subject:code=456). Note that each \_has parameter is processed independently of other \_has parameters.

The \_has parameter can be chained, like this:

GET [base]/Patient?\_has:Observation:patient:\_has:AuditEvent:entity:user=MyUserId

Fetch all the patients that have an Observation where the observation has an audit event from a specific user.

**STU Note:** the \_has search parameter is a new addition that has generated significant discussion. if adopted, it may replace the [\_list](file:///C:\temp\main-pages.html#has) parameter. Or not.

Feedback is welcome [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**Composite Search Parameters**

Composite search parameters supports joining single values with a $. For example, the result of the search operation is the intersection of the resources that match the criteria specified by each individual search parameter. If a parameter repeats, such as /Patient?language=FR&language=NL, then this matches a patient who speaks both languages. This is known as an AND search parameter, since the server is expected to respond only with results which match both values.

If, instead, the search is to find patients that speak either language, then this is a single parameter with multiple values, separated by a ,. For example, /Patient?language=FR,NL. This is known as an OR search parameter, since the server is expected to respond with results which match either value.

AND parameters and OR parameters may also be combined, for example: /Patient?language=FR,NL&language=EN would refer to any patient who speaks English, as well as either French or Dutch.

This approach allows for simple combinations of and/or values, but doesn't allow a search based on a pair of values, such as all observations with a sodium value >150 mmol/L (particularly as the end criteria of a chained search), or searching on Group.characteristic where you need find a combination of key/value, not an intersection of separate matches on key and value. Another example is spatial coordinates when doing geographical searches.

To allow these searches, a resource may also specify *composite* parameters that take sequences of single values that match other defined parameters as an argument. The matching parameter of each component in such a sequence is documented in the definition of the parameter. These sequences are formed by joining the single values with a $. Note: This sequence is a single value and itself can be composed into a set of values, so that, for example, multiple matching state-on-date parameters can be specified as state-on-date=new$2013-05-04,active$2013-05-05.

Note: Modifiers are not used on composite parameters.

Examples of using composite parameters:

|  |  |
| --- | --- |
| **Search** | **Description** |
| GET [base]/DiagnosticReport?result.code-value-quantity=http://loinc.org|2823-3$gt5.4|http://unitsofmeasure.org|mmol/L | Search for all diagnostic reports that contain on observation with a potassium value of >5.4 mmol/L (UCUM) |
| GET [base]/Observation?component-code-value-quantity=http://loinc.org|8480-6$lt60 | Search for all the observations with a systolic blood pressure < 60. Note that in this case, the unit is assumed (everyone uses mmHg) |
| GET [base]/Group?characteristic-value=gender$mixed | Search for all groups that have a characteristic "gender" with a text value of "mixed" |

**Handling Missing Data**

Consider the case of searching for all AllergyIntolerance resources:

GET [base]/AllergyIntolerance?clinical-status=active

This search will only return resources the have a value for clinicalStatus:

{

"resourceType" : "AllergyIntolerance",

"clinicalStatus" : "active"

}

Resources missing a clinicalStatus will not be returned. This is probably unsafe - it would not usually be appropriate to ignore AllergyIntolerance warnings with an unknown clinical status, and only return resources with an explicit clinicalStatus. Instead, it might be desired to return AllergyIntolerance resources with either an explicit value for clinicalStatus, or none:

GET [base]/AllergyIntolerance?clinical-status=active

GET [base]/AllergyIntolerance?clinical-status:exists=false

Note that this is 2 separate queries. They can be [combined in a batch](file:///C:\temp\http.html#transaction), but not in a single operation. This query will always return an empty list, as no resource can satisfy both criteria at once:

GET [base]/AllergyIntolerance?clinical-status=active&clinical-status:exists=false

There is no way to use the :exists modifier and mix with a value using the comma syntax documented above for for composite search parameters.

An alternative approach is to use the [\_filter](file:///C:\temp\main-pages.html#filter) parameter, for servers that support this parameter.

**Escaping Search Parameters**

In the rules described above, special rules are defined for the characters $, ,, and |. As a consequence, if these characters appear in an actual parameter value, they must be differentiated from their use as separator characters. When any of these characters appear in an actual parameter value, they must be prepended by the character \, which also must be used to prepend itself. Therefore, param=xxx$xxx indicates that it is a composite parameter, while param=xx\$xx indicates that the parameter has the literal value xx$xx. The parameter value xx\xx is illegal, and the parameter value param=xx\\xx indicates a literal value of xx\xx.

This specification defines this additional form of escape for a reason. The classic %xx escaping which is part of normal HTTP URLs ensures that the character appears at the FHIR server correctly, while the , versus \ becomes important once it has reached the server and the query is parsed. Therefore:

GET [base]/ValueSet?url=http://acme.org/fhir/ValueSet/123,http://acme.org/fhir/ValueSet/124%2CValueSet/125

uses url escaping to make sure the FHIR server received:

GET [base]/ValueSet?url=http://acme.org/fhir/ValueSet/123,http://acme.org/fhir/ValueSet/124,125

This request will compare the URL against three values: the last one being a relative and incorrect url, which is likely not the actual intent. However:

GET [base]/ValueSet?url=http://acme.org/fhir/ValueSet/123,http://acme.org/fhir/ValueSet/124\,125

is equivalent to:

GET [base]/ValueSet?url=http://acme.org/fhir/ValueSet/123,http://acme.org/fhir/ValueSet/124\%2C125

which would mean: url = http://.....123 OR http://....124,125.

**Text Search Parameters**

The special text search parameters, \_text and \_content, search on the narrative of the resource, and the entire content of the resource respectively. These parameters SHOULD support a sophisticated search functionality of the type offered by typical text indexing services. The value of the parameter is a text based search, which may involve searching multiple words with thesaurus and proximity considerations, and logical operations such as AND, OR etc. For example:

GET [base]/Condition?\_text=(bone OR liver) and metastases

This request returns all Condition resources with the word "metastases" and either "bone" or "liver" in the narrative. The server MAY choose to search for related words as well.

**STU Note:** The issues around standardizing text search are not fully resolved. During the trial use period for this specification, we recommend that systems use the rules specified by [the OData specification for the $search parameter](http://docs.oasis-open.org/odata/odata/v4.0/cs01/part1-protocol/odata-v4.0-cs01-part1-protocol.html#_The_$search_System). Typical implementations would use Lucene, an sql-based full text search, or some indexing service.

Feedback is welcome [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**Searching by list**

The \_list parameter allows for the retrieval of resources that are referenced by a [List](file:///C:\temp\list.html) resource.

GET [base]/Patient?\_list=42

This request returns all Patient resources that are referenced from the list found at [base]/List/42) in List.entry.item. While it is possible to retrieve the list, and then iterate the entries in the list fetching each patient, using a list as a search criteria allows for additional search criteria to be specified. For instance:

GET [base]/Patient?\_list=42&gender=female

This request will return all female patients in the list. The server can return the list referred to in the search parameter as an included resource, but is not required to do so. In addition, a system can support searching by lists by their logical function. For example:

GET [base]/AllergyIntolerance?patient=42&\_list=$current-allergies

This request will return all allergies in patient 42's "Current Allergy List". The server returns all relevant AllergyIntolerance resources, and can also choose to return the list. For further information, refer to the [definition of "$current-allergies"](file:///C:\temp\lifecycle.html#current), and the [List Operation "Find"](file:///C:\temp\list-operations.html#find). Note: Servers are not required to make these lists available to the clients as list resources, but may choose to do so.

**STU Note:** the [\_has](file:///C:\temp\main-pages.html#has) search parameter documented above is a new addition that has generated significant discussion. if adopted, it may replace the \_list parameter. Or not.

Feedback is welcome [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**Advanced filtering**

The search mechanism described above is flexible, and easy to implement for simple cases, but is limited in its ability to express combination queries. To complement this mechanism, a the "\_filter" search expression parameter can be used.

For example, "Find all the observations for patient with a name including peter that have a LOINC code 1234-5":

GET [base]/Observation?code=http://loinc.org|1234-5&subject.name=peter

Using the \_filter parameter, the search would be expressed like this:

GET [base]/Observation?\_filter=name eq http://loinc.org|1234-5 and subject.name co "peter"

The \_filter parameter is described in detail on the ["\_Filter Parameter" page](file:///C:\temp\search_filter.html).

**Specifying the type of Resource**

Normally, a search is initiated against a known type of resource, e.g.

GET [base]/Observation?params...

However in some circumstances, a search is executed where there is no fixed type of resource:

* Using search across all resource types (GET [base]?params...)
* Using search with [messaging](file:///C:\temp\messaging.html)
* Some search specifications inside other services e.g. decision support usage

In these circumstances, the search criteria may need to specify one or more resource types that the search applies to. This can be done by using the \_type parameter:

GET [base]/?\_type=Observation,Condition&other params...

Note that the only search parameters that be can be used in global search like this are the base parameters that apply to all resources. In other contexts, searches on multiple types may allow resource specific search parameters, but implementation experience will be required to determine the correct behaviour in these cases.

**Managing Returned Resources**

**Sorting**

The client can indicate which order to return the results by using the parameter \_sort, which can contains a comma-separated list of sort rules in priority order:

GET [base]/Observation?\_sort=status,-date,category

Each item in the comma separated list is a search parameter, optionally with a '-' prefix. The prefix indicates decreasing order; in it's absence, the parameter is applied in increasing order.

Notes:

* When sorting, the actual sort value used is not returned explicitly by the server for each resource, just the resource contents.
* To sort by relevance, use \_score.
* The server returns the sort it performs as part of the returned search parameters (see [below](file:///C:\temp\main-pages.html#conformance)).
* A search parameter can refer to an element that repeats, and therefore there can be multiple values for a given search parameter for a single resource. In this case, the sort is based on the item in the set of multiple parameters that comes earliest in the specified sort order when ordering the returned resources.
* When sorting on string search parameters, sorting SHOULD be performed on a case-insensitive basis. Accents may either be ignored or sorted as per realm convention.
* This specification does not specify exacts rules for consistency of sorting across servers. In general, this is deemed to be not as essential as consistency of filtering (though even that is a little variable). The purpose of sorting is to provide data in a "reasonable" order for end-users. "Reasonable" may vary by use case and realm, particularly for accented characters.

**Page Count**

In order to keep the load on clients, servers and the network minimized, the server may choose to return the results in a series of pages. The search result set contains the URLs that the client uses to request additional pages from the search set. For a simple RESTful search, the page links are[contained in the returned bundle as links](file:///C:\temp\http.html#paging).

Typically, a server will provide its own parameters in the links that it uses to manage the state of the search as pages are retrieved. These parameters do not need to be understood or processed by the client.

The parameter \_count is defined as a hint to the server regarding how many resources should be returned in a single page. Servers SHALL NOT return more resources than requested, even if they don't support paging, but are allowed to return less than the client requested. The server should repeat the original \_count parameter in its returned page links so that subsequent paging requests honor the original \_count. Note: It is at the discretion of the search engine as to how to handle ongoing updates to the resources while the search is proceeding.

Note: The combination of \_sort and \_count can be used to return only the latest resource that meets a particular criteria - set the criteria, and then sort by date in descending order, with \_count=1. This way, the last matching resource will be returned.

if \_count has the value 0, this shall be treated the same as \_summary=count: the server resturns a bundle that reports the total number of resources that match in Bundle.total, but with no entries, and no prev/next/last links. Note that the Bundle.total only include the total number of matching resources. It does not count extra resources such as [OperationOutcome](file:///C:\temp\operationoutcome.html) or [included](file:///C:\temp\main-pages.html#include) resources that may also be returned.

**Including other resources in result (\_include and \_revinclude)**

Clients may request that the engine return resources related to the search results, in order to reduce the overall network delay of repeated retrievals of related resources. This is useful when the client is searching on a clinical resource, but for every such resource returned, the client will also need the subject (patient) resource that the clinical resource refers to. The client can use the \_include parameter to indicate that the subject resources be included in the results. An alternative scenario is where the client wishes to fetch a particular resource, and any resources that refer to it. For example, the client may wish to fetch a MedicationRequest, and any provenance resources that refer to the prescription. This is known as a reverse include, and is specified by providing a \_revinclude parameter.

Both \_include and \_revinclude are based on search parameters, rather than paths in the resource, since joins, such as [chaining](file:///C:\temp\main-pages.html#chaining), are already done by search parameter.

Each \_include parameter specifies a search parameter to join on:

GET [base]/MedicationRequest?\_include=MedicationRequest:patient&criteria...

GET [base]/MedicationRequest?\_revinclude=Provenance:target&criteria...

The first search requests all matching MedicationRequests, to include any patient that the medication prescriptions in the result set refer to. The second search requests all matching prescriptions, return all the provenance resources that refer to them.

Parameter values for both \_include and \_revinclude have three parts, separated by a : character:

1. The name of the source resource from which the join comes
2. The name of the search parameter which must be of type *reference*
3. (Optional) A specific of type of target resource (for when the search parameter refers to multiple possible target types)

\_include and \_revInclude parameters do not include multiple values. Instead, the parameters are repeated for each different include criteria.

For each returned resource, the server identifies the resources that meet the criteria expressed in the join, and adds to the results, with the [entry.search.mode](file:///C:\temp\bundle-definitions.html#Bundle.entry.search.mode) set to "include" (in some searches, it is not obvious which resources are matches, and which are includes). If there is no reference, or no matching resource, the resource cannot be retrieved (e.g. on a different server), then the resource is omitted, and no error is returned.

The inclusion process can be recursive, if the modifier :recurse is included. For example, this search returns all [Medication Request](file:///C:\temp\medicationrequest.html) resources and their [prescribing Practitioner](file:///C:\temp\practitioner.html) Resources for the matching [Medication Dispense](file:///C:\temp\medicationdispense.html) resources:

GET [base]/MedicationDispense?\_include=MedicationDispense:authorizingPrescription

&\_include:recurse=MedicationRequest:prescriber&criteria...

This technique applies to circular relationships as well. For example, the first of these two searches includes any related observations to the target relationships, but only those directly related. The second search asks for the \_include based on related parameter to be executed recursively, so it will retrieve observations that are directly related, and also any related observations to any other included observation.

GET [base]/Observation?\_include=Observation:related-target&criteria...

GET [base]/Observation?\_include:recurse=Observation:related-target&criteria...

Both \_include and \_revInclude use the wild card "\*" for the search parameter name, indicating that any search parameter of type=reference be included. Though both clients and servers need to take care not to request or return too many resources when doing this. Most notably, using recursive inclusions might lead to the retrieval of the full patient's record, or even more: resources are organized into an interlinked network and broad \_include paths may eventually traverse all possible paths on the server. For servers, these recursive and wildcard \_includes are demanding and may slow the search response time significantly.

It is at the server's discretion how deep to recursively evaluate the inclusions. Servers are expected to limit the number of iterations done to an appropriate level and are not obliged to honor requests to include additional resources in the search results.

When search results are paged, each page of search results should include the matching includes for the resources in each page, so that each page stands alone as a coherent package.

**Contained Resources**

By default, search results only include resources that are not contained in other resources. A chained condition will be evaluated inside contained resources. To illustrate this, consider a MedicationRequest resource that has a contained Medication resource specifying a custom formulation that has ingredient with a itemCodeableConcept "abc" in "http://acme.com./medications". In this case, a search:

GET MedicationRequest?medication.ingredient-code=abc

will include the MedicationRequest resource in the results. However, this search:

GET Medication?ingredient-code=abc

will not include the contained Medication resource in the results, since either the wrong type of resource would be returned, or the contained resource would be returned without its container resource, which provides context to the contained resource.

Clients are able to modify this behavior using the \_contained parameter, which can have one of the following values:

* false (default): Do not return contained resources
* true: return only contained resources
* both: return both contained and non-contained (normal) resources

When contained resources are being returned, the server should return either the container resource, or the contained resource alone. The client can specify which by using the \_containedType parameter, which can have one of the following values:

* container (default): Return the container resources
* contained: return only the contained resource

When returning a container resource, the server simply puts this in the search results:

<Bundle>

...

<entry>

<resource>

<MedicationRequest>

<id value="23">

....

<contained>

<Medication>

<id value="m1">

...

</Medication>

<contained>

</MedicationRequest>

</resource>

<search>

<mode value="match"/>

</search>

</entry>

</Bundle>

In the case of returning container resources, the server SHALL populate the entry.search.mode element, as shown, so that the client can pick apart matches and includes (since the usual approach of doing it by type may not work).

If the return type is the contained resource, this must be done slightly differently:

<Bundle>

...

<entry>

<fullUrl value="http://example.com/fhir/MedicationRequest/23#m1"/>

<resource>

<Medication>

<id value="m1">

...

</Medication>

</resource>

<search>

<mode value="match"/>

</search>

</entry>

</Bundle>

In this case, the fullUrl informs the client that this is a contained resource, along with indicating the identity of the containing resource.

**External References**

If the \_include path selects a reference that refers to a resource on another server, the server can elect to include that resource in the search results for the convenience of the client.

If the \_include path selects a reference that refers to an entity that is not a Resource, such as an image attachment, the server may also elect to include this in the returned results as a [Binary](file:///C:\temp\binary.html) resource. For example, the include path may point to an attachment which is by reference, like this:

<content>

<contentType>image/jpeg</contentType>

<url>http://example.org/images/2343434/234234.jpg</url>

</content>

The server can retrieve the target of this reference, and add it to the results for the convenience of the client.

**STU Note:** Should additional rules about how \_include works be made?

Feedback based on implementation experience is sought [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**Paging**

When returning paged results for a search with \_include resources, all \_include resources that are related to the primary resources returned for the page SHOULD also be returned as part of that same page, even if some of those resource instances have previously been returned on previous pages. This approach allows both sender and receiver to avoid caching results of other pages.

**Summary**

The client can request the server to return a portion of the resources by using the parameter \_summary:

GET [base]/ValueSet?\_summary=true

The \_summary parameter requests the server to return a subset of the resource. It can contain one of the following values:

|  |  |
| --- | --- |
| [true](file:///C:\temp\main-pages.html#summary-true) | Return only those elements marked as "summary" in the base definition of the resource(s) (see [ElementDefinition.isSummary](file:///C:\temp\elementdefinition-definitions.html#ElementDefinition.isSummary)) |
| [text](file:///C:\temp\main-pages.html#summary-text) | Return only the "text" element, the 'id' element, the 'meta' element, and only top-level mandatory elements |
| [data](file:///C:\temp\main-pages.html#summary-data) | Remove the text element |
| [count](file:///C:\temp\main-pages.html#summary-count) | Search only: just return a count of the matching resources, without returning the actual matches |
| [false](file:///C:\temp\main-pages.html#summary-false) | Return all parts of the resource(s) |

The intent of the \_summary parameter is to reduce the total processing load on server, client, and resources between them such as the network. It is most useful for resources that are large, particularly ones that include images or elements that may repeat many times. The purpose of the summary form is to allow a client to quickly retrieve a large set of resources, and let a user pick the appropriate one. The summary for an element is defined to allow a user to quickly sort and filter the resources, and typically omit important content on the basis that the entire resource will be retrieved when the user selects a resource.

Servers are not obliged to return just a summary as requested. There are only a limited number of summary forms defined for resources in order to allow servers to store the summarized form(s) in advance. Servers SHOULD mark the resources with the tag [SUBSETTED](file:///C:\temp\v3\SecurityIntegrityObservationValue\cs.html#SUBSETTED) to ensure that the incomplete resource is not acidentally used to overwrite a complete resource.

Note that the \_include and \_revinclude parameters cannot be mixed with \_summary=text.

**Elements**

If one of the summary views defined above is not appropriate, a client can request a specific set of elements be returned as part of a resource using the \_elements parameter:

GET [base]/Patient?\_elements=identifier,active,link

The \_elements parameter consists of a comma separated list of base element names such as, elements defined at the root level in the resource. Only elements that are listed are to be returned. Clients SHOULD list all mandatory and modifier elements in a resource as part of the list of elements. The list of elements does not apply to [included resources](file:///C:\temp\main-pages.html#include).

Servers are not obliged to return just the requested elements. Servers SHOULD always return mandatory elements whether they are requested or not. Servers SHOULD mark the resources with the tag [SUBSETTED](file:///C:\temp\v3\SecurityIntegrityObservationValue\cs.html#SUBSETTED) to ensure that the incomplete resource is not actually used to overwrite a complete resource.

**Relevance**

Where a search specifies a non-deterministic sort, the search algorithm may generate some kind of ranking score to indicate which resources meet the specified criteria better than others. The server can return this score in [entry.score](file:///C:\temp\bundle-definitions.html#Bundle.entry.score):

<entry>

<score value=".45"/>

<Patient>

... patient data ...

</Patient>

</entry>

The score is a decimal number with a value between (and including) 0 and 1, where 1 is best match, and 0 is least match.

**Server Conformance**

In order to allow the client to be confident about what search parameters were used as criteria by the server, the server SHALL return the parameters that were actually used to process the search. Applications processing search results SHALL check these returned values where necessary. For example, if the server did not support some of the filters specified in the search, a client might manually apply those filters to the retrieved result set, display a warning message to the user or take some other action.

In the case of a RESTful search, these parameters are encoded in the self link in the bundle that is returned:

<link>

<relation value="self"/>

<url value="http://example.org/Patient?name=peter"/>

</link>

In other respects, servers have considerable discretion with regards to supporting search:

* Servers can choose which parameters to support (other than \_id above).
* Servers can choose when and where to implement parameter chaining, and when and where they support the \_include parameter.
* Servers are able to declare additional parameters in the profiles referenced from their Capability statements. Servers should define search parameters starting with a "-" character to ensure that the names they choose do not clash with future parameters defined by this specification.
* Servers are not required to enforce case sensitivity on parameter names, though the names are case sensitive (and URLs are generally case-sensitive).
* Servers may choose how many results to return, though the client can use \_count as above
* Servers can choose how to sort the return results, though they SHOULD honor the \_sort parameter.

**Advanced Search**

The search framework described above is a useful framework for providing a simple search based on indexed criteria, but more sophisticated query capability is needed to handle precise queries, complex decision support based requests, and direct queries that have human resolution.

More advanced search operations are specified by the \_query parameter:

GET [base]/Patient?\_query=name&parameters...

The \_query parameter names a custom search profile that describes a specific query operation. The named query may define additional named parameters that are used with that particular named query. Servers can define their own additional named queries to meet their own uses using an [OperationDefinition](file:///C:\temp\operationdefinition.html).

There can only ever be one \_query parameter in a set of search parameters. Servers processing search requests SHALL refuse to process a search request if they do not recognize the \_query parameter value.

**Search Result Currency**

The results of a search operation are only guaranteed to be current at the moment the operation is executed. After the operation is executed, ongoing actions performed on the resources against which the search was executed will render the results increasingly stale. The significance of this depends on the nature of the search, and the kind of use that is being made of the results.

This is particularly relevant when the server is returning the results in a series of pages. It is at the discretion of the search engine of how to handle ongoing updates to the resources while the search is proceeding.

Note: Performing a search operation does not change the set of resources on the server, with the exception of the creation of [Audit Event](file:///C:\temp\auditevent.html) resources auditing the search itself.

**Summary Tables**

|  |  |  |  |
| --- | --- | --- | --- |
| Common Parameters defined for all resources: | | | |
| **Name** | **Type** | **Description** | **Paths** |
| [\_id](file:///C:\temp\main-pages.html#id) | [token](file:///C:\temp\main-pages.html#token) | Resource id (not a full URL) | Resource.id |
| [\_lastUpdated](file:///C:\temp\main-pages.html#lastUpdated) | [date](file:///C:\temp\main-pages.html#date) | Date last updated. Server has discretion on the boundary precision | Resource.meta.lastUpdated |
| [\_tag](file:///C:\temp\main-pages.html#tag) | [token](file:///C:\temp\main-pages.html#token) | Search by a resource tag | Resource.meta.tag |
| [\_profile](file:///C:\temp\main-pages.html#profile) | [uri](file:///C:\temp\main-pages.html#uri) | Search for all resources tagged with a profile | Resource.meta.profile |
| [\_security](file:///C:\temp\main-pages.html#security) | [token](file:///C:\temp\main-pages.html#token) | Search by a security label | Resource.meta.security |
| [\_text](file:///C:\temp\main-pages.html#text) | [string](file:///C:\temp\main-pages.html#string) | Text search against the narrative |  |
| [\_content](file:///C:\temp\main-pages.html#content) | [string](file:///C:\temp\main-pages.html#string) | Text search against the entire resource |  |
| [\_list](file:///C:\temp\main-pages.html#list) | [string](file:///C:\temp\main-pages.html#string) | All resources in nominated list (by id, not a full URL) |  |
| [\_query](file:///C:\temp\main-pages.html#query) | [string](file:///C:\temp\main-pages.html#string) | Custom named query |  |
| Search Control Parameters: | | | |
| **Name** | **Type** | **Description** | **Allowable Content** |
| [\_sort](file:///C:\temp\main-pages.html#sort) | [string](file:///C:\temp\main-pages.html#string) | Order to sort results in (can repeat for inner sort orders) | Name of a valid search parameter |
| [\_count](file:///C:\temp\main-pages.html#count) | [number](file:///C:\temp\main-pages.html#number ) | Number of results per page | Whole number |
| [\_include](file:///C:\temp\main-pages.html#include) | [string](file:///C:\temp\main-pages.html#string) | Other resources to include in the search results that search matches point to | SourceType:searchParam(:targetType) |
| [\_revinclude](file:///C:\temp\main-pages.html#revinclude) | [string](file:///C:\temp\main-pages.html#string) | Other resources to include in the search results when they refer to search matches | SourceType:searchParam(:targetType) |
| [\_summary](file:///C:\temp\main-pages.html#summary) | [string](file:///C:\temp\main-pages.html#string) | Just return the summary elements (for resources where this is defined) | true | false (false is default) |
| [\_contained](file:///C:\temp\main-pages.html#contained) | [string](file:///C:\temp\main-pages.html#string) | Whether to return resources contained in other resources in the search matches | true | false | both (false is default) |
| [\_containedType](file:///C:\temp\main-pages.html#containedType) | [string](file:///C:\temp\main-pages.html#string) | If returning contained resources, whether to return the contained or container resources | container | contained |

Cross-map between search parameter types and Data types:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Type** | [**number**](file:///C:\temp\main-pages.html#number) | [**date**](file:///C:\temp\main-pages.html#date) | [**string**](file:///C:\temp\main-pages.html#string) | [**token**](file:///C:\temp\main-pages.html#token) | [**reference**](file:///C:\temp\main-pages.html#reference) | [**quantity**](file:///C:\temp\main-pages.html#quantity) | [**uri**](file:///C:\temp\main-pages.html#uri) |
| [boolean](file:///C:\temp\datatypes.html#boolean) |  |  |  | . true|false (System = http://hl7.org/fhir/special-values but not usually needed) |  |  |  |
| [code](file:///C:\temp\datatypes.html#code) |  |  |  | . (System, if desired, is defined in the underlying value set for eeach code) |  |  |  |
| [date](file:///C:\temp\datatypes.html#date) |  |  |  |  |  |  |  |
| [dateTime](file:///C:\temp\datatypes.html#dateTime) |  |  |  |  |  |  |  |
| [decimal](file:///C:\temp\datatypes.html#decimal) |  |  |  |  |  |  |  |
| [instant](file:///C:\temp\datatypes.html#instant) |  |  |  |  |  |  |  |
| [integer](file:///C:\temp\datatypes.html#integer) |  |  |  |  |  |  |  |
| [string](file:///C:\temp\datatypes.html#string) |  |  |  |  |  |  |  |
| [uri](file:///C:\temp\datatypes.html#uri) |  |  |  |  |  |  |  |
| [Address](file:///C:\temp\datatypes.html#Address) |  |  | search on any string element in the address |  |  |  |  |
| [Annotation](file:///C:\temp\datatypes.html#Annotation) |  |  |  |  |  |  |  |
| [CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept) |  |  |  |  |  |  |  |
| [Coding](file:///C:\temp\datatypes.html#Coding) |  |  |  |  |  |  |  |
| [ContactPoint](file:///C:\temp\datatypes.html#ContactPoint) |  |  |  |  |  |  |  |
| [Duration](file:///C:\temp\datatypes.html#Duration) |  |  |  |  |  |  |  |
| [HumanName](file:///C:\temp\datatypes.html#HumanName) |  |  | Search on any string element in the name |  |  |  |  |
| [Identifier](file:///C:\temp\datatypes.html#Identifier) |  |  |  |  |  |  |  |
| [Period](file:///C:\temp\datatypes.html#Period) |  |  |  |  |  |  |  |
| [Quantity](file:///C:\temp\datatypes.html#Quantity) |  |  |  |  |  |  |  |
| [Range](file:///C:\temp\datatypes.html#Range) |  |  |  |  |  |  |  |
| [Reference](file:///C:\temp\datatypes.html#Reference) |  |  |  |  |  |  |  |
| [SampledData](file:///C:\temp\datatypes.html#SampledData) |  |  |  |  |  |  |  |
| [Timing](file:///C:\temp\datatypes.html#Timing) |  |  |  |  |  |  |  |

**searchparameter-registry.html**

**Defined Search Parameters**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page lists all search parameters defined as part of this specification. For further details regarding using search parameters, see [searching](file:///C:\temp\search.html) on the [RESTful API](file:///C:\temp\http.html).

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Description** | **Paths** |

**search\_filter.html**

**\_filter Parameter**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Introduction**

\_filter is a parameter that can be used with the [Search Operation](file:///C:\temp\search.html). It uses the same set of standard parameters defined for the resources, and provides a syntax for expressing a set of query expressions on the underlying resources.

Examples of filters:

* Patient: name co "pet" - all patients with the characters "pet" in a given or family name
* Patient: given eq "peter" and birthdate ge 2014-10-10 - all patients with a given name of peter, born on or after 10-Oct 2014
* Observation: name eq http://loinc.org|1234-5 - all observations with the loinc code "1234-5"
* Observation: subject.name co "pet" - all observations on a patient with the characters "pet" in a given or family name
* Observation: related[type eq "has-component"].target pr true - all observations that have component observations (note: this uses one of the search parameters defined for this mechanism, see below)
* Observation: related[type eq has-component].target re Observation/4 - all observations that have Observation/v as a component

The \_filter syntax has the following features:

* A filter can be a logical one (x or x, or x and x, or not x).
* A filter can contain other filters in a set of parentheses : "()".
* A filter can be a test - path operation value, where operation is taken from the table below, and value is either a "true", "false", a JSON string, or a token (any sequence of non-whitespace characters, excluding ")" and "]". Values are never case sensitive.
* A 'path' is a name, with chained searches done by name.name etc.as per existing source. There can also be a filter: name[filter].name.
* The name is one of the defined search parameters that are used with the other search mechanism, with some special exemptions defined below.

Note: The only difference between a "string" value and a "token" value is that a string can contain spaces and ) and ]. There is otherwise no significant difference between them.

Formal grammar for the syntax:

filter = paramExp / logExp / ("not") "(" filter ")"

logExp = filter ("and" / "or" filter)+

paramExp = paramPath SP compareOp SP compValue

compareOp = (see table below)

compValue = string / numberOrDate / token

string = json string

token = any sequence of non-whitespace characters (by Unicode rules) except "]" and ")"

paramPath = paramName (("[" filter "]") "." paramPath)

paramName = ALPHA (nameChar)\*

nameChar = "\_" / "-" / DIGIT / ALPHA

numberOrDate = DIGIT (DateChar)\*

dateChar = DIGIT / "T" / "-" / "." / "+"

Notes about using the syntax:

* Logical expressions are evaluated left to right, with no precedence between "and" and "or". If there is ambiguity, use parentheses to be explicit.
* Rhe compareOp is always evaluated against the set of values produced by evaluating the param path.
* Rhe parameter names are those defined by the specification for search parameters, except for those defined below.
* Rhe date format is a standard XML (i.e. XSD) dateTime (including timezone).

**Operators**

This table summarizes the comparison operations available:

|  |  |
| --- | --- |
| Operation | Definition |
| eq | an item in the set has an equal value |
| ne | An item in the set has an unequal value |
| co | An item in the set contains this value |
| sw | An item in the set starts with this value |
| ew | An item in the set ends with this value |
| gt / lt / ge / le | A value in the set is (greater than, less than, greater or equal, less or equal) the given value |
| ap | A value in the set isis approximately the same as this value. Note that the recommended value for the approximation is 10% of the stated value (or for a date, 10% of the gap between now and the date), but systems may choose other values where appropriate |
| sa | The value starts after the specified value |
| eb | The value ends before the specified value |
| pr | The set is empty or not (value is false or true) |
| po | True if a (implied) date period in the set overlaps with the implied period in the value |
| ss | True if the value subsumes a concept in the set |
| sb | True if the value is subsumed by a concept in the set |
| in | True if one of the concepts is in the nominated value set by URI, either a relative, literal or logical vs |
| ni | True if none of the concepts are in the nominated value set by URI, either a relative, literal or logical vs |
| re | True if one of the references in set points to the given URL |

For detailed rules about the operators eq, ne, le, ge, lt, gt, sa, and eb see [Search Prefixes](file:///C:\temp\search.html#prefix).

The interpretation of the operation depends on the type of the search parameter it is being evaulated against. This table contains those details:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation | String | Number | Date | Token | Reference | Quantity |
| Eq | Character sequence is the same (case insensitive) | Number is the same incl same precision | Date is the same including same precision and timezone if provided | Token is the same, including namespace if specified (case insensitive) | n/a | Unit and value are the same |
| Ne | (same) | | | | | |
| Co | Character sequence matches somewhere (case insensitive) | An item in the set's implicit imprecision includes the stated value | An item in the set's implicit period includes the stated value | n/a | n/a | n/a? |
| Sw | Character sequence matches from first digit (left most, when L->R) (case insensitive) | n/a | n/a | n/a | n/a | n/a |
| ew | Character sequence matches up to last digit (right most, when L->R) (case insensitive) | n/a | n/a | n/a | n/a | n/a |
| gt / lt / ge / le | Based on Integer comparison of Unicode code points of starting character (trimmed) (case insensitive) | Based on numerical comparison | Based on date period comparison per 2.2.2.3 | n/a | n/a | Based on numerical comparison if units are the same (or are canonicalised) |
| pr |  |  |  |  |  |  |
| po | n/a | n/a | Based on date period comparison per 2.2.2.3 |  | n/a | n/a |
| ss | n/a | n/a | n/a | Based on logical subsumption; potentially catering for mapping between tx | n/a | n/a |
| sb | n/a | n/a | n/a | Based on logical subsumption; potentially catering for mapping between tx | n/a | n/a |
| in | n/a | n/a | n/a | Based on logical subsumption; potentially catering for mapping between tx | n/a | n/a |
| re | n/a | n/a | n/a | n/a | Relative or absolute url | n/a |

Note:

* For token, the format is the same as the existing search parameter.
* For convenience, the codes "loinc", "snomed", "rxnorm" and "ucum" are predefined and can be used in place of their associated full namespace.

**Additional Parameters**

Some additional parameters are defined for the filter parameter (*to do: move these into the standard parameters*):

|  |  |  |
| --- | --- | --- |
| Resource Type | Parameter Name | Children |
| Observation | related | target = related-target  Type = related-type |
| Group | characteristic | value = value  code = characteristic |
| DocumentReference | relatesTo | code = relation  target = relatesTo |
| ProcedureRequest | event | status = event-status date = event-date |
| ProcedureRequest | item | status = item-status  code = item-code  site = bodysite  event = item-event |
| ProcedureRequest | item-event | status = item-past-status date = item-date actor = actor |

Note:

* Any time these names are used in a parameter, they must have a filter and a chained name under them.
* The first column is the resource type against which this name can be used.
* The second column is the parameter name that is used.
* The third column defines the names that can be used in the chained parameter, and in the filter, and shows which existing search parameters they equate to.
* For example, you could search on Observation for '\_filter=related[type eq has-component].target re url'. "type" refers to the search parameter "related-type", and "target" to the search parameter "related-target".

**secpriv-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Security and Privacy Module**

**Introduction**

The Security and Privacy Module describes how to protect a FHIR server (through access control and authorization), how to document what permissions a user has granted (consent), and how to keep records about what events have been performed (audit logging and provenance). FHIR does not mandate a single technical approach to security and privacy; rather, the specification provides a set of building blocks that can be applied to create secure, private systems.

**Index**

The Security and Privacy module includes the following materials:

**FHIR Implementation Guidance and Principles:**

* [Security](file:///C:\temp\security.html)
* [Security Labels](file:///C:\temp\security-labels.html)

**FHIR Resources:**

* [Consent](file:///C:\temp\consent.html)
* [Provenance](file:///C:\temp\provenance.html)
* [Audit Event](file:///C:\temp\auditevent.html)

**FHIR Datatypes:**

* [Signature](file:///C:\temp\datatypes.html#signature)

**Security**

Security in FHIR includes the set of considerations required to ensure that data can be discovered, accessed, or altered only in accordance with expectations and policies. FHIR includes implementation guidance to ensure that:

* All communications can be encrypted to prevent unauthorized access
* No information leaks when errors occur
* No active script content can be injected into narrative resources
* Full audit trails can be constructed and used to detect anomalous access patterns
* For general considerations and principles, see [Security](file:///C:\temp\security.html).

**Privacy**

Privacy in FHIR includes the set of considerations required to ensure that individual data are treated according to an individual's preferences. FHIR includes implementation guidance to ensure that:

* Individual preferences can be communicated through standards-based protocols (e.g., OAuth, User-Managed Access) or using an explicit FHIR representation ([Consent](file:///C:\temp\consent.html))
* Resources can be tagged to indicate the sensitivity or confidentiality of the data they represent ([Security Labels](file:///C:\temp\security-labels.html))
* Data access records and audit logs can be shared with individuals, e.g. for accounting of disclosures ([Audit Event](file:///C:\temp\auditevent.html))

**Common Use Cases**

**Authorization and Access Control**

*Use case:* A FHIR server should ensure that API access is allowed for authorized requests, and denied for unauthorized requests.

*Approach:* Authorization details can vary according to local policy, and according to the access scenario (e.g. sharing data among institution-internal subsystems vs. sharing data with trusted partners vs. sharing data with third-party user-facing apps). In general, FHIR enables a separation of concerns between the FHIR REST API and standards-based authorization protocols like OAuth. For the use case of user-facing third-party app authorization, we recommend the OAuth-based SMART protocol (ref) as an externally-reviewed authorization mechanism with a real-world deployment base Ã¢â‚¬â€ but we note that community efforts are underway to explore a variety of approaches to authorization. For details, see [Security: Binding](file:///C:\temp\security.html#binding).

**Audit Logging**

*Use case:* "A FHIR server should keep a complete, tamper-proof log of all API access and other security- and privacy-relevant events".

*Approach:* Approach: FHIR provides an AuditEvent resource suitable for use by FHIR clients and servers to record when a security or privacy relevant event has occurred. This form of audit logging records as much detail as reasonable at the time the event happened. For details, see [Security: Audit](file:///C:\temp\security.html#audit) + add reference to ATNA for conseptual background (GG).

**Developmental Roadmap**

In the STU3 release, FHIR includes building blocks and principles for creating secure, privacy-oriented health IT systems; FHIR does not mandate a single technical approach to security and privacy.

In future releases, we are anticipate including:

* More specific implementation guidance for particular Authorization scenarios (for example, we may incorporate the SMART on FHIR authorization specification for user-authorized apps).
* More details about how to use digital Signatures for data integrity and non-repudiation, including an approach that supports some level of manipulation of resources (e.g. separating the entries in a bundle, or conversion between XML and JSON during processing)
* More detailed Consent management, including support for specific consent use cases.

**security-labels.html**

**Security Labels**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

A security label is a [concept](file:///C:\temp\datatypes.html#Coding) attached to a resource or bundle that provides specific security metadata about the information it is fixed to. The [Access Control decision engine](file:///C:\temp\security.html#access-control) uses the security label together with any provenance resources associated with the resource and other metadata (e.g. the resource type, resource contents, etc.) to

* approve read, change, and other operations
* determine what level of the resource can be returned
* determine what handling caveats must be conveyed with the data

Security Labels enable more data to flow as they enable policy fragments to accompany the resource data.

The intent of a security label is that the recipient of resources or bundles with security-tags is obligated to enforce the handling caveats of the tags and carry the security labels forward as appropriate.

Security labels are only a device to connect specific resources, bundles, or operations to a wider security framework; a full set of policy and consent statements and their consequent obligations is needed to give the labels meaning. As a consequence of this, security labels are most effective in fully trusted environments - that is, where all trading partners have agreed to abide by them in a Mutual Trust Framework. Note also that security labels support policy, and specific tagging of individual resources is not always required to implement policy correctly.

In the absence of this kind of pre-agreement, Security Labels may still be used by individual parties to assist with security role checking, but they may not all be recognized and enforced, which in turn limits what information is allowed to flow.

Local agreements and implementation profiles for the use security labels should describe how the security labels connect to the relevant consent and policy statements, and in particular:

* Which Security Labels are able to be used
* What to do if a resource has an unrecognized security label on it
* Authoring obligations around security labels
* Operational implications of security labels

This specification defines a basic set of labels for the most common use cases trading partners, and also a wider array of security labels that allow much finer grained management of the information.

**Representing Security Labels**

A security label is represented as a [Coding](file:///C:\temp\datatypes.html#Coding), with the following important properties:

|  |  |
| --- | --- |
| system | The coding scheme from which label is taken (see [code system URI](file:///C:\temp\terminologies-systems.html), and below) |
| code | a code from the coding scheme that identifies the security label and code is an value from the code system |
| display | The display form for the code (mostly for use when a system doesn't recognize the code) |

An XML patient resource with a "celebrity" tag associated with it, as represented in an HTTP response:

<Patient xmlns="http://hl7.org/fhir">

<meta>

<security>

<system value="http://hl7.org/fhir/v3/ActCode"/>

<code value="CEL"/>

<display value="Celebrity"/>

</security>

</meta>

... [snip] ...

</Patient>

A JSON search result that includes a resource that the receiving application must delete all copies of the resource after using it:

{

"resourceType" : "Bundle",

"type" : "searchset",

... other headers etc.....

"entry" : [

... other entries ....

{

"resource": {

"id" : "1",

"meta" : {

"security" : [{

"system" : "http://hl7.org/fhir/v3/ActCode",

"code" : "DELAU",

"display" : "delete after use"

}]

}

... other content etc.....

}

},

... other entries ....

]

}

Note: the actual terms used in these examples are described below.

The basic framework for security labels is described by the HL7 Healthcare Classification System (HCS; ref todo). This specification identifies how security labels are defined and provides a relatively comprehensive list of labels. All of the HCS defined labels (see below for the lists) can be used as security labels on FHIR resources and bundles (e.g. requests and responses).

In addition, other security labels not defined here or in the HCS can be defined by jurisdictions, vendors and/or projects and used as appropriate. However, note that:

* Defining additional security labels will increase costs associated with information and system portability
* Implementation guides and applications SHOULD always use the applicable label defined by the HCS if one exists

Note: The use of security labels and the expression of common shared security policies is a matter of ongoing discussion and development in several communities at this time.

**Core Security Labels**

This specification defines a set of core security labels for all FHIR systems. All conformant FHIR Applications SHOULD use these labels where appropriate. For all of these labels, how they are operationalised - their use and interpretation - is subject to the applicable Mutual Trust Framework agreement as described above. These codes all come from one of two code systems: http://hl7.org/fhir/v3/Confidentiality, and http://hl7.org/fhir/v3/ActCode,

|  |  |
| --- | --- |
| **Name/ Tag** | **Description** |
| **Context of Use** | |
| Confidentiality codes | These [confidentiality class](file:///C:\temp\v3\ConfidentialityClassification\vs.html) (system = http://hl7.org/fhir/v3/Confidentiality) can be applied to any resource or bundle. They are generally assigned by the author of the resource, but can be modified subsequently as a matter of operational management. The Confidentiality classifications describe the sensitivity of the information in a resource with regard to whether it should made available or disclosed to unauthorized individuals, entities, or processes.  Notes:   * In the absence of a confidentiality code, the basic confidentiality of a resource may be implied by its definition and content; e.g. a patient's condition is far more likely to be confidential than a practitioner resource, and a Diagnostic Report with an HIV test is always highly confidential, whereas a routine electrolytes is rarely particularly confidential * A few resources have a confidentiality code in the resource itself. This should always be understood as the original intended confidentiality, whereas a confidentiality tag is the current confidentiality of the content; e.g. the confidentiality may change in response to patient concern * The confidentiality of a bundle is always as confidential as the most confidential resource in the bundle   The additional more specific labels below are defined to support very specific fine-grained access control, and should always be used in association with an appropriate confidentiality label. |
| Celebrity / VIP: ActCode.[CEL](file:///C:\temp\v3\ActCode\cs.html#CEL) | Use on any resource to indicate that the subject/patient is a celebrity or well known to the staff in the institution.  Notes:   * This may be applied to the [Patient](file:///C:\temp\patient.html) resource, with implied behavior for the [entire patient compartment](file:///C:\temp\compartmentdefinition.html), or it may be applied to individual resources * Resources affected by this label are more likely to be the subject of active audit maintenance or additional security policy |
| Staff: ActCode.[EMP](file:///C:\temp\v3\ActCode\cs.html#EMP) | Use on a [Patient](file:///C:\temp\patient.html) resource and resources with a subject of that patient to indicate that the patient is a staff member of the institution. This is a variation on being a celebrity.  Notes:   * This may be applied to the [Patient](file:///C:\temp\patient.html) resource, with implied behavior for the [entire patient compartment](file:///C:\temp\compartmentdefinition.html), or it may be applied to individual resources * Resources affected by this label are (even) more likely to be the subject of active audit maintenance or additional security policy |
| Keep information from patient: ActCode.[TBOO](file:///C:\temp\v3\ActCode\cs.html#TBOO) | Used on [any](file:///C:\temp\resourcelist.html) resource to indicate that information is not to be made available to the patient or their relatives/carers, except by the personal decision of a physician assigned to the patient.  Notes:   * A common use for this is with [Flag](file:///C:\temp\flag.html) resources, when the alert records information on patient abuse or non-compliance * This label might also be used temporarily on laboratory or other test results where policy is for the results to be initially disclosed by direct discussion with the patient's physician |
| Contact/Employment Details Confidential: ActCode.[DEMO](file:///C:\temp\v3\ActCode\cs.html#DEMO) | Used on a [Patient](file:///C:\temp\patient.html) resource to indicate that the patient's address and contact details (phone numbers, email addresses) - including employment information - are sensitive and shouldn't be shared with the patient's family or others without specific authorization |
| Diagnosis-related confidentiality: ActCode.[DIA](file:///C:\temp\v3\ActCode\cs.html#DIA) | Used on [any](file:///C:\temp\resourcelist.html) resource to indicate that the resource relates to a diagnosis (or potential diagnosis) which is generally associated with confidentiality requirements - or is for this particular patient. This may be associated for diagnoses including STDs, psychiatric conditions, adolescent related issues, drug abuse, genetics conditions and others.  Notes:   * Generally, this security label cascades logically; e.g. any [Diagnostic Reports](file:///C:\temp\diagnosticreport.html) produced because of a [Procedure Request](file:///C:\temp\procedurerequest.html) with this security label should also have the same security label. * There may be additional labels classifying the diagnosis; such labels SHOULD always be accompanied by this label so that more systems will know that restrictions apply |
| Author Consent needed: ActCode.[ORCON](file:///C:\temp\v3\ActCode\cs.html#ORCON) | The author's consent is needed for disclosure. Typically, this is used by a treating practitioner to label portions of their own record confidential. Any such resource is only shared with the author or with other parties as arranged. |
| **Control of Flow** | |
| Delete After Use: ActCode.[DELAU](file:///C:\temp\v3\ActCode\cs.html#DELAU) | An application receiving a resource with this label must delete all copies after the immediate use for which the resource/feed was exchanged is complete.  Notes:   * This may imply a prohibition not storing the resource in any audit trail as well * Additional security labels are allowed to make exceptions to the blanket restriction this implies. This allows a resource to be exchanged with a blanket rule not to retain copies unless the exact rules for retaining it can be followed |
| Do Not Re-use: ActCode.[NOREUSE](file:///C:\temp\v3\ActCode\cs.html#NOREUSE) | An application receiving a resource with this label may only use it for the immediate purpose of use. In particular, the application is not authorized to re-distribute (i.e. exchange this resource with any other application).  Notes:   * The exact interpretation of "immediate purpose of use" and the boundaries of "the application" are determined by local policy * Additional security labels are allowed to make exceptions to the blanket restriction this implies. This allows a resource to be exchanged with a blanket rule not to re-use unless the exact rules for doing so can be followed |

**Break The Glass**

There is a special security label to support the commonly encountered "break-the-glass" protocol, where a clinician (usually in an emergency context) requests emergency unauthorized access to the patient's record. This specification does not make any policy recommendations or rules about the operation, merely provides support for it. See [this paper](http://www.hl7.org/search/viewSearchResult.cfm?search_id=393442&search_result_url=%2Fdocumentcenter%2Fpublic%2Fwg%2Fsecure%2FHL7%20Emergency%20Access%2Edoc) for discussion of the issues involved in break-the-glass operations.

When the operation occurs, it is represented as a security label on the request, rather than on a resource, and so is represented differently. The break the glass tag needs to be used as part of an agreed policy and protocol. FHIR does not attempt to define this policy or protocol, it must be agreed on an implementation by implementation basis. For example as a URL:

|  |  |  |
| --- | --- | --- |
| Break The Glass | http://hl7.org/fhir/security-label#break-the-glass | The requester is asking for emergency access for patient treatment. Typically, this means that the patient is unconscious and not able to provide relevant information or consent. |

The URL is represented in the request as a [web category](https://tools.ietf.org/html/draft-johnston-http-category-header-02):

HTTP/1.1 GET fhir/Patient/482735/condition

Content-Type: text/xml

Access-Control-Allow-Origin: \*

Last-Modified: Thu, 19 Nov 2013 07:07:32 +1100

ETag: 24

Category: http://hl7.org/fhir/security-label#break-the-glass; scheme="http://hl7.org/fhir/tag/security"; label="Break The Glass"

**Healthcare Privacy and Security Classification System (HCS)**

The security labels described above are a subset of the full set of security labels defined by the HL7 Healthcare Privacy and Security Classification System (HCS; ref todo). The HCS defines 5 categories of security labels that may be applied to a resource:

|  |  |  |  |
| --- | --- | --- | --- |
| **Security Label** | **Card.** | **Values** | **Description** |
| Confidentiality Classification | 0..1 | [ConfidentialityClassification](file:///C:\temp\v3\ConfidentialityClassification\vs.html) | Security label metadata classifying an IT resource (clinical fact, data, information object, service, or system capability) according to its level of sensitivity, which is based on an analysis of applicable privacy policies and the risk of financial, reputational, or other harm to an individual or entity that could result if made available or disclosed to unauthorized individuals, entities, or processes.  Example Uses: Unrestricted, Normal, Very restricted |
| Sensitivity Category | 0..\* | [InformationSensitivityPolicy](file:///C:\temp\v3\InformationSensitivityPolicy\vs.html) | Security label metadata that "segments" an IT resource by categorizing the value, importance, and vulnerability of an IT resource perceived as undesirable to share.  Example Uses: STDs, Psychiatric care, Celebrity status |
| Compartment Category | 0..\* | [Compartment](file:///C:\temp\v3\Compartment\vs.html) | Security label metadata that "segments" an IT resource by indicating that access and use is restricted to members of a defined community or project  Note: this is a different use of "Compartment" to the [Patient Compartment](file:///C:\temp\compartmentdefinition.html) use.  Example Uses: Research, HR records |
| Integrity Category | 0..\* | [SecurityIntegrityObservationValue](file:///C:\temp\v3\SecurityIntegrityObservationValue\vs.html) | Security label metadata that "segments" an IT resource by conveying the completeness, veracity, reliability, trustworthiness, and provenance of an IT resource  Example Uses: Anonymized, signed, patient reported |
| Handling Caveat | 0..\* | [SecurityControlObservationValue](file:///C:\temp\v3\SecurityControlObservationValue\vs.html) | Security label metadata conveying dissemination controls and information handling instructions such as obligations and retention policies to which an IT resource custodian or receiver must comply. This type of handling caveat SHALL be assigned to a clinical fact if required by jurisdictional or organizational policy, which may be triggered by a patient consent directive  Example Uses: do not disclose, various restrictions on use, and policy marks |

Each of these security labels identifies a [ValueSet](file:///C:\temp\valueset.html) that lists a set of possible codes for the security label.

**Jurisdiction Specific Security Labels**

The HL7 Healthcare Classification System also allows for Realm-specific privacy law or policy category codes for use in security labels in particular domains. These domains are included with this specification:

|  |  |  |  |
| --- | --- | --- | --- |
| **Security Label** | **Card.** | **Values** | **Description** |
| US Privacy Law | 0..\* | [ActUSPrivacyLaw](file:///C:\temp\v3\ActUSPrivacyLaw\vs.html) | Security label metadata that "segments" an IT resource by indicating the legal provisions to which the assignment of a Confidentiality Classification complies in the US. |

**security.html**

**FHIR Security**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Fast Healthcare Interoperability Resources (FHIR) is not a security protocol, nor does it define any security related functionality. However, FHIR does define exchange protocols and content models that need to be used with various security protocols defined elsewhere. This section gathers all information about security in one section. A summary:

* Time Keeping - all clocks should be synchronised using NTP/SNTP, and the design of the system should be robust against a system clock with the wrong value
* Communications Security - all exchange of production data should be secured using TLS/SSL (e.g. https)
* Authentication - Users/Clients may be authenticated in any way desired. For web-centric use, OAuth is recommended
* Authorization/Access Control - FHIR defines a Security Label infrastructure to support access control management. FHIR may also define a set of resources to administer access control management, but does not define any at present
* Audit - FHIR defines [provenance](file:///C:\temp\provenance.html) and [audit event](file:///C:\temp\auditevent.html) resources suitable for tracking the origins, authorship, history, status, and access of resources
* Digital Signatures - FHIR includes several specifically reserved locations for digital signatures
* Attachments - FHIR allows for binary resources and attachments. These have their own concerns
* [Labels](file:///C:\temp\security-labels.html) - FHIR allows for set of security related tags that affect that way resources are handled
* Data Management Policies - FHIR defines a set of capabilities to support data exchange. Not all the capabilities that FHIR enables may be appropriate or legal for use in some combinations of context and jurisdiction (e.g. HIPAA for exchange between institutions). It is the responsibility of implementers to ensure that relevant regulations and other requirements are met
* Narrative - Care must be taken when displaying the narrative from FHIR resources

Time critical concerns regarding security flaws in the FHIR specification should be addressed to the [FHIR email list](http://wiki.hl7.org/index.php?title=FHIR_email_list_subscription_instructions) for prompt consideration.

Implementers should track the developing IHE IUA Profile for additional security considerations.

**General Considerations**

A production FHIR system will need some kind of security sub-system that administers users, user authentication, and user authorization. Where this subsystem fits into the deployment architecture is a matter for system design:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  |  | | --- | --- | |  | The consumer that is using a healthcare related system | |  | The client application the user is using (application, mobile app, website, etc.) | |  | The security system (authentication and access control) | |  | The clinical/healthcare repository | |

In this diagram, the red lines represent FHIR interfaces. From the perspective of the FHIR API, the client (consumer of FHIR services) may either interact with a security system that manifests as a FHIR server, and which depends on a subsequent FHIR interface to provide the actual storage, or either the client or server interacts with the security system independently. In each of these 3 scenarios, the different components may be assembled into applications or network components differently, but the same logical layout applies. The FHIR specification assumes that a security system exists, and that it may be deployed in front of or behind the FHIR API.

The security system includes the following subsystems:

* Authentication: identifies and authenticates the user
* Access Control decision engine: decides whether FHIR operations are allowed
* Audit Log: records actions to allow for subsequent review and detection of intrusion or inappropriate usage

Because there are a plethora of standards relating to the administration and functionality of the security system, FHIR does not provide user, profile, or other such administration resources. Instead, the FHIR resources are the targets of the policies expressed in these other approaches. What FHIR does specify is a way to apply [security labels](file:///C:\temp\security-labels.html) to resources so that a security system may use these (along with the contents of the resources if appropriate) to determine whether a user is authorized to perform a particular FHIR operation or not.

**Communications**

For the [RESTful API](file:///C:\temp\http.html), normal HTTP security rules apply. Please follow the [HTTP specification Security Considerations section 15](http://www.w3.org/Protocols/rfc2616/rfc2616-sec15.html). The [Service Base URL](file:///C:\temp\http.html#root) will specify whether SSL is required. Client authentication may be required by the server, possibly including the requirement for client certificates.

TLS/SSL SHOULD be used for all production data exchange. The TLS/SSL communications are established prior to any HTTP command/response, so the whole FHIR interaction is protected by the SSL/TLS communications. The security of the endpoints of the TLS/SSL communications must be risk-managed, so as to prevent inappropriate risks (e.g. audit logging of the GET parameters into an unprotected audit log).

To support browser-based client applications, servers SHOULD implement [cross-origin resource sharing](http://enable-cors.org/) for the [REST operations](file:///C:\temp\http.html).

**Authentication**

Other than testing systems, FHIR servers should authenticate the clients. The server may choose to authenticate the client system and trust it, or to authenticate the individual user by a variety of techniques. For web-centric use, [OpenID Connect](http://openid.net/connect/) may be used to authenticate users and [OAuth](http://oauth.net/) may be used to authenticate and/or authorize the users. The [Smart-On-FHIR](http://docs.smarthealthit.org/) profile on OAuth is tightly integrated with FHIR and is the preferred method for using OAuth.

The [HEART Working Group](https://openid.net/wg/heart/) has developed a set of privacy and security specifications that enable an individual to control the authorization of access to RESTful health-related data sharing APIs, and to facilitate the development of interoperable implementations of these specifications by others.

**Authorization/Access Control**

Correctly identifying people, devices, locations and organizations is one of the foundations that any security system is built on. Most applications of security protocols, whether authentication, access control, digital signatures, etc. rely on the correct mapping between the relevant resources and the underlying systems. Note that this isn't necessary. There is nothing in FHIR that requires or relies on any security being in place, or any particular implementation. However, real world usage will generally require this.

A holder of data should not allow the data to be communicated unless there are sufficient assurances that the other party is authorized to receive it. This is true for a client creating a resource through a PUT/POST, as much as it is true for a server returning resources on a GET. The presumption is that without proper authorization, to the satisfaction of the data holder, the data does not get communicated.

Two of the classic Access Control models are: Role-Based Access Control (RBAC), and Attribute-Based Access Control (ABAC).

In Role-Based Access Control (RBAC), permissions are operations on an object that a user wishes to access. Permissions are grouped into roles. A role characterizes the functions a user is allowed to perform. Roles are assigned to users. If the userÃ¢â‚¬â„¢s role has the appropriate permissions to access an object, then that user is granted access to the object. FHIR readily enables RBAC, as FHIR Resources are object types and the CRUDE (Create, Read, Update, Delete, Execute) events (the FHIR equivalent to permissions in the RBAC scheme) are operations on those objects.

In Attribute-Based Access Control (ABAC), a user requests to perform operations on objects. That user's access request is granted or denied based on a set of access control policies that are specified in terms of attributes and conditions. FHIR readily enables ABAC, as instances of a Resource in FHIR (again, Resources are object types) can have attributes associated with them. These attributes include security tags, environment conditions, and a host of user and object characteristics, which are the same attributes as those used in ABAC. Attributes help define the access control policies that determine the operations a user may perform on a Resource (in FHIR) or object (in ABAC). For example, a tag (or attribute) may specify that the identified Resource (object) is not to be further disclosed without explicit consent from the patient.

The rules behind the access control decision are often very complex, and potentially depends on information sourced from:

* Client, such as user identity, user role, location, level of assurance
* Resource, such as confidentiality, sensitivity, type of data, date ranges covered by the data, author of the data
* Patient, such as the patient identity, patient relationship to the user, patient consent policies
* Context of the transaction, system identity, time-of-day, purpose of use, workflow state, and transport security

For one source of further information, see the [IHE Access Control white paper](http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_WhitePaper_AccessControl_2009-09-28.pdf)

Access control constraints may result in data returned in a read or search being redacted or otherwise restricted. See [Variations between Submitted data and Retrieved data](file:///C:\temp\updates.html).

**Access Control Considerations**

The FHIR RESTful API provides a number of ways that a client may request or create information. When designing a system to authorize access to information, all potential access methods must be considered. They include the following:

* The basic CRUD methods on resources. A security implementation must evaluate whether a client can read, update create or delete a given resource.
* Chained search provides the ability to disclose information on related resources. A security implementation must consider whether a client has the permission to access the resource being searched on, as well as the chained resource(s)
* \_include and \_revinclude search parameters allow client to request related resources. A security implementation must determine if the client has access to the included resources.
* [security labels](file:///C:\temp\security-labels.html)
* Several resources, including Bundle, Composition, Group and List, are designed to contain other resources. A security implementation should consider whether access to an individual resource, such as a Bundle, should permit access to all resources contained within the resource.
* FHIR defines several operations that may be supported by a server. Security implementations must evaluate whether a client has the ability to invoke these operations and what information should be returned from them. Fetch Encounter Record, Evaluate Measure, Observation Statistics, Find patient matches using MPI based logic and Fetch Patient Record specifically provide the ability to disclose patient information.
* Batch and transaction processing provide ways for clients to create and update information in bulk. Security implementations should consider whether a client has the ability to initiate one of these interactions and make authorization decisions on each action in the batch/transaction.
* Security implementations must be aware of the [Break the Glass protocol](file:///C:\temp\security-labels.html#break-the-glass) (e.g. break the glass) ([example](file:///C:\temp\operationoutcome-example-break-the-glass.html)).

**Access Denied Response Handling**

A web-server, especially hosting FHIR, must choose the response carefully when an Access Denied condition exists. Returning too much information may expose details that should not be communicated. The Access Denied condition might be because of missing but required Authentication, the user is not authorized to access the endpoint, the user is not authorized to access specific data, or other policy reasons.

To balance usability of the returned result vs appropriate protection, the actual result method used needs to be controlled by policy and context. Typical methods of handling Access Denied used are:

**Return a Success with Bundle containing zero results** Ã¢â‚¬â€œ This result is indistinguishable from the case where no data is known. When consistently returned on Access Denied, this will not expose which patients exist, or what data might be blinded. This method is also consistent with cases where some results are authorized while other results are blinded. This can only be used when the returning a Bundle is a valid result.

**Return a 404 Ã¢â‚¬Å“Not FoundÃ¢â‚¬Â** Ã¢â‚¬â€œ This also protects from data leakage as it is indistinguishable from a query against a resource that doesnÃ¢â‚¬â„¢t exist. It does however leak that the user authentication is validated.

**Return a 403 Ã¢â‚¬Å“ForbiddenÃ¢â‚¬Â** Ã¢â‚¬â€œ This communicates that the reason for the failure is an Authorization failure. It should only be used when the client and/or user is well enough known to be given this information. Thus this method is most used when the user is allowed to know that they are forbidden access. It doesnÃ¢â‚¬â„¢t explain how the user might change things to become authorized.

**Return a 401 Ã¢â‚¬Å“UnauthorizedÃ¢â‚¬Â** Ã¢â‚¬â€œ This communicates that user authentication was attempted and failed to be authenticated.

**Audit Logging**

FHIR provides an [AuditEvent](file:///C:\temp\auditevent.html) resource suitable for use by FHIR clients and servers to record when a security or privacy relevant event has occurred. This form of audit logging records as much detail as reasonable at the time the event happened.

When used to record security and privacy relevant events, the AuditEvent can then be used by properly authorized applications to support audit reporting, alerting, filtering, and forwarding. This model has been developed and used in healthcare for a decade as [IHE-ATNA profile](http://wiki.ihe.net/index.php?title=Audit_Trail_and_Node_Authentication). ATNA log events can be automatically converted to AuditEvent resources, and from there, client applications are able to search the audit events, or subscribe to them.

With regard to HTTP logs, implementers need to consider the implications of distributing access to the logs. HTTP logs, including those that only contain the URL itself, should be regarded as being as sensitive as the resources themselves. Even if direct PHI is kept out of the logs by careful avoidance of search parameters (e.g. by using POST), the logs will still contain a rich set of information about the clinical records.

**Digital Signatures**

This specification recommends the use of [W3C Digital Signatures](http://www.w3.org/TR/xmldsig-core/) for signatures. Resources can be signed using the [Provenance](file:///C:\temp\provenance.html) resource to carry a [detached digital signature](http://www.w3.org/TR/xmldsig-core/#def-SignatureDetached). The [Signature datatype](file:///C:\temp\datatypes.html#signature) is available to support various signature types including non-repudiation purposes. Further details on creation and validation of [Signatures are defined.](file:///C:\temp\datatypes.html#signature)

In addition, [documents may be signed](file:///C:\temp\documents.html#signatures) using an [enveloped](http://www.w3.org/TR/xmldsig-core/#def-SignatureEnveloped) signature. A specification for enveloped signature is profiled in the [IHE DSG profile](http://wiki.ihe.net/index.php?title=Document_Digital_Signature).

Neither of these definitions prohibits the use of other ways of using digital signatures.

**STU Note:** the use of signatures with RESTful interfaces is a poorly understood area, and we would welcome reports of implementation experience.

Feedback [here](http://wiki.hl7.org/index.php?title=FHIR_Specification_Feedback_(DSTU_2)).

**Attachments**

Several FHIR resources include attachments. Attachments can either be references to content found elsewhere or included inline encoded in base64. Attachments represent security risks in a way that FHIR resources do not, since some attachments contain executable code. Implementers should always use caution when handling resources.

**Security Labels**

See [Security Labels](file:///C:\temp\security-labels.html).

**Narrative**

FHIR resources include an XHTML narrative, so that applications can display the contents of the resource to users without having to fully and correctly process the data in the resource. However, displaying HTML is associated with several known security issues that have been observed in production systems in other contexts (e.g. [with CDA](http://smartplatforms.org/2014/04/security-vulnerabilities-in-ccda-display/)). For this reason, the [FHIR narrative is not allowed to contain active content](file:///C:\temp\narrative.html#security). However, care is still needed when displaying the narrative:

* Validate the narrative (the standard FHIR schemas do not allow active content, and the reference implementations won't handle it). Note, though, that external references could still be included in CSS, and these are outside the scope of schemas and reference implementations.
* Ensure that any external references to images or anchors (e.g. outside the resource) do not cause the display software to [leak sensitive information in headers](http://smartplatforms.org/2014/04/security-vulnerabilities-in-ccda-display/)
* Do not allow external links to run in a privileged context such as the EHR unless you are sure they can be trusted
* Care should be taken to differentiate HTTP RESTful (API) from browser based server content. Specifically, one should separate user session cookies, as an attacker could create content that serves up with content-type "text/html" and has content like "<script>send\_to\_attacker(document.cookie);</script>".

Also note that the inclusion of an external reference to an image can allow the server that hosts the image to track when the resource is displayed. This may be a feature or a problem depending on the context.

In addition to narrative, [Documents](file:///C:\temp\documents.html) may also contain stylesheets. Unlike with CDA, the stylesheets are simple CSS stylesheets, not executable XSLT, so the same security risks do not apply. However CSS stylesheets may still reference external content (e.g. background images), and applications displaying documents should ensure that CSS links are not automatically followed without checking their safety first, and that session/identifying information does not leak with any use of external links.

**servicelist.html**

**Services Defined by the FHIR specification**

This page lists all the services defined by FHIR. Services are business level aggregations of [Resources](file:///C:\temp\resourcelist.html) and [Operations](file:///C:\temp\operationslist.html) to provide a defined package of services that correspond to an identified business need.

Note that in addition to the services defined in this specification, many [implementation guides](http://fhir.org/implementation_guides) define business level services.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Links** |
| [Terminology Service](file:///C:\temp\terminology-service.html) | A service that lets healthcare applications make use of codes, code systems, and value sets without having to become experts in the fine details of the code system, value set and concept map resources, and the underlying code systems and terminological principles. The service provides the following functionality:   * Defining and managing code systems, value sets, and concept maps * Code lookup & Validation * Value Set Expansion (including for UI data entry support) * Value Set Validation * Translations between code systems * Subsumption testing and other logical analyses * Maintaining a closure table | [Capability statement](file:///C:\temp\capabilitystatement-terminology-server.html) [module](file:///C:\temp\terminology-module.html) |
| Conformance Service | A service that provides the underlying services needed to test whether resources conform to the rules defined by the FHIR specification and applicable implementation guides, and to help author implementation guides. The service provides the following functionality:   * Defining and managing structure definitions, data elements, Capability statements, search parameters, and operation & compartment definitions * Resource Validation * Comparing and subsetting Capability statements | To be developed |
| Knowledge Repository | A service that provides basic retrieval and maintenance functionality for clinical knowledge artifacts. The service provides the following functionality:   * Defining and managing plan and activity definitions (e.g. order sets, protocols, decision support rules, documentation templates, etc.), libraries, and measures * Search and retrieval of knowledge artifacts * Data requirements analysis of knowledge artifacts | [Capability statement](file:///C:\temp\capabilitystatement-knowledge-repository.html) [module](file:///C:\temp\clinicalreasoning-module.html) |
| Measure Processor | A service that provides measure evaluation functionality. The service provides the following functionality:   * Searching and retrieval of measure definitions * Evaluation of measures for patients and populations * Data requirements analysis of measure definitions | [Capability statement](file:///C:\temp\capabilitystatement-measure-processor.html) [module](file:///C:\temp\clinicalreasoning-module.html) |

Possible candidate Business Services for future versions:

* Conversion Service (using mapping language, and also iso-semantic transforms in a resource)
* Questionnaire related finctionality (though see SDC)
* Patient Registration & Reconciliation service (per IXS)
* Medication Management service (home specific variant?)
* Clinical Data Repository
* Clinical Task Manager
* Comminications Manager
* ...?

Note that there is already past & current work on some of these in HL7 (though not necessarily FHIR specific).

**services.html**

**Using Resources with Services and Service-oriented Architecture**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Though FHIR resources are primarily designed for the [RESTful HTTP-based implementation](file:///C:\temp\http.html), it is not necessary to use a RESTful interface when exchanging the resources; one way that they can be exchanged is in the context of services. Service Oriented Architecture (SOA) is an architecture pattern using services to encapsulate and provide discrete pieces of application functionality to each other. Services communicate by invoking public interfaces and exchanging information (as parameters and outputs) in accordance with a well-defined service contract.

Generally speaking, services are cohesive sets of functions that maintain responsibility for both data and "state" for the scope of their responsibility. Services have a unity of function, such as Terminology Management, Identity Management, or so on, working with other services in collaboration as part of an orchestrated workflow. Data within a service typically follows a "black box" model and is not classically exposed, instead using pre-defined interfaces to make specific behavioral requests for the service to fill.

To relate this to FHIR, resources in a service-oriented implementation can serve two discrete roles: either as a payload parameter, specifying information flowing into or out of a service; or in a behavioral sense, using FHIR APIs as the invocation mechanism for engaging a service. Service APIs are often realized using either RESTful or SOAP-based interfaces.

Of note, there are potential benefits in considering usage of FHIR and SOA together. FHIR allows for ease of implementation and ready access to data payload in an open way. SOA has maturity around transactional integrity across distributed systems instances, providing a framework for loose-coupling and addressing pre-conditions, exception handling, and other implementation considerations of complex distributed systems.

This section illustrates different contexts to allow implementers to make informed decisions at implementation time, considering potential impacts and solution patterns to benefit specific situations. In doing so, this section will clarify what is meant to apply SOA principles and specifications within a FHIR environment/implementation, and the converse. It will identify the situational factors to consider, and the elements of the SOA discipline that can be applied for more effective FHIR implementation, providing a roadmap on how to optimize your implementation toolkit to make you and your team most effective.

This section is being included as FHIR offers a highly flexible framework for interoperability that can be deployed that can be used in a variety of different ways, and in ways that have the potential to hinder interoperability if not done thoughtfully or consistently. Use of SOA patterns and approaches can mitigate this risk, limiting the potential liability resulting from the extreme flexibility of FHIR and fostering an overall framework that leads to deployment and consistency. SOA provides guidance for how components interact, how to partition responsibilities, and how to manage workflows among different parts of systems, all of which have potential utility in FHIR implementation settings.

**Service References and Patterns**

Generally, the definition of particular services is a domain or context specific task, and it is anticipated that this would be done as separate specifications that make use of the underlying facilities defined in this specification. HL7 has already defined a portfolio of services that take advantage of REST- and SOAP-based interfaces, but the parameters passed to and from those services are not at present resource-based. These are potential extension points that may lend well for FHIR implementation. Services defined like this are able to build on the common underlying platform features defined in this specification such as REST or messaging, and add specific interactions where appropriate. Alternatively, the service interfaces can build an entirely separate implementation.

Note that for each of the items below, there are paired sets of standards, an HL7 Standard defining the functional specification, and an OMG Technical Standard containing SOAP bindings (for all of the items below) and REST bindings (for about half of the items below).

The portfolio of healthcare SOA service specifications includes:

* IXS (Identification and Cross-Reference Services) that identifies the characteristic of a generic identification management services ([www.omg.org/spec/IXS](http://www.omg.org/spec/IXS))
* RLUS (Retrieve, Locate, Update Service) that supports the generic management resource in a SOA environment. ([www.omg.org/spec/RLUS](http://www.omg.org/spec/RLUS))
* ServD (Service directory) that supports the registration and discovery of health and human-related services ([www.omg.org/spec/ServD](http://www.omg.org/spec/ServD))
* CDSS (Clinical Decision Support Service) that defines a standard access interface to a clinical decision support system ([www.omg.org/spec/CDSS](http://www.omg.org/spec/CDSS))
* CTS2 (Common Terminology Service V 2) that defines the capabilities, responsibilities, inputs, outputs, and expected behavior to support the management, maintenance, and interaction with ontologies and medical vocabulary systems ([www.omg.org/spec/CTS2](http://www.omg.org/spec/CTS2))
* CCS (Care Coordination Service) that provides the capabilities to support the coordination of patient care across the care continuum spanning multiple organization [emerging specification]
* OS (Order Service) that provides functional capabilities for ordering pharmacy, laboratory, radiology, consult and nutritional services individually or part of an order set [emerging specification].
* EPSS (Event Publication & Subscription Service) that provides a Service Functional Model (SFM) for services, components and systems to subscribe to clinical events of interest and receive notice when new data are available [emerging specification].
* UCS (Unified Communications Service) facilitates the bi-directional communication between participants (both SOA services and human participants) that is so essential for the efficient delivery and coordination of patient care [emerging specification].

Note: FHIR-enablement of the services above is already occurring on a limited basis. The [Clinical Quality Improvement Framework](http://hl7.org/fhir/2015Sep/cqif/cqif.html) (CQIF) portion of the FHIR specification is an example of a technical interface and resources that have been influenced by prior SOA work. Similarly, this specification defines a [Terminology Service](file:///C:\temp\terminology-service.html) which is tightly integrated with the RESTful API (note: this terminology service has different goals than the CTS2 service, so they are not functionally equivalent). FHIR itself is a framework for the instantiation of an RLUS service. Definition of other additional services based on the service interface definitions provided by the HL7 SOA work group will be considered if there is sufficient interest in this.

**Implementation Approaches**

Given that there are a set of available services that can be applied directly or used as a pattern in conjunction with FHIR, it merits consideration that there are alternative ways to bring together FHIR and SOA, each of which has benefits and drawbacks. Three specific styles of implementation have been identified:

* FHIR + REST ("RESTful FHIR"), the predominant approach for implementing FHIR today
* FHIR + WS\* represents FHIR implementations using the web services stack as the communication protocol instead of REST. This would include use of FHIR resources as payload parameters in SOAP calls, for example
* FHIR + SOA Pattern illustrates the impact of applying interaction patterns, exception handing and role definition, guided by SOA practices, atop an implementation technology (SOAP, REST, or others)

Determining which of the above implementation styles is best suited to a particular situation is context-dependent. The following table presents these alternatives, deliberately using qualitative measures to help navigate to the best-fit based upon influencing factors. "Harvey Balls" in the table below indicate relative strengths and weaknesses of the implementation alternatives.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Solution Quality** | **Short Description** | **#1:FHIR + REST** (RESTful FHIR) | **#2 FHIR + WS\*** | **#3 FHIR + SOA Pattern** | **Comments** |
| Support of transactional integrity | Transactions may involve multiple steps composed together into a single unit | ◒ | ◒ | ● | Transactional integrity is not assured in #1 or #2 but can be hand-coded. #3 assures that this is addressed |
| Support of stateless transactions | Transactions are independent and do not require supporting context | ● | ● | ● | Each of the implementation approaches are capable of supporting stateless processing |
| Support of "loose coupling" | Minimum of dependencies between the calling and responding system; reduces burden on client to manage context of interaction with service | ◒ | ◒ | ● | The notion of "loose coupling" is fundamental to SOA and inherent in that architecture. Loose coupling can be supported by FHIR, but it does not innately do so |
| Use of FHIR Resources as Data Payload | Representation of FHIR resources for use as query or return parameters | ● | ● | ● | Data payload includes input data as well as data returned |
| Support of Dynamic Service Discovery | Provides for service registration, search, discovery, and late binding | ◒ | ◒ | ● | This is part of SOA specification and customary in implementations. For the others it is up to the approach |
| Provides resource-oriented operations | Supports ability to create, read, update, delete resources; | ● | ◒ | ◒ | Direct access to resources/data revisions in a strength of REST and FHIR. Direct access to fine-grained transactional operations are generally not supported within SOA |
| Suitability for atomic transactions | Fitness to support fine-grained transactions, such as data access or targeted update | ● | ◒ | ◒ | REST is ideally suited for point access or updates of specific data elements/resources - functions discouraged within SOA and WS\* |
| Suitability for composite transactions | Fitness to support complex transactions: context-sensitivity, multi-step workflows, etc. | ◒ | ● | ● | SOA provides for complex event processing, multi-step sequencing, orchestration |

**Architectural Considerations**

FHIR resources are designed to be used in a wide variety of contexts. In particular, FHIR resources are required to be suitable for use in a REST environment. This means that there are number of design requirements and choices that impact how suitable resources are for use with services.

If service use was the only consideration, different decisions would be made, and resources would be more suitable for use with services. However this would curtail their usefulness and reusability in other contexts.

**API/Interface Design**. Services' API design is explicit and exposes service capabilities which are a part of a service contract to consuming applications. This contrasts with what is the default interface into a FHIR server of a REST-based interface exposing CRUD operations and stateless access. Good SOA design practice indicates loosely-coupled access (e.g., external clients do not have visibility into the inner workings of a service) improves implementation flexibility over time, permitting design and performance improvements with the maturation of the implementation. As such, interfaces are typically exposed and aligned with business functionality (such as "Register Patient" or "Validate Identity").

Bridging what is not necessarily a natural-fit between SOA operations and RESTful operations are key to interface design. Implementers may elect to apply finer grained FHIR interactions to be encapsulated in coarse-grained functions, delegating those responsibilities to either implementations within a FHIR instance or to external code not visible to the service consumer.

Principal within the design objective is to minimize or eliminate responsibilities on behalf of the service consumer to understand or apply knowledge of the service implementation. This frees up the service provider to be able to evolve and improve the inner workings of the service without a "ripple effect" adversely impacting service consumers. In doing so, technical dependencies are minimized. Several good sources of SOA Design Patterns are available from HL7 and other public sources.

**Data Storage and Coherence**. The approach to and complexities associated with data storage and overall data coherence will significantly depend upon whether an implementation is subject to enterprise policies and whether data consistency is the responsibility of the service provider or consumer. In Enterprise settings these issues are typically governed by organizational policy. FHIR servers and their data persistence would need to fit within that policy, with data access being in compliance with that policy and services interacting with "policy enforcement points" to assure that appropriate permissions are in place.

Where data consistency is the responsibility of the service (or server) use of SOA-friendly interface protocols, such as SOAP and potentially REST, create the access channel for integration. This may necessitate defining minimum sets of data that are required to comply with data coherence expectations (in other words, a service contract that specifies the collection of resources that must be updated in tandem, and the scope of what constitutes a Ã¢â‚¬Ëœsingle transactionÃ¢â‚¬â„¢ - more on this in the Transactional Integrity section).

In alternative implementations data storage is managed within the FHIR server, consistent with non-SOA FHIR implementations. Data consistency is the responsibility of the Server but responsibility may carry to consuming applications (if they are provided with sufficient update rights so as to make that a concern). Data validation also has a shared responsibility, first with the FHIR Server, and additionally with the authorized updating applications. Data visibility is a product of the REST interface, typically allowing CRUD operations to the data element level.

**Transactional Integrity**. When encapsulating functionality within a SOA-type service, data and transactional integrity are essential in making sure that data received and stored is accurate and consistent with what was received. An issue particularly with update functions, the ability to update multiple different resources and maintain harmony among them as part of a broader transaction is how this is typically realized.

Should one part of the update Ã¢â‚¬ËœfailÃ¢â‚¬â„¢, one would expect all of the other updates to be "rolled back" to a prior, stable state. Rollback is not expressly specified as part of the FHIR. Data update and integrity becomes the responsibility of each service. The FHIR specification does not inherently provide for concurrency management (e.g., managing of "deadlock" conditions where two resources each await update permission from the other). As a result, within the FHIR server itself it is assumed that fault tolerance and server availability is being managed (beyond the scope of the specification).

Applied use of SOA patterns have benefits in terms of distributed transactional integrity, as SOA has substantially removed burdens from calling (client) applications, instead levying these responsibilities upon the service provider. This has relevance because it is customary in larger scale implementations (such as large enterprises) that service consumers are varied and not necessarily know in advance to the service. As a result, these implementation patterns make for more robust infrastructure, mitigating potential implementation risk and ultimately improving data quality. We note that these elements can be manually accommodated in FHIR implementations, absent use of SOA patterns, though that can lead to inconsistent interpretation, delegated responsibility to client application(s), and the potential for incompatible implementations.

**Modularity**. The modularity of resources - which resources are defined, and why - is driven by a wide set of considerations around how they are used. Resources are a platform on which a set of business/clinical services of various kinds are provided. Accordingly, the resources are defined for general use, and they can be expected to be less suitable for a particular service than custom defined structures. The pay back is wider re-use of the information that the service deals with.

Effective "service-orientation" establishes modularity at both a macro and micro-level. At the macro-level, services themselves are modular, componentized, and composable. Since each service has a singularity of purpose with a well-defined interface, they are naturally suited to working in tandem with other services and consistent with modular design. At a more fine-grained level, the interfaces into and out of services also allow for modularity, and the ability to leverage the structure of FHIR resources as a payload descriptor for carrying data structures and corresponding semantics across APIs is modularly based.

**Explicit State**. All resources represent the various states of the record and real world entities involved in a transaction explicitly. This is necessary for use in RESTful and document contexts, where there is no explicit transaction. Service interactions are typically associated with implicit semantics, such as a request to change the status of a particular resource to something else, for example. The fact that resources that carry this state explicitly as well as the transaction fixing state implicitly creates duplication between the two, and this will need to be managed.

**Error Handling**. SOA provides an implementation approach providing consistency in error handling, escalation, and error management. These tools can be leveraged to identify likely error conditions and exceptions based upon prior SOA work, helping to put into place infrastructure within FHIR to manage those exceptions. Moreover, this alignment creates future opportunity to more effectively integrate FHIR resources into an enterprise SOA fabric should that become a need. It is not anticipated that there is a need to change error management for FHIR, except to address payload-specific errors that may arise.

The FHIR specification does not specifically address error handling between and across servers and clients. When implementing within larger or more complex environments, particularly in situations where multiple FHIR servers may be involved (especially if they are provided by different vendors/implementations), error handing and management can quickly become complex and untenable.

**Resource References**. The most obvious impact is that resources [refer to each other using full URL based references](file:///C:\temp\references.html), and there are a number of rules around how these references are resolved. In the context of a service, this means that the references between modules carry this extra weight of choice and obligation, even when it might not be necessary.

FHIR [resources](file:///C:\temp\resourcelist.html) or [bundles](file:///C:\temp\bundle.html) may be used as the parameters or outputs of service interfaces.

**Other Considerations**

**Orchestration**. *Orchestration* is a term typically used within SOA to describe the steps, sequencing, and dynamic adjustment of workflow to meet a process need. Orchestrations may be entirely automated and fulfilled in a short timeline, or longer and multistep in fulfillment of a business process that may involve manual steps and human intervention. In the context of FHIR implementation, orchestration would refer to the sequencing of collections of FHIR (or service) calls that are used in tandem to fulfil a specific need.

Orchestration is neither natively supported nor unsupported in FHIR, save the availability of the FHIR Batch mode which allows for some degree of compounding of operations. In a SOA environment, orchestration is typically realized by documenting in some formalism, such as Business Process Modeling Notation (BPMN), a sequence of steps and flows, and inherent logic or decision-points affecting that flow. An engine is then capable of executing the process flow, receiving inputs during execution to adjust or adapt those processes based upon situational need as part of delivery fulfillment.

There are many examples where this approach is advantageous. For example, in a Care Management situation, test results, current problems, chief complaint, and potentially even resource availability might affect subsequent steps in fulfilling a care plan. Based upon changes to any of these factors, the sequence of calls and ultimately the systems or FHIR servers involved would vary. The role of automated clinical decision support is another example that naturally ties to workflow orchestration, adjusting care pathways based upon patient evaluation and affecting process flows within a health system.

At present, to support this complexity within a FHIR setting without the use of SOA tools, these flows would need to be manually coded, either by having a directed sequence of calls, or by creating an independent capability effectively acting as an orchestrator. This can be supported within FHIR exclusively, but all of the steps and corresponding state management would need to be done by hand. FHIR Batch provides some of this capability, allowing for aggregation of multiple steps, but does not necessarily support the event processing needed in complex.

**Security**. Security is inherently a dimension of any enterprise SOA architecture, meaning that the responsibility within a service implementation is to provide the "hooks" to interact with that architecture. In other words, the service does not need to create or enforce security, it needs to interact with those enterprise components that have that responsibility. This includes topics such as identity management, access control, or other dimensions of a secure solution. SOA guidance can foster an effective design involving FHIR in support of authentication, policy enforcement, role-based access controls, and a host of other provisions and protections. It is important to note that within SOA, security is an established, mainstream, and mature offering. Reuse of these concepts can help prevent inadequate, incomplete, or ineffective security measures within FHIR, and eliminate the need to re-invent solutions exclusively for FHIR.

Enterprises typically have security architecture into which a SOA environment will have already integrated. Service implementations would rely upon this existing infrastructure as part of authentication, access control, etc. Policy enforcement is a function of the architecture and not a specific service itself. Services would rely upon policy enforcement points to govern access to information.

While technical dimensions of security would be addressed by the security architecture directly, FHIR implementers should anticipate that data-specific access control policies would need to be captured and formalized so that they are enforceable within the architecture. For example, if a new service is handling protected health information, the policy enforcement point within the architecture would need to know that the service has that nature of data.

**Resource Identity**. All resources have a single identity (the full URL), and a logical id which may be maintained as the resource moves from server to server (see [Managing Resource Identity](file:///C:\temp\managing.html) for further discussion). Since this identity is used by any reference to the resource, it must be maintained when the resource is exchanged so that references from other resources to the one being exchanged can still be resolved. Any use of resources in a service environment needs to address how these references can be resolved. This can be achieved by delegating the reference resolution to a RESTful framework, by ensuring that all the relevant resources are contained in the service call, or by making some service based arrangement by which additional resources can be retrieved. Services that exchange resources SHOULD maintain resource identity. Further, as explicit version tracking is not a guaranteed capability of a service call, services that exchange resources should include version information associated with resources being exchanged.

**Capability Statement**. When using [RESTful exchange](file:///C:\temp\http.html), [messaging](file:///C:\temp\messaging.html), and [document](file:///C:\temp\documents.html) based exchange, the Capability statement allows authoring and reading applications to describe how they use a resource. The [Capability Statement](file:///C:\temp\capabilitystatement.html) supports trading partner negotiation from specification time through to run-time discovery. The Capability statement doesn't provide any equivalent way to make declarations about services, though this might be added in the future if common requirements emerge. Services are expected to make appropriate arrangements around discovery and compatibility, though it is expected that these will vary considerably.

Services that exchange resources MAY choose to provide support for describing and changing descriptions of service implementations.

**Relating SOA and FHIR: Rationale**

At its core, FHIR is based upon the elemental capabilities of Create, Read, Update, Delete (CRUD), allowing access to resources where interaction is primarily based upon these operations. Of note, multi-step or complex processing levies requirements upon the calling, client application. Sequencing of events, particularly when those events involve orchestration of complex or dynamic processes, can benefit from the application of SOA techniques and patterns. This is common where multi-step processing is involved, and particularly important when adjustments to a process flow can be made based upon situational context.

As these areas are both commonplace within health settings and not inherent strengths of FHIR, the marriage of FHIR and SOA techniques provides a viable and beneficial path forward to improving implementation and mitigating risks by leveraging proven and established industry best-practices.

SOA design patterns are frequently used to govern interactions for dynamic processes, non-sequential workflows, or dynamic workflows involving human intervention and/or consideration of external factors as part of processing. The ability to model interaction patterns using languages such as SOAml, and to define roles and role interactions, are tools that may be useful to FHIR development in these circumstances, and would be indicators signaling that SOA has the potential to add value to a FHIR implementation.

SOA design principles can provide guidance to a FHIR implementer resulting in reduction of co-dependencies between components, promoting "loose coupling" and minimizing potential impacts resulting from changes to inner workings of one component adversely affecting others. (For those familiar with the term, this encourages "black box" implementation).

SOA provides guidance around data persistence, durability, and expectations in support of the data lifecycle.

**Summary**

FHIR has proven itself as beneficial in that it provides an easy-to-implement and coherent approach to accessing healthcare resources using a modern protocol stack. SOA has proven itself over time in multiple vertical market segments as a way to divide responsibilities and authoritatively manage information across distributed systems. Marrying the discipline of SOA with the implementation ease of FHIR is a winning combination when the situation warrants. Implementers should consider evaluation of the contextual landscape to determine where FHIR and SOA have the opportunity to benefit from complementary implementation leverage points.

Recognizing that there are situations where the coming-together of FHIR implementation and SOA techniques are advantageous, there are several implementation approaches available. Note that the selection of which approach is advantageous for any given situation will depend upon a variety of factors: existing legacy implementation and available APIs (particularly for interfacing applications), the strategic direction/IT roadmap of the organization, and so on. Noting that there are different styles of realizing FHIR and SOA together, a recap of the principal alternatives would include:

* "Full Support in FHIR" Approach. This alternative involves simplification of the exposed behaviors of the service/server so as to be supported within the REST protocol, thus allowing full support within FHIR. Behind a FHIR technology stack, SOA design principles such as loose coupling and "black box" API design would be coded to create service autonomy.
* "Classic SOA with FHIR Payload" Approach. This alternative is attractive particularly in enterprises that have or are making investments in SOA infrastructure. This alternative relies upon traditional SOA service definition, leveraging use of FHIR Resources as the parameters into and out of the service.

It is important to note that electing to apply SOA patterns to FHIR implementation does not fundamentally change the nature of what is being implemented. Applied correctly, it results in minor adaptation to coding and interface design resulting in enhanced consistency across implementations as well as enhanced interoperability and robustness.

These considerations are summarized in this table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Capability** | **#1:FHIR + REST** (RESTful FHIR) | **#2 FHIR + WS\*** | **#3 FHIR + SOA Pattern** | **Comments** |
| Supports dynamic adjustment in workflows | ○ | ◒ | ● | In a native FHIR environment, this must be done by hand. The WS\* stack provides for some interaction patterns, but limited. Use of formal notation (BPMN) and service orchestration is most robust option |
| Supports ability to batch multiple operations | ◒ | ◒ | ● | FHIR Batch allows for concatenation of multiple process steps. SOA Orchestration allows for fine grained control, grouping and sequencing of multiple operations |
| Provides for management of data coherence (e.g. Deadlocks, transactions) | ○ | ○ | ● | SOA principles define data governance, assigning responsibility for all data management within a service contract |
| Provide support for orchestration languages ([BPMN](http://www.bpmn.org/), [SOAml](http://www.omg.org/spec/SoaML/)) | ○ | ◒ | ● | SOA applys a systematic framework for orchestration using industry accepted formalisms, avoiding costs and complexities associated with an ad-hoc approach |

**SOA in a FHIR Environment** considers the benefits of applying SOA patterns and best-practices to a FHIR implementation community. In these cases, developers are faced with implementation decisions that have the potential to benefit from existing case studies, design patterns, or guidance that may either help provide consistency among FHIR implementations, or which may address gaps - either known or unidentified - resulting from implicit assumptions around a FHIR implementation.

**FHIR in a SOA Environment** considers the implications of using FHIR in a large, established enterprise that has or is making investments in SOA infrastructure. In these cases, shared services, enterprise policies, and existing infrastructure is common, and FHIR implementations would need to fit within the fabric of that environment.

Finally, it merits mention that not every FHIR implementation benefits from SOA, and vice-versa. The goal of this section is to help implementers navigate based upon their specific considerations to help determine if and to what extent these approaches provide benefits.

**signatures.html**

**Digital Signatures**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

todo - digital signatures in FHIR.

**snomedct-usage.html**

**SNOMED CT in FHIR**

This page summarizes the use of SNOMED CT in the base FHIR Specification. It is used by HL7 and IHTSDO to help with SNOMED CT licensing management.

In principle, this base FHIR Specification only uses concepts defined in the International distribution of SNOMED CT, while FHIR implementation guides that use SNOMED CT are generally published by HL7 Affiliates under agreements with National distribution centers. However there are some US specific codes on this page; HL7 is working with IHTSDO towards remove these from the specification (hopefully they will be elevated to the international SNOMED CT distribution).

**Value Sets**

The following value sets reference SNOMED CT. The table lists the value sets, and then the places they are used from, and includes the binding strengths

**Concept References**

The value sets that reference SNOMED CT explicitly reference the following SNOMED CT concepts. The table lists the concepts, and then the value sets and places they are used from, and includes the binding strengths. The table does not include codes published in the expansions.

**snomedct.html**

**Using SNOMED CT with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Summary**

|  |  |
| --- | --- |
| Source | SNOMED CT is owned, maintained and distributed by [IHTSDO](http://www.ihtsdo.org/). IHTSDO is an international organization with countries as members. IHTSDO publishes the international release of SNOMED CT (which is used in FHIR international standards and resources). Each of the IHTSDO member countries may also distribute their own SNOMED CT edition, which contains the international release plus local extension content and derivatives. |
| System | The URI <http://snomed.info/sct> identifies the SNOMED CT code system |
| Version | Where a version is used, it should be a full URI that represents the specific distribution (International Release, or National Release and version), following the [SNOMED CT URI Specification](http://snomed.org/uristandard.pdf) (see note below) |
| Code | The following SNOMED CT artifacts are valid in the *code* element for the http://snomed.info/sct namespace: [Concept IDs](http://www.snomed.org/gl?t=glsct_st_ConceptId), [Expressions](http://www.snomed.org/gl?t=glsct_se_Expression) ([grammar](http://snomed.org/compgrammar.pdf)) and [SNOMED Legacy codes](https://confluence.ihtsdotools.org/display/DOCCMG/4.6.2+Migration+from+SNOMED+International).  SNOMED CT Terms (Description Ids) are not valid as codes in FHIR, nor are other alternate identifiers associated with SNOMED CT Concepts using the RF2 identifier infrastructure. Expressions SHOULD NOT contain terms, only concept IDs (note that when SNOMED CT terms must be exchanged, use the [Description Id Extension)](file:///C:\temp\extension-coding-sctdescid.html) |
| Display | The correct display for a SNOMED CT concept is one of the synonyms for the concept. The best display is the preferred term in the relevant language reference set (if applicable). Preferred terms are case sensitive. The Fully Specified Name is not an appropriate choice. The source of preferred name comes from a Language Reference Set. SNOMED CT does not define displays for expressions; if no display has been associated with the expression through a value set or other mechanism, the full expression syntax with preferred terms embedded may be used |
| Inactive | Inactive codes are identified using the 'inactive' property (see below) |
| Subsumption | SNOMED CT Subsumption testing is based in the |is a| relationship defined by SNOMED CT |
| Filter Properties | Several properties are defined as described below |

This specification publishes a [canonical SNOMED CT](file:///C:\temp\codesystem-snomedct.html) [code system](file:///C:\temp\codesystem.html) resource. See also the [SNOMED CT Usage Summary](file:///C:\temp\snomedct-usage.html).

Note: The [IHTSDO glossary](http://snomed.org/gl) explains some of these SNOMED CT specific terms.

**Version Issues**

There is no single distribution that contains all defined SNOMED CT codes in all contexts of use. Instead the international release contains all concepts shared and agreed to be internationally relevant and each national release centre distributes this international release plus additional national content (to extend that international set). In addition, other release authorities may be designated. The [SNOMED CT URI Specification](http://snomed.org/uristandard) describes how to unambiguously reference a particular version of a distribution:

http://snomed.info/sct/[sctid]/version/[YYYYMMDD]

where [sctid] is the concept id of the SNOMED CT distribution (e.g. 32506021000036107 for Australia), and the tail is the date of release (by custom, this is usually the last day of the month). Note that many implementations are in the habit of simple using the date of release in the form YYYYMMDD (e.g. "20140531"), and assuming that the distribution is known. However this is not always safe, so implementations that populate the *version* element SHOULD use the full URI form.

**Copyright/License Issues**

The use of SNOMED CT codes, display names and value sets in this specification is subject to an MOU signed between HL7 International and IHTSDO. This does not convey the right to use SNOMED CT to any users of this specification; implementers must acquire a license to use SNOMED CT in their own right. See [HL7 Policies and Guidance documents Ã¢â‚¬â€œ Licensing](https://www.hl7.org/special/committees/termauth/docs.cfm).

**RDF**

The IHTSDO URI specifications uses the namespace http://snomed.info/sct for the code system, and the URI http://snomed.info/id for the individual concepts in the code system. This means that when a SNOMED CT concept is converted from the system::code pair, where the system is http://snomed.info/sct, to the [RDF ontological form](file:///C:\temp\rdf.html), the representation is http://snomed.info/id/[concept-id]. Expressions are represented the same way, except that for this use, expressions SHALL not contain whitespace, terms, or comments.

**SNOMED CT Properties**

In addition to the [standard properties](file:///C:\temp\terminology-service.html#standard-props), the following properties are defined for SNOMED CT:

|  |  |  |
| --- | --- | --- |
| inactive | boolean | Whether the code is active or not (defaults to false). This is derived from the active column in the Concept file of the RF2 Distribution (by inverting the value) |
| sufficientlyDefined | boolean | True if the description logic definition of the concept includes sufficient conditions (i.e., if the concept is not primitive - found in the value of definitionStatusId in the concept file). |
| moduleId | code | The SNOMED CT concept id of the module that the concept belongs to. |
| normalForm | string | Generated Normal form expression for the provided code or expression, with terms |
| normalFormTerse | string | Generated Normal form expression for the provided code or expression, conceptIds only |

In addition, any SNOMED CT relationships where the relationship type is subsumed by Attribute (246061005) automatically become properties. For example, laterality:

|  |  |  |
| --- | --- | --- |
| Laterality | code | In this case, the URI (See the [code system definition](file:///C:\temp\codesystem-snomedct.html)) is http://snomed.info/id/272741003, which can be used to unambiguously map to the underlying concept |

Note that when a [$lookup](file:///C:\temp\codesystem-operations.html#lookup) operation is performed on a SNOMED CT concept, servers SHALL return the full URI for the edition and version being used (see above) in the version property. Other properties are at the discretion of the server and the client.

**SNOMED CT Filters**

This section documents the property filters that can be used with the SNOMED CT code system in value set composition statements.

For implementer convenience, some of the property filters are documented in terms of the [SNOMED CT Query Language](http://www.ihtsdo.org/fileadmin/user_upload/Docs_01/News/SNOMED_CT_Query_Specification_-__v0.08_-_20121213.doc), but this does not imply that its use is required. *To Do: what's the correct link for this>*

**By Subsumption**

|  |  |
| --- | --- |
| Description | Select a set of concepts based on subsumption testing |
| Property Name | concept |
| Operations Allowed | is-a |
| Values Allowed | [concept id] |
| Comments | Includes all concept ids that have a transitive is-a relationship with the concept Id provided as the value (including the concept itself) |
| Example | [Administation Methods](file:///C:\temp\valueset-administration-method-codes.html) |
| [SCT ECL](http://snomed.org/expressionconstraints.pdf) | << [concept] (or the long form: descendantOrSelfOf [concept]) |

**By Reference Set**

|  |  |
| --- | --- |
| Description | Select a set of concepts based on their membership of a SNOMED CT reference set |
| Property Name | concept |
| Operations Allowed | in |
| Values Allowed | [concept id] |
| Comments | Includes all concept ids that are active members of the reference set identified by the concept Id provided as the value |
| [SCT ECL](http://snomed.org/expressionconstraints.pdf) | ^ [concept] (or the long form: memberOf [concept]) |

**By Snomed Query Expression**

|  |  |
| --- | --- |
| Description | Select a set of concepts based on a formal expression statement |
| Property Name | constraint |
| Operations Allowed | = |
| Values Allowed | [expression] |
| Comments | The result of the filter is the result of executing the given [SNOMED CT Expression (Expression Constraint Language)](http://snomed.org/expressionconstraints.pdf).  Example:  "compose": {  "include": [  {  "system": "http://snomed.info/sct",  "filter": [  {  "property": "constraint",  "op": "=",  "value": "<< 30506011000036107 |Australian product|: 700000101000036108 |hasTP| = 17311000168105 |Panadol|"  }  ]  }  ]  } |

**By whether post-coordination is allowed**

|  |  |
| --- | --- |
| Description | Specify whether post-coordination is allowed or not |
| Property Name | expressions |
| Operations Allowed | = |
| Values Allowed | true or false |
| Comments | Expressions, if allowed, are subject to the same rules as pre-coordinated concepts. Note: simple reference sets do not include expressions, but Query specification reference sets might). |
| Example | [Administation Methods](file:///C:\temp\valueset-administration-method-codes.html) |
| [SCT ECL](http://snomed.org/expressionconstraints.pdf) | n/a |

**Implicit Value Sets**

Implicit value sets are those whose specification can be predicted based on the grammar of the underlying code system, and the known structure of the URL that identifies them. SNOMED CT has two common sets of implicit value sets defined: By Subsumption, and By Reference Set. These implicit value sets do not use complex queries. This allows a single URL to serve as a value set definition that defines a value set, and can serve as the basis for the [$expansion](file:///C:\temp\valueset-operations.html#expand) operation.

If any value set resources exist with an identifier that conforms to the URL patterns specified below, the content of the resource must conform to the template provided. Profiles and other value set references are allowed to reference these value sets directly (by reference as a URI, rather than by a value set reference, which is a literal reference).

A SNOMED CT implicit value set URL has two parts:

* the base URL is either "http://snomed.info/sct", or the URI for the edition version, in the format specified by the IHTSDO the [SNOMED CT URI Specification](http://snomed.org/uristandard.pdf)
* a query portion that specifies the scope of the content

"http://snomed.info/sct" should be understood to mean an unspecified edition/version. This defines an incomplete value set whose actual membership will depend on the particular edition used when it is expanded. If no version or edition is specified, the terminology service SHALL use the latest version available for it's default edition (or the international edition, if no other edition is the default).

For the second part of the URL (the query part), the 4 possible values are:

* ?fhir\_vs - all Concept IDs in the edition/version. If the base URI is http://snomed.info/sct, this means all possible SNOMED CT concepts
* ?fhir\_vs=isa/[sctid] - all concept IDs that are subsumed by the specified Concept.
* ?fhir\_vs=refset - all concept ids that correspond to real references sets defined in the specified SNOMED CT edition
* ?fhir\_vs=refset/[sctid] - all concept IDs in the specified reference set

A value set with a URL that follows the pattern "[edition/version]?fhir\_vs=isa/[sctid]" follows this template:

<ValueSet xmlns="http://hl7.org/fhir">

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml">

[Some HTML that describes this value set as all concepts subsumed by conceptid]

</div>

</text>

<url value="[edition/version]?fhir\_vs=isa/[sctid]"/>

<version value="[edition/version]"/>

<name value="SNOMED CT Concept [conceptid] and descendants"/>

<description value="All SNOMED CT concepts for [concept id or preferred description]"/>

<copyright value="This value set includes content from SNOMED CT, which is copyright Ã‚Â© 2002+ International Health Terminology Standards Development Organisation (IHTSDO), and distributed by agreement between IHTSDO and HL7. Implementer use of SNOMED CT is not covered by this agreement"/>

<status value="active"/>

<compose>

<include>

<system value="http://snomed.info/sct"/>

<filter>

<property value="concept"/>

<op value="is-a"/>

<value value="[sctid]"/>

</filter>

</include>

</compose>

</ValueSet>

The value set with a url that follows the pattern "[edition/version]?fhir\_vs=refset" follows this template:

<ValueSet xmlns="http://hl7.org/fhir">

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml">

[Some HTML that describes this value set as all concepts with associated reference sets]

</div>

</text>

<url value="[edition/version]?fhir\_vs=refset"/>

<version value="[edition/version]"/>

<name value="SNOMED CT Reference Sets"/>

<description value="All SNOMED CT concepts associated with a reference set"/>

<copyright value="This value set includes content from SNOMED CT, which is copyright Ã‚Â© 2002+ International Health Terminology Standards Development Organisation (IHTSDO), and distributed by agreement between IHTSDO and HL7. Implementer use of SNOMED CT is not covered by this agreement"/>

<status value="active"/>

<compose>

<include>

<system value="http://snomed.info/sct"/>

<!-- repeat: one concept element with a code for each concept that has an associated reference set -->

<concept>

<code value="[sctid]"/>

</concept>

<!-- end repeat -->

</include>

</compose>

</ValueSet>

For each concept that is associated with a reference set, there will be one concept element with a contained code that contains the concept id.

A value set with a url that follows the pattern "[edition/version]?fhir\_vs=refset/[conceptid]" follows this template:

<ValueSet xmlns="http://hl7.org/fhir">

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml">

[Some HTML that describes this value set as all concepts in the reference set identified by conceptid]

</div>

</text>

<url value="[edition/version]?fhir\_vs=refset/[sctid]"/>

<version value="[edition/version]"/>

<name value="SNOMED CT Reference Set [conceptid]"/>

<description value="All SNOMED CT concepts in the reference set [concept id or preferred description]"/>

<copyright value="This value set includes content from SNOMED CT, which is copyright Ã‚Â© 2002+ International Health Terminology Standards Development Organisation (IHTSDO), and distributed by agreement between IHTSDO and HL7. Implementer use of SNOMED CT is not covered by this agreement"/>

<status value="active"/>

<compose>

<include>

<system value="http://snomed.info/sct"/>

<filter>

<property value="concept"/>

<op value="in"/>

<value value="[conceptid]"/>

</filter>

</include>

</compose>

</ValueSet>

**Implicit Concept Maps**

Implicit concept maps are those whose specification can be predicted based on the grammar and/or content of the underlying code system, and the known structure of the URL that identifies them. This allows a single URL to serve as a concept map definition that defines a mapping between two sets of concepts, and which can serve as the basis for the [$translate](file:///C:\temp\conceptmap-operations.html#translate) operation. SNOMED CT has two common sets of implicit concept maps defined:

* Association Reference Sets
* Simple Map Reference Sets

[Association Reference Sets](https://confluence.ihtsdotools.org/display/DOCRELFMT/4.2.5.+Association+Reference+Set) are part of the core SNOMED CT distribution. The following standard Association Reference sets are mapped to implicit Concept Maps:

|  |  |  |
| --- | --- | --- |
| **Name** | **Concept Id** | **Relatioship** |
| POSSIBLY EQUIVALENT TO | 900000000000523009 | inexact |
| REPLACED BY | 900000000000526001 | equivalent |
| SAME AS | 900000000000527005 | equal |
| ALTERNATIVE | 900000000000530003 | inexact |

If any concept map resources exist with an identifier that conforms to the URL pattern specified below, the content of the resource must conform to the template provided. Canonical references to concept maps are allowed to reference these concept maps directly by referring to their URI.

A SNOMED CT implicit concept map URL has two parts:

* the base URL is either "http://snomed.info/sct", or the URI for the edition version, in the format specified by the IHTSDO the [SNOMED CT URI Specification](http://snomed.org/uristandard.pdf)
* a query portion that specifies the scope of the content

"http://snomed.info/sct" should be understood to mean an unspecified edition/version. This defines an incomplete concept map whose actual membership will depend on the particular edition used when it is expanded. If no version or edition is specified, the terminology service SHALL use the latest version available for it's default edition (or the international edition, if no other edition is the default).

For the second part of the URL (the query part), there is only one possible value:

* ?fhir\_cm=[sctid] - where [sctid] is a value from the table above

A concept map with a URL that follows the pattern "[edition/version]?fhir\_cm=[sctid]" follows this template, where [name], [sctid] and [relationship] are taken from the table above:

<ConceptMap xmlns="http://hl7.org/fhir">

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml">

[Some HTML that describes this concept map]

</div>

</text>

<url value="[edition/version]?fhir\_cm=[sctid]"/>

<version value="[edition/version]"/>

<name value="SNOMED CT [name] Concept Map"/>

<description value="The concept map implicitly defined by the [name] Association Reference Set"/>

<copyright value="This value set includes content from SNOMED CT, which is copyright Ã‚Â© 2002+ International Health Terminology Standards Development Organisation (IHTSDO), and distributed by agreement between IHTSDO and HL7. Implementer use of SNOMED CT is not covered by this agreement"/>

<status value="active"/>

<sourceUri value="[edition/version]?fhir\_vs"/>

<targetUri value="[edition/version]?fhir\_vs"/>

<group> <!-- 0..\* Same source and target systems -->

<source value="http://snomed.info/sct"/>

<sourceVersion value="[edition/version]"/>

<target value="http://snomed.info/sct"/>

<targetVersion value="[edition/version]"/>

<!-- a mapping for each member of the reference set -->

<element>

<code value="[member]"/>

<target>

<code value="[reference set value]"/>

<equivalence value="[relationship]"/>

</target>

</element>

</group>

</ConceptMap>

[Simple Map Reference Sets](https://confluence.ihtsdotools.org/display/DOCRELFMT/4.2.8.+Simple+Map+Reference+Set) (reference sets which are descendants of 900000000000496009 "Simple map") also define an implicit concept map. However at this time, these cannot be converted to Concept Maps because there is no source for the target code system. Another reference set has been proposed to IHTSDO, and this section will be revisited sonce it is adopted.

**summary.html**

**Introducing HL7 FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

FHIRÃ‚Â® Ã¢â‚¬â€œ Fast Healthcare Interoperability Resources (hl7.org/fhir) Ã¢â‚¬â€œ is a next generation standards framework created by HL7. FHIR combines the best features of HL7's [v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185), [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) and [CDA](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7) product lines while leveraging the latest web standards and applying a tight focus on implementability.

FHIR solutions are built from a set of modular components called "Resources". These resources can easily be assembled into working systems that solve real world clinical and administrative problems at a fraction of the price of existing alternatives. FHIR is suitable for use in a wide variety of contexts Ã¢â‚¬â€œ mobile phone apps, cloud communications, EHR-based data sharing, server communication in large institutional healthcare providers, and much more.

**Why FHIR is better**

FHIR offers many improvements over existing standards:

* A strong focus on implementation Ã¢â‚¬â€œ fast and easy to implement (multiple developers have had simple interfaces working in a single day)
* Multiple implementation libraries, many examples available to kick-start development
* Specification is free for use with no restrictions
* Interoperability out-of-the-boxÃ¢â‚¬â€œ base resources can be used as is, but can also be adapted for local requirements
* Evolutionary development path from HL7 Version 2 and CDA Ã¢â‚¬â€œ standards can co-exist and leverage each other
* Strong foundation in Web standardsÃ¢â‚¬â€œ XML, JSON, HTTP, OAuth, etc.
* Support for RESTful architectures and also seamless exchange of information using messages or documents
* Concise and easily understood specifications
* A Human-readable serialization format for ease of use by developers
* Solid ontology-based analysis with a rigorous formal mapping for correctness

**Flexibility**

A central challenge for healthcare standards is how to handle variability caused by diverse healthcare processes. Over time, more fields and optionality are added to the specification, gradually adding cost and complexity to the resulting implementations. The alternative is relying on custom extensions, but these create many implementation problems too.

FHIR solves this challenge by defining a simple framework for extending and adapting the existing resources. All systems, no matter how they are developed, can easily read these extensions and extension definitions can be retrieved using the same framework as retrieving other resources.

In addition, each resource carries a human-readable text representation using html as a fallback display option for clinical safety. This is particularly important for complex clinical information where many systems take a simple textual/document based approach.

**Example Resource: Patient**

This simple example shows the important parts of a resource: a local extension, the human readable HTML presentation, and the standard defined data content.

FHIR has resources for administrative concepts such as patient, provider, organization and device as well as a wide variety of clinical concepts covering problems, medications, diagnostics, care plans, financial concerns and more.

**The FHIR development process**

FHIR is published as a Standard for Trial Use. During the Trial Use phase, HL7 actively monitors implementations in order to continue to improve the specification, and is able to be responsive to their needs. Due to the many advantages FHIR offers, trial use is already beginning right now.

<http://hl7.org/fhir>. Follow us on Twitter using [#FHIR](https://twitter.com/search?q=%23FHIR)

**template-abstract-book.html**

**Resource Definition:**

.

**Resource Content**

Alternate definitions: Master Definition ([XML](file:///C:\temp\.profile.xml.html), [JSON](file:///C:\temp\.profile.json.html)), [Schema](file:///C:\temp\.xsd)/[Schematron](file:///C:\temp\.sch) (for [XML](file:///C:\temp\xml.html)), [ShEx](file:///C:\temp\.shex.html) (for [Turtle](file:///C:\temp\rdf.html))

**template-abstract-definitions.html**

**Resource - Detailed Descriptions**

Detailed Descriptions for the elements in the .

**template-abstract-examples.html**

Usage note: every effort has been made to ensure that the examples are correct and useful, but they are not a normative part of the specification.

**template-abstract.html**

**Resource Content**

* [Structure](file:///C:\temp\main-pages.html#tabs-struc)
* [UML](file:///C:\temp\main-pages.html#tabs-uml)
* [XML](file:///C:\temp\main-pages.html#tabs-xml)
* [JSON](file:///C:\temp\main-pages.html#tabs-json)
* [Turtle](file:///C:\temp\main-pages.html#tabs-ttl)
* [All](file:///C:\temp\main-pages.html#tabs-all)

**Structure**

**UML Diagram** ([Legend](file:///C:\temp\formats.html#uml))

**XML Template**

**JSON Template**

**Turtle Template**

**Structure**

**UML Diagram** ([Legend](file:///C:\temp\formats.html#uml))

**XML Template**

**JSON Template**

**Turtle Template**

**template-book-defn.html**

**Resource Detailed Descriptions:**

The Detailed Descriptions for the resource.

**template-book-ex.html**

**Examples:**

Examples for the resource.

**template-book.html**

**Resource - Content**

|  |  |  |
| --- | --- | --- |
|  |  |  |

.

**Resource Content**

See also the [Examples](file:///C:\temp\Ex.html) and the [Definitions](file:///C:\temp\Defn.html).

Alternate definitions: Master Definition ([XML](file:///C:\temp\.profile.xml.html), [JSON](file:///C:\temp\.profile.json.html)), [Schema](file:///C:\temp\.xsd)/[Schematron](file:///C:\temp\.sch) (for [XML](file:///C:\temp\xml.html)), [ShEx](file:///C:\temp\.shex.html) (for [Turtle](file:///C:\temp\rdf.html))

**template-comparison-set.html**

**Comparison**

This is a comparison between the 2 implementation guides and , generated to help compare the interoperability of implementations based on the two implementation guides, or, alternatively, for implementers to understand what is required to conform to both implementation guides.

The comparison consists of a series of comparisons between the constraint definitions defined in the implementation guide. For each comparison, implementers can review:

* A series of messages from the comparison algorithm. Errors indicate that solutions cannot be interoperable across both implementation guides (or that there are structural flaws in the definition of at least one)
* The intersection of the 2 constraint statements. This is what resource authors (either client or server) would need to conform to produce content valid against both implementation guides
* The union of the 2 constraint statements. This is what resource authors (either client or server) would need to be able to handle to accept content valid against either implementation guides

**Comparisons**

**ValueSets**

**template-comparison.html**

**vs**

**Messages**

A series of messages from the comparison algorithm. Errors indicate that solutions cannot be interoperable across both implementation guides (or that there are structural flaws in the definition of at least one).

**Intersection**

The intersection of the 2 constraint statements. This is what resource authors (either client or server) would need to conform to produce content valid against both implementation guides.

**Union**

The union of the 2 constraint statements. This is what resource authors (either client or server) would need to be able to handle to accept content valid against either implementation guides.

**template-compartment-book.html**

**Compartment**

|  |  |
| --- | --- |
| Description |  |
| Identity |  |
| Membership |  |

Resource based membership rules:

See [information about compartments](file:///C:\temp\compartmentdefinition.html).

**template-compartment.html**

**Compartment**

|  |  |
| --- | --- |
| Formal URI |  |
| Description |  |
| Identity |  |
| Membership |  |
| Formal Definition | [CompartmentDefinition](file:///C:\temp\compartmentdefinition.html)resource: [XML](file:///C:\temp\compartmentdefinition-.xml.html) or [JSON](file:///C:\temp\compartmentdefinition-.json.html) |

Resource based membership rules:

See [information about compartments](file:///C:\temp\compartmentdefinition.html).

**template-conformance-pack.html**

**Content**

**template-cs-book.html**

**Value Set**

This is a value set defined by the FHIR project

Detailed Descriptions: [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html).

The OID for the value set is (OIDs are not used in FHIR, but may be used in v3, or OID based terminology systems).

See [the full registry of value sets](file:///C:\temp\terminologies-valuesets.html) defined as part of FHIR.

**template-cs-ig-book.html**

**Value Set**

Detailed Descriptions: [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html).

**template-cs-ig.html**

**Value Set**

Detailed Descriptions: [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html).

**template-cs.html**

**Code System**

This is a value set defined

**Summary**

|  |  |
| --- | --- |
| Defining URL: |  |
| Name: |  |
| Definition: |  |
| Committee: |  |
| OID: | (for OID based terminology systems) |
| Source Resource | [XML](file:///C:\temp\.xml.html) / [JSON](file:///C:\temp\.json.html) |

**Content**

See [the full registry of value sets](file:///C:\temp\terminologies-systems.html) defined as part of FHIR.

Explanation of the columns that may appear on this page:

|  |  |
| --- | --- |
| Level | A few code lists that FHIR defines are hierarchical - each code is assigned a level. In this scheme, some codes are under other codes, and imply that the code they are under also applies |
| Source | The source of the definition of the code (when the value set draws in codes defined elsewhere) |
| Code | The code (used as the code in the resource instance) |
| Display | The display (used in the *display* element of a [Coding](file:///C:\temp\datatypes.html#Coding)). If there is no display, implementers should not simply display the code, but map the concept into their application |
| Definition | An explanation of the meaning of the concept |
| Comments | Additional notes about how to use the code |

**template-definitions.html**

**Resource - Detailed Descriptions**

Detailed Descriptions for the elements in the resource.

**template-dictionary.html**

**template-example-json.html**

Raw JSON (canonical form)

Usage note: every effort has been made to ensure that the examples are correct and useful, but they are not a normative part of the specification.

**template-example-shex.html**

Raw ShEx

Usage note: every effort has been made to ensure that the ShEx files are correct and useful, but they are not a normative part of the specification.

**template-example-ttl.html**

Raw Turtle, JSON-LD

Usage note: every effort has been made to ensure that the examples are correct and useful, but they are not a normative part of the specification.

**template-example-xml.html**

Raw XML (canonical form)

Usage note: every effort has been made to ensure that the examples are correct and useful, but they are not a normative part of the specification.

**template-example.html**

This is the narrative for the resource. See also the [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html) format.

Usage note: every effort has been made to ensure that the examples are correct and useful, but they are not a normative part of the specification.

**template-examples.html**

Usage note: every effort has been made to ensure that the examples are correct and useful, but they are not a normative part of the specification.

**template-explanations.html**

**Resource - Design Notes**

to do

**template-extension-definitions.html**

**Extension - Detailed Descriptions**

Detailed Descriptions for the elements in the extension.

**template-extension-mappings.html**

Mappings for the extension.

**template-extension.html**

**Extension:**

|  |  |  |
| --- | --- | --- |
|  |  |  |

URL for this extension:

Status: .

**Extension Content**

* [Summary](file:///C:\temp\main-pages.html#tabs-struc1)
* [Full Structure](file:///C:\temp\main-pages.html#tabs-struc2)
* [XML](file:///C:\temp\main-pages.html#tabs-xml)
* [JSON](file:///C:\temp\main-pages.html#tabs-json)
* [All](file:///C:\temp\main-pages.html#tabs-all)

**Summary**

**Fulle Structure**

**XML Template**

**JSON Template**

**Summary**

**Full Structure**

**XML Template**

**JSON Template**

**template-ig-operations.html**

**Operations defined by**

This Implementation guide defines :

For more information about operations, including how they are invoked, see [Operations](file:///C:\temp\operations.html).

**template-ig-toc.html**

**Table of Contents**

**template-logical-definitions.html**

**Resource - Detailed Descriptions**

Detailed Descriptions for the elements in the resource.

**template-logical-examples.html**

**Logical Model - Examples**

**template-logical-mappings.html**

Mappings for the resource.

**template-logical.html**

**Logical Model - Content**

**Logical Model Content**

* [Structure](file:///C:\temp\main-pages.html#tabs-struc)
* [UML](file:///C:\temp\main-pages.html#tabs-uml)
* [Both](file:///C:\temp\main-pages.html#tabs-all)

**Structure**

**UML Diagram** ([Legend](file:///C:\temp\formats.html#uml))

**Structure**

**UML Diagram** ([Legend](file:///C:\temp\formats.html#uml))

**template-mappings.html**

Mappings for the resource.

**template-operations.html**

**Resource - Operations**

This resource has associated with it:

For more information about operations, including how they are invoked, see [Operations](file:///C:\temp\operations.html).

**template-print.html**

**Resource Definition:**

.

**Resource Content**

Alternate definitions: Master Definition ([XML](file:///C:\temp\.profile.xml.html), [JSON](file:///C:\temp\.profile.json.html)), [Schema](file:///C:\temp\.xsd)/[Schematron](file:///C:\temp\.sch) (for [XML](file:///C:\temp\xml.html)), [ShEx](file:///C:\temp\.shex.html) (for [Turtle](file:///C:\temp\rdf.html))

**Examples**

|  |
| --- |
|  |

**Detailed Descriptions**

The Detailed Descriptions for the elements above

**template-profile-constraint.html**

**template-profile-definitions.html**

**- Detailed Descriptions**

Definitions for the Profile.

**template-profile-example-json.html**

**- JSON**

**template-profile-example-xml.html**

**- XML**

**template-profile-examples.html**

**Examples**

**template-profile-mappings.html**

**- Mappings**

Mappings for the Profile.

**template-profile-questionnaire.html**

**- Example Form**

This is an example form based on a questionnaire generated from the definition, to help visualise the content. See also the [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html) questionnaire.

**template-profile.html**

|  |  |  |
| --- | --- | --- |
|  |  |  |

The official URL for this profile is:

This profile was published on by .

**Formal Views of Profile Content**

[Description of Profiles, Differentials, Snapshots, and how the XML and JSON presentations work](file:///C:\temp\profiling.html#representation).

* [Text Summary](file:///C:\temp\main-pages.html#tabs-summ)
* [Differential Table](file:///C:\temp\main-pages.html#tabs-diff)
* [Snapshot Table](file:///C:\temp\main-pages.html#tabs-snap)
* [XML Template](file:///C:\temp\main-pages.html#tabs-xml)
* [JSON Template](file:///C:\temp\main-pages.html#tabs-json)
* [All](file:///C:\temp\main-pages.html#tabs-all)

This structure is derived from .

This structure is derived from .

This structure is derived from .

**Differential View**

This structure is derived from .

**Snapshot View**

**XML Template**

**JSON Template**

Other representations of profile:

.

**template-profiles.html**

**Resource - Extensions & Profiles**

This table lists profiles and extensions for the resource. For background information, see [Profiling Resources](file:///C:\temp\profiling.html#resources) and [Extensibility](file:///C:\temp\extensibility.html). Additional profiles and resources may be found in published [Implementation Guides](http://fhir.org/guides/registry), or in the [Conformance resource registry](http://registry.fhir.org).

**Profiles**

|  |
| --- |
|  |

**Extensions**

|  |
| --- |
|  |

Extensions that reference this resource:

|  |
| --- |
|  |

**Search Extensions**

|  |
| --- |
|  |

**template-questionnaire.html**

**- Example Form**

This is an example form based on a questionnaire generated from the definition, to help visualise the content. See also the [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html) questionnaire.

**template-search-parameter.html**

Search Parameter (type: ) defined as part of [the Profile](file:///C:\temp\.html).

Other Details:

* Full URL :
* Paths :
* Targets :

**template-tx-book.html**

**Codes defined in**

Formal value Set definition (identifier ): [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html).

See [the full registry of codes](file:///C:\temp\terminologies-codes.html) defined as part of FHIR.

The OID for the value set is (OIDs are not used in FHIR, but may be used in [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186), or OID based terminology systems).

Explanation of the columns that may appear on this page:

|  |  |
| --- | --- |
| Id | The internal identifier for the concept (when the value set defines its own codes) |
| Level | A few code lists that FHIR defines are hierarchical - each code is assigned a level. In this scheme, some codes are under other codes, and imply that the code they are under also applies |
| Source | The source of the definition of the code (when the value set draws in codes defined elsewhere) |
| Code | The code (used as the code in the resource instance) |
| Display | The display (used in the *display* element of a [Coding](file:///C:\temp\datatypes.html#Coding)). If there is no display, implementers should not simply display the code, but map the concept into their application |
| Definition | An explanation of the meaning of the concept |
| Comments | Additional notes about how to use the code |

**template-tx.html**

**Value Set for codes in**

This template should not be used any more

**Summary**

|  |  |
| --- | --- |
| Code System URL: |  |
| Value Set URL: |  |
| Definition: |  |

Formal value Set definition : [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html).

**OIDs**

|  |  |
| --- | --- |
| Code System OID: |  |
| Value Set OID: |  |
| Note: these OIDs are not used in FHIR, but may be used in [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186), or OID based terminology systems | |

See [the full registry of value sets](file:///C:\temp\terminologies-valuesets.html) defined as part of FHIR.

Explanation of the columns that may appear on this page:

|  |  |
| --- | --- |
| Level | A few code lists that FHIR defines are hierarchical - each code is assigned a level. In this scheme, some codes are under other codes, and imply that the code they are under also applies |
| Source | The source of the definition of the code (when the value set draws in codes defined elsewhere) |
| Code | The code (used as the code in the resource instance) |
| Display | The display (used in the *display* element of a [Coding](file:///C:\temp\datatypes.html#Coding)). If there is no display, implementers should not simply display the code, but map the concept into their application |
| Definition | An explanation of the meaning of the concept |
| Comments | Additional notes about how to use the code |

In addition, this page will include mappings to [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) or [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) code where these have been defined.

**template-version-maps.html**

**Resource - Detailed Descriptions**

R2 : R3 Converson maps for .

Functional status for this map: ([see documentation)](file:///C:\temp\r2maps.html)

**R2 to R3**

**R3 to R2**

**template-vs-book.html**

**Value Set**

This is a value set defined by the FHIR project

Detailed Descriptions: [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html).

The OID for the value set is (OIDs are not used in FHIR, but may be used in v3, or OID based terminology systems).

See [the full registry of value sets](file:///C:\temp\terminologies-valuesets.html) defined as part of FHIR.

**template-vs-ig-book.html**

**Value Set**

Detailed Descriptions: [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html).

**template-vs-ig.html**

**Value Set**

Detailed Descriptions: [XML](file:///C:\temp\.xml.html) or [JSON](file:///C:\temp\.json.html).

**template-vs.html**

**Value Set**

This is a value set defined

**Summary**

|  |  |
| --- | --- |
| Defining URL: |  |
| Name: |  |
| Definition: |  |
| Committee: |  |
| OID: | (for OID based terminology systems) |
| Source Resource | [XML](file:///C:\temp\.xml.html) / [JSON](file:///C:\temp\.json.html) |

**Content Logical Definition**

See [the full registry of value sets](file:///C:\temp\terminologies-valuesets.html) defined as part of FHIR.

Explanation of the columns that may appear on this page:

|  |  |
| --- | --- |
| Level | A few code lists that FHIR defines are hierarchical - each code is assigned a level. In this scheme, some codes are under other codes, and imply that the code they are under also applies |
| Source | The source of the definition of the code (when the value set draws in codes defined elsewhere) |
| Code | The code (used as the code in the resource instance) |
| Display | The display (used in the *display* element of a [Coding](file:///C:\temp\datatypes.html#Coding)). If there is no display, implementers should not simply display the code, but map the concept into their application |
| Definition | An explanation of the meaning of the concept |
| Comments | Additional notes about how to use the code |

**template.html**

**Resource - Content**

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Resource Content**

* [Structure](file:///C:\\temp\\main-pages.html" \l "tabs-struc)
* [UML](file:///C:\temp\main-pages.html#tabs-uml)
* [XML](file:///C:\temp\main-pages.html#tabs-xml)
* [JSON](file:///C:\temp\main-pages.html#tabs-json)
* [Turtle](file:///C:\temp\main-pages.html#tabs-ttl)
* [R2 Diff](file:///C:\temp\main-pages.html#tabs-diff)
* [All](file:///C:\temp\main-pages.html#tabs-all)

**Structure**

**UML Diagram** ([Legend](file:///C:\temp\formats.html#uml))

**XML Template**

**JSON Template**

**Turtle Template**

**Changes since DSTU2**

**Structure**

**UML Diagram** ([Legend](file:///C:\temp\formats.html#uml))

**XML Template**

**JSON Template**

**Turtle Template**

**Changes since DSTU2**

Alternate definitions: Master Definition ([XML](file:///C:\temp\.profile.xml.html), [JSON](file:///C:\temp\.profile.json.html)), [XML](file:///C:\temp\xml.html) [Schema](file:///C:\temp\.xsd)/[Schematron](file:///C:\temp\.sch) (for ) + [JSON](file:///C:\temp\json.html) [Schema](file:///C:\temp\.schema.json), [ShEx](file:///C:\temp\.shex.html) (for [Turtle](file:///C:\temp\rdf.html))

**terminologies-bindings.html**

**Terminology Bindings**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This table contains a list of all the terminology bindings in FHIR.

**terminologies-conceptmaps.html**

**Mappings between Value Sets Defined in FHIR**

This table contains a list of all the mappings between value sets that are defined as part of the FHIR specification.

**terminologies-systems.html**

**Code Systems**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The following names (URIs) may be used in the *system* element of the [Coding](file:///C:\temp\datatypes.html#Coding) data type. If a URI is defined here, it SHALL be used in preference to any other identifying mechanisms. If a code system is not listed here, the correct URI may be determined by working through the following list, in order:

* the HL7 OID Registry
* the documentation associated with the code system
* consulting the owner of the code system
* asking on the HL7 vocabulary mailing list

See also the [list of known identifier systems](file:///C:\temp\identifier-registry.html) that can be used in the *system* element of the [Identifier](file:///C:\temp\datatypes.html#Identifier) data type. Additional identifier systems may be registered on the HL7 FHIR registry at <http://hl7.org/fhir/registry>.

**Important Notes:**

* This list of names is incomplete and subject to change. Some values may be dropped and others will likely be added in the coming months as HL7 institutes formal processes around URIs in vocabulary
* Note that some of the URNs in this list follow the URN specification in [RFC 5141](http://tools.ietf.org/html/rfc5141) for referring to standards published by ISO, such as urn:iso:std:iso:11073:10101. Where ISO standards define codes with meanings, and there is no entry in the list above, and they are not registered in the HL7 OID registry, the default URN for the code system is that defined by the RFC 5141.
* For several of the code systems in this list, multiple systems are given. This means that the variants identified are different code systems, not just variants of the same code system
* Any URL in http://example.org is reserved for testing and documentation purposes.
* [External](file:///C:\temp\main-pages.html#tabs-ext)
* [Internal (FHIR)](file:///C:\temp\main-pages.html#tabs-fhir)
* [HL7 V3](file:///C:\temp\main-pages.html#tabs-v3)
* [HL7 v2](file:///C:\temp\main-pages.html#tabs-v2)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **URI** | | **Source** | | | **Comment** | **OID (for non-FHIR systems)** |
| **Externally Published code systems** | | | | | | |
| http://snomed.info/sct | | SNOMED CT ([IHTSDO](http://ihtsdo.org)) | | | See [Using SNOMED CT with FHIR](file:///C:\temp\snomedct.html) | 2.16.840.1.113883.6.96 |
| http://www.nlm.nih.gov/research/umls/rxnorm | | RxNorm ([US NLM](http://www.nlm.nih.gov/)) | | | See [Using RxNorm with FHIR](file:///C:\temp\rxnorm.html) | 2.16.840.1.113883.6.88 |
| http://loinc.org | | LOINC ([LOINC.org](http://loinc.org)) | | | See [Using LOINC with FHIR](file:///C:\temp\loinc.html) | 2.16.840.1.113883.6.1 |
| http://unitsofmeasure.org | | UCUM: ([UnitsOfMeasure.org](http://unitsofmeasure.org)) Case Sensitive Codes | | | See [Using UCUM with FHIR](file:///C:\temp\ucum.html) | 2.16.840.1.113883.6.8 |
| http://ncimeta.nci.nih.gov | | [NCI Metathesaurus](http://ncimeta.nci.nih.gov) | | | See [Using NCI Metathesaurus with FHIR](file:///C:\temp\ncimeta.html) | 2.16.840.1.113883.3.26.1.2 |
| http://www.ama-assn.org/go/cpt | | [AMA CPT codes](http://www.ama-assn.org/go/cpt) | | | See [Using CPT with FHIR](file:///C:\temp\cpt.html) | 2.16.840.1.113883.6.12 |
| http://hl7.org/fhir/ndfrt | | [NDF-RT (National Drug File Ã¢â‚¬â€œ Reference Terminology)](http://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/NDFRT/) | | | See [Using NDF-RT with FHIR](file:///C:\temp\ndfrt.html) | 2.16.840.1.113883.6.209 |
| http://fdasis.nlm.nih.gov | | [Unique Ingredient Identifier (UNII)](http://www.fda.gov/Drugs/InformationOnDrugs/ucm142438.htm) | | | See [Using UNII with FHIR](file:///C:\temp\unii.html) | 2.16.840.1.113883.4.9 |
| http://hl7.org/fhir/sid/ndc | | [NDC/NHRIC Codes](http://www.fda.gov/Drugs/InformationOnDrugs/ucm142438.htm) | | | See [Using NDC with FHIR](file:///C:\temp\ndc.html) | 2.16.840.1.113883.6.69 |
| http://hl7.org/fhir/sid/cvx | | [CVX (Vaccine Administered)](http://www2a.cdc.gov/vaccines/iis/iisstandards/vaccines.asp?rpt=cvx) | | | See [Using CVX with FHIR](file:///C:\temp\cvx.html) | 2.16.840.1.113883.12.292 |
| urn:iso:std:iso:3166 | | [ISO 2 letter Country Codes](http://www.iso.org/iso/country_codes.htm) | | | a few country codes have been reused (e.g. CS). If a version is needed, simply use the year of publication e.g. 1998 | 1.0.3166.1.2.2 |
| http://www.nubc.org/patient-discharge | | [NUBC](http://www.nubc.org) code system for Patient Discharge Status | | | National Uniform Billing Committee, manual UB-04, UB form locator 17 | 2.16.840.1.113883.6.301.5 |
| http://www.radlex.org | | [RadLex](http://www.radlex.org) | | | (includes play book codes as well) | 2.16.840.1.113883.6.256 |
| ICD-9, ICD-10 | | [WHO](http://www.who.int/classifications/icd/en/)) & National Variants | | | See [Using ICD-[x] with FHIR](file:///C:\temp\icd.html) | See ICD page for details |
| http://hl7.org/fhir/sid/icpc-1  http://hl7.org/fhir/sid/icpc-1-nl  http://hl7.org/fhir/sid/icpc-2 | | ICPC (International Classification of Primary Care) ([PH3C](http://www.ph3c.org/)) | | |  | 2.16.840.1.113883.2.4.4.31.1  2.16.840.1.113883.6.139 |
| http://hl7.org/fhir/sid/icf-nl | | ICF (International Classification of Functioning, Disability and Health) ([WHO](http://www.who.int/classifications/icf/en/)) | | |  | 2.16.840.1.113883.6.254 |
| http://hl7.org/fhir/v2/[X](/v) | | [Version 2 tables](file:///C:\temp\terminologies-v2.html) | | | [X] is the 4 digit identifier for a table; e.g. http://hl7.org/fhir/v2/0203 Note: only [some tables](file:///C:\temp\terminologies-v2.html) may be treated in this fashion. For some tables, the meaning of the code is version dependent, and so additional information must be included in the namespace, e.g. http://hl7.org/fhir/v2/0123/2.3+, as defined in the [v2 table namespace list](file:///C:\temp\terminologies-v2.html). Version 2 codes are case sensitive. | 2.16.840.1.113883.12.[X] |
| http://hl7.org/fhir/v3/[X] | | [A](file:///C:\temp\terminologies-v3.html)[HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) code system | | | [X] is the code system name; e.g. http://hl7.org/fhir/v3/GenderStatus. HL7 v3 code systems are case sensitive. | see [v3 list](file:///C:\temp\terminologies-v3.html) |
| http://www.whocc.no/atc | | Anatomical Therapeutic Chemical Classification System ([WHO](http://www.whocc.no/atc/structure_and_principles/)) | | |  | 2.16.840.1.113883.6.73 |
| urn:ietf:bcp:47 | | IETF language (see [Tags for Identifying Languages - BCP 47](http://tools.ietf.org/html/bcp47)) | | | This is used for identifying langauge throughout FHIR. Note that usually these codes are in a code and the system is assumed |  |
| urn:ietf:bcp:13 | | Mime Types (see [Multipurpose Internet Mail Extensions (MIME) Part Four - BCP 13](http://tools.ietf.org/html/bcp13)) | | | This is used for identifying the mime type system throughout FHIR. Note that these codes are in a code (e.g. [Attachment.contentType](file:///C:\temp\datatypes.html#Attachment) and in these elements the system is assumed). This system is defined for when constructing value sets of mime type codes |  |
| urn:iso:std:iso:11073:10101 | | Medical Device Codes defined in ISO 11073-10101 (see [RTM Management service](http://hit-testing.nist.gov:13110/rtmms/index.html#rosetta)) | | | The ISO/IEEE 11073-10201 Terminology Code is used in the *code* element, and the Terminology Reference ID is used in the *display* element (CODE10 and REFID in the reference) | 2.16.840.1.113883.6.24 |
| [http://nema.org/dicom/dicm](file:///C:\temp\codesystem-dicom-dcim.html) | | DICOM Code Definitions | | | The meanings of codes defined in DICOM, either explicitly or by reference to another part of DICOM or an external reference document or standard | 1.2.840.10008.2.16.4 |
| http://hl7.org/fhir/sid/ca-hc-din | | [Health Canada Drug Identification Number](http://www.hc-sc.gc.ca/dhp-mps/prodpharma/activit/fs-fi/dinfs_fd-eng.php) | | | A computer-generated eight digit number assigned by Health Canada to a drug product prior to being marketed in Canada. It uniquely identifies all drug products sold in a dosage form in Canada and is located on the label of prescription and over-the-counter drug products that have been evaluated and authorized for sale in Canada. A DIN uniquely identifies the following product characteristics: manufacturer; product name; active ingredient(s); strength(s) of active ingredient(s); pharmaceutical form; route of administration.  [Canada Health Drug Product Database](http://www.hc-sc.gc.ca/dhp-mps/prodpharma/databasdon/index-eng.php) contains product specific information on drugs approved for use in Canada. The database is managed by Health Canada and includes human pharmaceutical and biological drugs, veterinary drugs, radiopharmaceutical drugs and disinfectant products. It contains approximately 47,000 products that are currently approved, marketed or cancelled. Updates to the database are ongoing. If a version is needed, the correct version is YYYYMMDD for the date of download. | 2.16.840.1.113883.5.1105 |
| http://nucc.org/provider-taxonomy | | [NUCC Provider Taxonomy](http://www.nucc.org/index.php/code-sets-mainmenu-41/provider-taxonomy-mainmenu-40/csv-mainmenu-57) | | | The Health Care Provider Taxonomy code is a unique alphanumeric code, ten characters in length. The code set is structured into three distinct "Levels" including Provider Type, Classification, and Area of Specialization. | 2.16.840.1.113883.6.101 |
| **Code Systems for Genetics** | | | | | | |
| http://www.genenames.org | | HGNC: Human Gene Nomenclature Committee | | |  | 2.16.840.1.113883.6.281 |
| http://www.ensembl.org | | ENSEMBL reference sequence identifiers | | | Maintained jointly by the European Bioinformatics Institute and Welcome Trust Sanger Institute | *not assigned yet* |
| http://www.ncbi.nlm.nih.gov/nuccore | | REFSEQ : National Center for Biotechnology Information (NCBI) Reference Sequences | | |  | 2.16.840.1.113883.6.280 |
| http://www.ncbi.nlm.nih.gov/clinvar | | ClinVar | | | NCBI central respository for curating pathogenicity of potentially clinically relevant variants | *not assigned yet* |
| http://sequenceontology.org | | Sequence Ontology | | |  | *not assigned yet* |
| http://www.hgvs.org/mutnomen | | HGVS : Human Genome Variation Society | | |  | 2.16.840.1.113883.6.282 |
| http://www.ncbi.nlm.nih.gov/projects/SNP | | DBSNP : Single Nucleotide Polymorphism database | | |  | 2.16.840.1.113883.6.284 |
| http://cancer.sanger.ac.uk/ cancergenome/projects/cosmic | | COSMIC : Catalogue Of Somatic Mutations In Cancer | | |  | 2.16.840.1.113883.3.912 |
| http://www.lrg-sequence.org | | LRG : Locus Reference Genomic Sequences | | |  | 2.16.840.1.113883.6.283 |
| http://www.omim.org | | OMIM : Online Mendelian Inheritance in Man | | |  | 2.16.840.1.113883.6.174 |
| http://www.ncbi.nlm.nih.gov/pubmed | | PubMed | | |  | 2.16.840.1.113883.13.191 |
| http://www.pharmgkb.org | | PHARMGKB : Pharmacogenomic Knowledge Base | | |  | 2.16.840.1.113883.3.913 |
| http://clinicaltrials.gov | | ClinicalTrials.gov | | |  | 2.16.840.1.113883.3.1077 |
| http://www.ebi.ac.uk/ipd/imgt/hla/ | | European Bioinformatics Institute | | |  | 2.16.840.1.113883.13.252 |
| **URI (all prefixed with http://hl7.org/fhir/)** | **Description** | | **OID** |

**terminologies-valuesets.html**

**Value Sets Defined in FHIR**

This table contains a list of all the value sets defined as part of the FHIR specification. Some of these value sets include codes defined elsewhere, some contain their own inline code system definitions, and some do both. Any implicit code systems are case sensitive, though FHIR will never define codes that only differ by case.

* [FHIR](file:///C:\temp\main-pages.html#tabs-fhir)
* [HL7 V3](file:///C:\temp\main-pages.html#tabs-v3)
* [HL7 v2](file:///C:\temp\main-pages.html#tabs-v2)

|  |  |  |
| --- | --- | --- |
| **URI** | **Description** | **OID** |

**terminologies.html**

**Using Codes in Resources**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Many elements in the FHIR resources have a **coded value**: some fixed string (a sequence of characters) assigned elsewhere that identifies some defined "concept". The sequence of characters and its meaning may be defined in one of several places:

* As one of a set of fixed values defined in this specification
* In an internet RFC (e.g. mime type, language)
* An HL7 specification ([HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) code system, or [HL7 v2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) table)
* Some external terminology or ontology such as [LOINC](http://loinc.org), or [SNOMED CT](http://www.ihtsdo.org)
* A locally maintained dictionary, look up table or enumeration in an application (for further discussion of locally defined value sets, see ["Profiling FHIR"](file:///C:\temp\profiling.html)

All of these kinds of ways of defining codes are collectively called "code systems". This list is far from complete; there are many ways to define code systems, and they vary widely in sophistication and size.

Throughout this specification, coded values are always treated as a pair composed of "system" and "code", where the system is a URL that identifies the code system that defines the codes. Note that system values are always case sensitive. Different code systems make their own rules as to whether the codes they define are case sensitive or not. Note that all the codes defined by FHIR itself are case sensitive and SHALL be used in the provided case (usually, but not always, lowercase).

The FHIR framework for using coded values is based on the fundamental framework defined in section 5 of the [HL7 v3 Core Principles](http://www.hl7.org/documentcenter/public/standards/V3/core_principles/infrastructure/coreprinciples/v3modelcoreprinciples.html) document, including the separation between code systems and value sets.

When codes are carried in resources, one of 4 different data types is used:

|  |  |
| --- | --- |
| [code](file:///C:\temp\datatypes.html#code) | The instance represents the *code* only. The *system* is implicit - it is defined as part of the definition of the element, and not carried in the instance. |
| [Coding](file:///C:\temp\datatypes.html#Coding) | A data type that has a *code* and a *system* element that identifies where the definition of the code comes from |
| [CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept) | A type that represents a concept by plain text and/or one or more *Codings* |
| [Quantity](file:///C:\temp\datatypes.html#Quantity) | Special case: has *system* and *code* elements for carrying a code for the type of unit |

Notes:

* generally the choice of data type is dictated by the resource itself. When choosing a data type for an [extension](file:///C:\temp\extensibility.html), see the FHIR wiki for [advice about data type choice](http://wiki.hl7.org/index.php?title=FHIR_Guide_to_Designing_Resources#Choice_of_coding_data_type).
* the types [string](file:///C:\temp\datatypes.html#string) and [uri](file:///C:\temp\datatypes.html#uri) can also be bound - [see below](file:///C:\temp\main-pages.html#strings)

**Coded Values, Systems, and Value Sets**

The set of coded values that is allowed to be used in an element of one of these 4 data types is known as a ["value set"](file:///C:\temp\valueset.html). Anywhere these data types are used, the specification "binds" a value set to the element.

The difference between a code system and a value set is an important distinction that is easily missed by implementers, since the difference is often overlooked in system design. For instance, it's not unusual to see an application table that is a mixed list of codes, containing some LOINC codes and also some additional in-house codes. Quite often, there is no explicit differentiation between them; only the fact that a code happens to look like a LOINC code betrays its origin.

For data exchange, on the other hand, explicitly tracking the source of the code is both important and necessary. In order to do this, each code system that defines codes is assigned a URL that identifies it, and all the codes it defines are actually a pair ("Code Pair": a name with a namespace). So in the case of this mixed list example from the previous paragraph, there are two code systems: [LOINC](file:///C:\temp\loinc.html) (http://loinc.org) and a local one (let's say it has been given the URL: http://example.com/codesystems/additional-test-codes). The application table is a single value set (a set of Code Pairs) that includes codes from each of those two namespaces. The value set itself is given its own URL as an identifier (e.g. "http://example.com/fhir/ValueSet/test-codes)") - this identifies the set of Code Pairs, but is never used as the namespace in a actual code pair, or in an instance. In FHIR, Code Pairs are always represented as "code" and "system", except for the simple [data type "code"](file:///C:\temp\datatypes.html#code) data type where the namespace (e.g. the system element/property) is fixed in the schema and not represented explicitly.

Note that for some code systems, there is a single correct machanism by which to represent codes defined by the system as a single URL. These single URLs are used in the context of the [RDF](file:///C:\temp\rdf.html#concept) format to enable ontological reasoning. The URL is often a direct reference to a web source that can provide additional definitional material about the concept. Where the mechanism is known and defined by the code system, it is described in this specification.

**Choosing a system**

The URL in a system is always a reference to a code system, not to a value set. The system ensures that codes can be unambiguously traced back to their original definition, and that logical comparisons, matching and inferences can be performed consistently by different systems. For this reason, choice of the correct URI for the system attribute is critical.

The correct value to use in the *system* for a given code system can be determined by working through the following list, in order:

* the specification [Code System Registry](file:///C:\temp\terminologies-systems.html) - if a code system is listed here, it SHALL be used
* A system URI or OID defined as the correct value to use in FHIR by the publisher of the code system
* the FHIR [community code system registry](http://registry.fhir.org/NamingSystem) - if a code system is listed here with [status = active](file:///C:\temp\namingsystem-definitions.html#NamingSystem.status), it SHALL be used
* an OID registered in the [HL7 OID registry](http://hl7.org/oid) - if a code system is registered here, the OID SHOULD be used (using the syntax urn:oid:[oid])

If a code system is not resolved by this list, and there is no publisher to consult, implementers will have to choose a URI to use. The priority should be to choose a unique value that won't accidently be used by another implementer for a different purpose - or a very similar purpose with a different scope.

For publishers of code systems, the following considerations should be kept in mind when defining the correct URI to use:

* Once defined, the URI will require agreement from all implementers to change, and some may not be able to change (stored resources). If the set of users is not closed, it will usually not be possible to change the URI
* Implementers strongly prefer a human readable URI. http://acme.com/patients/mrn is a great deal easier to work with than urn:oid:1.2.3.4.5.6.7
* An http: address SHOULD resolve to some useful description of the code system. Ideally, if a user makes a request of the address with the media type set to a FHIR media type, the server will respond with a FHIR Value Set resource with an inline [code system](file:///C:\temp\valueset-definitions.html#ValueSet.codeSystem), but some other human or computable definition is allowed
* HTTP addresses should be permalinks which may re-direct to the current correct content
* A scope of the code system URI and the correct usage of codes and displays in its namespace SHOULD be clearly defined. See examples for [SNOMED CT](file:///C:\temp\snomedct.html), [RxNorm](file:///C:\temp\rxnorm.html), [LOINC](file:///C:\temp\loinc.html), [NDC](file:///C:\temp\ndc.html)
* Generally, allocation of URLs is hierarchical, and most care is required in choosing the Base URL. Once sub-URL policies are clearly defined, URIs can often be automatically assigned

Note: if the code system is made available packaged inside a [ValueSet](file:///C:\temp\valueset.html) resource, the correct URL for the system value is *ValueSet.codeSystem.system*, not *ValueSet.uri*.

**Controlling the use of Coded Values**

When an element is bound to a value set, the binding has these properties:

|  |  |
| --- | --- |
| Name | A descriptive name used when presenting information about the binding |
| Strength | How the binding should be understood - see below |
| Reference | A URL that defines the value set. Usually, this is a direct reference to a [ValueSet](file:///C:\temp\valueset.html) resource, but can be a more indirect reference, where the value set is inferred |
| Description | A text description of the use of the codes. If there is no reference, this must be populated. When there is a reference, this can be used to make additional notes about the use and implementation of the value set |

In the FHIR declarative datatypes, a binding is always represented using an [ElementDefinition.binding](file:///C:\temp\elementdefinition-definitions.html#ElementDefinition.binding).

**Value Set References**

There are a number of places in the specification where value sets are referenced in order to bind a coded value to a value set:

|  |  |
| --- | --- |
| [ElementDefinition](file:///C:\temp\elementdefinition.html).binding.valueSet[x] | Used to bind a defined element to a value set |
| [ConceptMap](file:///C:\temp\conceptmap.html).source[x] and .target[x] | used to indicate the scope of the mapping in the Concept Map - from one value set to another |
| [Questionnaire](file:///C:\temp\questionnaire.html).group.question.options | Indicates that answers to a set of questions come from a value set |
| [ValueSet](file:///C:\temp\valueset.html).compose.import | The content of a value set includes the content in the imported value set too |
| [ValueSet Reference Extension](file:///C:\temp\extension-valueset-reference.html) | Indicates that a particular coded value was chosen from the specified value set |

There are two types of value set references in this list, direct and logical.

**Direct Value Set references**

A direct value set reference has the type [Reference](file:///C:\temp\references.html#Reference), and refers directly to a ValueSet based on a URL, usually to a terminology server running a [FHIR RESTful API](file:///C:\temp\http.html). When accessing a value set based on this kind of reference, a system should access the URL directly (after converting a relative reference to an absolute reference according to the local context). If this process fails, the system is unable to resolve the value set and must handle the error appropriately.

Example:

GET fhir/Questionnaire/234

<Questionnaire>

...

<question>

<options>

<reference value="ValueSet/234234"/>

</options>

</question>

....

</Questionnaire>

This specifies that the values for a particular questionnaire come from the ValueSet with id 234234 on the same FHIR end-point. To resolve this, the system would GET fhir/ValueSet/234234

Typically, a direct reference like this is good for in-process references, in closed or carefully managed eco-systems. In a more general context, these references tend to be fragile over time because web URLs - including RESTful API URLS - are easily reassigned. For this reason, systems are encouraged to use logical value set references.

**Logical Value Set references**

A logical value set reference has the type [uri](file:///C:\temp\datatypes.html#uri), where an absolute URI is provided that matches the one in ValueSet.url. The value set URL can - and is preferred to be - a web address that actually resolves directly to a fixed web address that serves as the authoritative source for that value set. Alternatively, the system can query its terminology server(s) to resolve a value set with that URL as its identity.

Example:

<StructureDefinition>

...

<element>

...

<binding>

...

<valueSetUri value="http://hl7.org/fhir/ValueSet/clinical-findings"/>

</binding>

...

</element>

....

</StructureDefinition>

This specifies that the element is bound to the value set with a ValueSet.url of <http://hl7.org/fhir/ValueSet/clinical-findings>. One way to accees this value set is to try GET http://hl7.org/fhir/ValueSet/clinical-findings - which works, for this value set - http://hl7.org/fhir/ValueSet/clinical-findings returns the authoritative value set for this URL.

Alternatively, the value set could be resolved using a local terminology server. If that's running a [FHIR Terminology Server](file:///C:\temp\terminology-service.html), then this would work like this:

GET fhir/ValueSet?url=http://hl7.org/fhir/ValueSet/clinical-findings

if the terminology server knows the value set, then it will return the value set. If the URL doesn't resolve to an authoritative value set, and the terminology server(s) don't know the value set, the system is unable to resolve the value set and must handle the error appropriately.

The value set URL is allowed to be a URI such as a UUID (e.g. urn:uuid:c0e0d027-1250-4278-8f44-33a49dc67916). These value sets can never be accessed directly, and must come from a terminology server. Note that this specification defines many value sets that have a logical URL that is not resolvable (examples for [SNOMED CT](file:///C:\temp\snomedct.html#implicit), [RxNorm](file:///C:\temp\rxnorm.html#implicit), [LOINC](file:///C:\temp\loinc.html#implicit))

Using a logical reference which is a direct reference to the authoritative value set is the easiest and most reliable approach. However this requires suitable hosting arrangements, and cannot always be guaranteed, so it is not required.

**Version specific Logical References**

A value set has a two part identifier: a url, and a version. Some value sets only ever have a single 'version'; a revision of the value set contents will cause a new url to be assigned. Others, however, maintain the same URL, and change the version. A terminology server may have multiple value sets for the same ValueSet.url with different versions.

To be precise about which version of a value set is being referred to in a value set reference, append the version to the canonical url with a '|' like this:

<valueSetUri value="http://hl7.org/fhir/ValueSet/clinical-findings|0.8"/>

This is a version specific reference to a value set. Searching for this on a terminology server would look like this:

GET fhir/ValueSet?url=http://hl7.org/fhir/ValueSet/clinical-findings&version=0.8

Note that if a value set reference does not have a version, and the server finds multiple versions for the value set, the system using the value set should pick the latest version of the value set and use that. Note that this applies to all [conformance resources](file:///C:\temp\references.html#canonical).

**Unbound**

Note that as a matter of ongoing development, a few elements that have coded data types are not bound to any value set at all. Bindings are to be provided for these elements.

**Binding Strengths**

Almost all of the elements that have a coded data type are bound to a value set. The bindings are associated with various degrees of flexibility as to how closely the value set should be followed:

|  |  |
| --- | --- |
| required | To be conformant, codes in this element SHALL be from the specified value set |
| extensible | To be conformant, instances of this element must include a code from the specified value set if any of the codes within the value set can apply to the concept being communicated. If the value set does not cover the concept (based on human review), alternate codings (from different code systems, including local ones) or (data type allowing) text) may be included instead. |
| preferred | Instances are encouraged, to draw from the specified codes for interoperability purposes but are not required to do so to be considered conformant |
| example | Instances are not expected or even encouraged to draw from the specified value set. The value set merely provides examples of the types of concepts intended to be included |

The precise conformance criteria for 'required' and 'extensible' binding strengths vary by the data type to which they are applied, as described in the paragraphs below.

Irrespective of the binding strength, when a [StructureDefinition](file:///C:\temp\structuredefinition.html) is used to describe local usage, it can bind the element to a different value set in order to be much more precise about exactly which coded values can be used for these elements, and/or increase the strength of the binding. There are different rules for this, depending on the binding strength, as discussed below. Generally it is expected that jurisdictions, projects and vendors will work together to choose actual working value sets.

**Required**

*To be conformant, codes in this element SHALL be from the specified value set.*.

In the standard, this is generally used for elements where the value needs to be strictly controlled so that everyone can interpret it with confidence. Generally, this is used for elements with type [code](file:///C:\temp\datatypes.html#code):

* the element is bound to a value set that contains a list of distinct codes with a specified system (and version, where required)
* the element is bound to some external standard that defines the set of valid codes that can be used (typical examples of references are [Mime Types](http://www.rfc-editor.org/bcp/bcp13.txt), [Language Codes](http://tools.ietf.org/html/bcp47), [UCUM](http://unitsofmeasure.org), etc.)

The other place where this is used is when [profiling resources](file:///C:\temp\profiling.html), and there is agreement within a particular context of use that a particular set of codes are the only ones that can be used. In these cases, the data type SHALL contain one of the values in the value set.

The following rules apply when required bindings are used with the [CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept) data type:

* at least one Coding element SHALL be present
* one of the Coding values SHALL be from the specified value set
* text can be provided as well, and is always recommended, but is not an acceptable substitute for the required code

The following rules apply when required bindings are used with the [code](file:///C:\temp\datatypes.html#code) data type:

* Where the value set is defined by FHIR, the list of allowed codes will be fixed in the XML schema
* Comparison between codes is always case sensitive for codes unless the codes are selected by reference (e.g. ValueSet.compose), and the referenced specification clearly states otherwise
* The list of codes that can be used can only be extended in subsequent releases of the FHIR specification

When an element is bound to a required value set, [derived profiles](file:///C:\temp\profiling.html) may state rules on which codes can be used, but cannot select new or additional codes for these elements.

**Extensible**

*To be conformant, codes in this element SHALL be from the specified value set if any of the codes within the value set can apply to the concept being communicated. If the value set does not cover the concept (based on human review), an alternate system.code may be used instead.*

If the data type is [CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept), then one of the *coding* values SHALL be from the specified value set if a code applies, but if no suitable code exists in the value set, alternate code(s) may be provided in its place. If no codes, including local codes, are available, then just text may be used.

If the data type is [Coding](file:///C:\temp\datatypes.html#Coding), then the code/system SHALL be from the specified value set if a code applies, but if no suitable code exists in the value set, an alternate code may provided in its place.

Identified gaps in value sets should be submitted to the organization administering the value set in order to improve interoperability in the future.

Extensible bindings are used when there is consensus at the specification or profiling level about the coded values that should be used, but it is impossible to create a bounded list of codes that are known to cover all use cases, including ones that are yet to arise.

When an element is extensibly bound to value set, [derived profiles](file:///C:\temp\profiling.html) may state rules on which codes can be used, but cannot select new or additional codes for these elements unless no codes with appropriate meanings are found in the base value set.

**Preferred**

*Instances are encouraged to draw from the specified codes for interoperability purposes but are not required to do so to be considered conformant.*

If the data type is [CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept), then one of the *coding* values SHOULD be from the specified value set, but another code and/or text can be used in its place.

Preferred bindings are used when there is consensus at the specification level about the coded values that are the best to be used, but there is recognition that some implementation contexts are unable to use the recommended codes for a variety of reasons. Applications should consider adopting the preferred value set where ever possible, as these preferred value sets are the most likely to server interoperability purposes in the future.

When an element is bound to a preferred value set, [derived profiles](file:///C:\temp\profiling.html) may bind the element to any value set they choose.

**Example Bindings**

*Instances are not expected or even encouraged to draw from the specified value set. The value set merely provides examples of the types of concepts intended to be included.*

Example bindings are used when an element has a very broad meaning (such as [List](file:///C:\temp\list.html).code), or there is no consensus over the correct codes to be used. For these bindings:

* **Coding**: the *system*/*code* values MAY be one of the codes in the value set.
* **CodeableConcept**: one of the *coding* elements MAY contain a *system*/*code* that is in the value set.

Some other coded value MAY be used, or (for a CodeableConcept), a text alternative MAY be provided. Example value sets are provided to assist implementers to understand the correct use of an element. Value sets based on code systems such as SNOMED CT that have restrictive license terms will only be used as example bindings in the base FHIR specification, though implementation guides for particular jurisdictions may adopt value sets that require licenses.

When an element is bound to an example value set, [derived profiles](file:///C:\temp\profiling.html) may bind the element to any value set they choose.

**Other notes**

* Subsequent versions of FHIR may replace example value sets with preferred bindings if enough consensus emerges in the relevant sphere.
* Bindings to value sets provided as part of the specification are always specific to the version of the value set published with the specification. The value set may be sealed by defining a simple list of enumerated codes, or it may include codes by their properties, along with a non-version specific reference to an underlying code system, in which case the list of valid concepts may change over time.

**Binding String Values**

In a few special cases, humans customarily use codes directly for elements that have type "string". A typical case is codes for states, and there are several places where a URI must come from a set of controlled values. An element of type [string](file:///C:\temp\datatypes.html#string) or [uri](file:///C:\temp\datatypes.html#uri) can also be bound to a value set. When a string or URI is bound to a value set, the value property SHALL contain the code specified by the value set, and the system and display values are ignored.

**terminology-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Terminology Module**

**Introduction**

The Terminology Module provides an overview and guide to the FHIR resources, operations, coded data types and externally-defined standard and FHIR-defined terminologies that are used for representing and communicating coded, structured data in the FHIR core specification and profiles. Collectively, these capabilities are used to provide the terminology service functionality required for supporting the use of coded data in FHIR resources throughout the specification as described in the other modules.

The terminology resources and their relationships are shown below:

|  |
| --- |
|  |

The ElementDefinition type (shown with a dotted box) is described elsewhere in the specification in the [Foundation](file:///C:\temp\foundation-module.html) and [Conformance](file:///C:\temp\conformance-module.html) modules.

**Index**

The Terminology Module covers the following:

**Resources**

|  |  |  |
| --- | --- | --- |
| * [CodeSystem](file:///C:\temp\codesystem.html) * [ValueSet](file:///C:\temp\valueset.html) | * [ConceptMap](file:///C:\temp\conceptmap.html) * [NamingSystem](file:///C:\temp\namingsystem.html) | * [ExpansionProfile](file:///C:\temp\expansionprofile.html) |

**Terminology Service**

|  |
| --- |
| * [Terminology Service Documentation](file:///C:\temp\terminology-service.html) |

**Operations**

|  |  |  |
| --- | --- | --- |
| * **CodeSystem** * [$lookup](file:///C:\temp\codesystem-operations.html#lookup) * [$subsumes](file:///C:\temp\codesystem-operations.html#subsumes) * [$infer](file:///C:\temp\codesystem-operations.html#infer) | * **ValueSet** * [$expand](file:///C:\temp\valueset-operations.html#lookup) * [$validate-code](file:///C:\temp\valueset-operations.html#validate-code) | * **ConceptMap** * [$translate](file:///C:\temp\conceptmap-operations.html#lookup) * [$closure](file:///C:\temp\conceptmap-operations.html#lookup) |

**Coded Data Types**

|  |  |  |
| --- | --- | --- |
| * [code](file:///C:\temp\datatypes.html#code) | * [Coding](file:///C:\temp\datatypes.html#Coding) | * [CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept) |

**Documentation**

|  |  |
| --- | --- |
| * [Using Codes in Resources](file:///C:\temp\terminologies.html) * [Code Systems Defined/Referenced in FHIR](file:///C:\temp\terminologies-systems.html) * [Value Sets Defined in FHIR](file:///C:\temp\terminologies-valuesets.html) | * [Mappings (FHIR-defined) Between Related Value Set Concepts](file:///C:\temp\terminologies-conceptmaps.html) * [Known Identifier Systems](file:///C:\temp\identifier-registry.html) |

**Security and Privacy**

For security considerations for terminology services, see the [Terminology Service page Security section](file:///C:\temp\terminology-service.html#4.6.1). For more general considerations, see the [Security and Privacy module](file:///C:\temp\secpriv-module.html).

**Common Use Cases**

* Expand a value set
* Validate a code
* Look up a display term for a code
* Translate a code from one value set to another
* Maintain a client-side [transitive closure table](https://en.wikipedia.org/wiki/Transitive_closure#In_graph_theory) on subsumption relationships
* Test subsumption between concepts
* For a set of property/concept pairs return the set of concepts for the requested properties

**Developmental Roadmap**

A subset of the terminology resources have been tested and are being used in production tooling, and as such have reached a maturity level where changes become less likely. These are:

* [ValueSet](file:///C:\temp\valueset.html) [3](file:///C:\temp\resource.html#maturity)
* [ConceptMap](file:///C:\temp\conceptmap.html) [2](file:///C:\temp\resource.html#maturity)

Other resources are still under development:

* [NamingSystem](file:///C:\temp\namingsystem.html) [1](file:///C:\temp\resource.html#maturity)
* [CodeSystem](file:///C:\temp\codesystem.html) [0](file:///C:\temp\resource.html#maturity)
* [ExpansionProfile](file:///C:\temp\expansionprofile.html) [0](file:///C:\temp\resource.html#maturity)

Over the next 18 months, we will continue to advance the resources through the [Maturity Levels](file:///C:\temp\resource.html#maturity) through further development and testing. We anticipate more widespread implementation of NamingSystem, CodeSystem and ExpansionProfile and testing in Connectathons and with implementation experience.

**terminology-service.html**

**Terminology Service**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This specification includes support for the provision of a terminology service - that is, a service that lets healthcare applications make use of codes and value sets without having to become experts in the fine details of the code system, value set and concept map resources, and the underlying code systems and terminological principles. A server that supports all the functionality described here can be described as a "FHIR Terminology Service", and SHALL conform to the [terminology server Capability statement](file:///C:\temp\capabilitystatement-terminology-server.html). Note that servers can declare that they provide terminology services in their Capability statement:

<CapabilityStatement xmlns="http://hl7.org/fhir">

<!-- snip -->

<instantiates value="http://hl7.org/fhir/CapabilityStatement/terminology-server"/>

<!-- snip -->

</CapabilityStatement>

**Security**

Generally, SSL SHOULD be used for all production health care data exchange. Even though terminology servers do not generally handle patient information directly, observers may still be able to infer information about patients by observing the codes and concepts that the terminology service is asked about, so encryption is still recommended.

A terminology server may choose not to authenticate the clients/users in any fashion, but can do so in order to limit or account for usage. For a value set maintenance server that allows terminologies to be edited, some form of [authorization and/or authentication would be appropriate](file:///C:\temp\security.html). This specification does not require any particular approach to security.

**Basic Concepts**

A FHIR terminology service is simply a set of functions built on the definitions provided by a collection of [CodeSystem](file:///C:\temp\codesystem.html), [ValueSet](file:///C:\temp\valueset.html) and [ConceptMap](file:///C:\temp\conceptmap.html) resources, with additional inherently known terminologies providing support.

The terminology service builds on the basic principles for using terminologies in FHIR. Implementers should be familiar with:

* [Using codes in FHIR](file:///C:\temp\terminologies.html)
* The [CodeSystem](file:///C:\temp\codesystem.html) resource
* The [ValueSet](file:///C:\temp\valueset.html) resource
* The [ConceptMap](file:///C:\temp\conceptmap.html) resource

In addition, implementers should be familiar with the [operations framework](file:///C:\temp\operations.html). Further useful information about terminologies may be found in:

* Underlying Principles: [HL7 v3 Core Principles](http://www.hl7.org/documentcenter/public/standards/V3/core_principles/infrastructure/coreprinciples/v3modelcoreprinciples.html)
* SNOMED CT [technical documentation](http://ihtsdo.org/fileadmin/user_upload/doc/). Note: "namespace" is used differently here from the way it is used by IHTSDO (see [discussion here](file:///C:\temp\terminologies.html))

**External Code Systems**

In order to be used with a value set, code systems and their content must be defined somewhere. They can be defined explicitly using the [code system resource](file:///C:\temp\codesystem.html), or they can be defined elsewhere, and then used in a value set by quoting the correct system url. The FHIR specification defines a [set of namespaces](file:///C:\temp\terminologies-systems.html) for commonly encountered code systems, and defines how some work with FHIR (e.g. [SNOMED CT](file:///C:\temp\snomedct.html), [LOINC](file:///C:\temp\loinc.html), [RxNorm](file:///C:\temp\rxnorm.html)). These code systems are often large, and have many internally defined properties that are part of their formal definitions. The code systems resource is not an appropriate way to distribute the contents of these code systems; the standard FHIR code system resource simply represents the properties of the code system. Instead, these terminologies provide their own distribution formats, and it is assumed that the content of these code systems are externally known to the terminology server.

Most useful terminology servers will make one or more of these external code systems available for use within the value sets that they manage. The list of additional terminologies that a terminology server supports beyond those defined in its value sets is published to clients by referencing code system resources in the server's [Capability Statement](file:///C:\temp\capabilitystatement.html).

{

"resourceType" : "CapabilityStatement",

"extension" : [

{

"url" : "http://hl7.org/fhir/StructureDefinition/capabilitystatement-supported-system",

"valueUri" : "http://loinc.org"

}]

}

This extension is added to the root [Capability Statement](file:///C:\temp\capabilitystatement.html).

**Implementation Note**

When a terminology server exposes an external code system, it makes a set of services available internally that serve the operational interfaces below. The internal server depends on the following logical information for a terminology:

* its URL (namespace, and how versioning works)
* what codes are valid
* what properties can be used to select codes
* what implicit value sets exist

The FHIR specification itself defines these things for common terminologies (including [SNOMED CT](file:///C:\temp\snomedct.html), [LOINC](file:///C:\temp\loinc.html), [RxNorm](file:///C:\temp\rxnorm.html)), and provides the [CodeSysten](file:///C:\temp\codesystem.html) infrastructure for supporting typical relatively simple small code systems.

*Implementers Note: Implementers interested in working with existing published terminologies for which the CodeSystem infrastructure is not suitable should discuss their needs with HL7 to get the list above extended.*

Note: A terminology service may choose to expose additional external code system specific related functionality such as summation, or structured search, but these services are outside the scope of the FHIR terminology service.

**Operations across all value sets**

For some of the operations below, it can be useful to perform them across all value sets known to the system. For example, $expand using a text filter, and searching all value sets at once. A special value set is defined that means "all value sets known to the server":

http://hl7.org/fhir/ValueSet/@all

Technically, this value set automatically imports all the existing value sets on the server. Note that this URL has no fixed meaning - its interpretation is server specific (e.g. whether it includes all versions of all value sets). This URL can only be used as a parameter to the operations described on this page.

**Terminology Maintenance**

The terminology service uses the code systems and value set resources defined on the system - both the implicit ones associated with the external code systems and those explicitly available at the /CodeSystem and /ValueSet endpoints - to serve the operational interface defined below. As code systems and value sets are created, updated or deleted, the outcomes of the operational services change. A terminology server should validate incoming resources, and ensure integrity of the terminology services. Typically, servers provide a test and production environment, but there is no explicit notion of this in the interface itself.

**Value Set Expansion**

A value set describes a set of rules for what codes or concepts are considered to be in the value set. These rules might be simple (e.g. a direct list of codes from a specified version of a code system), or they might be quite complex (e.g. all codes with a particular property from an unspecified version of a code system).

A FHIR enabled application can simply ask a terminology server to figure out all the details, and return it a list of the current codes in the value set. This is known as ["expanding" the valueset](file:///C:\temp\valueset-operations.html#expand). As a summary, the client passes the server the following information:

* the value set (either by its URL on the RESTful interface, by its logical identifier [(ValueSet.url)](file:///C:\temp\valueset-definitions.html#ValueSet.url), or directly as a parameter to the call)
* (Optionally) a text filter to use to restrict the codes that are returned (e.g. user input text). It is left to server discretion to choose how to apply the text filter
* (Optionally) a date at which the expansion should be evaluated (usually, this is the current date/time, but there are circumstances where that is not appropriate)
* (Optionally) which page to retrieve - asking the server to break the expansion into a set of chunks
* (Optionally) a reference to an [ExpansionProfile](file:///C:\temp\expansionprofile.html) resource that supplies additional information about how to perform the expansion

The server returns a value set that contains the current list of codes that meet the filter criteria (or an [OperationOutcome](file:///C:\temp\operationoutcome.html) with an error if the expansion fails). Note that some value sets expand to many thousands of codes, or even an infinite number, and for these, the server SHOULD return an [error code *too-costly*](file:///C:\temp\valueset-issue-type.html#too-costly). In these cases the client can try again with a more specific text filter to reduce the number of codes returned - this may result in a valid expansion.

For further information, consult the [definition of the operation](file:///C:\temp\valueset-operations.html#expand).

The $expand operation has support for paging - for a client to retrieve a big expansion in a set of partial views, in order to present the most optimal user experience. The client specifies both an offset and a count - how many codes per page, and where in the sequence to start. The return expansion specifies the number of concepts in the expansion, and the offset at which this partial view starts. Note that all expansions SHOULD include the total code count, but the offset element SHALL only exist when paging is being used. Expansions that are hierarchical trees of concepts are not subject to paging and the server simply returns the entire expansion.

Some example uses for the expansion operation:

* get a list of codes to display in a User interface (e.g. a drop down interface)
* a variation on this is to offer the user a text box to type in. As the user types, call the expand operation to provide the user with a list of matching codes/concepts (like a browser search)
* fetch a list of codes to use when generating software programming instructions
* get a list of codes so that software can check whether a code is valid or not in a particular context

**Examples**

Expanding a value set that is already registered on the server as "23", with a text filter of "abdo":

GET [base]/ValueSet/23/$expand?filter=abdo

Expanding a value set that is specified by the client (using JSON):

POST [base]/ValueSet/$expand

[other headers]

{

"resourceType" : "Parameters",

"parameter" : [

{

"name" : "valueSet",

"resource" : {

"resourceType" : "ValueSet",

[value set details]

}

}

]

}

The server responds with a value set (this example in XML):

HTTP/1.1 200 OK

[other headers]

<ValueSet xmlns="http://hl7.org/fhir">

<!-- the server SHOULD populate the id with a newly created UUID

so clients can easily track a particular expansion -->

<id value="43770626-f685-4ba8-8d66-fb63e674c467"/>

<!-- no need for meta, though it is allowed for security labels, profiles -->

<!-- other value set details -->

<expansion>

<!-- when expanded -->

<timestamp value="20141203T08:50:00+11:00"/>

<contains>

<!-- expansion contents -->

</contains>

</expansion>

</ValueSet>

**Concept Lookup / Decomposition**

A system can ask a terminology server to return a set of information about a particular system/code combination using [the lookup operation](file:///C:\temp\codesystem-operations.html#lookup). The server returns information for both display and processing purposes. The client passes the server the following information:

* the code value (either a code, or a Coding data type)
* (Optionally) the id or the url of the code system in which the code is being checked
* (Optionally) a date at which the code information should be returned (usually, this is the current date/time, but there are circumstances where that is not appropriate)
* (Optionally) a set of properties to return about the code

The server returns some or all of the following information:

* a human description of the system
* a recommended display for the code
* properties of the code (e.g. status)
* other designations for the code (a value, optionally with language and/or a use code)
* relationships between this code and other codes (parent/child properties, etc)
* Component properties of the specified code (e.g. to support reasoning) (e.g. decomposition)

The recommended display for the code is a text representation of the code that the terminology server recommends as the default choice to show to the user, though a client may choose out of the other designations if it has reason to.

If the client does not ask for any particular properties to be returned, it is at the discretion of the server to decide which properties to return (though note that the "version" property is always returned if the code system has a version).

**Examples**

Looking up a code in a code system:

GET [base]/CodeSystem/loinc/$lookup?code=1963-8

Note that the logical id "loinc" is not a reliable identifer across systems; each server assigns logical ids to code system resources however it sees fit. A more reliable query is this:

GET [base]/CodeSystem/$lookup?system=http://loinc.org&code=1963-8&property=code&property=display&property=designations

Lookup the code system using a Coding (this example in XML):

POST [base]/CodeSystem/$lookup

[other headers]

<Parameters xmlns="http://hl7.org/fhir">

<parameter>

<name value="coding"/>

<valueCoding>

<system value="http://loinc.org"/>

<code value="1963-8"/>

</valueCoding>

</parameter>

</Parameters>

The server responds with a set of information (JSON this time):

HTTP/1.1 200 OK

[other headers]

{

"resourceType" : "Parameters",

"parameter" : [

{

"name" : "name",

"valueString" : "LOINC"

},

{

"name" : "version",

"valueString" : "2.56"

},

{

"name" : "display",

"valueString" : "Bicarbonate [Moles/volume] in Serum"

},

{

"name" : "abstract",

"valueString" : "false"

}

]

}

**Standard Properties**

The following properties are defined for all code systems:

|  |  |
| --- | --- |
| **Name** | **Usage** |
| system | The name of the code ssystem |
| version | The version of the code system used for the look up operation |
| display | The recommended display for the code, if one is known |
| definition | The definition for the code |
| designation | Other designations for the code |
| lang.X | Designations in language X (where X is an IETF Language code, see [BCP-47](http://tools.ietf.org/html/bcp47) |
| parent | Parent codes for this code (for code systems with a defined heirarchy) |
| child | child codes of this code (for code systems with a defined heirarchy) |

In addition, any property codes defined by the code system (CodeSysem.property.code) can be used (and see the definitions for [SNOMED CT](file:///C:\temp\snomedct.html#props), [LOINC](file:///C:\temp\loinc.html#props), and [RxNorm](file:///C:\temp\rxnorm.html#props)).

**Value Set Validation**

One way to determine whether a code is in a value set is to expand the value set (as described above), and then look at the returned codes to see if the code is in the expansion. However this is not an efficient way to test whether a code is valid, and for some value sets (e.g. with infinite number of members), it cannot work. Instead, a FHIR terminology server provides [a "validate-code" operation](file:///C:\temp\valueset-operations.html#validate-code). The client passes the server the following information:

* the value set (either by its URL on the RESTful interface, by its logical identifier [(ValueSet.url)](file:///C:\temp\valueset-definitions.html#ValueSet.url), or directly as a parameter to the call)
* the code value (either a code + system, a Coding data type, or a CodeableConcept
* (Optionally) a date at which the expansion should be evaluated (usually, and by default, this is the current date/time, but there are circumstances where that is not appropriate)

The server returns a true/false indicating whether the code/concept is valid, and a list of errors and warnings associated with it. The server should also return an appropriate display for the concept for use in a UI context.

Note that if the server is passed a CodeableConcept, the server is able to check whether any of the codes are valid against the value set, and also check whether multiple codings are allowed and/or the codings provided are consistent with each other.

Every code system has an implicit value set that is "all the concepts defined in the code system" (CodeSystem.valueSet). For some code systems, these value set URIs are defined in advance (e.g. for [LOINC](file:///C:\temp\loinc.html), it is http://loinc.org/vs). However for some code systems, they are not known. Clients can refer to these implicit value sets by providing the URI for the code system itself.

**Examples**

Simple validation of a code/system against a known value set:

GET [base]/ValueSet/23/$validate-code?system=http://loinc.org&code=1963-8&display=test

Validate a CodeableConcept against a client specified value set (this example in JSON):

POST [base]/ValueSet/$validate-code

[other headers]

{

"ResourceType" : "Parameters",

"parameter" : [

{

"name" : "coding",

"valueCodeableConcept" : {

"coding" : {

"system" : "http://loinc.org",

"code" : "1963-8",

"display" : "test"

}

}

},

{

"name" : "valueSet",

"resource": {

"resourceType" : "ValueSet",

[etc]

}

}

]

}

The server responds with validation information (JSON this time):

HTTP/1.1 200 OK

[other headers]

{

"resourceType" : "Parameters",

"parameter" : [

{

"name" : "result",

"valueBoolean" : "false"

},

{

"name" : "message",

"valueString" : "The display \"test\" is incorrect"

},

{

"name" : "display",

"valueString" : "Bicarbonate [Moles/volume] in Serum"

}

]

}

**Subsumption testing**

To test the subsumption relationship between *code/Coding A* and *code/Coding B*, perform a $subsumes operation. Subsumption testing is based on [the CodeSystem definition of subsumption](file:///C:\temp\codesystem.html#subsumption). The client passes the server the following information:

* the system that identifies the code system in which subsumption testing is to be performed (either by invoking the operation on the code system directly, or referring to it by it's canonical URL)
* Concepts A and B - either as codes, or Codings
* (Optionally) the version of the code system to use (mostly, this should not matter)

If the client passes Codings, it is allowed to use code system values that are different from the code system in which subsumption testing is to be performed. In this case, the server SHALL return an error unless the relationships between the various code systems is well defined.

If the concepts can be compared, then the server returns an outcome code:

|  |  |
| --- | --- |
| equivalent | Concepts A and B are equivalent |
| subsumes | Concept A subsumes Concept B |
| subsumed-by | Concept A is subsumed by Concept B |
| not-subsumed | Concepts A and B are not related by any subsumption relationship |

**Examples**

Test whether a SNOMED CT Concept 'Disorder of liver' (235856003) subsumes 'Viral hepatitis' (3738000):

GET [base]/CodeSystem/$subsumes?system=http://snomed.info/sct&codeA=235856003&codeB=3738000

Or using Codings:

POST [base]/CodeSystem/$subsumes

[other headers]

<Parameters xmlns="http://hl7.org/fhir">

<!-- Subsumption testing - use SNOMED CT rules -->

<parameter>

<name value="system"/>

<valueUri value="http://snomed.info/sct"/>

</parameter>

<!-- Australian distribution -->

<parameter>

<name value="version"/>

<valueString value="http://snomed.info/sct/32506021000036107/version/20160430"/>

</parameter>

<parameter>

<name value="codingA"/>

<valueCoding>

<system value="http://snomed.info/sct"/>

<code value="235856003"/>

</valueCoding>

</parameter>

<parameter>

<name value="codingB"/>

<valueCoding>

<system value="http://snomed.info/sct"/>

<code value="3738000"/>

</valueCoding>

</parameter>

</Parameters>

Server response:

HTTP/1.1 200 OK

[other headers]

{

"resourceType" : "Parameters",

"parameter" : [

{

"name" : "outcome",

"valueCode" : "subsumes"

},

]

}

**Batch Validation**

It is also possible to validate a set of concepts against their relevant value sets by using the $validate-code operation in a [Batch](file:///C:\temp\http.html#batch) interaction.

**Example**

A request to validate 2 concepts from a [CDA](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7) document, with OIDs for value set identifiers:

POST [base]

[other headers]

{

"ResourceType": "Bundle",

"type": "batch",

"entry": [{

"request": {

"method": "Get",

"url": "ValueSet/$validate-code?system=http://loinc.org&code=2324-4&uri=urn:oid:1.2.3.4.6"

}

},

{

"request": {

"method": "GET",

"url": "ValueSet/$validate-code?system=http://snomed.info/sct&codes=22298006&uri=urn:oid:1.2.3.4.7"

}

}]

}

The server responds with a series of validation outcomes (JSON this time):

HTTP/1.1 200 OK

[other headers]

{

"ResourceType": "Bundle",

"type": "batch-response",

"entry": [{

"resource": {

"resourceType": "Parameters",

"parameter": [{

"name": "result",

"valueBoolean": "false"

},

{

"name": "message",

"valueString": "'2324-4' is not a valid LOINC code"

}]

}

},

{

"resource": {

"resourceType": "Parameters",

"parameter": [{

"name": "result",

"valueBoolean": "false"

},

{

"name": "message",

"valueString": "The concept is not in the specified value set (\"Organisms\")"

},

{

"name": "display",

"valueString": "Myocardial infarction"

}]

}]

}

**Translations**

A client can ask a server to translate a concept from one value set to another. Typically, this is used to translate between code systems (e.g. from LOINC to SNOMED CT, or from a [HL7 V3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) code to a [HL7 V2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185) code). The client calls [the translate operation](file:///C:\temp\conceptmap-operations.html#translate) and passes the following parameters:

* a code + system, Coding, or CodeableConcept
* a Concept Map to use for the translation
* the value set for the context of the source
* the value set for the destination

The client passes either a concept map, or the value sets for the source and destination context. If there is no concept map, then the server may determine the appropriate map to use from context provided in the value sets. If there is no particular context, the appropriate value sets would be the value sets for the entire coding system at question (e.g. from http://snomed.info/sct to http://loinc.org/vs). The server performs the translation as it is able based on the concept maps that it knows about. If no single mapping can be determined then the server returns an error. Some servers may require a concept map to use for the translation.

**Example**

Translate from FHIR Composition status to [HL7 v3](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186) Act Status (based on [this defined concept map](file:///C:\temp\cm-composition-status-v3.html):

GET [base]/ConceptMap/$translate?system=http://hl7.org/fhir/composition-status

&code=preliminary&source= http://hl7.org/fhir/ValueSet/composition-status

&target=http://hl7.org/fhir/ValueSet/v3-ActStatus

The server responds with validation information:

HTTP/1.1 200 OK

[other headers]

{

"resourceType" : "Parameters",

"parameter" : [

{

"name" : "result",

"valueBoolean" : "true"

},

{

"name" : "outcome",

"valueCoding" : {

"system" : "http://hl7.org/fhir/v3/ActStatus",

"code" : "active",

}

}

]

}

**Batch Translation**

It is also possible to translate a set of concepts against their relevant value sets by using the $translate operation in a [Batch](file:///C:\temp\http.html#batch) interaction.

**Example**

A request to translate 2 concepts from a [CDA](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7) document, with OIDs for value set identifiers:

POST [base]

[other headers]

{

"ResourceType": "Bundle",

"type": "batch",

"entry": [{

"resource": {

"ResourceType": "Parameters",

"parameter": [{

"name": "concept",

"valueCodeableConcept": {

"system": "http://loinc.org",

"code": "2324-4"

}

},

{

"name": "target",

"valueUri": "urn:oid:1.2.3.4.6"

}]

},

"request": {

"method": "POST",

"url": "ConceptMap/$translate"

}

},

{

"resource": {

"ResourceType": "Parameters",

"parameter": [{

"name": "concept",

"valueCodeableConcept": {

"system": "http://snomed.info/sct",

"code": "22298006"

}

},

{

"name": "target",

"valueUri": "urn:oid:1.2.3.4.7"

}]

},

{

"request": {

"method": "POST",

"url": "ConceptMap/$translate"

}

}]

}

The server responds with a series of translation outcomes:

HTTP/1.1 200 OK

[other headers]

{

"ResourceType": "Bundle",

"type": "batch-response",

"entry": [{

"resource": {

"resourceType": "Parameters",

"parameter": [{

"name": "result",

"valueBoolean": "false"

},

{

"name": "message",

"valueString": "'2324-4' is not a valid LOINC code"

}]

}

},

{

"resource": {

"resourceType": "Parameters",

"parameter": [{

"name": "result",

"valueBoolean": "true"

},

{

"name": "outcome",

"valueCodeableConcept": {

"coding": {

"system": "http://example.com/codesystems/example",

"code": "xxxx"

}

}

}]

}

}]

}

**Maintaining a Closure Table**

The 5 operations Expand, Lookup, Validate, Subsumes, and Translate account for most operational requirements associated with terminology use. However there is one difficult but important use case that they do not address, which is integrating terminologically based logic into application searches.

A typical example of this is a user that wants to find any observations for male patients over the age of 50 who attended a particular clinic within a particular 2 week period, with a diagnosis of gout, and who had an elevated serum creatinine.

In this case, both "diagnosis of gout" and "serum creatinine" involve value set and/or subsumption queries (e.g. against SNOMED CT and LOINC respectively). This search has to be executed by some logical processing engine that knows how to find patient related in a given persistence store. Often, this is some kind of SQL query, though many other technological choices are available. However this is done, the challenge with an operation like this is to integrate the terminological knowledge into a search execution that also covers other relationships expressed in the search criteria.

One approach to this problem would be to using the expand operation above, so that the system executing the search could generate expansions, and then search for these expansions. This has a couple of problems:

* the list of subsumed codes could be very long, and the search operation becomes correspondingly inefficient
* the expansion of the subsumption might not be closed, and so the search operation cannot be correct

An alternative approach is to generate a [transitive closure table](https://en.wikipedia.org/wiki/Transitive_closure#In_graph_theory) which lists all the possible transitive subsumption relationships, and [allows for rapid execution of these kind of queries](http://karwin.blogspot.com.au/2010/03/rendering-trees-with-closure-tables.html) . However this has other problems:

* the subsumption table can be very large (>500000 records for SNOMED CT), even though very few of the codes are used
* subsumption tables are generally built up front, and do not deal with new codes as they are encountered very well
* they still do not offer a solution for non-closed expansions

This is the main reason why most systems do not support post-coordination or other forms of coded expressions.

In FHIR, this problem is solved by building a closure table on the fly as new codes are seen. This technique leaves the FHIR terminology server responsible for the terminological reasoning, and the client responsible for the closure table maintenance. To the client, it doesn't matter whether the concept is post-coordinated or not. Here's a description of how the process works:

1. the client defines a name associated with a particular context in which it wishes to maintain a subsumption based closure table
2. the client registers this name with the FHIR Terminology server using the $closure operation (described below), with only one parameter, the name of the context
3. any time the client system encounters a new Coding that is not entered in the closure table, it calls the $closure operation with the context name, and the Coding value it has encountered
4. the server returns a ConceptMap resource with a list of new entries (code : system -> code : system) that the client should add to its closure table
   * the server can indicate that entries should be removed from the table by providing a (code : system -> code : system) with equivalence "unmatched" (though it's not known why that would be needed)
5. the client makes these entries into its closure table
6. to facilitate the initialization process, a client can call $closure with multiple Coding values

The [$closure operation](file:///C:\temp\conceptmap-operations.html#closure) takes 2 parameters:

* closure table context name
* concepts to enter into the table (0 or more - 0 codings is a request to (re-)initialise the table)

The operation returns a concept map which has a list of mappings that represent new entries to make in the closure table. The subsumption testing performed when building a closure table is the same as for the $subsumes operation, and is based on [the CodeSystem definition of subsumption](file:///C:\temp\codesystem.html#subsumption).

The closure table can be resynchronized by passing an additional "version" parameter, which is a value taken from the version in one of the delta responses. This is a request to replay all the mapping changes since that delta was sent.

**Initializing a Closure Table**

Before it can be used, a closure table has to be initialised. To initialise a closure table, POST the following to [base]/ConceptMap/$closure:

{

"resourceType" : "Parameters",

"parameter" : [{

"name" : "name",

"valueString" : "[name]"

}]

}

A successful response is a 200 OK from the server, with an associated ConceptMap:

{

"resourceType": "ConceptMap",

"id": "[name]",

"version": "0",

"name": "Closure Table [name] Creation",

"status": "active",

"experimental": true,

"date": "2015-12-20T23:10:55Z"

}

If there is an error Ã¢â‚¬â€œ usually involving the closure name, the server returns a HTTP status 400 with an operation outcome:

{

"resourceType": "OperationOutcome",

"text": {

"status": "generated",

"div": "<div xmlns=\"http://www.w3.org/1999/xhtml\"><p>invalid closure name \"invalid-id!\":</p></div>"

},

"issue": [

{

"severity": "error",

"details": {

"text" : "invalid closure name \"invalid-id!\""

}

}

]

}

What closure names are valid is at the discretion of the server.

**Adding to a Closure Table**

When the consumer (client) encounters a new code, it POSTs the following to [base]/ConceptMap/$closure:

{

"resourceType" : "Parameters",

"parameter" : [{

"name" : "name",

"valueString" : "[name]"

}, {

"name" : "concept",

"valueCoding" : {

"system" : "http://snomed.info/sct",

"code" : "22298006",

"display" : "Myocardial infarction"

}

}]

}

Note that this example only includes one concept, but more than one is allowed:

{

"resourceType" : "Parameters",

"parameter" : [{

"name" : "name",

"valueString" : "[name]"

}, {

"name" : "concept",

"valueCoding" : {

"system" : "http://snomed.info/sct",

"code" : "22298006",

"display" : "Myocardial infarction"

}

}, {

"name" : "concept",

"valueCoding" : {

"system" : "http://snomed.info/sct",

"code" : "128599005",

"display" : "Structural disorder of heart"

}

}]

}

The response varies depending on the conditions on the server. Possible responses: If the closure table has not been initialised: Return a 404 Not Found with

{

"resourceType": "OperationOutcome",

"text": {

"status": "generated",

"div": "<div xmlns=\"http://www.w3.org/1999/xhtml\"><p>invalid closure name \"[name]\":</p></div>"

},

"issue": [

{

"severity": "error",

"details": {

"text" : "invalid closure name \"[name]\""

}

}

]

}

If the closure table needs to be reinitialised: Return a 422 Unprocessable Entity with

{

"resourceType": "OperationOutcome",

"text": {

"status": "generated",

"div": "<div xmlns=\"http://www.w3.org/1999/xhtml\"><p>closure \"[name\" must be reinitialised</p></div>"

},

"issue": [{

"severity": "error",

"details": {

"text" : "closure \"[name]\" must be reinitialised"

}

}

]

}

The server should only send this when itÃ¢â‚¬â„¢s underlying terminology conditions have been changed (e.g. a new version of SNOMED CT has been loaded). When a client gets this, itÃ¢â‚¬â„¢s only choice is to initialise the closure table, and process all the codes in the closure table again (the assumption here is that the system has some external source of Ã¢â‚¬Ëœall the codesÃ¢â‚¬â„¢ so it can rebuild the table again). If the concept(s) submitted are processed ok, but thereÃ¢â‚¬â„¢s no new concepts, or no new entries in the table, return a 200 OK with :

{

"resourceType": "ConceptMap",

"id": "[name]",

"version": "[version]",

"name": "Updates for Closure Table [name]",

"status": "active",

"experimental": true,

"date": "2015-12-20T23:12:55Z"

}

If thereÃ¢â‚¬â„¢s new entries in the closure table, the server returns a 200 OK with:

{

"resourceType": "ConceptMap",

"id": "b87db127-9996-4d0c-bda9-a278d7a24a69",

"version": "[version]",

"name": "Updates for Closure Table [name]",

"status": "active",

"experimental": true,

"date": "2015-12-20T23:16:24Z",

"element": [{

"codeSystem": "http://snomed.info/sct",

"code": "22298006",

"target": [{

"codeSystem": "http://snomed.info/sct",

"code": "128599005",

"equivalence": "subsumes"

}]

}]

}

Notes

* The server can return multiple elements, each with 1 or more targets
* servers may return the relationship represented in either direction
* itÃ¢â‚¬â„¢s important to understand the relationship the right way around. From the spec: The equivalence is read from target to source (e.g. the target is Ã¢â‚¬ËœwiderÃ¢â‚¬â„¢ than the source). So in this case, 128599005 (Structural disorder of heart) subsumes 22298006 (Myocardial infarction)
* In the $closure operation, the response never explicitly states that a code is subsumed by itself. Clients should assume that this is implicit
* The version is important. Each new invocation of the $closure operation returns a new version of the concept map. The server must keep track of the versions is has issued for replay (see below)
* As well as entering codes that are actually used, the client also enters search terms into the closure table
* The combination of the system and code is the key to the closure table; if the server encounters two different codes that have the same meaning (e.g. syntactical variation), it should create an "equals" relationship between them

**Re-running Closure operation**

Given the way that the closure operation functions, itÃ¢â‚¬â„¢s possible for a client to lose a response from the server before it is committed to safe storage (or the client may not have particularly safe storage). For this reason, when a client is starting up, it should check that there has been no missing operations. It can do this by passing the last version (from the Concept Map response) it is sure it processed in the request:

{

"resourceType" : "Parameters",

"parameter" : [{

"name" : "name",

"valueString" : "[name]"

}, {

"name" : "version",

"valueString" : "3"

}]

}

ThatÃ¢â‚¬â„¢s a request to return all the additions to the closure table since version 3. The server returns its latest version in the concept map, along with anything added to the closure table since version 3 (not including version 3)

Notes:

* The client can pass a concept or version, but not both
* These examples (and my server) use a serially incrementing sequential integer, but this is not required, and clients should not assume that there is any meaning or order in the version. Just recall the last version, and treat it as a magic value. There is, however, one special value: Ã¢â‚¬Ëœ0Ã¢â‚¬â„¢. Passing a last version of 0 should be understood as resyncing the entire closure table

**Making use of the Closure Table**

The client uses the result of the closure operation to maintain a closure table. Simplistically, it might look like this:

|  |  |  |  |
| --- | --- | --- | --- |
| **Scope** | **Source** | **Target** |  |
| patient-problems | http://snomed.info/sct|22298006 | http://snomed.info/sct|128599005 |  |
| patient-problems | http://snomed.info/sct|24595009 | http://snomed.info/sct|90560007 |  |
| obs-code | http://loinc.org|14682-9 | http://loinc.org|LP41281-4 |  |

The client can then use a table like this as part of its general search conditions. Using the example from above: "Find any observations for male patients over the age of 50 who attended a particular clinic within a particular 2 week period, with a diagnosis of gout, and who had an elevated serum creatinine." This query could be done, for instance, with an SQL query like this:

Select \* from Observations, Patients, Encounters, Conditions, Observations as Obs2 where

Observations.patient = Patients.Key and Patients.Age > 50 and

Observations.encounter = Encounters.Key and Encounter.clinic = [key]

and encounter.date >= [date] and encounter.date <= [date] and

Conditions.patient = Patients.Key and Conditions.code

in (select Source From ClosureTable

where Scope = "patient-problems" and Target = "http://snomed.info/sct|90560007") and

Obs2.patient = Patients.Key and Obs2.value > 0.19 and Obs2.code

in (select Source From ClosureTable

where Scope = "obs-code" and Target = "http://loinc.org|LP41281-4")

Note that in real clinical systems, tables are usually far more structured than this example implies, and the query is correspondingly more complex. The closure table would usually be normalised - this example is kept simple to demonstrate the concept.

**testing.html**

**Testing FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [Draft](file:///C:\temp\history.html#pubs) |

The FHIR specification describes a set of [resources](file:///C:\temp\resource.html), and several different frameworks for exchanging resources between different systems. Because of its general nature and wide applicability, the rules made in this specification are generally fairly loose. As a consequence and in order to insure interoperability between applications claiming conformance to this specification, a testing framework has been established within the FHIR specification. To this end, the [TestScript](file:///C:\temp\testscript.html) resource provides an implementation agnostic description of tests that allows test execution engines to evaluate if a FHIR implementation conforms with the FHIR specification. Providing a clear and concise test methodology for the FHIR specification through the TestScript resource helps to enable interoperability among various FHIR server and client implementations.

Furthermore, the TestScript resource provides clear examples of the appropriate use of the FHIR specification through test-based documentation. The TestScript resource stands as a form of executable documentation allowing developers to examine the operations defined by the tests in order to understand how various RESTful API interactions and resources should be used in coordination. The tests can also be automatically executed against systems under development to determine how well the systems adhere to the specification.

The TestScript resource contains:

* Name and description detailing the purpose of the test suite
* Links describing how the test suite relates to the FHIR specification
* A list of server interactions required to execute the test suite
* A list of server interactions that the test suite validates the correctness of
* The fixtures (required data or resources) the tests use during execution
* A set of operations to set up the test suite environment
* A list of tests each containing
  + Name and description of the test
  + Links describing how the test relates to the FHIR specification
  + A list of server interactions required to execute the test
  + A list of server interactions that the test validates the correctness of
  + A list of operations that provide the execution logic of the test
  + A list of assertions that provide the verification logic of the test
* A set of operations to tear down the test environment

**Execution**

**Workflow**

**Pre-Processing**

The TestScript execution workflow begins by determining if the test suite is appropriate for the server under test. This can be determined by evaluating if the interactions listed in the TestScript metadata "capabilities" section are supported by the server's conformance resource. If the capabilities are supported by the server, then the TestScript can be executed. Otherwise, a test engine will provide a comparison of the system under testâ€™s conformance capabilities and allow a TestScript as a whole to be skipped based on this comparison.. See [How to specify metadata capabilities](file:///C:\temp\main-pages.html#howToSpecifiyCapabilities).

If the server supports the requirements of the TestScript instance, any specified fixtures are loaded or retrieved. If the fixtures are marked as 'autocreate' then they are automatically created on the server(s) under test using 'create' operations. If any of the autocreate operations fail, then the tests in the TestScript are skipped.

**Setup Execution**

After the fixtures are loaded and autocreates are executed, then the setup section is executed to establish the testing environment. The purpose of the setup section is typically to pre-load data (if it was not autocreated) or delete data required for the execution of the tests against the FHIR system(s) under test. The setup operations are executed once before all the tests are run (see [Operation Execution](file:///C:\temp\main-pages.html#operation-execution)). All operations in a setup section (including assertions) must complete successfully for the subsequent tests to be executed. If an assertion operation in the setup section fails, then execution and evaluation of the tests in the TestScript should be skipped. Technically, any operation (see the [operations table](file:///C:\temp\main-pages.html#operations-table) for a complete listing) can be included in the setup section, but typical operations will be create, update, read, and vread.

Due to the possibility that the setup actions are not required on the server under test, the TestScript execution workflow MAY provide the capability of skipping or ignoring the setup section of the TestScript.

**Test Execution**

Once setup is complete, each test is executed. Tests contain a set of operations, and executing a test involves the evaluation of each operation listed in the test in the order defined by the test (see [Operation Execution](file:///C:\temp\main-pages.html#operation-execution) and the [list of operations](file:///C:\temp\main-pages.html#operations-table)).

**Teardown Execution**

After all the tests have completed execution, the teardown section is executed. The purpose of the teardown section is to revert the FHIR server under test to a pre-test clean state. This requires removing any resources or artifacts generated as part of test suite setup or test execution. Technically, any operation (see the [operations table](file:///C:\temp\main-pages.html#operations-table) for a complete listing) can be included in the teardown section, but the most often used operation will be delete. Assertions are not supported in the teardown section.

Due to the possibility that the teardown actions are not required on the server under test, the TestScript execution workflow MAY provide the capability of skipping or ignoring the teardown section of the TestScript.

**Post-Processing**

After the teardown section is executed, any fixtures that were marked 'autodelete' are removed from the server(s) under test. After this final stage, the execution of the TestScript is complete.

**Fixtures**

The fixtures section of the TestScript defines a set of resource instances that will be used as part of the setup, test, and teardown sections during TestScript execution. All defined fixtures are expected to be required in order for the test script to execute. Each fixture defines a resource instance by URI, and must be identified by an ID. The URI can be local or remote (i.e. another server than the one the TestScript resource resides), absolute or relative. The ID on the fixture is considered the "source" identifier of the fixture -- it is not the same thing as the resource ID on the server where it was hosted. The "source" identifier is used to define the fixture instance within the context of the TestScript. Operations reference the ID of a fixture to uniquely identify the fixture instance the operation is using ("sourceId") or acting against ("targetId"). Once a fixture has been instantiated on a server (typically by the use of a create operation), the fixture ID is mapped to the ID of the corresponding resource instance on the server. TestScript execution engines must maintain this relationship between fixture IDs and server resource IDs. The TestScript execution engine is responsible for translating the fixture IDs (whether provided to the operation as "source" or "target") to the ID of the resource on the server during execution.

Using the optional "autocreate" and "autodelete" elements (missing values default to false), fixtures can be configured to automatically be created during TestScript setup and automatically deleted during TestScript teardown. This means that additional "create" and "delete" operations in the TestScript.setup and TestScript.teardown sections are unnecessary.

**Profiles**

The profiles section of the TestScript defines a set of FHIR profiles (see [StructureDefinition](file:///C:\temp\structuredefinition.html) resource) that will be used as part of the setup or test sections during TestScript execution. Each profile defines a StructureDefinition instance by URI, and must be identified by an ID. The URI MUST reference a known FHIR profile corresponding to the FHIR specification version being tested. As with Fixtures, the ID on the profile is considered the "source" identifier of the profile. Operations reference the ID of a profile to uniquely identify the profile instance the operation is using ("validateProfileId").

See [Use Profiles](file:///C:\temp\main-pages.html#howToUseProfiles) for more information.

**Variables**

The variables section of the TestScript defines a set of expressions whose evaluations will be used in substitutions. These substitutions are performed in operation request headers and URL paths as well as assertion values.

Without variables, search parameters and request headers (such as If-Modified-Since) would be specified in outgoing requests as literal values. Variables allow the values to be managed externally in fixtures or dynamically in server response fixtures. They would be defined to hold path expressions against the fixtures. The path expressions would not change from one server to another but the fixture data could.

Using variables allows for the same test scripts to be executed against the same servers by different clients at the same time. Each client would change the fixture data (external to the test script) to make the data unique to that client. This ensures that the same delete/create/read operations executed concurrently by one client does not interfere with those of a another client. That can be important within the context of a testing event such as a Connectathon. It can be very useful in year-round testing against public servers as well.

See [Use Variables](file:///C:\temp\main-pages.html#howToUseVariables) for more information.

**Operation Execution**

The setup, test, and teardown sections of a TestScript can contain operation elements. Operations are intended to be executed in sequence and they represent the logic defined by the TestScript. Operations define a type, sourceId, targetid, destination, responseId, contentType, and other parameters. The type of the operation aligns with a corresponding RESTful API interaction. The sourceId and targetId of an operation define the context of the fixture data the operation is acting against (see [Fixtures](file:///C:\temp\main-pages.html#fixtures)). The destination defines the server the operation is executed on and is only required for tests that include multiple servers. The responseId specifies a fixture ID to use to map to the server response. The contentType defines the format (XML or JSON) and the corresponding mime-type (application/fhir+xml and application/fhir+json) to use in the RESTful operation (defaulting to XML). The parameters of an operation allow providing additional data required for execution.

TestScript execution engines must load the operation details and execute the operation against the server(s) under test. Operations that are expected to result in an error or exception, shall immediately be followed by one or more "assertion" operations (that test for those error conditions), otherwise the test fails. This allows for "negative" testing (for example, the test script may perform operations that should return a 4XX or 5XX HTTP response code). Test execution is halted as soon as an operation or assertion fails. If an operation or assertion fails, then the test ends in failure and the test script execution proceeds to the next test. Once all tests have completed execution, the teardown section is executed. Once teardown completes, the suite execution is complete. If any setup or test operation or assertion failed, the test script is marked in failure. Failures in teardown are ignored.

**Assertion Execution**

The "assertion" evaluates the results of previous operations to determine if the server under test behaves appropriately. In order to evaluate an assertion, the request, response, and results of the most recently executed operation must always be maintained by the TestScript execution engine.

The TestScript execution engine must implement the behavior of each assertion to determine if the response of the last operation meets the conditions of the assertion.

If the conditions of the assertion are met execution of the test continues. If the conditions of the assertion are not met, the test being executed fails and execution of the test is halted.

**How Tos**

**Test create operation**

To test if [create](file:///C:\temp\http.html#create) operation is properly supported on a server, run the operation as part of [TestScript.test](file:///C:\temp\testscript-definitions.html#TestScript.test).

First, define the [fixture](file:///C:\temp\testscript-definitions.html#TestScript.fixture) as a [reference](file:///C:\temp\references.html) at the top of the test script. The fixture will hold the body of the POST request:

<fixture id="example-patient">

<resource>

<reference value="Patient/patient-example.xml"/>

</resource>

</fixture>

Note that it is illegal for the fixture to contain a resource id in a create operation.

Point the [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.sourceId) element of the create operation to the fixture id just defined:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="create"/>

</type>

<sourceId value="example-patient"/>

</operation>

</action>

There are two ways to verify that the create operation returned the right status code:

1. Use [assert.response](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.response):
2. <action>
3. <assert>
4. <response value="created" />
5. </assert>
6. </action>

See [response codes](file:///C:\temp\valueset-assert-response-code-types.html) for complete list.

1. Use [assert.responseCode](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.responseCode) explicitly:
2. <action>
3. <assert>
4. <responseCode value="201" />
5. </assert>
6. </action>

**Test search operation**

To test if [search](file:///C:\temp\http.html#search) operation is properly supported on a server, run the operation as part of [TestScript.test](file:///C:\temp\testscript-definitions.html#TestScript.test).

Use the [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) element to specify the resource type and the [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element to specify the search parameters:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="search"/>

</type>

<resource value="Patient" />

<contentType value="json" />

<params value="?given=John&amp;family=Doe" />

<responseId value="R1" />

</operation>

</action>

The [contentType](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.contentType) element is optional and will default to "xml" which will translate to HTTP request header "Content-Type" being set to "application/fhir+xml" by test engines. In this case, though, it was used to set it to "application/fhir+json".

The [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId) element was used to store the response in a reference called "R1". This reference will hold both the response headers and the response body.

Verify that the search operation returned the right status code:

<action>

<assert>

<response value="okay" />

</assert>

</action>

See [response codes](file:///C:\temp\valueset-assert-response-code-types.html) for complete list.

Verify that the search operation returned the right resource type:

<action>

<assert>

<resource value="Patient" />

</assert>

</action>

There are many ways to verify that the search operation returned the right Patient:

1. Explicitly compare the elements to known value:
2. <action>
3. <assert>
4. <path value="fhir:Patient/fhir:birthDate/@value"/>
5. <sourceId value="R1"/>
6. <value value="1974-12-31" />
7. </assert>
8. </action>

The [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.sourceId) element is pointed to the [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId) value of the search operation. If no [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.sourceId) is specified, then test engines will use the response of the last operation in the test script even if [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId) was not specified in the operation.

The [path](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.path) element holds an XPath or JSONPath expression against the response body contents.

1. Compare the elements in response to elements in another fixture that is either dynamically set by [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId) or defined statically by the [fixture](file:///C:\temp\testscript-definitions.html#TestScript.fixture) element at the top of the script:
2. <action>
3. <assert>
4. <compareToSourceId value="F1"/>
5. <compareToSourcePath value="fhir:Patient/fhir:birthDate/@value"/>
6. <path value="fhir:Patient/fhir:birthDate/@value"/>
7. </assert>
8. </action>

This time the birthDate value in the response is compared to the birthDate value in a fixture called 'F1'.

1. Verify that the response contains all the element/content in another fixture pointed to by [minimumId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.minimumId).
2. <action>
3. <assert>
4. <minimumId value="F1"/>
5. <sourceId value="R1"/>
6. </assert>
7. </action>

Test engines will parse the 'body' of the F1 fixture and verify that each element and its value matches the corresponding element in the R1 response body. In other words, R1 is verified to be a 'superset' of F1. The resource id element in the body will be ignored during comparison. The headers will also be ignored.

F1 can be statically defined or it can be the [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId) for another operation. If [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.sourceId) is not specified, then test engines will use the response of the last operation. So the previous assertion could have been defined as:

<action>

<assert>

<minimumId value="F1"/>

</assert>

</action>

**Perform delete operation in teardown**

Test scripts should clean up resources created as part of execution. The [TestScript.teardown](file:///C:\temp\testscript-definitions.html#TestScript.teardown) operations will get executed once before the test script execution completes.

Here are a couple of ways to run delete operation in [TestScript.teardown](file:///C:\temp\testscript-definitions.html#TestScript.teardown):

1. Use [conditional delete](file:///C:\temp\http.html#2.1.0.12.1) operation in [TestScript.teardown](file:///C:\temp\testscript-definitions.html#TestScript.teardown):
2. <action>
3. <operation>
4. <type>
5. <system value="http://hl7.org/fhir/testscript-operation-codes"/>
6. <code value="delete"/>
7. </type>
8. <resource value="Patient" />
9. <params value="?given=John&amp;family=Doe" />
10. </operation>
11. </action>
12. Use delete operation with targetId fixture.

To do that, the resource must have been created during [TestScript.setup](file:///C:\temp\testscript-definitions.html#TestScript.setup) or [TestScript.test](file:///C:\temp\testscript-definitions.html#TestScript.test):

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="create"/>

</type>

<responseId value="create-response"/>

<sourceId value="example-patient"/>

</operation>

</action>

As part of [TestScript.teardown](file:///C:\temp\testscript-definitions.html#TestScript.teardown), run the delete operation with [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) value pointed to [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.sourceId) value of the create operation:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="delete"/>

</type>

<targetId value="create-response"/>

</operation>

</action>

Test engines will keep track of response headers and body of all operations.

The delete operation's targetId value is expected to correspond to the responseId of a GET operation (such as search or read) or the sourceId of a POST/PUT operation (such as create).

For targetId value corresponding to responseId of GET operations (such as search or read), test engines will use the resource type and id returned in the GET response body's resource to set the [type] and [id] in delete operation's URL, respectively.

For targetId value corresponding to responseId of POST/PUT operations (such as create), test engines will use the resource type and id returned in the POST/PUT response "Location" header to set the [type] and [id] in delete operation's URL, respectively. This is the case in the example above.

The targetId value cannot point to a statically defined fixture as the id in the fixture cannot be relied upon.

**Perform delete operation in setup**

Deletion of resources created during test script execution should be done using [delete](file:///C:\temp\http.html#delete) operation in [TestScript.teardown](file:///C:\temp\testscript-definitions.html#TestScript.teardown). See [Perform delete operation in teardown](file:///C:\temp\main-pages.html#howToDeleteResourceInTeardown) for details.

There might be left-over resource instances though on the server from prior executions of the script that terminated prematurely through an error. Resources can be deleted in [TestScript.setup](file:///C:\temp\http.html#setup) as well to ensure reliable test execution.

To delete a resource in setup, the server is required to support [Conditional Delete](file:///C:\temp\http.html#2.1.0.12.1) operation.

Use the [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element to specify the search criteria for the delete operation:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="delete"/>

</type>

<resource value="Patient" />

<params value="?family=Doe&amp;given=Joe"/>

</operation>

</action>

Test engines will append the contents of the params element to url after [type]: "[base]/[type]**?[search parameters]"**. The [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) element value ("Patient") will be used to replace [type] in the url.

**Test conditional delete operation**

To test if a server supports [conditional delete](file:///C:\temp\http.html#2.1.0.12.1) operation, run a create operation prior to the delete using a sourceId that points to a [fixture](file:///C:\temp\testscript-definitions.html#TestScript.fixture) defined at the top of the script:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="create"/>

</type>

<sourceId value="example-patient"/>

</operation>

</action>

Then use the [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element to specify the search criteria for the delete operation:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="delete"/>

</type>

<resource value="Patient" />

<params value="?family=Doe&amp;given=Joe"/>

</operation>

</action>

Test engines will append the contents of the [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element to url after [type]: "[base]/[type]**?[search parameters]"**. The [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) element value ("Patient") will be used to replace [type] in the url.

To verify that the delete operation returned the right status code:

<action>

<assert>

<operator value="in" />

<responseCode value="200,204" />

</assert>

</action>

To verify that the resource was indeed deleted on the server, run a search using the same parameters and verify that the status code is 404 (not found):

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="search"/>

</type>

<resource value="Patient" />

<params value="?given=John&amp;family=Doe" />

</operation>

</action>

<action>

<assert>

<response value="notFound" />

</assert>

</action>

**Test conditional create operation**

To test if a server supports [conditional create](file:///C:\temp\http.html#2.1.0.13.1) operation, use the 'If-None-Exist' request header:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="create"/>

</type>

<requestHeader>

<field value="If-None-Exist" />

<value value="Patient?given=John&amp;Doe&amp;birthdate=1974-12-31" />

</requestHeader>

<sourceId value="F1" />

</operation>

</action>

The response code of 200 verifies that the resource already exists and did not get created:

<action>

<assert>

<responseCode value="200"/>

</assert>

</action>

**Test update operation**

[Update](file:///C:\temp\http.html#update) operations require a resource id. The id must be present in the fixture (PUT body contents) as well as the URL. The values must match.

Because resource ids cannot be predicted on the server, it is best to retrieve the id on a resource freshly created as part of the script

There are many ways to do that. Below is a couple:

1. Use update operation with [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) fixture pointing to create operation's [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId):
2. <action>
3. <operation>
4. <type>
5. <system value="http://hl7.org/fhir/testscript-operation-codes"/>
6. <code value="create"/>
7. </type>
8. <responseId value="create-response" />
9. <sourceId value="example-patient"/>
10. </operation>
11. </action>
12. <action>
13. <operation>
14. <type>
15. <system value="http://hl7.org/fhir/testscript-operation-codes"/>
16. <code value="update"/>
17. </type>
18. <resource value="Patient"/>
19. <responseId value="R3" />
20. <sourceId value="john-doe-update"/>
21. <targetId value="create-response"/>
22. </operation>
23. </action>

Test engines will keep track of response headers and body of all operations.

The update operation's targetId value is expected to correspond to the responseId of a GET operation (such as search or read) or the sourceId of a POST/PUT operation (such as create).

For targetId value corresponding to responseId of GET operations (such as search or read), test engines will use the resource type and id returned in the GET response body's resource to set the [type] and [id] in update operation's URL, respectively. This is the case in the next example below.

For targetId value corresponding to responseId of POST/PUT operations (such as create and update), test engines will use the resource type and id returned in the POST/PUT response "Location" header to set the [type] and [id] in update operation's URL, respectively. This is the case in the example above.

The targetId value cannot point to a statically defined fixture as the id in the fixture cannot be relied upon.

1. Use update operation with [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) fixture pointing to search operation's [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId):
2. <action>
3. <operation>
4. <type>
5. <system value="http://hl7.org/fhir/testscript-operation-codes"/>
6. <code value="create"/>
7. </type>
8. <responseId value="R1"/>
9. <sourceId value="john-doe-patient"/> <!-- Fixture must be defined at the top of the script -->
10. </operation>
11. </action>
12. <action>
13. <operation>
14. <type>
15. <system value="http://hl7.org/fhir/testscript-operation-codes"/>
16. <code value="search"/>
17. </type>
18. <resource value="Patient"/>
19. <params value="?family=Doe&amp;given=Joe"/>
20. <responseId value="R2" />
21. </operation>
22. </action>
23. <action>
24. <operation>
25. <type>
26. <system value="http://hl7.org/fhir/testscript-operation-codes"/>
27. <code value="update"/>
28. </type>
29. <resource value="Patient"/>
30. <responseId value="R3" />
31. <sourceId value="john-doe-update"/>
32. <targetId value="R2"/>
33. </operation>
34. </action>

After the update operation, test scripts would perform at least one more read/search operation to retrieve the contents of the updated resource and then perform assertions to verify that the data did indeed get updated on the server:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="search"/>

</type>

<resource value="Patient"/>

<params value="?family=Doe&amp;given=Joe"/>

<responseId value="R4" />

</operation>

</action>

Verify that the birthdate got updated and is being returned properly:

<action>

<assert>

<path value="fhir:Patient/fhir:birthDate/@value"/>

<sourceId value="R4"/>

<value value="1974-12-31" />

</assert>

</action>

**Test conditional update operation**

Unlike a regular [update](file:///C:\temp\http.html#update) operation, a [conditional update](file:///C:\temp\http.html#2.1.0.10.1) operation does not require a resource id in the URL (or the body of the PUT).

To test conditional update, use [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element in the operation instead of [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId). The [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) element will be required in this case.

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="update"/>

</type>

<resource value="Patient"/>

<params value="?family=Doe&amp;given=Joe"/>

<responseId value="R3" />

<sourceId value="john-doe-update"/> <!-- john-doe-update fixture will have the 'correct' birthDate -->

</operation>

</action>

Test engines will append the contents of the [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element to url after [type]: "PUT [base]/[type]**?[search parameters]"**. The [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) element value ("Patient") will be used to replace [type] in the URL.

Verify that the birthdate got updated and is being returned properly:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="search"/>

</type>

<resource value="Patient"/>

<params value="?family=Doe&amp;given=Joe"/>

<responseId value="R4" />

</operation>

</action>

<action>

<assert>

<path value="fhir:Patient/fhir:birthDate/@value"/>

<sourceId value="R4"/>

<value value="1974-12-31" />

</assert>

</action>

**Test read operation**

The [read](file:///C:\temp\http.html#read) operation operation requires the resource id in the URL. Since resource ids are unpredictable on servers, it's best to create the resource within the test script prior to executing the read operation:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="create"/>

</type>

<responseId value="create-response"/>

<sourceId value="example-patient"/>

</operation>

</action>

One way to execute the read operation is to run the read operation with [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) value pointed to [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId) value of the create operation:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="read"/>

</type>

<targetId value="create-response"/>

</operation>

</action>

Test engines will keep track of response headers and body of all operations.

The read operation's targetId value is expected to correspond to the responseId of a GET operation (such as search or read) or the sourceId of a POST/PUT operation (such as create).

For targetId value corresponding to responseId of GET operations (such as search or read), test engines will use the resource type and id returned in the GET response body's resource to set the [type] and [id] in read operation's URL, respectively.

For targetId value corresponding to responseId of POST/PUT operations (such as create), test engines will use the resource type and id returned in the POST/PUT response "Location" header to set the [type] and [id] in read operation's URL, respectively. This is the case in the example above.

The targetId value cannot point to a statically defined fixture as the id in the fixture cannot be relied upon.

**Test vread operation**

The [vread](file:///C:\temp\http.html#vread) operation operation requires the resource id as well as the resource version id in the URL. Since resource ids and version ids are unpredictable on servers, it's best to create the resource within the test script prior to executing the vread operation:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="create"/>

</type>

<responseId value="create-response"/>

<sourceId value="example-patient"/>

</operation>

</action>

One way to execute the vread operation is to run the vread operation with [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) value pointed to [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId) value of the create operation:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="vread"/>

</type>

<targetId value="create-response"/>

</operation>

</action>

Test engines will keep track of response headers and body of all operations.

The vread operation's targetId value is expected to correspond to the responseId of a GET operation (such as search or read) or the sourceId of a POST/PUT operation (such as create).

For targetId value corresponding to responseId of GET operations (such as search or read), test engines will use the resource type, id and version id returned in the GET response body's resource to set the [type], [id] and [vid] in vread operation's URL, respectively.

For targetId value corresponding to responseId of POST/PUT operations (such as create), test engines will use the resource type, id and version id returned in the POST/PUT response "Location" header to set the [type], [id] and [vid] in vread operation's URL, respectively. This is the case in the example above.

The targetId value cannot point to a statically defined fixture as the id in the fixture cannot be relied upon.

**Test history operation**

The [history](file:///C:\temp\http.html#history) operation can be executed in the following ways:

1. GET [base]/[type]/[id]/\_history{?[parameters]&\_format=[mime-type]}

Here the resource id is required in the URL. This is similar to read operation if targetId elemet is used. See [Test read operation](file:///C:\temp\main-pages.html#howToTestReadOperation) for details.

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="history"/>

</type>

<targetId value="create-response"/> <!-- or search-response or update-response-->

</operation>

</action>

<action>

<assert>

<resource value="Bundle"/>

</assert>

</action>

1. GET [base]/[type]/\_history{?[parameters]&\_format=[mime-type]}

Here the resource id is not required in the URL.

Instead of [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) element, the [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element can be used to specify the search criteria for the history operation.

In the following example, all history entries for John Doe patient would be returned by server:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="history"/>

</type>

<resource value="Patient"/>

<params value="?family=Doe&amp;given=Joe"/>

</operation>

</action>

1. GET [base]/\_history{?[parameters]&\_format=[mime-type]}

Here neither the resource type nor the resource id is required in the url. In the following example, no more than 50 history entries would be returned by server:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="history"/>

</type>

<params value="?\_count=50"/>

</operation>

</action>

**Specify Accept header in request**

The default "Accept" header that will be set on all GET operations (such as read, vread, search, history, etc.) will be "application/fhir+xml".

There are two ways to change the default "Accept" header:

1. Use the [accept](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.accept) element:
2. <action>
3. <operation>
4. <type>
5. <system value="http://hl7.org/fhir/testscript-operation-codes"/>
6. <code value="read"/>
7. </type>
8. <accept value="json" />
9. <targetId value="F1" />
10. </operation>
11. </action>

Test engines will set the Accept header to "application/fhir+json" if "json" is specified and will use "application/fhir+xml" if "xml" is specified.

1. Use the [requestHeader](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.requestHeader) element to set "Accept" **explicitly**:
2. <action>
3. <operation>
4. <type>
5. <system value="http://hl7.org/fhir/testscript-operation-codes"/>
6. <code value="read"/>
7. </type>
8. <requestHeader>
9. <field value="Accept" />
10. <value value="application/fhir+json" />
11. </requestHeader>
12. <targetId value="F1" />
13. </operation>
14. </action>

Test engines will take values specified for [requestHeader](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.requestHeader) "as-is" and not transform them. This might be useful for negative testing e.g. the value can be set explicitly to "applcation/xml" or an invalid value and verify server response.

**Specify Content-Type header in request**

The default "Content-Type" header that will be set on all POST/PUT operations (such as create, update, etc.) will be "application/fhir+xml".

There are two ways to change the default "Content-Type" header:

1. Use the [contentType](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.contentType) element:
2. <action>
3. <operation>
4. <type>
5. <system value="http://hl7.org/fhir/testscript-operation-codes"/>
6. <code value="create"/>
7. </type>
8. <contentType value="json" />
9. <targetId value="F1" />
10. </operation>
11. </action>

Test engines will set the Content-Type header to "application/fhir+json" if "json" is specified and will use "application/fhir+xml" if "xml" is specified.

1. Use the [requestHeader](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.requestHeader) element to set Content-Type **explicitly**:
2. <action>
3. <operation>
4. <type>
5. <system value="http://hl7.org/fhir/testscript-operation-codes"/>
6. <code value="create"/>
7. </type>
8. <requestHeader>
9. <field value="Content-Type" />
10. <value value="application/fhir+json" />
11. </requestHeader>
12. <targetId value="F1" />
13. </operation>
14. </action>

Test engines will take values specified for [requestHeader](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.requestHeader) "as-is" and not transform them. This might be useful for negative testing e.g. the value can be set explicitly to "applcation/xml" or an invalid value and verify server response.

**Verify Content-Type header in response**

There are two ways to verify the "Content-Type" header in response:

1. Use the [contentType](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.contentType) element:
2. <action>
3. <assert>
4. <contentType value="json" />
5. </assert>
6. </action>

Test engines will verify that "application/fhir+json" is present in Content-Type header if "json" is specified and will verify that "application/fhir+xml" is present if "xml" is specified.

1. Use the [requestHeader](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.requestHeader) element to verify Content-Type **explicitly**:
2. <action>
3. <assert>
4. <headerField value="Content-Type" />
5. <value value="application/fhir+json" />
6. </assert>
7. </action>

Test engines will take values specified for [headerField](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.headerField) "as-is" and not interpret them.

Note that test engines will not verify contentType in response if assertions for contentType are missing.

**Use Profiles**

[Profiles](file:///C:\temp\main-pages.html#profiles) are defined against known FHIR profiles (see [StructureDefinition](file:///C:\temp\structuredefinition.html) resource) corresponding to the FHIR specification version being tested. They are used in the "assert.validateProfileId" and, when present in an assert, will invoke validation of the FHIR resource payload in either the request or response message.

Profiles would be defined at the top of the script.

Below is a profile that is defined against the FHIR Patient resource type and referenced by "P1":

<profile id="P1">

<reference value="http://hl7.org/fhir/StructureDefinition/Patient" />

</profile>

Test engines will not evaluate this at this point. They will store the profile reference in "P1" and will look for "P1" in an "assert.validateProfileId" element values during operation calls.

Here is a read operation for a Patient with a resource id of "example" followed by an assert that performs a FHIR validation using the base FHIR profile for the Patient resource type referenced by the "P1" profile. The validation will be performed against the response payload by default if the "assert.direction" is not present. If the validation returns any "fatal" or "error" messages, then this assert will be failed. Otherwise, if the validation returns any "warning" messages, then this assert will pass but with warnings. Finally, if the validation returns only "information" messages or no messages, then this assert will pass.

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="read"/>

</type>

<resource value="Patient"/>

<accept value="xml"/>

<params value="/example"/>

</operation>

</action>

<action>

<assert>

<validateProfileId value="P1"/>

</assert>

</action>

**Use Variables**

[Variables](file:///C:\temp\main-pages.html#variables) can be defined against static fixtures and dynamic operation responses. They can be used in "operation.params", "operation.requestHeader.value", "operation.url", and "assert.value" element values. As such they allow for the data used in operations and assertions to be externally defined. The data could be unique to each client involved in interactions with a server or could be unique to a given server database instance. This allows for multiple clients to execute the same test scripts concurrently against the same server.

Variables would be defined at the top of the script.

Below is a variable that is defined as the Location header to the response referenced by "R1":

<variable>

<name value="V1" />

<headerField value="Location" />

<sourceId value="R1" />

</variable>

Test engines will not evaluate this at this point. They will store the expresssion in "V1" and will look for "${}" in "operation.params", "operation.requestHeader.value", and "operation.url" element values during operation calls.

Here is a read operation that will use the V1 variable. The variable expression was "Location against R1 response" (defined above). If a prior operation has not set R1 to be the responseId of the operation, then test engine will error out. Otherwise, V1 will be set to the Location header value of R1 response and that value will be substituted for ${V1} below. In other words, the read will be performed against the Location header value of R1 response.

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="read"/>

</type>

<accept value="json" />

<responseId value="R2" />

<url value="${V1}" />

</operation>

</action>

Below are three variables defined as path expressions against the static fixture referenced by "F1". The expressions are against the given name, family name, and birthDate of a patient resource. The resource data will be managed external to the test script.

<variable>

<name value="PatientGivenName" />

<path value="fhir:Patient/fhir:name/fhir:given/@value" />

<sourceId value="F1" />

</variable>

<variable>

<name value="PatientFamilyName" />

<path value="fhir:Patient/fhir:name/fhir:family/@value" />

<sourceId value="F1" />

</variable>

<variable>

<name value="PatientDOB" />

<path value="hir:Patient/fhir:birthDate/@value" />

<sourceId value="F1" />

</variable>

Again, test engines will not evaluate the path expression at this point. They will look for anything wrapped in '${}' in "operation.params", "operation.requestHeader.value", "operation.url", and "assert.value" element values and substitute the placeholders with the evaluated expressions.

Here is a conditional create operation that will set the requestHeader using the PatientGivenName, PatientFamilyName, and PatientDOB variables defined above. The variable expressions were path expressions against the statically defined F1 fixture. They will be evaluated against the fixture body (containing resource) and the corresponding values will be extracted from the fixtures and used to substitute the variables in the requestHeader value below.

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="create"/>

</type>

<requestHeader>

<field value="If-None-Exist" />

<value value="Patient?given=${PatientGivenName}&amp;${PatientFamilyName}&amp;birthdate=${PatientDOB}" />

</requestHeader>

<sourceId value="F1" />

</operation>

</action>

Here is a search operation that will perform a search using the PatientGivenName, PatientFamilyName, and PatientDOB variables defined above. The variable expressions were path expressions against the statically defined F1 fixture. They will be evaluated against the fixture body (containing resource) and the corresponding values will be extracted from the fixtures and used to substitute the variables in the params value below.

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="search"/>

</type>

<resource value="Patient" />

<accept value="json" />

<params value="?given=${PatientGivenName}&amp;family=${PatientFamilyName}&amp;birthdate=${PatientDOB}" />

<responseId value="R3" />

</operation>

</action>

Here are the assertions that verify that the search was successful:

<action>

<assert>

<path value="Patient/name/given" />

<value value="${PatientGivenName}" />

</assert>

</action>

<action>

<assert>

<path value="Patient/name/family" />

<value value="${PatientFamilyName}" />

</assert>

</action>

<action>

<assert>

<path value="Patient/birthdate" />

<value value="${PatientDOB}" />

</assert>

</action>

**Test server support for '\_format'**

Servers are required to support "\_format" in the request url to determine the response mime-type. See [Content Type and Encodings](file:///C:\temp\main-pages.html#http.html#2.1.0.6)

Use the [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element to specify the \_format:

<action>

<operation>

<type>

<system value="http://hl7.org/fhir/testscript-operation-codes"/>

<code value="search"/>

</type>

<resource value="Patient"/>

<params value="?family=Doe&amp;given=Joe&amp;\_format=application/fhir+json"/>

<responseId value="R1" />

</operation>

</action>

Use the [requestHeader](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.requestHeader) element to verify Content-Type **explicitly**:

<action>

<assert>

<headerField value="Content-Type" />

<value value="application/fhir+json" />

</assert>

</action>

**How to specify metadata capabilities**

If the capabilities are supported by the server, then the TestScript can be executed. Otherwise, a test engine will provide a comparison of the system under testâ€™s conformance capabilities and allow a TestScript as a whole to be skipped based on this comparison..

Here's how to specify that the test script requires the server to support Patient create and delete operations:

<metadata>

<capabilities>

<required value="true"/>

<description value="Patient Create and Delete Operation"/>

<conformance>

<reference value="/Conformance/PatientCreateDelete.xml" />

</conformance>

</capabilities>

</metadata>

The contents of PatientCreateDelete.xml would be a minimal capability statement to indicate what sections need to be present in server capability statement:

<Conformance xmlns="http://hl7.org/fhir">

<rest>

<mode value="server"/>

<resource>

<type value="Patient"/>

<interaction>

<code value="create"/>

</interaction>

<interaction>

<code value="delete"/>

</interaction>

</resource>

</rest>

</Conformance>

When the metadata capabilities section is defined at [TestScript.metadata](file:///C:\temp\testscript-definitions.html#TestScript.metadata) and the server's capability statement does not contain the elements defined in the minimal capability statement, then all the tests in the TestScript are skipped.

The "metadata.capabilities.required" and "metadata.capabilities.validated" elements only indicate whether the capabilities are the primary focus of the test script or not. The do not impact the skipping logic. Capabilities whose "metadata.capabilities.validated" flag is true are the primary focus of the test script.

**Operations**

This table presents a summary of the constraints applicable to TestScript.setup.action.operation, TestScript.test.action.operation, and TestScript.teardown.action.operation elements. The operation elements should be configured consistently with the [FHIR RESTful API summary](file:///C:\temp\http.html).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **read** | **vread** | **search** | **history** | **create** | **update** | **transaction** | **conformance** | **delete** |
| **resource** | The [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) element is required to specify the resource ([type]) in the request URL when [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element is used. Will be ignored if [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) or [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) are specified. In the case of [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId), the resource type will be extracted from the fixture. | | | | If [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) is specified, then [type] for request URL will be determined from targetId's fixture and [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) element will be ignored. Otherwise, resource type will be extracted from [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.sourceId)'s fixture if specified. For [conditional updates](file:///C:\temp\http.html#2.1.0.10.1), [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) is required. | | N/A | N/A | If [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) is specified, then [type] for request URL will be determined from [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) and [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) element will be ignored. For [conditional deletes](file:///C:\temp\http.html#2.1.0.12.1), [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) is required. |
| **accept** | The [accept](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.accept) element can be used to specify the "Accept" header in the outgoing HTTP request. If "json" is specified, then "Accept" value of "application/fhir+json" will be set in the request header. If "xml" is specified, then "application/fhir+xml" will be used. | | | | | | | | N/A |
| **contentType** | The [contentType](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.contentType) element can be used to specify the "Content-Type" header in the outgoing HTTP request. If "json" is specified, then "Content-Type" value of "application/fhir+json" will be set in the request header. If "xml" is specified, then "application/fhir+xml" will be used. | | | | | | | | N/A |
| **destination** | If the TestScript is testing more than one FHIR server simultaneously, the [destination](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.destination) identifies which test system the operation will receive this request message using zero-based indexing. The destination value MUST be equal to a defined [TestScript.destination"](file:///C:\temp\testscript-definitions.html#TestScript.destination). | | | | | | | | |
| **encodeRequestUrl** | The [encodeRequestUrl](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.encodeRequestUrl) element, if "true", can be used to specify that the destination test system will receive this request message URL query parameters in HTML (Percent) encoded UTF-8 format. The default value for this element if not present is "false". | | | | | | | | |
| **origin** | If the TestScript is testing more than one FHIR server simultaneously, the [origin](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.origin) identifies which test system the operation will send this request message using zero-based indexing. The destination value MUST be equal to a defined [TestScript.origin"](file:///C:\temp\testscript-definitions.html#TestScript.origin). | | | | | | | | |
| **params** | The [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element can be used to specify the [id] using variable substitutions and the rest of the highlighted portion in the request URL:  [base]/[type]**/[id] {?\_format=[mime-type]}**  If used, then [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) is required and [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified. | The [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element can be used to specify the [id] and [vid] using variable substitutions and the rest of the highlighted portion in the request URL:  [base]/[type]**/[id]/\_history/[vid] {?\_format=[mime-type]}**  If used, then [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) is required and [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified. | The [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element can be used to specify the highlighted portion in the request URL:  [base]/[type]**{?[parameters]{&\_format=[mime-type]}}**  If used, then [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) is optional and [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified. | The [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element can be used to specify the [id] using variable substitutions and the rest of the highlighted portion in the following request URLs: [base]/[type]**/[id]/\_history{?[parameters]&\_format=[mime-type]}** [base]/[type]**/\_history{?[parameters]&\_format=[mime-type]}** [base]**/\_history{?[parameters]&\_format=[mime-type]}**  If used, then [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) is optional and [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified. | N/A | The [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element can be used in [conditional update](file:///C:\temp\http.html#2.1.0.10.1) operation to specify the highlighted portion of the request URL:  [base]/[type]**?[search parameters]**  If used, then [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) is required and [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified. | N/A | The [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element can be used to specify the highlighted portion in the request URL:  [base]**/metadata {?\_format=[mime-type]}**  If used, then [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) is ignored and [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified. | The [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element can be used to specify the [id] using variable substitutions in the request URL:  [base]/[type]**/[id]**  If used, then [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.resource) is required and [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified. |
| **requestHeader** | The [requestHeader](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.requestHeader) element allows for request headers to be specified **explicitly**. Test engines will take values specified for [requestHeader](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.requestHeader) "as-is" and not transform them. This allows for testing using:   * "If-Modified-Since" and "If-None-Match" headers. See [Conditional read](file:///C:\temp\http.html#2.1.0.5.1). * "If-Match" header. See [Managing resource contention](file:///C:\temp\http.html#2.1.0.11). * Conditional Create using "If-None-Exist". See [Conditional Create](file:///C:\temp\http.html#2.1.0.13.1). * Invalid "Content-Type" header for negative testing. * etc. | | | | | | | | |
| **requestId** | N/A | N/A | N/A | N/A | The [requestId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.requestId) element can be used to reference the operation request containing response body and headers. If specified, the value can later be used in assertion [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.sourceId) to evaluate [path](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.path) (XPath/JSONPath) and [headerFields](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.headerField) against the request sent for an operation. | | | N/A | N/A |
| **responseId** | The [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId) element can be used to reference the operation response containing response body and headers. If specified, the value can later be used in assertion [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.sourceId) to evaluate [path](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.path) (XPath/JSONPath) and [headerFields](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.headerField) against the response received for an operation. | | | | | | | | N/A |
| **sourceId** | N/A | N/A | N/A | N/A | The [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.sourceId) element points to a fixture to be used for the created resource. The fixture cannot contain the id element. | The [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.sourceId) element points to a fixture to be used for the updated resource. Has to correspond to the [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId) of an operation executed upstream in the test script. The response body must contain a resource with a resource id. The sourceId fixture cannot be statically defined because the id cannot be relied upon. | Fixture to be used for the transaction. Has to be a [Bundle](file:///C:\temp\main-pages.html#bundle.html). | N/A | N/A |
| **targetId** | The [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) element can be used to specify the [type] and [id] in the request URL.  If used, then [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified.  The targetId value has to correspond to the responseId of an operation executed upstream in the test script. The response body must contain a reosurce with a resource id. The targetId fixture cannot be statically defined because the id cannot be relied upon. | The [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) element can be used to specify the [type], [id], and [vid] in the request URL.  If used, then [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified.  The targetId value has to correspond to the responseId of an operation executed upstream in the test script. The response body must contain a reosurce with a resource id. The targetId fixture cannot be statically defined because the id and vid cannot be relied upon. | The [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) element cannot be used as it's not allowed with [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) element | The [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) element can be used to specify the [type], [id], and [vid] in the request URL.  If used, then [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified.  The targetId value has to correspond to the responseId of an operation executed upstream in the test script. The response body must contain a reosurce with a resource id. The targetId fixture cannot be statically defined because the id and vid cannot be relied upon. | N/A. The [type] for the request URL will be extracted from [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.sourceId). | N/A. The [type] for the request URL will be extracted from [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.sourceId). | N/A. | N/A. | The [targetId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.targetId) element can be used to specify the [type] and [id] in the request URL.  If used, then [params](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.params) and [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) must not be specified.  The targetId value has to correspond to the responseId of an operation executed upstream in the test script. The response body must contain a reosurce with a resource id. The targetId fixture cannot be statically defined because the id cannot be relied upon. |
| **url** | The [url](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.url) element will contain the full HTTP URL for the operation. This should rarely be used in test scripts. One possible application would be to test if the Location header returned in a response is pointing to an expected resource. See [testscript-example-search example](file:///C:\temp\testscript-example-search.xml.html). | | | | | | | | |

**Assertions**

|  |  |  |
| --- | --- | --- |
| **Assertion** | **Valid operator values** | **Description** |
| compareToSourcePath | equals | notEquals | Asserts that [compareToSourcePath](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.compareToSourcePath) against the response body of [compareToSourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.compareToSourceId) fixture evaluates to a value that is equal or notEqual to the evaluated value of [path](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.path) which must be present also. |
| contentType | equals | notEquals | contains | notContains | Asserts that the "Content-Type" in response header is or is not the specified value for [contentType](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.contentType) element depending on the operator used. |
| headerField | equals | notEquals | in | notIn | greaterThan | lessThan | empty | notEmpty | contains | notContains | Asserts that the header specified for [headerField](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.headerField) element in the response contains, not contains, is equal, not equal, in, not in, greater than, or less than the value specified for [value](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.value) element if present. If the [operator](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.operator) is "empty" or "notEmpty" then [value](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.value) will be ignored. If [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.sourceId) is not specified, then [headerField](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.headerField) will be evaluated against the last operation's response headers. |
| minimumId | N/A | Asserts that the response contains all the element/content in another fixture pointed to by [minimumId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.minimumId) element. This can be a statically defined [fixture](file:///C:\temp\testscript-definitions.html#TestScript.fixture) or one that is dynamically set via [responseId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.operation.responseId). |
| navigationLinks | N/A | Asserts that the response Bundle contains or does NOT contain first, last, and next links depending on whether or not [navigationLinks](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.navigationLinks) element is set to true or false. |
| path | equals | notEquals | in | notIn | greaterThan | lessThan | empty | notEmpty | contains | notContains | Asserts that [path](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.path) against the response body evaluates to a value that contains, not contains, is equal, not equal, in, not in, greater than, or less than the value specified for [value](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.value) element if present. If the [operator](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.operator) is "empty" or "notEmpty" then [value](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.value) will be ignored. If [sourceId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.sourceId) is not specified, then [path](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.path) will be evaluated against the last operation's response body. |
| resource | equals | notEquals | Asserts that the resource returned in the response body is or is not of the specified value for [resource](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.resource) element. |
| response | equals | notEquals | Asserts that status code in the response is or is not one of the enumerated values in [response abbreviations](file:///C:\temp\valueset-assert-response-code-types.html). |
| responseCode | equals | notEquals | in | notIn | greaterThan | lessThan | Asserts that status code in the response is equal, notEqual, in, not in, greater than, or less than the specified value(s) for [responseCode](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.responseCode) element |
| validateProfileId | N/A | Asserts that the response is valid according to the profile specified by [validateProfileId](file:///C:\temp\testscript-definitions.html#TestScript.setup.action.assert.validateProfileId) element. |

**timelines.html**

**FHIR Timelines**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

FHIR is a standard. In order to be useful, standards need to evolve. At the same time, the evolution of standards needs to be predictable and manageable for the implementation community. This section provides forward-looking statements about the expected pattern of FHIR releases as well as the degree of stability and change implementers should expect from the standard as it continues to evolve.

**Standard Levels**

FHIR has three descriptive terms that describe the level of stability and implementation readiness associated with different aspects of the specification. They are as follows:

|  |  |
| --- | --- |
| **Standard Level** | **Description** |
| **Draft** | This portion of the specification is not considered to be complete enough or sufficiently reviewed to be safe for implementation. It may have known issues or still in the "in development" stages. It is included in the publication as a place-holder, to solicit feedback from the implementation community and/or to give implementers some insight as to functionality likely to be included in future versions of the specification. Content at this level should only be implemented by the brave or desperate and is very much "use at your own risk". The content that is draft that will usually be elevated to STU once review and correction is complete after ballot |
| **Standard for Trial Use (STU)** | This content has been well reviewed and is considered by the authors to be ready for use in production systems. It has been subjected to ballot and approved as an official standard. However, it has not yet seen widespread use in production across the full spectrum of environments it is intended to be used in. In some cases, there may be documented known issues that require implementation experience to determine appropriate resolutions for.  For these reasons, STU content is not subject to FHIR [Inter-version Compatibility Rules](file:///C:\temp\compatibility.html). Future versions of FHIR may make significant changes to STU-level content that are not compatible with previously published content. See [STU suggestions](file:///C:\temp\main-pages.html#stu) for implementation strategies to help manage the risk of non-compatible future changes.  STU content can have a wide range of stability ranging from never before implemented even in test systems through to widely implemented in production and well reviewed. FHIR makes use of a [FHIR Management Group](file:///C:\temp\resource.html#maturity>Maturity rating</a>      that provides insight into how well polished a given resource or other artifact is.  Artifacts with higher maturity      ratings (4 or 5) will be subject to somewhat tighter constraints on non-backward compatible changes, though these      are still possible when deemed to be in the best interests of the implementer community.      </p>      <p>      The maturity model is significantly influenced by the degree and type of implementation activity using an artifact.      For this reason, we encourage implementers to share with the <a href=) what artifacts they have implemented. A detailed analysis of the maturity metrics for FHIR artifacts can be found [here](https://docs.google.com/spreadsheets/d/18HfXF7mUCUV7jACCG0oejFp6D-ibtvbmcgywNhn76lw).  Note: STU specifications were previously called DSTU (Draft Standard for Trial Use). The name has been changed due to the reluctance of implementers to make use of a "draft" specification - when the whole purpose of the STU phase is for implementers to exercise the specification. |
| **Normative** | This content has been subject to review and production implementation in a wide variety of environments. The content is considered to be stable and has been 'locked', subjecting it to FHIR [Inter-version Compatibility Rules](file:///C:\temp\compatibility.html). While changes are possible, they are expected to be infrequent and are tightly constrained. |

The above statuses can apply to both the standard overall as well as to individual components of the FHIR specification. Specification components cannot ever have a "higher" standard level than the overall specification, but they can have lower levels. For example, at STU, FHIR can include draft content. When it reaches Normative, FHIR may include some content that remains at STU or draft levels if that content has not yet reached the criteria for normative. All content that is at a different ballot level than the overall specification will be clearly identified.

**FHIR timeline**

New versions of FHIR will be published on a release cycle of aproximately 18-24 months. This frequency is based on the timelines necessary to develop, implement and review new content as well as the time necessary to undertake the formal balloting and reconciliation processes required for ANSI-approved standards. This release cycle also ensures an opportunity to incorporate implementer feedback from earlier versions of the specification into subsequent versions.

In some situations, the FHIR Management Group may authorize a limited-scope release on a shorter timeline where necessary to meet implementer requirements and is achievable with available HL7 resources.

The forthcoming release (STU 3) may be the final publication of the specification that is entirely at the STU level. The subsequent publication of the specification (targeted for 2018) will hopefully take the core aspects of the specification and many of the most broadly used resources to Normative level. Whether this timeline will be met will be dependent on uptake and feedback from implementers. This feedback will also govern exactly which resources, profiles and other content become normative. Only content that has been successfully implemented in a wide variety of implementation environments with minimal divergence from the STU specification will be candidates for normative.

Once FHIR has reached normative status, subsequent publications will continue on the 18-24 month schedule with subsequent releases introducing additional resources, capabilities and other content as well as migrating existing content from draft to STU and STU to normative, based on the level of implementation.

**STU Suggestions**

The FHIR specification is a "Standard for Trial Use" (STU). It has been subject to significant review through ballot and other HL7 processes and many aspects of it have been implemented and subjected to interoperability testing through Connectathons and early adoption. However, the degree of testing has varied. Some resources have been well tested in a variety of environments. Others have received relatively little real-world exercise. In general, the infrastructure should be considered to be more stable than the resources themselves. In some cases, there are issues on which input is specifically requested during the STU period (see the [Outstanding Issue List](file:///C:\temp\todo.html), and known issues will arise after publication (refer to the [FHIR Change Request](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemBrowse&tracker_id=677) tracker for details.) Guidance from early implementation will help address these areas.

Regardless of the degree of prior implementation, all aspects of the FHIR specification are potentially subject to change. These changes may be minor (clarifications of definitions, etc.) or major (refactoring of resources, changes to serialization rules, eliminating or adding data types, etc.) There is no commitment to backward or forward compatibility during the STU process. Changes will not be made without cause, however the interests of long-term implementability will generally trump the impact on early adopters when determining what changes should be made. This balance will shift more towards early adopters as maturity levels increase. I.e. Impact on existing implementations will be weighted more highly for an FMM-level 5 artifact than they would for an FMM-level 1 artifact.

This specification has been promoted to STU because it is felt that the specification, as is, is implementable and that more value can be gleaned from implementer experience than from subsequent review as part of the ballot process. Implementers who are willing to accept the risk of change (perhaps for the benefit of early implementation experience, first mover advantage and the ability to leverage FHIR's intrinsic benefits) are encouraged to implement FHIR in real-world systems. However, those implementers should be aware that local adaptations may be necessary to meet real-world requirements. Furthermore, such implementers should architect their solutions to be tolerant of changes to the specification and, where necessary, to manage interoperability with systems that may be using different versions of the specification or different local adaptations.

During the STU period, requests for change may be submitted using the HL7 gForge tracker which can be found [here](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemBrowse&tracker_id=677). Where possible, updates to the "development" version of the specification will be made in a timely fashion. A list of these proposed changes will be published as a continuously updated supplement to the official STU publication. Implementers should be aware that the changes are not considered "official" until such time as they are balloted and approved as part of a subsequent STU or Normative publication. Change requests might be fixes to allow implementation, clarifications or enhancements. In addition, HL7 will be developing and introducing additional resources and profiles as part of the FHIR specification.

SDOs and regulatory bodies that are interested in making use of the FHIR specification should feel free to do so, but should consider and plan for the possibility that the specification will evolve and change prior to becoming normative.

A key objective of the STU process is gaining feedback from implementers making use of the specification. As well, HL7 has a need to monitor which portions of FHIR are being implemented in what sorts of environments so as to make an informed decision on when the specification is ready to proceed to Normative status. For this reason, all FHIR implementers are asked to complete a short survey which can be found [here](https://www.surveymonkey.com/s/PXZTY7Z).

This survey will capture contact and other information that will allow the FMG to perform appropriate monitoring of FHIR STU usage. Survey information will be kept confidential unless the participant authorizes inclusion of their project in an HL7-maintained wiki page of early implementers. Confidential submissions will be reported in aggregate only.

While implementation of this STU release is occurring, development will be progressing on the next (hopefully Normative) release. This next release will include additional resources, profiles and quality enhancements over the current release. It will also incorporate fixes for issues raised with the [FHIR change tracker](http://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemBrowse&tracker_id=677). It may be useful for implementers of the STU to browse the development release to get a sense of what changes are likely coming and perhaps to find more robust definitions and guidance than are available in the first release. The FHIR development release can be found at [build.fhir.org](http://build.fhir.org/). Some implementers who are dependent on content that exists in a draft release may choose to implement based on a particular snapshot of the development release, though in doing so, they will limit their potential communication partners and would not be considered to be completely FHIR conformant.

**toc.html**

**Full Table of Contents**

**todo.html**

**Outstanding Issues**

This specification is currently approaching its third round of trial use. While some parts of the specification are mature and stable, much work remains to be done. The following general areas of functionality have been deferred to a future version:

* Adverse Event Reporting
* An alarm resource to represent current issues with the patient (e.g. device created)
* Concern Tracking
* Clinical Studies and Protocols
* Aggregated Data Reporting including Public Health Reporting
* Payment related resources, and specifically, an Account resource for payment tracking
* One or more resources for Advance Care Directive / Power of Attorney
* A full server side query framework

For some of these, some draft content is included in the specification for implementer consideration.

In addition, there are a number of specific notes in the specification requesting feedback from implementers:

* [AllergyIntolerance](file:///C:\\temp\\allergyintolerance.html" \l "dstu0): new codes needed for certainty?
* [AllergyIntolerance](file:///C:\temp\allergyintolerance.html#dstu): How should "No Known Allergies" be represented?
* [Appointment](file:///C:\temp\appointment.html#dstu): Values for Appointment.priority: how interoperable are they
* [BodySite](file:///C:\temp\bodysite.html#dstu): Should it be an independent resource or a datatype?
* [CarePlan](file:///C:\temp\careplan.html#dstu): Combination concepts of "Care Plan" and "Care Team", and patient being optional
* [ClinicalImpression](file:///C:\temp\clinicalimpression.html#dstu): General Questions about use
* [Composition](file:///C:\temp\composition.html#dstu): Is title different to type? and open questions about document signatures
* [Condition](file:///C:\temp\condition.html#dstu): Questions about Condition.category
* [Condition](file:///C:\temp\condition.html#dstu-1): How should "No Known Problems" be represented?
* [DataElement](file:///C:\temp\dataelement.html#dstu): Should constraints be common across Questionnaire, DataElement, and StructureDefinition?
* [DiagnosticReport](file:///C:\temp\diagnosticreport.html#dstu): Relationship between DiagnosticReport and Composition?
* [Extensibility](file:///C:\temp\extensibility.html#dstu): Do we need to support modifier extensions on extensions?
* [Managing Resource Identity](file:///C:\temp\managing.html#dstu): Mandating Identification practices for cross-system interoperability?
* [Markdown](file:///C:\temp\datatypes.html#dstu): Should we support CommonMark>
* [Messaging](file:///C:\temp\messaging.html#dstu): What additional events should be defined?
* [Operations](file:///C:\temp\operations.html#dstu): Do we need a way to execute operations asynchronously?
* [Patient](file:///C:\temp\patient.html#dstu): Should linking/merging affect the RESTful API?
* [Patient](file:///C:\temp\patient.html#dstu-2): Comment sought on MPI matching
* [PractitionerRole](file:///C:\temp\practitionerrole.html#dstu): Practitioner.role to be removed?
* [Profiling Resources](file:///C:\temp\profiling.html#dstu): Need feedback on using system profiles
* [References between Resources](file:///C:\temp\references.html#dstu): Do we need to allow contained resources that reference the container?
* [RESTful API](file:///C:\temp\http.html#dstu-2): Consquence of side effects in batches and transactions?
* [RESTful API](file:///C:\temp\http.html#dstu-3): Should servers maintain ids when processing transactions?
* [RESTful API](file:///C:\temp\http.html#dstu-4): What are the documentation requirements around transaction integrity?
* [RiskAssessment](file:///C:\temp\riskassessment.html#dstu): several open issues
* [Search](file:///C:\temp\search.html#dstu): Feedback required on \_has parameter
* [Search](file:///C:\temp\search.html#dstu2): does text search need to be standardized?
* [Search](file:///C:\temp\search.html#dstu4): Do we need additional rules about \_include?
* [Security](file:///C:\temp\security.html#dstu): Feedback about signatures on RESTful interfaces sought
* [Subscription](file:///C:\temp\subscription.html#dstu): messaging details still to be resolved
* [Workflow](file:///C:\temp\workflow.html#issues): several implementation questions

**ucum.html**

**Using UCUM with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The [Units of Measure](http://unitsofmeasure.org) Coding System (UCUM) is recommended for use with the [Quantity](file:///C:\temp\datatypes.html#Quantity) data type.

**Summary**

|  |  |
| --- | --- |
| Source | UCUM is made available by the [Regenstrief Institute, Inc](https://www.regenstrief.org/) and The UCUM Organization at <http://unitsofmeasure.org> |
| System | The URI <http://unitsofmeasure.org> identifies UCUM codes |
| Version | Where a version is used, it should be the standard UCUM version e.g. 1.9. There is no need to use version in the [Coding](file:///C:\temp\datatypes.html#Coding) data type, only in [Value sets](file:///C:\temp\valueset.html) that use UCUM codes |
| Code | Valid expressions using the case sensitive symbols |
| Display | There is no defined display; the UCUM code is used directly for the display |
| Inactive | A few codes in UCUM are marked as deprecated (e.g. ppb, pptr) |
| Subsumption | No Subsumption relationships are defined by UCUM |
| Filter Properties | Two filter properties are defined as described below |

**Copyright**

UCUM is Copyright Ã‚Â© 1999-2013 Regenstrief Institute, Inc. and The UCUM Organization, Indianapolis, IN. All rights reserved. See [TermsOfUse](http://unitsofmeasure.org/trac/wiki/TermsOfUse) for details.

**UCUM Filter Properties**

This section documents the property filters that can be used with the UCUM code system in value set composition statements.

**Property filter**

|  |  |
| --- | --- |
| Description | Restricts the expression to a describe a particular UCUM base property |
| Property Name | property |
| Operations Allowed | = |
| Values Allowed | [string name of property] |
| Comments | Restricts expressions to any expression that is comparable to a base unit with a matching property value. Note: this does not match the property on units, only base units |

**Canonical Filter**

|  |  |
| --- | --- |
| Description | Allows for any expression that is comparable to the named unit |
| Property Name | canonical |
| Operations Allowed | = / in |
| Values Allowed | UCUM expression |
| Comments | This allows any expression that is comparable to the given unit(s) |

**Implicit Value Sets**

Implicit value sets are those whose specification can be predicted based on the grammar of the underlying code system, and the known structure of the URL that refers to them. There is one set of implicit value sets defined for UCUM: By Canonical Value.

If any value set resources exist with an identifier that conforms to the URL patterns specified below, the content of the resource must conform to the template provided. Profiles and other value set references are allowed to reference these value sets directly.

The value set identifier http://unitsofmeasure.org/vs is a value set that contains all UCUM codes.

**UCUM Comparable Value Set**

A value set with an identifier of "http://unitsofmeasure.org/vs/[expression]" must conform to this template, where [expression] is a valid UCUM expression:

<ValueSet xmlns="http://hl7.org/fhir">

<text>

<status value="generated"/>

<div xmlns="http://www.w3.org/1999/xhtml">

[some html that identifies that this value set

includes all UCUM expressions that are comparable to the provided expression]

</div>

</text>

<identifier value="http://unitsofmeasure.org/vs/[expression]"/>

<version value="[optional - but recommended - UCUM version]"/>

<name value="Ucum Expressions comparable to [expression]"/>

<description value="Ucum Expressions comparable to [expression]"/>

<status value="active"/>

<date value="[optional date of UCUM release]"/>

<compose>

<include>

<system value="http://unitsofmeasure.org"/>

<filter>

<property value="comparable"/>

<op value="="/>

<value value="[expression]"/>

</filter>

</include>

</compose>

</ValueSet>

**unii.html**

**Using Unique Ingredient Identifier (UNII) with FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

**Summary**

|  |  |
| --- | --- |
| Source | [FDA Substance Regsitration System](http://fdasis.nlm.nih.gov) |
| System | The URI to identify UNII codes is http://fdasis.nlm.nih.gov |
| Version | If it is desired to exchange the version, use the date of the file in the UNII list download in the format YYYYMMDD |
| Code | Use the value of the UNII column in the UNII List |
| Display | Use the value of the PT (Preferred Term) column in the UNII List |
| Inactive | Todo: Describe how it is determined which concepts are inactive |
| Subsumption | No Subsumption relationships are defined by UNII |
| Filter Properties | None are described yet |

**Version Issues**

To be investigated

**Copyright/License Issues**

UNII has no copyright acknowledgement needed, nor are there any license terms to adhere to.

**NDF-RT Filter Properties**

No need for filters identified yet.

**Implicit Value Sets**

No need for implicit value sets identified yet.

**updates.html**

**Variations between Submitted data and Retrieved data**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Some of the time when using a FHIR interface, requests to create or update resource instances will behave exactly as the initiator requested. The desired record(s) will be created or revised within the target system and a subsequent query of the data would show the exact same information as was submitted. However, FHIR systems are not guaranteed to behave this way. Without any other agreement between exchange partners, FHIR systems are not obligated to store and return data as it was received. In fact, for some interoperability paradigms, they're not obligated to store any data at all. This page discusses some of the considerations around system behavior, including differences in expectations for systems interoperating using REST, messaging, documents and services.

**Data element support**

The most common reason for differences between what data is submitted to a system and what data can be extracted from it is that the system doesn't support all of the data elements present in the instance received. In the base resource, no systems are required to support any particular set of extensions or even any particular subset of core elements.

For elements that are part of the resource, the expectation is that "most" systems will support the element. i.e. most systems will support capturing a patient's name, gender and date of birth. But "most" does not mean "all". It is possible to be fully FHIR conformant and claim to support the Patient resource, but be incapable of storing any of those data elements. There are use-cases where names may be unnecessary (e.g. agricultural veterinary systems, anonymized reporting, etc.) and similar use-cases for almost every data element. No FHIR resource elements start off as [mustSupport](file:///C:\temp\elementdefinition-definitions.html#ElementDefinition.mustSupport) and very few resource elements start off with a minimum cardinality other than "0".

In the case of extensions, the very nature of a concept being an extension means that the designers of the specification expected that fewer than "most" systems would support the element, though support might vary widely by context. An extension might be used by 100% of systems in some country, discipline, etc. and not used by any systems in another context. In any event, there is no guarantee that an arbitrary receiver will recognize and be able to persist any given extension.

In order to know whether a particular data element is likely to be stored by a given server, a client should check the [Capability statement](file:///C:\temp\capabilitystatement.html) of that server. If, for a given resource, the [StructureDefinition](file:///C:\temp\structuredefinition.html) pointed to indicates that the element or extension is "mustSupport=true", and the server is capable of storing and returning data in general, then it would be expected that the system will be capable of storing and returning that data element. (Some servers such as decision support systems might not be capable of storing or returning any received data.)

All of these concerns around possibly not storing resource elements or extensions can hold whether the data is sent using [REST](file:///C:\temp\http.html), [Messaging](file:///C:\temp\messaging.html) or [Services](file:///C:\temp\services.html). However, with [documents](file:///C:\temp\documents.html), a consuming system is expected to accept the entire contents of the document without losing any information or altering it in any way.

**System behavior**

Even if a system supports all of the data elements provided, not all systems will actually persist the data received or be capable of returning it in response to a query. *mustSupport* indicates that a system supports an element but does not prescribe exactly what the system must do with supported elements. Data might be persisted, displayed, relayed, analyzed, tabulated or used in a variety of other fashions. The behavior of a given system should be unsurprising given its context, but it is still important to recognize that not all systems will persist the data they receive.

**Access permissions**

Even if a system stores a given data element, that does not mean it will always include that element when responding to queries. Systems will have access permssion rules that restrict who can see a given resource instance and, occasionally, who can see a particular data element within a resource. Systems responding to queries might suppress records or may adjust the content of resource instances to exclude data elements the querying system is not permitted to see.

**Generated and inferred data**

Some servers may add additional data elements (or more commonly, extensions) based on information they have generated or inferred from data in the resource, from other resources or other information of which the server is aware. i.e. an instance queried after being created or updated might have *more* information present than was included on the originally submitted record

**Data integration**

Servers receiving updates from multiple sources may choose to be selective about what sources they choose to trust for updates to certain information. For example, a patient registry system might choose to only allow updates to name, gender and date of birth from administrative systems but not clinical systems. If a system filters out patient address information from being disclosed to a system when it queries, it will likely choose to not replace or eliminate the addresses it has on file when it receives an update from that system.

One approach commonly followed by HL7 version 2 messaging interfaces is to not update any elements not included in an instance. For example, if a Patient instance were received with no telecom or contact information, all existing telecom and contact information would be retained and only those elements included in the instance would be updated. This same approach can be followed with FHIR, however, unlike [HL7 V2](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185), FHIR does not have a defined construct to use to indicate that a particular data element should explicitly be set to empty. Systems wishing to fully emulate HL7 v2 behavior will need to use an extension to mirror this behavior.

**Ramifications of storage/retrieval variations**

While changing data prior to storage or prior to returning query results is possible, it should not be considered "normal" behavior. It creates several challenges:

* Digital signatures will no longer be valid. Any change to submitted data will break standard signatures. In theory, this could be mitigated by using a custom canonicalization for the signature that excludes maleable elements. However, this would require both sender and receiver to agree on the alternate signature canonicalization. The signature would still be considered invalid by systems that were not party to the agreement.
* Workflows can be made more complex. A system that sends a create and then sends an update presuming that it has seen the most "current" version, could accidentally overwrite information set by the server as part of the create process.
* Audit becomes more complicated. The [AuditEvent](file:///C:\temp\auditevent.html) resource points to a specific resource version when tracking what data was returned by a query. If the record is modified when it is returned, then the audit record will likely need to be supplemented with information about what elements were filtered.
* Multiple views of the same information by different people (or even by the same people in different contexts) may cause confusion, particularly when attempting to coordinate care across a team. Providing OperationOutcome instances containing warnings if information has been suppressed may be helpful, though doing this may not be possible, depending on security policy rules.
* If data is lost as part of the transmission process, then multiple stages of transmission (or even round-tripping of data) could result in significant degradation as information passes through multiple systems.

**Mitigation approaches for storage/retrieval variations**

FHIR does provide a couple of mechanisms that can help with the issue of a system that has received only partial data overwriting data that was filtered from its record:

* For systems that enforce the use of [ETags](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.19) for updates, the server can choose to not include an ETag when returning resource instances that have been modified in some way. This will prevent updates from being performed.
* The [SUBSETTED](file:///C:\temp\v3\vs\SecurityIntegrityObservationValue\cs.html#SUBSETTED) [Security Label](file:///C:\temp\security-labels.html) can be used to flag data that has had information removed.

Note that using either of the above mechanisms may constitute a security breach in circumstances where the individual or system accessing the data should not know that the record being reviewed has been modified in any way. (The presence of digital signatures would present a similar issue.)

**use.html**

**How to Use Resources**

**Service Orientated use of Resources**

While FHIR Resources are designed with a simple [RESTful HTTP-based implementation](file:///C:\temp\http.html) in mind, it is not necessary to use this implementation framework. This specification also defines a [messaging based implementation framework](file:///C:\temp\messaging.html) for FHIR resources and a [document-based framework](file:///C:\temp\documents.html).

Alternatively, it is not necessary to use any of these approaches. Resources can be exchanged or persisted using any technical means that is appropriate to the context at hand. A common use of FHIR resources or [bundles](file:///C:\temp\bundle.html) is as parameters of service interfaces. FHIR itself does not define any particular service interface. Instead, other standards and implementations define their own service interfaces and architecture that use FHIR resources and optionally build extra features on top of the base repository-mediated exchange that the FHIR RESTful specification provides. As long as the resources that are used are conformant with this specification and the rules for authoring and reading applications are followed, then the implementation can claim conformance to "FHIR Resources". Such implementations will need to resolve several issues:

* Resource identity (the "id" metadata property) must be maintained. Resources all have an identity, which is how other resources refer to them, and these references need to be able to be resolved. However resources are exchanged, their identity - which is not included inside the resource - needs to be included with the resource
* Resource references need to be resolvable. There are a variety of solutions to this, from ensuring that all the relevant resources are bundled together or that all relevant resources are passed as parameters in a service call, through to having a resource repository in the background that provides access to all referenced resources.
* The [Resource metadata](file:///C:\temp\resource.html#metadata) items "Version Id" and "Last Modified Date" are provided for use in resolving resource versioning and concurrency issues, both from a technical and human perspective. Most contexts of use will require at least one if not both of these attributes for some uses, and the implementation framework will need to resolve how and when they are exchanged.
* The [Capability statement](file:///C:\temp\capabilitystatement.html) allows authoring and reading applications to describe their rules concerning the use and contents of a resource. The implementation will need to describe how this Capability statement or some other equivalent fits into the exchange/persistence context.
* How transactional information such as data enterer, author(s), responsible party, consent and approvals is treated

The resolution to these issues should be documented and published with the service specification.

**Managing Resource Identity**

Each resource has a known identity. The identity is not stored inside the resource, but must be tracked by systems handling resources. For RESTful systems, the resource identity is the same as the URL by which it is found. When a resource is packaged in a [bundle](file:///C:\temp\bundle.html), the id is included along with the resource. Real-world use of FHIR resources creates the need to manage resource identification.

Resources are used in a variety of circumstances. Generally, these can be categorized into 3 different scenarios:

1. **Closed Trading System**: the resources are only ever exchanged between fixed systems in a tightly controlled community, such as a hospital. There is only one master server for each resource type, and resources are managed by that server. In this context, the logical id of a resource is sufficient to fully identify the resource
2. **World-wide RESTful system**: there are many peer servers, each managing a set of resources of different types. In order to identify resources, a full URL reference to the origin server is required
3. **Partially closed, inter-linked systems**: a mixture of both - trading communities that are tightly managed, but have managed interactions with other closed trading systems, or with the world-wide RESTful system, or both. In fact, this combination appears to be the most likely scenario for current real-world healthcare business solutions

These combinations show why either relative (logical) or absolute references are allowed, and why a logical id is always required, in order to enable seamless exchange amongst partially closed trading systems.

**Copying Resources and re-identification**

When resources are exchanged between systems, they may need to be re-identified (i.e. assigned a new resource). When a resource is re-identified, nothing in the resource changes, but any references that point to the resource need to be updated. Whether re-identification is required or not depends on the context, as does how resource references are updated.

The normal case is that a client/receiving system accepts the server/sender's identification of a resource at the face value, whether it is a relative or absolute reference. When the client/receiver wants to follow resource references, the server id is used (typically either by http calls or locating them in a [bundle](file:///C:\temp\bundle.html)). In such cases, there is no need for re-identification.

Another scenario is for a client to retrieve a resource from a server, and make its own local persistent copy. If the local resource has a life-cycle of its own (i.e. it is not just a cached resource), then it needs to have its own identity; i.e. the resource must be re-identified. The simplest case is that the client is only keeping local copies of resources from a single server. In these cases, the client can simply replace the Base URL and keep the logical id of the resource the same. In fact, if the server is using relative references, then this change doesn't involve any actual changes to the resources, only a re-interpretation of the references.

In some cases, however, the client may deal with multiple servers. In this case, the logical id of the resource is not guaranteed to be unique (unless all resources have a UUID for the logical id, which is allowed but not required). When the client cannot be sure that the resource identities are unique, it will have to re-identify the resources. In practice this means that the client needs to keep an identity translation table, and update references to the resources it has copied locally when other resources are received.

The case of a gateway system that migrates resources from one ecosystem to another is very similar. In some limited cases, it can leave the logical id of the resources unchanged as resources are copied from one closed system to another. However in more complicated cases, it will have to modify the resource references as resources pass across the gateway.

**Workflow with resources**

There are many ways to implement any particular workflow and there are many ways to use resources to build working systems:

* A RESTful paradigm where resources are exchanged separately using http transactions directly as defined in this specification. Implementations can use both push and pull or a mix of the two
* The resources can be exchanged in messages or some other SOA implementation where the resources form the contents/parameters that are exchanged
* The resources can be "bundled" into documents that are self-contained and complete collections of linked resources and then these documents can be exchanged and/or persisted
* The resources can be embedded in HTML pages or other web content such as content feeds

**usecases.html**

**Common Example Scenarios in FHIR**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

FHIR is a framework standard that defines a common way to solve healthcare problems and provides a set of resources that can be used in many different ways. This page describes how certain common usage scenarios are implemented using the capabilities that FHIR defines. The provided scenarios are examples of usage and are not in any way exhaustive. FHIR can and will be used in a wide variety of circumstances.

In addition, to the information on this page, see also [Resource Guide](file:///C:\temp\resourceguide.html).

**Personal Health Record (PHR)**

In the PHR scenario, an Electronic Medical Record system (EMR, though many other names and acronyms are also used) provides a RESTful API that allows patients to access their own medical record via a common web portal or mobile application, usually provided by a third party. In this scenario, the PHR provider:

* Provides the patient with a login that identifies them (or links the patient record to an external identity provided by OpenID, Facebook, Google, etc.)
* Authenticates the client using an appropriate OAuth server for the login (possibly their own) and restricts the client to viewing records associated with the specific patient (or patients, where appropriate access has been arranged)

The EMR exposes a FHIR server that supports the [search](file:///C:\temp\http.html#search) and [read](file:///C:\temp\http.html#read) operations on the following resources:

1. the [Patient](file:///C:\temp\patient.html) resource in order to provide demographics to the client. When a client searches patients with no search criteria, they get a list of all patients they have access to
2. [search](file:///C:\temp\http.html#search) and [read](file:///C:\temp\http.html#read) on the [Document Reference](file:///C:\temp\documentreference.html) resource to provide access to general patient documents in the form of PDFs etc. (PDFs are preferred)
3. [search](file:///C:\temp\http.html#search) and [read](file:///C:\temp\http.html#read) on a set of [clinical resources](file:///C:\temp\clinicalsummary-module.html)

Here is the Capabilities Statement for this scenario: [XML](file:///C:\temp\capabilitystatement-phr-example.xml.html) or [JSON](file:///C:\temp\capabilitystatement-phr-example.json.html).

The EMR may also choose to provide additional functionality, such as shared access to patient records by relatives/carers, to allow the patient to upload their own documents, medication statements, observations (e.g. from patient monitoring devices) and/or to allow the patient to make appointments. This additional functionality will involve additional API capabilities to be implemented and exposed. The EMR server may also choose to expose the [search](file:///C:\temp\http.html#search), [read](file:///C:\temp\http.html#read) and [history](file:///C:\temp\http.html#history) operation on the [AuditEvent](file:///C:\temp\auditevent.html) resource for the patient-specific records to allow patient review of record access. Note that all usage of the RESTful API should be logged in an AuditEvent resource.

**Document Sharing (XDS)**

One common way to integrate healthcare information from a variety of sources is to build a repository of documents around a patient record. Building a repository of documents allows for less stringent alignment around policy, procedures and record-keeping/informatics standards.

The most widely adopted framework for sharing documents within institutions, regions, states or countries is IHE Cross-Enterprise Document Sharing (XDS). XDS allows for a federated system of repositories with a registry to provide coordinated access to the documents.

FHIR provides equivalent functionality to XDS that can be used to implement XDS behind the existing XDS.b interface, to provide a simpler mobile-friendly interface to an existing XDS ecosystem, or to link document sharing into other functionality provided through a FHIR interface.

The following FHIR Resources are involved in the XDS functionality:

* The [DocumentReference](file:///C:\temp\documentreference.html) resource describes a document that is located elsewhere. A document registry is a system that maintains a set of Document References
* The [Binary](file:///C:\temp\http.html#binary) support can be used to store the actual documents on a FHIR server. A repository is a system that stores the binary document in addition to Document References (or sometimes without)
* [Patient](file:///C:\temp\patient.html), [Practitioner](file:///C:\temp\practitioner.html) and [Organization](file:///C:\temp\organization.html) resources provide support for identifying people and organizations
* The [AuditEvent](file:///C:\temp\auditevent.html) resource tracks usage of the document registry and repository
* The [DocumentManifest](file:///C:\temp\documentmanifest.html) resource describes a set of documents that are grouped together in original publication

At present, IHE is working with the FHIR project team to use FHIR for Mobile Health Documents (MHD).

**Decision Support**

One common use of healthcare information systems is to integrate some form of decision support software into clinical systems. Common uses of clinical decision support are:

* Drug-drug interaction checking, and more generally, prescription safety checks
* Suggesting commonly missed diagnostic data interpretations (including delta checking)
* Patient surveillance for early warning of deteriorating patient health (both acute and ambulatory care)
* Identifying candidates for alternative treatment plans for improved efficacy

Note that in addition to clinical decision support, there are also infrastructural uses, such as managing access control.

The various forms of decision support each involve different interaction patterns, so there is no single decision support implementation in the FHIR specification. Generally, the patterns fall into several classes:

1. The decision comes from an engine entirely hidden behind a system interface and has no direct impact on the data exchange
2. The decision support engine uses existing data and generates alarm messages concerning patient state that are visible on the FHIR interface
3. The decision support engine is consulted through a described interface; it accepts a request for, and returns a decision

Any decision support may fall into multiple categories at once, depending on the perspective of a particular system.

1. There is no particular support required from the FHIR specification, though there will be ongoing review of the contents of the resources to ensure that they support common decision support practices appropriately
2. There is no suitable resource for this use yet. The [Flag](file:///C:\temp\flag.html) resource is intended for clinical notes about the patient, and is not intended for this use. A resource called "Alarm" is under preparation for this purpose
3. A request for a decision support is understood as a [search](file:///C:\temp\search.html) using a named \_query that takes a set of parameters. See below

**Explicit Requests for Decisions**

When a query is initiated in order to get a decision made, the following considerations apply:

**Request**

* The request for a decision is made using one of the interaction patterns described for [search/query](file:///C:\temp\search.html): A RESTful search, a query posted to [/Mailbox](file:///C:\temp\messaging.html#mailbox), a query [message](file:///C:\temp\messaging.html), or the Asynchronous query pattern
* The request has a \_query parameter that identifies the decision that is being requested
* The request also has a set of parameter values. These parameter values may be the data that describes the decision being made or they may be references to specific resources that contain the request. In general, the more complex the decision request is, the more likely it is that a full resource is appropriate, particularly since this provides a ready made way to record and manage the requests themselves.
* In some of the query interaction patterns, the resources identified in the parameter value can be bundled up with the request. In others, only the references can be passed
* Which of the query patterns is most appropriate depends on the complexity of the decision support input, and the length of time the decision is expected to take. As either of these increases, the more complex query patterns become more appropriate

**Response**

* If the decision support engine is unable to provide the requested decision, it returns an [OperationOutcome Resource](file:///C:\temp\operationoutcome.html) describing the issue
* Otherwise it returns a resource that represents the decision, along with other resources as supporting information, as described by the resource, or applicable profiles
* In principle, the decision provider can choose to make a copy of the returned decision resource available through a normal RESTful interface, or it can choose not to. This decision may be constrained by applicable profiles, policy decision, or the innate nature of the query
* If either the decision provider or the requester choose to retain a copy of the decision, they must ensure that the (lack of) currency of the decision is appropriately considered when it is used

It follows from this then, that decisions that may be requested need at least a response resource defined, and possibly a request resource. This table summarizes known decisions for which resources have been defined.

|  |  |  |
| --- | --- | --- |
| **Decision** | **Resources** | **Invocation** |
| What immunizations should this patient have? | Response: [ImmunizationRecommendation](file:///C:\temp\immunizationrecommendation.html) | The exact way to invoke this decision is not yet defined |

Implementers are allowed to use existing resources for decisions not documented here, but there is no guarantee that they will be suitable. Improving decision support will be a focus for ongoing development during the Trial Use period.

**validation.html**

**Validating Resources**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

This page provides a quick overview of how the FHIR specification supports validation of resources.

Validating a resource means, checking that the the following aspects of the resource are valid:

* **Structure**: Check that all the content in the resource is described by the specification, and nothing extra is present
* **Cardinality**: Check that the cardinality of all properties is correct (min & max)
* **Value Domains**: Check that the values of all properties conform to the rules for the specified types (including checking that enumerated codes are valid)
* **Coding/CodeableConcept bindings**: Check that codes/displays provided in the [Coding](file:///C:\temp\datatypes.html#Coding)/[CodeableConcept](file:///C:\temp\datatypes.html#CodeableConcept) types are valid
* **Invariants**: Check that the [invariants](file:///C:\temp\conformance-rules.html#constraints) (co-occurance rules, etc) have been followed correctly
* **Profiles**: Check that any rules in [profiles have been followed](file:///C:\temp\profiling.html) (including those listed in the [Resource.meta.profile](file:///C:\temp\resource.html#Meta), or in [CapabilityStatement](file:///C:\temp\profiling.html#profile-uses), or in an [ImplementationGuide](file:///C:\temp\implementationguide.html), or otherwise required by context)
* **Questionnaires**: Check that a [QuestionnaireResponse](file:///C:\temp\questionnaireresponse.html) is valid against it's matching [Questionnaire](file:///C:\temp\questionnaire.html)
* **Business Rules**: Business rules are made outside the specification, such as checking for duplicates, checking that references resolve, checking that a user is authorized to do what they want to do, etc.

There are multiple ways to validate resources. This table summarizes the options described in this specification, and which of the aspects above they can validate:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Method** | **XML** | **JSON** | **RDF** | **Structure** | **Cardinality** | **Values** | **Bindings** | **Invariants** | **Profiles** | **Questionnaires** | **Business Rules** |
| [XML Schema](file:///C:\temp\main-pages.html#schema) |  |  |  |  |  |  |  |  |  |  |  |
| [XML Schema + Schematron](file:///C:\temp\main-pages.html#schema) |  |  |  |  |  |  |  |  | **1** |  |  |
| [JSON Schema](file:///C:\temp\main-pages.html#json-schema) |  |  |  |  |  |  |  |  | **2** |  |  |
| [ShEx](file:///C:\temp\rdf.html#schema) |  |  |  |  |  |  | **3** |  |  |  |  |
| [Validator](file:///C:\temp\main-pages.html#jar) |  |  |  |  |  |  |  |  |  |  |  |
| [Validation Operation](file:///C:\temp\main-pages.html#op)**4** |  |  |  |  |  |  |  |  |  |  |  |

Notes:

1. Schematron generated for a profile can test cardinality and invariants, but not bindings, and slicing is not really supported well
2. JSON schema generated for a profile can test cardinality, and slicing is partially supported
3. ShEx can enforce some bindings for well understood terminologies, but this is an ongoing area of development
4. It is at the discretion of the server how much validation to perform, but most servers use the validation jar, or code derived from it, and offer the same services. Some servers also offer [a web interface](file:///C:\temp\main-pages.html#web)

Note that all these validation methods are incomplete; they can only validate the computable aspects of conformance, but there are always additional rules made in narrative that they are not able to check (e.g. a rule such as "All the clinically important content in the data SHALL be in the narrative", which might be made in an implementation guide, but could never be checked by a conformance tool).

In case of disagreement between these conformance methods, note that:

* The schema/schematron is the least capable - mainly because it is not connected to a terminology server
* The java validator is only as good as the underlying definitions, and in particular depends on whether the underlying terminology server supports all the relevant terminologies
* In general, the server validation operations use or derive from the java validation code, so have the same caveats
* The final arbiter is human inspection of the content of the resources, and the relevant implementation guides and base specification

Also, note that static testing of resource content is not enough to prove conformance to the specification. For further information, see [FHIR Conformance Testing](http://fhir.org/conformance-testing).

**Using the XML schema**

The XML schema can be used to validate XML representations of the resources. When validating a resource, you can nominate one of the following schema:

* fhir-all.xsd: links in all the individual modular schemas
* fhir-single.xsd: a single large file, mainly provided for schema processors that can't support circular references

In addition, the validation schema includes schematron that can be initiated with transform "iso\_svrl\_for\_xslt2.xsl" included in the [XML Tools](file:///C:\temp\downloads.html#refimpl) download. Note that XSLT2 is required to run the schematrons.

When running the schematron, use the file "fhir-invariants.sch". This includes all the schematrons. The individual schematron files for each resource are provided to allow implementers to build their own smaller combined file that covers the relevant resource types for them.

**Using the JSON schema**

The JSON schema can be used with JSON schema validation software. Links:

* [JSON Schema](file:///C:\temp\fhir.schema.json.zip)
* [JSON Validation Software](http://json-schema.org/implementations)

**Using the FHIR Validator**

The FHIR Validator is a Java jar that is provided as part of the specification, and that is used during the publication process to validate that all the published examples. To execute the FHIR validator, follow the following steps:

* [Download](file:///C:\temp\downloads.html) the FHIR Validator
* (optional) [Download](file:///C:\temp\downloads.html) One of the FHIR definitions (with or without text)
* Execute the validator, providing the path to the definitions, and a reference to the resource to validate

Here is an example windows batch file that demonstrates the process (using the common utilities [wget](http://gnuwin32.sourceforge.net/packages/wget.htm) and [7z](http://www.7-zip.org/):

@ECHO OFF

ECHO get the validator and unzip it

wget /validator.zip

7z.exe x validator.zip

ECHO Get the validation source file (dictionary)

wget /igpack.zip

ECHO 1. First example shows how to validate against the base spec:

ECHO a. get an example to validate

wget /patient-example.xml -O pat-ex.xml

ECHO b. validate it

java -jar org.hl7.fhir.validator.jar pat-ex.xml -defn igpack.zip

ECHO 2. Second example shows how to validate against a profile in the spec:

ECHO a. get an example to validate

wget /observation-example-heart-rate.xml -O obs-ex.xml

ECHO b. validate it

java -jar org.hl7.fhir.validator.jar obs-ex.xml -defn igpack.zip -profile http://hl7.org/fhir/StructureDefinition/heartrate

ECHO 3. Third example shows how to validate against a profile in an implementation guide:

ECHO a. get an example to validate

wget /observation-example-heart-rate.xml -O obs-ex.xml

ECHO b. validate it. note that you have to tell the validator where to get the implementation guide information

java -jar org.hl7.fhir.validator.jar obs-ex.xml -defn igpack.zip -ig http://hl7.org/fhir/us/core -profile http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient

ECHO Press Any Key to Close

pause

Note that it is not necessary to download the resource first; the http address can be supplied directly:

java -jar org.hl7.fhir.validator.jar /patient-example.html -defn validation-min.xml.zip -profile http://hl7.org/fhir/StructureDefinition/Patient

The validator requires an underlying terminology server. By default, this is http://fhir3.healthintersections.com.au. Use the parameter -tx to specify another server. For a full list of parameters and options, just run the validator.jar without any parameters.

**Asking a FHIR Server**

The [operation](file:///C:\temp\operations.html) [validate](file:///C:\temp\resource-operations.html#validate) can be used to check whether a resource conforms to a profile. The simplest way to execute this operation is to post the resource to a server:

POST [base]/Observation/$validate?profile=http://hl7.org/fhir/StructureDefinition/heartrate

[other HTTP headers]

<Observation>... resource to check as the body

The server will return an [OperationOutcome](file:///C:\temp\operationoutcome.html) resource listing issues found in the resource.

There are several things to consider when using this operation:

* Not all servers support the $validate operation, though some of the public test servers do
* Servers support the $validate operation generally will only validate against profiles already registered with the server
* Servers may choose to support either XML, JSON, or both

**Via a web interface**

Some servers expose the $validate functionality though a web page. For known public implementations, see the [FHIR wiki](http://wiki.hl7.org/index.php?title=Public_FHIR_Validation_Services)

**w5.html**

**FHIR W5 Report**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Concise

Full

**wglist.html**

**FHIR Workgroup List Report**

[See also the QA Page](file:///C:\temp\qa.html)

**workflow-ad-hoc.html**

* [Overview](file:///C:\temp\workflow.html)
* [Communication Patterns](file:///C:\temp\workflow-communications.html)
* [Ad-hoc Workflow Patterns](file:///C:\temp\main-pages.html)
* [Workflow Management Patterns](file:///C:\temp\workflow-management.html)
* [Examples](file:///C:\temp\workflow-examples.html)

**Ad-Hoc Workflow Communication Patterns**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

*TODO: Discussiion (or reference to one) on Polling and Subscription*

**Option A: Simple RESTful POST or PUT**

**Steps**

1. The placer makes a RESTful call to [create](file:///C:\temp\http.html#create) or [update](file:///C:\temp\http.html#update) a record or a POST to invoke an [operation](file:///C:\temp\operations.html#executing) over HTTP
2. The receiver responds with a 2xx HTTP response indicating whether the request was successfully processed or not and, if appropriate, provides the response to the request in the payload of the HTTP response

**Benefits**

* Simplest of all the possible workflow architectures
* Placer knows whether the request was accepted or not and knows when the task has been done

**Limitations**

* Only works for automated execution where the decision to perform the request and the execution of the request can be done synchronously within the HTTP timeout period (generally on the order of 10s of seconds).
* Requires that the placer have authority to post directly to the placer's system
* Requires that the "request" be expressible as a simple creation, update or operation invocation
* Only works for "fulfillment" requests for Request resources - can't handle request for state changes or information

**Usage Recommendations**

This is by far the most common pattern in FHIR for simple changes as it requires the least overhead. However, it covers only automated responses, and does not support any actual workflows of more than one step. If human processing is involved in the request execution, then this approach won't suffice. This approach is listed here to make sure that implementers consider whether they can make this one work first before falling back to one of the more sophisticated patterns.

**Usage Examples**

*???*

**Option B: Direct POST of request to fulfiller's system**

**Steps**

1. Placer system invokes a [create](file:///C:\temp\http.html#create) by POSTing a 'request' resource (e.g. [MedicationRequest](file:///C:\temp\medicationrequest.html), [ProcedureRequest](file:///C:\temp\procedurerequest.html), [ReferralRequest](file:///C:\temp\referralrequest.html), etc.) to the appropriate RESTful resource endpoint (e.g. [base]/MedicationRequest) on the filler system and places an [actionable](file:///C:\temp\codesystem-common-tags.html#common-tags-actionable) [tag](file:///C:\temp\resource.html#simple-tags) on the resource that indicates the request is intended to be acted upon, not merely stored.
2. The filler synchronously responds with a "201" indicating that that they have received and stored (created) the resource on their system
3. At some later point, the filler POSTs an 'event' resource (e.g. [MedicationDispense](file:///C:\temp\medicationdispense.html), [DiagosticReport](file:///C:\temp\diagnosticreport.html), [Encounter](file:///C:\temp\encounter.html), etc.) to the appropriate resource endpoint on the placer system, including a basedOn link to the 'request' resource that the action was performed in fulfillment of.
4. The placer system synchronously responds with a "201" indicating they've received and store (created) the resource on their system

**Benefits**

* Lowest amount of overhead. No need for [Task](file:///C:\temp\task.html). No need for polling or subscriptions
* Explicit acknowledgement that filler has received the request

**Limitations**

* Can only use when requesting fulfillment (can't use to request status change or other updates)
* Placer and filler must be able to communicate directly (i.e. know each other's respective endpoints) and must each have a FHIR server and must have "write" permissions to each other's servers. This could become unmanageable if there are a large (or dynamic) number of placers and fillers that need to communicate
* No indication of agreement to act on the request
* There's no ability to negotiate fulfillment - no ability to say "no"
* This completely reverses the usual provenance of resources, as the request only exists on the fulfiller's system, and the event only exists on the placer's system. This greatly limits the use of these resources for any other purpose.

**Usage Recommendations**

Use this approach when there's no ability to have queue servers and no support/need for complexity of Task, polling or pub/sub (and no need for negotiation or the ability for the filler to say "no"). This is a pseudo-messaging architecture that doesn't actually use messaging architecture.

**Usage Examples**

**Option C: POST of request to placer/queue server system, receiver uses polling or subscribtion**

**Steps**

1. Placer system creates a 'request' resource (e.g. MedicationRequest, DiagnosticOrder, ReferralRequest, etc.) by a system action or a POST on either its own system or a third party queue server system and places an "actionable" tag on the resource that indicates the request is inteneded to be acted upon. The request explicitly identifies the intended fullfiller
2. The filler system uses polling (or a subscription) at the placer or queue server system to see if there are any "new" requests that: are tagged as "actionable", have the filler identified as the intended performer, and are a type of request "of interest" to the filler.
3. At some later point, the filler creates an 'event' resource (e.g. MedicationDispense, DiagosticReport, Encounter, etc.) via a system action, or a POST on either its own system, the same queue server as the request was placed on, or some alternate queue server, including a link to the 'request' resource that the action was performed in fulfillment of
4. The placer system uses polling (or a subscription) to the filler or queue server system to see if there are any "new" events that are tied to any outstanding requests the placer has initiated

**Benefits**

* Placer and fulfiller don't have to communicate directly (can act through queue server). This can reduce the number of point-to-point interfaces that need to be supported
* Preserves the conventional management and ownership of the request and event at the placer and fulfiller's systems respectively
* No need for Task

**Limitations**

* Can only use when requesting fulfillment (can't use to request status change or other updates)
* Additional complexity of using subscription or polling
* Polling by the placer for "anything related to these 500 open orders" could be onerous, especially if some orders never get closed.
* Placer and fulfiller must know where to poll or subscribe for content - this could be a large number of systems
* No indication of agreement to act on the request
* There's no ability to negotiate fulfillment - no ability to say "no"
* Placer may not know when (or if) filler system has retrieved the request

**Usage Recommendations**

This pattern could be used when there's no support/need for complexity of Task, usually for simple two step wrkflows. This is a more typically RESTful approach where data resides on the server "owned" by the data creator and is accessed by other systems.

**Usage Examples**

**Option D: Messaging request from placer to filler & acknowledgment**

*TODO: needs review and update*

**Steps**

1. Placer sends message to filler system including Request resource (and other relevant resources) along with a MessageHeader with an "event" code saying "please fulfill" and "data" element pointing to the Request resource as the item to fulfill. Message could potentially use Task instead of MessageHeader.event to convey desired action (ongoing discussion)
2. Filler system sends a response indicating receipt of the message and, optionally an indication of their intention to fulfill the request
3. Filler system may send incremental messages to the placer showing progress (e.g. specimen collected, preliminary results, final results)

**Benefits**

* Reduced number of communications
* All relevant data sent in one package
* Responses can be asynchronous and content may routed
* There's an ability to negotiate fulfillment - i.e. the ability to say "no"
* Can request things other than just fulfillment (e.g. please suspend)
* Explicit acknowledgement that filler has received and agreed to act on the request (though no need for the placer to check)

**Limitations**

* Messaging is "heavy"
* Need to negotiate what allowed responses are and what data can be present in request and response messages
* Additional complexity of setting up and maintaining a subscription or polling infrastructure
* Additional complexity of using Task
* Need message delivery infrastructure in place

**Usage Recommendations**

Existing messaging infrastructure (e.g. v2 LTP, MLTP, WSI Web Services, Direct, VISA, REST, etc.) and a need to stay consistent with that architecture

**Option E: Services request from placer to filler & acknowledgment**

*TODO: Needs review, update and more work - there's not a lot of experience using FHIR services to manage the fulfillment process*

**Steps**

1. Placer may create and store a Request resource on their own system or a queue server.
2. Placer invokes a service on the filler system saying "please fulfill this order", including the content or a reference to the request resource and any other relevant data
3. Filler system responds (synchronously if using HTTP, but may be asynchronous if using SOAP or other transport mechanisms) with conformation of receipt and, optionally indication of intention to fulfill and/or results

**Benefits**

* ???

**Limitations**

* ???

**Usage Recommendations**

TBD

**Additional Scenarios**

*TODO: needs review and udate. Posibly add options about usiing messaging and/or services instead of polling/subscription in above scenarios*

**POST of "request" resource for filler system, response via Task**

This is a variation of[Option H](file:///C:\temp\workflow-management.html#optionh), where the Workflow broker is essentially merged with the fulfiller. It still allows the placer to only use a POST of the request, and be made aware of the changes to the other resources via subscpriotn or polling.

1. Placer system invokes a "create" action by POSTing a 'request' resource (e.g. MedicationRequest, DiagnosticOrder, ReferralRequest, etc.) to the appropriate RESTful resource endpoint (e.g. [base]/MedicationRequest) on the filler, placer or queue server system and sets a "tag" on the resource that indicates the request is "actionable"
2. Filler POSTs a Task resource to its own system or a queue server system, pointing to the request resource and indicating intent to fulfill or refusal to fulfill
3. Placer system uses either polling or pub/sub to become aware of the existence of the task and fulfillment intent
4. Fulfiller may update the Task to indicate interim progress notes
5. Placer either polls the Task to note acceptance and changes or uses a subscription to determine the same
6. Fulfiller POSTs an event resource to its own system or to a queue server system
7. Fulfiller Updates the Task resource to change its status to completed and to point to the event resource
8. Placer system becomes aware of the update via polling or subscription
9. Placer system retrieves the event
10. Placer system marks the request as "complete"

**Querying the status of a workflow using REST**

1. Placer sends query for Task(s) that have a focus of the request of interest to a system (placer system, queue server or filler) that holds tasks related to their request.
2. System returns a query response showing all related tasks (typically just one). Task shows current status.

**Querying the status of a workflow using services**

1. Placer invokes a "what's the status of this order" service, passing the request business identifier or URL of the request
2. Services responds with a Task showing the current state of the fulfillment of the request

**Cancellation of a Task using REST - placer owns**

1. Placer sends an update to the Task setting the status to "cancelled" signaling they are no longer concerned about the outcome of the task
2. Filler receives notification of the update (because the task is on their system or because they poll it or are subscribed to it) and ceases work if they are able

**Cancellation of a Task using REST - filler owns**

1. Placer creates a new task requesting cancellation of the original fulfillment task  
   Fulfillment of the "cancellation task" can be requested using any of the mechanisms above
2. Filler decides whether they are able to cancel the task and update the "cancellation" task to indicate either cancellation is complete or has been refused

**workflow-communications.html**

* [Overview](file:///C:\temp\workflow.html)
* [Communication Patterns](file:///C:\temp\main-pages.html)
* [Ad-hoc Workflow Patterns](file:///C:\temp\workflow-ad-hoc.html)
* [Workflow Management Patterns](file:///C:\temp\workflow-management.html)
* [Examples](file:///C:\temp\workflow-examples.html)

**Workflow Execution and Communication Patterns**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

As described in previous sections, many FHIR resources can be used within workflows without using FHIR to manage the execution of the workflows. When we want to manage workflow execution with FHIR, there is a number of mechanisms available. In addition to managing workflow with FHIR, there are also cases where management of workflow execution is not necessary, and supporting workflow becomes simply the use of the [resource paterns](file:///C:\temp\workflow.html#respatterns) in an ad-hoc fashion. In deciding how best to interoperate around workflow with FHIR, there are a number of considerations:

* Is sharing of the state of the workflow necessary among the participants?
* Which paradigm do you want to use (REST, messaging, services, a mix)?
* Who owns/manages the various resources involved in the workflow (placer, filler, another participant)?
* Is there infrastructure in place to support polling, push notifications via subscriptions or both?
* Is there a need for confirmation that the desired performer agrees to act, or can that be presumed?
* Is there a need to negotiate whether/how the requested action will be performed?
* Can the requesting and performing system communicate directly? Are they able to post to each other's servers (if using REST)?
* Is there an ability/need to have a queue server to facilitate workflow execution?
* How many potential actors are involved?
* Will the workflow always be directed or is there a pool of potential performers who could choose to perform the requested action?

The answers to these (and other) questions will guide the selection of communication patterns to be used for a specific interoperability use case. It is recommended that domain workgroups analyze the support for workflow execution within their domains and provide recommendations for an appropriate subset of patterns, with a discussion on where and how to use them. Implementation guides, by their nature, should prescribe which patterns to use for workflow execution management.

**Communication Patterns Conventions**

This section highlights some of the more common patterns and identifies their characteristics and limitations and provides recommendations on when each approach may be most useful or relevant. Please note that this list of patterns is not exhaustive. Patterns can be combined in various ways and there are likely some possibilities we haven't thought about yet (feel free to submit additional patterns using the 'submit a change' link at the bottom of the page). As well, the recommendations given here are just that - recommendations. Implementers are free to choose which patterns they wish to support. Because of this, tight interoperability around workflow execution (as with any other tight interoperability using FHIR) will depend on communicating participants doing some up-front negotiation around how they plan to support workflow execution or all communicating partners will need to adhere to an implementation guide that sets out clear interoperability expectations.

Prior to reviewing this list of options, readers are encouraged to be familiar with the following pages and resources: [REST](file:///C:\temp\http.html), [messaging](file:///C:\temp\messaging.html), [operations](file:///C:\temp\operations.html), [services](file:///C:\temp\services.html) and the [Subscription](file:///C:\temp\subscription.html) resource.

The scenarios used to illusttrate the patterns below make use of a few conventions:

* The focus here is on a "request" and the actioning of that request. Almost all workflows can be broken down to a sequence of these steps, though the responsibilities of the different parties may shift for each interaction and there can be more than two parties involved in the overall workflow
* The request could be as simple as "please look at this information" and the response could be as simple as an implicit "it's been looked at" or the request could be for some more involved action that may include reporting back multiple interim and final steps
* The requester is referred to as the "placer" and the performer is referred to as the "filler", which are often seen as order-specific terms. However, in this context, the terms hold whether the request is expressed as a proposal, plan or full-blown order
* Each of the patterns defines the set of steps involved in processing the request, lists some of the benefits and limitations associated with the approach, provides a scenario to illutrate the use of the pattern, and then makes recommendations about when the pattern is most appropriate
* The descriptions of these patterns focuses on the notion of requesting fulfillment of a request. However most of these patterns are also applicable to requests for status change, requests for information, etc. If a pattern is limited in the types of execution it can trigger, this will be noted in the "limitations" section.

**Workflow State**

One of the key distinguishing characteristics in the patterns below is whether the pattern supports the tracking of the wrokflow's state by both the placer and the filler. The workflow state is represented by the [Task](file:///C:\temp\task.html) resource. The [Task state machune](file:///C:\temp\task.html#statemachine), which is also presented here, shows the use of [Task.status](file:///C:\temp\task-definitions.html#Task.status) to represent the general infrastrcuture state of the Task resource.

The combination of [Task.status](file:///C:\temp\task-definitions.html#Task.status), [Task.statusReason](file:///C:\temp\task-definitions.html#Task.statusReason) and [Task.businessStatus](file:///C:\temp\task-definitions.html#Task.businessStatus) provide the mechanism for tracking the workflow state. The following two examples illustrate how that could be accomplished with regards of specific workflows. Note that these examples avoid any specificity on where the described resources exist, or any reference to a specific communication pattern. The goal of the examples is to demostrate the use of the Task attributes to represent the workflow state.

**Example: Workflow states for a lab order for a blood test**

The following table shows the steps for a simple blood test order, and the corresponding values of the Task attributes. A lot of details on other attributes have been omitted in order to focus on the state changes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Workflow step** | [**Task.status**](file:///C:\temp\task-definitions.html#Task.status) | [**Task.statusReason**](file:///C:\temp\task-definitions.html#Task.statusReason) | [**Task.businessStatus**](file:///C:\temp\task-definitions.html#Task.businessStatus) |
| Provider orders a blood test for a patient. A [ProcedureRequest](file:///C:\temp\procedurerequest.html) resource is created to represent the order, and a [Task](file:///C:\temp\task.html) resource is created to represent the request to the lab. [Task.baasedOn](file:///C:\temp\task-definitions.html#Task.basedOn) points to the DiagnsticRequest instance. | Requested | New order | Ordered | N/A |
| The lab accepts the order. | Accepted | Able to perform the test. | Accepted | N/A |
| The patient is at the lab, and blood is drawn. | In progress | Obtained the specimen | Specimen available |  |
| Work on the blood sample is in progress, preliminary results are available, a [DiagnosticReport](file:///C:\temp\diagnosticreport.html) resource is created, and [Task.output.value[1]](file:///C:\temp\task-definitions.html#Task.output.value_x_) references the DiagnosticReport resource. | In progress | Preliminary results available | Preliminary results |  |
| Final results are available | Complete | Final results are available | Final results |  |

**Example: Workflow states for a referral from a PCP to a specialist**

The following table shows the steps for a referral workflow, and the corresponding values of the Task attributes. A lot of details on other attributes have been omitted in order to focus on the state changes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Workflow step** | [**Task.status**](file:///C:\temp\task-definitions.html#Task.status) | [**Task.statusReason**](file:///C:\temp\task-definitions.html#Task.statusReason) | [**Task.businessStatus**](file:///C:\temp\task-definitions.html#Task.businessStatus) |
| Provider A is seeing a patient with a specific complaint. Based on the exam, Provider A refers the patient to a specialist at clinic B. It is an urgent referral, requesting to see the specialist within 4 days. A [ReferralRequest](file:///C:\temp\referralrequest.html) resource is created to represent the referral order, and a [Task](file:///C:\temp\task.html) resource is created with [Task.priority](file:///C:\temp\task-definitions.html#Task.priority) set to Urgent, and [Task.restriction.period.end](file:///C:\temp\task-definitions.html#Task.restriction.period) set to 4 days in the future. | Requested | New referral | Referred | N/A |
| Clinic B receives the referral and puts it on the urgent queue to be reviewd. | Received | Received by organization | Pending | N/A |
| Upon reviewing the referral, the clinic determines they have no capacity to see the patient within the specified time period, and they decline the referral. | Rejected | Provider unavailable | Declined | N/A |
| Provider A changes the referral request to clinic C. This creates a new [Task](file:///C:\temp\task.html) resource, with the appropriate information | Requested | Changed the referred to provider | Referred | N/A |
| Clinic C accepts the referral and it is placed on a schedular's queue. | Accepted | Able to provide the service. | Accepted | N/A |
| An appointment is scheduled for the patient at Clinic C. | In progress | Appointment scheduled | Scheduled |  |
| Soon after the appointment is scheduled, the patient calls provider A to ask for a different location for the referral appointment, as he lost his transportation, and he cannot reach Clinic C. The provider requests a cancel for the referral. | Suspended | Patient unable to make appointment. | Cancel requested |  |
| Clinic C confirms tha cancelation of the existing task. | In progress Failed | Referral cancelled | Cancelled as requested |  |
| Provider A changes the referral to Clinic D, which is located close to where the patient lives, and can be conveniently reached using public transportation. This creates a new [Task](file:///C:\temp\task.html) resource, wioth the appropriate information. | Requested | Changed the referred to provider | Referred | N/A |
| Clinic D accepts the referral and it is placed on a schedular's queue. | Accepted | Able to provide the service. | Accepted | N/A |
| An appointment is scheduled for the patient at Clinic D. | In progress | Appointment scheduled | Scheduled |  |
| Patient is seen at Clinic D. A diagnosis is made by the specialist, a prescription is given, and a follow-up appointment is made at clinic D. [Task.output.value[1]](file:///C:\temp\task-definitions.html#Task.output.value_x_) references a [Compositions](file:///C:\temp\composition.html) resource, which contains the encounter summary. | In progress | Intermediary consultation note available | Preliminary notes |  |
| Patient comes for follow-up appointment at clinic D. The specialist is satisfied by the progress the patient has made and creates the final consultation note. [Task.output.value[2]](file:///C:\temp\task-definitions.html#Task.output.value_x_) references a [Compositions](file:///C:\temp\composition.html) resource, which contains consultation note. | Completed | Final consultation note available | Final notes |  |

**List of patterns**

The patterns that facilitate the execution of workflow using the [Task](file:///C:\temp\task.html) resource are grouped in the Workflow Management Patterns section. The patterns where no Task resource is used are grouped in the Ad-hoc Workflow Patterns section. The list of patterns is as follows:

|  |
| --- |
| [**Ad-Hoc Workflow Patterns**](file:///C:\temp\workflow-ad-hoc.html) |
| [Option A: Simple RESTful POST or PUT](file:///C:\temp\workflow-ad-hoc.html#optiona) |
| [Option B: Direct POST of request to fulfiller's system](file:///C:\temp\workflow-ad-hoc.html#optionb) |
| [Option C: POST of request to placer/queue server system, receiver uses polling or subscription](file:///C:\temp\workflow-ad-hoc.html#optionc) |
| [Option D: Messaging request from placer to filler & acknowledgment](file:///C:\temp\workflow-ad-hoc.html#optiond) |
| [Option E: Services request from placer to filler & acknowledgment](file:///C:\temp\workflow-ad-hoc.html#optione) |
| [**Workflow Management Patterns**](file:///C:\temp\workflow-management.html) |
| [Option F: Creation of Task on placer's system](file:///C:\temp\workflow-management.html#optionf) |
| [Option G: POST of Task on fulfiller's system](file:///C:\temp\workflow-management.html#optiong) |
| [Option H: POST of Task to a workflow broker](file:///C:\temp\workflow-management.html#optionh) |
| [Option I: POST of Task on fulfiller's system, POST of sub-Task on placer's system](file:///C:\temp\workflow-management.html#optioni) |
| [Option J: Messaging Task from placer to fulfiller](file:///C:\temp\workflow-management.html#optionj) |
| [Option K: Service request referencing Task from placer to fullfiller](file:///C:\temp\workflow-management.html#optionk) |
| [Additional Scenarios and Variations](file:///C:\temp\workflow-ad-hoc.html#additional) |

TODO: Insert Jose's decision tree here?

**workflow-examples.html**

* [Overview](file:///C:\temp\workflow.html)
* [Communication Patterns](file:///C:\temp\workflow-communications.html)
* [Ad-hoc Workflow Patterns](file:///C:\temp\workflow-ad-hoc.html)
* [Workflow Management Patterns](file:///C:\temp\workflow-management.html)
* [Examples](file:///C:\temp\main-pages.html)

**Usage Examples for Workflow Patterns**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

*Future home for examples*

**workflow-management.html**

* [Overview](file:///C:\temp\workflow.html)
* [Communication Patterns](file:///C:\temp\workflow-communications.html)
* [Ad-hoc Workflow Patterns](file:///C:\temp\workflow-ad-hoc.html)
* [Workflow Management Patterns](file:///C:\temp\main-pages.html)
* [Examples](file:///C:\temp\workflow-examples.html)

**Workflow Management Communication Patterns**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

*TODO: Discussiion (or reference to one) on Polling and Subscription*

**Option F: Creation of Task on placer's system**

**Steps**

1. Placer creates the request in its own system via POST or an internal action, or POSTs it to a queue server system
2. Placer creates a Task resource in its own system via POST or an internal action, or POSTs it to a queue server system, pointing to the request resource and seeking fulfillment.  
   The Task may have a specified performer, in which case step 3 is expected to be done by that performer.  
   If the Task does not have a specified "performer" (although may have performer type), then this is a case of an "open" task, where any number of fulfillers may attempt to "claim" the task. Who succeeds is determined by local policies and procedures.
3. Fulfiller system uses either polling or pub/sub to become aware of the existence of the task
   1. A common case may be the conveyance of the Task id to a fullfiller by other means. For example, a lab test is ordered, and the patient takes a requisition to the lab of his choice. The requisition contains the Task id (as a bar code, or stored in the patient's healthcre smart card), and the lab system can execute a direct GET for the Task, thus eliminating the need for subscription or polliing.
4. Fulfiller system queries to retrieve the referenced request, and updates the Task to indicate "acceptance" and agreement to fulfill
5. Fulfiller may update the Task to indicate interim progress notes
6. Placer is aware of the acceptance of and changes to the Task either through the ownership of the resource, or using polling or a subscription to a queue server system to determine the same
7. Fulfiller creates an event resource in its own system via POST or internal action, or POSTs it to a queue server system
8. Fulfiller PUTs an update to the Task resource to change its status to completed and to point to the event resource
9. Placer is aware of the completion of the Task either through the ownership of the resource, or via polling or subscription to a queue server system, and retrieves the referenced event resource
10. Placer updates the request resource to indicate completion via PUT or an internal action, or PUTs the update to a queue server system

**Benefits**

* Uses the FHIR REST API for managing the workflow
* Both placer and fulfiller can track the state of the workflow
* Can use this approach for request other than just fulfillment (e.g. to request status change or other updates)
* There's an ability to negotiate fulfillment - i.e. the ability to say "no"
* Explicit acknowledgement that filler has received and agreed to act on the request

**Limitations**

* Additional complexity of using Task
* Additional complexity of setting up and maintaining a subscription or polling infrastructure
* Placer and fulfiller must have a FHIR server, and placer's FHIR server pust allow "write" updates to the Task resource
* Placer and filler may need to be able to communicate directly (i.e. know each other's respective endpoints)
  + This could become unmanageable if there are a large (or dynamic) number of placers and fillers that need to communicate
  + Placer and fulfiller must know where to subscribe for content - this could be a large number of systems
  + May not apply if there's a queue server

**Usage Recommendations**

* When following the state of the workflow is important
  + When there is a need to know that a fulfiller has accepted to perform the request
  + When there is a need to be able to change the performance of the request
  + When there are multiple steps in fulfilling the request
* When requests are not (always) directed to a specific filler
* When the placer needs to manage the Task resource

**Usage examples**

**Lab order to known performer**

A provider orders a bilirubin test for a patient, to be done at the provider's facility lab....

**"Open" lab order**

A patient is scheduled for their physical, and the provider sends her a secure message to have a fasting blood test performed before the office visit. The message has a link to a requisition, and a list of possible lab loacations where the test can be done. The patient chooses a lab location near her place of employment, and a test is scheduled. The lab gets the requisition from the patient, and using the information on how to retrieve the Task from the provider's system, claims the test to be performed...

**Option G: POST of Task on fulfiller's system**

**Steps**

1. Placer creates the request in its own system via POST or an internal action, or POSTs it to a queue server system
2. Placer POSTs a Task resource to the filler system, pointing to the request resource and seeking fulfillment
3. Fulfiller system GETs the referenced request
4. Fulfiller updates the Task to indicate acceptance of the task
5. Placer either polls the Task to note acceptance or uses a subscription to determine the same
6. Fulfiller may further update the Task to reflect the progeess made. Using the same method as in step 5, the placer becomes aware of these updates
7. Fulfiller creates an event resource in its own system via POST or an internal action, or POSTs it to a queue server system
8. Fulfiller Updates the Task resource to change its status to completed and to point to the event resource
9. Placer either polls the Task to note completion and changes or uses a subscription to determine the same
10. Placer system queries to retrieve the referenced event resource
11. Placer updates the request resource to indicate completion via PUT or an internal action, or PUTs the update to a queue server system

**Benefits**

* Uses the FHIR REST API for managing the workflow
* Subscription/polling is targeted to a specific resource instance - much simpler than general subscription/polling infrastructure
* Both placer and fulfiller can track the state of the workflow
* Can use this approach for request other than just when requesting fulfillment (e.g. to request status change or other updates)
* There's an ability to negotiate fulfillment - i.e. the ability to say "no"
* Explicit acknowledgement that filler has received and agreed to act on the request

**Limitations**

* Additional complexity of using Task
* Additional complexity of using subscription or polling
* Placer and fulfiller must have a FHIR server, and fulfiller's FHIR server must allow "write" creation of the Task resource by the placer
* Placer and filler may need to be able to communicate directly (i.e. know each other's respective endpoints)
  + This could become unmanageable if there are a large (or dynamic) number of placers and fillers that need to communicate
  + May not apply if there's a queue server
* Placer may not know immediately when filler system has retrieved the request

**Usage Recommendations**

* When following the state of the workflow is important
  + When there is a need to know that a fulfiller has accepted to perform the request
  + When there is a need to be able to change the status of the workflow by either the placer or the fulfiller
  + When there are multiple steps in fulfilling the request
* When the fulfiller needs to manage the Task resource

**Usage Examples**

**Outpatient prescription**

A patient is suffering from a poison ivy rash, and he is prescribed a steroid medication to manage the reaction. The provider asks the patient for his preferred pharmacy and creates the Task at the specified pharmacy's system. The patient is running late and by the time he arrives at the pharmacy, it is closed. He calls the clinic where the provider practices, and requests that the pharmacy where the order is to be fulfilled is changed to one that is open around the clock. The staff at the clinic cancels the Task at the original pharmacy, and creeates a new Task at the requested new one. By the time the patient arrives at the new pharmacy, the medicatin is ready for pickup. Once the request is fulfilled, the clinic receives the notification that the Task is complete, and the prescription is also marked as complete.

**Option H: POST of Task to a workflow broker**

*TODO: Still needs review and update*

**Steps**

1. Placer POSTs the request to its own system or to a queue server system
2. Broker detects that new un-assigned request (without a Task yet created and falling within the scope of the Broker to ensure fulfillment) via polling or subscription
3. Broker POSTs a Task resource to its own system or a queue server system, pointing to the request resource and seeking fulfillment from a specific filler  
   Task does not have a specified "performer" (but may have performer type)
4. If the Task is rejected by one potential recipient, the broker may create a new task to seek fulfillment from others
5. Continue as per [Option G](file:///C:\temp\main-pages.html#optiong)

**Benefits**

* Offloads responsibility for seeking fulfillment from the placer system, but more actively solicits fulfillment than a simple "post the task and see who takes it". Also allows prioritized assignment of tasks (i.e. some fillers may be preferred over others)

**Limitations**

* Requires a broker to exist
* Broker must know all available fillers and their capabilities to allow appropriate assignment

**Usage Recommendations**

Appropriate in environments that have a workflow engine that takes on responsibility for ensuring fulfillment

**Usage Examples**

**Option I: Option I: POST of Task on fulfiller's system, POST of sub-Task on placer's system**

**Steps**

1. Placer creates the request in its own system via POST or an internal action
2. Placer POSTs a Task resource to the filler system, pointing to the request resource and seeking fulfillment
3. Fulfiller system GETs the referenced request
4. Fulfiller updates the Task to indicate acceptance of the task
5. Fulfiller POSTs a Task, which also points to the request resource, and also uses the [Task.partOf](file:///C:\temp\task-definitions.html#Task.partOf) attribute to point to the Task from step 2 (indicating it is a sub-Task)
6. Fulfiller may further update both Tasks to reflect the progeess made. Since the sub-Task is on the placer's system, the placer is aware of these updates
7. Fulfiller creates an event resource in its own system via POST or an internal action
8. Fulfiller Updates the Task resource to change its status to completed and to point to the event resource
9. Fullfiler updates the sub-Task resource as completed, and points to the event resource. Since the sub-Task is on the palcer's system, the placer is aware of the completion and changes

.

1. Placer system queries to retrieve the referenced event resource
2. Placer updates the request resource to indicate completion via PUT or an internal action

**Benefits**

* Uses the FHIR REST API for managing the workflow
* No need to use subscription or polling
* Both placer and fulfiller can track and change the state of the workflow
* Can use this approach for request other than just when requesting fulfillment (e.g. to request status change or other updates)
* There's an ability to negotiate fulfillment - i.e. the ability to say "no"
* Explicit acknowledgement that filler has received and agreed to act on the request

**Limitations**

* Additional complexity of using Task
* Need to keep the two Task states in sync
* Placer and fulfiller must each have a FHIR server, and give the other side "write" access to create and update Task resources.
* Placer and filler may need to be able to communicate directly (i.e. know each other's respective endpoints)
  + This could become unmanageable if there are a large (or dynamic) number of placers and fillers that need to communicate
  + May not apply if there's a queue server

**Usage Recommendations**

* When following the state of the workflow is important
  + When there is a need to know that a fulfiller has accepted to perform the request
  + When there is a need to be able to change the status of the workflow by either the placer or the fulfiller
  + When there are multiple steps in fulfilling the request
* When there is no need for polling or subscription infrastructure
* When us of the FHIR REST API is preferred to any other method of communications
* When the intended fulfiller is known in advance
* When messaging-like functionality is needed, but a messaging infrastructure is not available

**Usage Examples**

**Outpatient referral**

A patient is seen by her PCP for shortness of breath, and is the provider decides to refer her to a cardiologist. As a result of the referral, the provider's system creates ReferralRequest resource, and POSTs a Task resource at a cardiologist's practice system, requesting a referral. THis places the referral request on a referral coordinator's work list to review. Upon accepting the referral, the Task is updated as accepted, and a new sub-Task, referencing both the ReferralRequest and the Task created by the provider's system, is created on the provider's system. The patient gets an appointment with the cardiologist, and the sub-Task is updated with that information, allowing the provider's system to inform the provider of the change.

The patient misplaces the information about the appointment, and misses the visit with the cardiologist. The cardiologist's system updates the sub-Task to indicate the no-show, and the PCP is alerted of the missed appointment. The provider's staff gets a hold of the patient, and schedules another appointment for her with the cardiologist. As a result of the cardiologist's system scheduling the appointment, the sub-Task is updated with that information. The patient is seen, and the sub-Task is updated accordingly. The cardiologists documents the findings about the patient, and updates the sub-Task as complete, adding references to the resources representing the findings. The PCP is alerted of the updates, and marks the referral as complete, which updates the original Task, and the ReferralRequest as complete as well.

**Option J: Messaging Task from placer to fulfiller**

*TODO: needs more details*

**Steps**

1. Placer sends message to filler with a MessageHeader, where the "data" element points to the Taks resource, also contained in the message. The massage may or may not contain any other relevant resources (e.g the actual request resource), or an "event" code saying "please fulfill"
2. Filler system sends a response containing the same Task resource, indicating receipt of the message and, optionally an indication of their intention to fulfill the request
3. Filler system may send incremental messages to the placer showing progress (e.g. specimen collected, preliminary results, final results) by including an updated Task resource
4. Placer system may also send messages to the fulfiller conatining the Task resource and updatingh the state of the workflow

**Benefits**

* Reduced number of communications
* All relevant data can be sent in one package
* Responses can be asynchronous and content may be routed
* There's an ability to negotiate fulfillment - i.e. the ability to say "no"
* Can request things other than just fulfillment (e.g. please suspend)
* Explicit acknowledgement that filler has received and agreed to act on the request (though no need for the placer to check)

**Limitations**

* Messaging is "heavy"
* Need to negotiate what allowed responses are and what data can be present in request and response messages
* Additional complexity of using Task
* Need message delivery infrastructure in place

**Usage Recommendations**

Appropriate when existing messaging infrastructure can be used (e.g. HL7 over HTTP, v2 LTP, MLTP, WSI Web Services, Direct, VISA, REST, etc.), and a need to stay consistent with that architecture.

**Usage examples**

**Option K: Service request referencing Task from placer to fullfiller**

*TODO: This scenario needs work - there's not a lot of experience using FHIR services to manage the fulfillment process*

**Steps**

1. Placer creates a request resource on their own system or a queue server
2. Placer may create a Task resource on their own system or a queue server
3. Placer invokes a service on the filler system saying "please fulfill this order", including the content or a reference to the request resource and any other relevant data
4. Filler system responds (synchronously if using HTTP, but may be asynchronous if using SOAP or other transport mechanisms) with conformation of receipt and, optionally indication of intention to fulfill and/or results

**Benefits**

* ???

**Limitations**

* ???

**Usage Recommendations**

TBD

**Usage examples**

**workflow-module.html**

|  |  |
| --- | --- |
| Work Group | [Ballot Status](file:///C:\temp\help.html#status): [n/a](file:///C:\temp\history.html#pubs) |

**Workflow Module**

The workflow module focuses on the coordination of activities within and across systems. This includes three primary aspects:

* How do we ask for another person, device or system to do something?
* How do we track the linkages and dependencies between activities - actions to their authorizations, complex activities to individual steps, protocols to plans to orders, etc.?
* How do we define what activities are possible and the expected order and dependencies of the steps within those activities? I.e. process/orchestration definition

In addition, this module supports resources to support two specific workflows - the requesting and delivery of supplies and the requesting and delivery of information.

**Introduction**

Workflows can be performed using the [Task](file:///C:\temp\task.html) resource, [messaging](file:///C:\temp\messaging.html) or [services](file:///C:\temp\services.html). This specification includes a [workflow](file:///C:\temp\workflow.html) page that describes the concepts underlying the discussion of workflows, and points to a number of different communication and architectural [workflow patterns](file:///C:\temp\workflow-communications.html).

In addition to the [Task](file:///C:\temp\task.html) resource, this specification defines three logical models - [Definition](file:///C:\temp\definition.html), [Request](file:///C:\temp\request.html) and [Event](file:///C:\temp\event.html) that define the patterns for resources that are typically involved in workflow. These patterns include elements defining relationships between requests, between events, and across requests and events. These relationships are summarized on the [workflow](file:///C:\temp\workflow.html#relationships) page, along with a complete [list](file:///C:\temp\workflow.html#list) of resources that follow (or are hoped to soon follow) the request and event patterns.

Finally the [PlanDefinition](file:///C:\temp\plandefinition.html) and [ActivityDefinition](file:///C:\temp\activitydefinition.html) resources combine to support the creation of protocols, orders sets, guidelines and other workflow definitions.

**Common use Cases**

Workflow manifests in many places in the healthcare environment:

* Creating a [lab order](file:///C:\temp\procedurerequest.html), [drug prescription](file:///C:\temp\medicationrequest.html), [referral](file:///C:\temp\referralrequest.html) or other clinical order or an [insurance claim](file:///C:\temp\claim.html), [EnrollmentRequest](file:///C:\temp\enrollmentrequest.html) or similar administrative request and asking for it to be actioned by a specific organization or practitioner
* Negotiating a fulfillment process, such as requesting further information before accepting a claim or referral or proposing an alternative therapy when processing an order
* Letting an ordering physician know what the current progress is in fulfilling an order (e.g. Blood has been drawn, sample is being processed, preliminary results are in, etc.)
* Defining a [plan](file:///C:\temp\careplan.html) or recommendation for a set of clinical and/or administrative activities to manage a patient's care and then tracking how those plans and recommendations are (or are not) acted upon over time.
* Communicating a state change to a request or order (e.g. suspension, update, cancellation, etc.) to a filling system so that they can take appropriate action
* Asking for a state change, merge of a couple of patients or the invoking of some operation or decision supporting an asynchronous manner - for example, one where human intervention is required
* Designing or adhering to a study protocol, chemotherapy protocol, instantiating an order set or other [plan definition](file:///C:\temp\plandefinition.html)

FHIR provides multiple ways to enable all of these scenarios (and many others). Common mechanisms, along with their pros and cons can be found in the workflow sections on [patterns](file:///C:\temp\workflow-communications.html#commpatternslist).

**Security and Privacy**

Resources related to workflow need to adhere to the same [security and privacy guidelines](file:///C:\temp\security.html) that apply to all FHIR resource, including specific considerations for those that may contain personally-identifying information. There are a couple of additional security and privacy considerations specific to workflow:

1. Some workflows are ad-hoc without pre-defined participants or flows. These can be challenging for security and privacy processes to manage appropriately

2. Workflow can drive automated behavior. I.e. The mere existence of an electronic record can cause information to flow, procedures to be performed, records to be changed and money to be transferred, potentially without any intervention, oversight or sanity checking by a human being. As such, even greater care must be taken to ensure that:

* constraints are placed on what systems (and users) can initiate workflow processes
* requests for action are appropriately authenticated before action is taken
* patient consents and other relevant policies are enforced either by the system storing the request or the system acting upon it (and that if enforcement is not performed by the actor, that they are confident that relevant policies have been enforced on the request prior to action)

For more general considerations, see [the Security and Privacy module](file:///C:\temp\secpriv-module.html).

**Developmental Roadmap**

The principles of requesting action and reporting results have been present in FHIR since its initial version. However, significant work has happened (or is planned) as part of this STU 3 release to increase the consistency of resources involved in workflow and to improve the documentation expressing the different ways workflow can be managed depending on architectural and other constraints. This has included the introduction of the [Task](file:///C:\temp\task.html) resource (which replaced the old Order and OrderResponse resources), the definition of the [Request](file:///C:\temp\workflow.html#request) and [Event](file:///C:\temp\workflow.html#event) patterns and the creation of the [PlanDefinition](file:///C:\temp\plandefinition.html) and [ActivityDefinition](file:///C:\temp\activitydefinition.html) resources. Some of this work has had preliminary testing at connectathon but has had little production experience as yet. Most of the discussion has been on how workflow is best managed from a [RESTful](file:///C:\temp\http.html) paradigm perspective.

Looking forward to the next release, we'll be seeking and incorporating feedback from the implementer community about both the workflow resources as well as the patterns and architectural approaches documented in this specification. We'll also be increasing focus on the messaging and services paradigms and seeing how much consistency in approach is possible/desirable across REST, messaging and services. Finally we'll be working with the workgroups responsible for the [Administration](file:///C:\temp\administration-module.html) and [Financial](file:///C:\temp\financial-module.html) modules to increase the consistency of approach to the design of workflow-related resources, regardless of domain. Whether any of the resources defined beneath this module will be candidates for normative status in the next release of FHIR will depend on the degree and type of implementer feedback between now and that release.

Additional topics for future work include:

* Resolving the overlap between the [SupplyRequest](file:///C:\temp\supplyrequest.html), [DeviceRequest](file:///C:\temp\devicerequest.html) and [VisionPrescription](file:///C:\temp\visionprescription.html) resources
* Improving mapping and alignment of the elements and status codes of the [Task](file:///C:\temp\task.html) resource with the WS-HumanTask specification
* Creating "best practice" guides for how to implement workflow for different business patterns
* Examining how workflow is used for "compensating actions", for example account transactions and reversals

**workflow.html**

* [Overview](file:///C:\temp\main-pages.html)
* [Communication Patterns](file:///C:\temp\workflow-communications.html)
* [Ad-hoc Workflow Patterns](file:///C:\temp\workflow-ad-hoc.html)
* [Workflow Management Patterns](file:///C:\temp\workflow-management.html)
* [Examples](file:///C:\temp\workflow-examples.html)

**Workflow Description**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): N/A | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

Workflow is an essential part of healthcare - orders, care protocols, referrals are the drivers of most activity within in-patient settings and a great deal of activity in community care as well. FHIR is concerned with workflow when there's a need to share information about workflow state or relationships, when there's a need to coordinate or drive the execution of workflow across systems and when there's a need to define allowed actions, dependencies and conditions on behavior.

**Workflow state & relationships**

FHIR does not need to be used for the execution of workflow. Orders, care plans, lab results, hospital admissions, claim payments and other records can all be shared using FHIR roesources without the process to actually solicit fulfillment of those orders or requesting payment of those claims being driven by a FHIR transaction. Interoperable support for workflow execution is actually a more advanced FHIR activity because it requires a higher degree of standardization. Rather than merely standardizing the data to exchange, interoperable workflow execution requires standardization of the processes, roles and activities of the different systems. However, even without using FHIR for workflow execution, there's still a need to standardize the data elements related to workflow: how does an event or a result point to the order that authorized it? How do parent steps and child steps get linked together? How does a care plan identify what protocol it's adhering to?

FHIR defines three categories of resources that are involved in activities - [requests](file:///C:\temp\main-pages.html#request), [events](file:///C:\temp\main-pages.html#event) and [definitions](file:///C:\temp\main-pages.html#definition). Each of these different types has a "pattern" associated with it. Ã‚Â Resources that fall into that type are encouraged to adhere to their respective pattern. These patterns provide standard elements that are typical for most resources of each type. Strict adherence is not required as work groups are expected to align with typical domain behavior and requirements as more authoritative than "desired" architectural patterns. In some cases, capabilities might be supported with extensions rather than core elements where a pattern capability is deemed to be "not common, but still relevant" for a given resource.

A full description of the patterns and their interrelationships can be found in the [Workflow Resource Patterns](file:///C:\temp\main-pages.html#respatterns) section of this page.

**Workflow execution**

In addition to defining patterns for resources used in workflow processes, FHIR supports the execution of those processes as well. However, FHIR does not define a "one size fits all" solution for workflow architecture. FHIR supports a variety of interoperability paradigms and most of them ([REST](file:///C:\temp\http.html), [Messaging](file:///C:\temp\messaging.html) and [Services](file:///C:\temp\services.html)) provide support for driving workflow execution. (The [Document](file:///C:\temp\documents.html) paradigm does not directly support driving behavior, though it can be combined with one of the other patterns to do so.) In addition, several of these paradigms allow multiple approaches to supporting workflow, depending on the context and needs of the workflow process.

The [Workflow Execution and Communication Patterns](file:///C:\temp\workflow-communications.html) section describes a number of options for workflow execution, summarizes their respective pros and cons and makes recommendations for the circumstances in which they might best be used.

**Workflow definition**

The definition of protocols, order sets, guidelines and other structures that define what sorts of activities should occur, what order they should occur on, what dependencies they have, in what circumstances they should start or end, etc. is handled by a pair of resources:

* [PlanDefinition](file:///C:\temp\plandefinition.html) defines the interrelationships of steps and the rules around their execution
* [ActivityDefinition](file:///C:\temp\activitydefinition.html) defines an activity to be performed as a single step

The use of these two artifacts is documented TODO.

**Workflow Resource Patterns**

Not all resources in FHIR are related to workflow - many are used to describe entities and roles (patients, medications, etc.) or infrastructure (structure definitions, value sets, etc.). However, a large proportion of the FHIR resources are devoted to the description of activities in one fashion or another and almost all of these fall into the realm of workflow - they describe things that can be done (definitions), are desired to be done (requests) or that have been done (events). The table below summarizes the list of workflow-relevant resources:

**Workflow resources**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requests** | Resources that ask for or express a desire/intention for something to be done | | |
| * [Appointment](file:///C:\temp\appointment.html)[\*](file:///C:\temp\main-pages.html#listnotes) * [CarePlan](file:///C:\temp\careplan.html) * [Claim](file:///C:\temp\claim.html) * [CommunicationRequest](file:///C:\temp\communicationrequest.html) | * [DeviceRequest](file:///C:\temp\devicerequest.html) * [EligibilityRequest](file:///C:\temp\eligibilityrequest.html) * [EnrollmentRequest](file:///C:\temp\enrollmentrequest.html) * [ImmunizationRecommendation](file:///C:\temp\immunizationrecommendation.html) | * [MedicationRequest](file:///C:\temp\medicationrequest.html) * [NutritionOrder](file:///C:\temp\nutritionorder.html) * [ProcedureRequest](file:///C:\temp\procedurerequest.html) * [ProcessRequest](file:///C:\temp\processrequest.html)[†](file:///C:\temp\main-pages.html#listnotes) | * [ReferralRequest](file:///C:\temp\referralrequest.html) * [Task](file:///C:\temp\task.html)[‡](file:///C:\temp\main-pages.html#listnotes) * [SupplyRequest](file:///C:\temp\supplyrequest.html) * [VisionPrescription](file:///C:\temp\visionprescription.html) |
| **Events** | Resources that express that something has been done and which can potentially be done as a result of a request | | |
| * [AppointmentResponse](file:///C:\temp\appointmentresponse.html)[\*](file:///C:\temp\main-pages.html#listnotes) * [ClaimResponse](file:///C:\temp\claimresponse.html) * [ClinicalImpression](file:///C:\temp\clinicalimpression.html) * [Communication](file:///C:\temp\communication.html) * [Composition](file:///C:\temp\composition.html) * [Condition (aka Problem)](file:///C:\temp\condition.html) * [Consent](file:///C:\temp\consent.html) * [Contract](file:///C:\temp\contract.html) * [Coverage](file:///C:\temp\coverage.html) | * [DeviceUseStatement](file:///C:\temp\deviceusestatement.html) * [DiagnosticReport](file:///C:\temp\diagnosticreport.html) * [DocumentManifest](file:///C:\temp\documentmanifest.html) * [DocumentReference](file:///C:\temp\documentreference.html) * [EligibilityResponse](file:///C:\temp\eligibilityresponse.html) * [Encounter](file:///C:\temp\encounter.html) * [EnrollmentResponse](file:///C:\temp\enrollmentresponse.html) * [EpisodeOfCare](file:///C:\temp\episodeofcare.html) * [ExplanationOfBenefit](file:///C:\temp\explanationofbenefit.html) | * [FamilyMemberHistory](file:///C:\temp\familymemberhistory.html) * [GuidanceResponse](file:///C:\temp\guidanceresponse.html) * [ImagingStudy](file:///C:\temp\imagingstudy.html) * [Immunization](file:///C:\temp\immunization.html) * [MeasureReport](file:///C:\temp\measurereport.html) * [Media](file:///C:\temp\media.html) * [MedicationAdministration](file:///C:\temp\medicationadministration.html) * [MedicationDispense](file:///C:\temp\medicationdispense.html) * [MedicationStatement](file:///C:\temp\medicationstatement.html) | * [Observation](file:///C:\temp\observation.html) * [PaymentNotice](file:///C:\temp\paymentnotice.html) * [PaymentReconciliation](file:///C:\temp\paymentreconciliation.html) * [Procedure](file:///C:\temp\procedure.html) * [ProcessResponse](file:///C:\temp\processresponse.html)[†](file:///C:\temp\main-pages.html#listnotes) * [QuestionnaireResponse](file:///C:\temp\questionnaireresponse.html) * [RiskAssessment](file:///C:\temp\riskassessment.html) * [SupplyDelivery](file:///C:\temp\supplydelivery.html) * [Task](file:///C:\temp\task.html)[‡](file:///C:\temp\main-pages.html#listnotes) |
| **Definitions** | Resources that define something that can potentially happen in a patient and time-independent manner | | |
| * [ActivityDefinition](file:///C:\temp\activitydefinition.html) * [DataElement](file:///C:\temp\dataelement.html) | * [HealthcareService](file:///C:\temp\healthcareservice.html) * [Measure](file:///C:\temp\measure.html) | * [OperationDefinition](file:///C:\temp\operationdefinition.html) * [PlanDefinition](file:///C:\temp\plandefinition.html) | * [Questionnaire](file:///C:\temp\questionnaire.html) |

|  |  |
| --- | --- |
| \* | The [Appointment](file:///C:\temp\appointment.html) and [AppointmentResponse](file:///C:\temp\appointmentresponse.html) resources do not follow the same sort of request/response pattern as the other resources. Their design is based on iCal conventions, so their model won't reflect the same alignment as most other resources. They are included here for completeness. |
| † | [ProcessRequest](file:///C:\temp\processrequest.html) and [ProcessResponse](file:///C:\temp\processresponse.html) are candidates for retirement with their function subsumed by [Task](file:///C:\temp\task.html) |
| ‡ | The [Task](file:///C:\temp\task.html) resource takes on characteristics of both "requests" and "events" and thus shares characteristics from both patterns |

Note that requests, events and definitions don't exist in a 1:1:1 relationship. Some requests and events have obvious pairings. For example, a [SupplyRequest](file:///C:\temp\supplyrequest.html) will generally always pair with a [SupplyDelivery](file:///C:\temp\supplydelivery.html). The same goes for [EnrollmentRequest](file:///C:\temp\enrollmentrequest.html)/[EnrollmentResponse](file:///C:\temp\enrollmentresponse.html), etc. On the other hand, for other resources there isn't a strict pairing. A [ReferralRequest](file:///C:\temp\referralrequest.html) might be responded to by an [Encounter](file:///C:\temp\encounter.html), [DiagnosticReport](file:///C:\temp\diagnosticreport.html), [Procedure](file:///C:\temp\procedure.html), [RiskAssessment](file:///C:\temp\riskassessment.html), etc. Similarly, a [Procedure](file:///C:\temp\procedure.html) might be triggered by a [ProcedureRequest](file:///C:\temp\procedurerequest.html) or [ReferralRequest](file:///C:\temp\referralrequest.html). The set of common linkages should be asserted in their respective resources. The specific types of responses for a given request will be governed by the Request.code, any workflow definitions/protocols referenced and local convention.

**Workflow Resource Relationships**

These three patterns of resources have a standard set of relationships, both with themselves, as well as with each other.

Specifically:

* both requests and events can point to their respective definitions
* events and requests can point to the proposals, plans or orders they are based on
* events and definitions can be organized into parent-child relationships of parents and components
* definitions and requests can both replace prior versions of the same type of artifact

This list of relationships is not exhaustive, but covers those that are "standardized" as part of the patterns. Further description and guidance on these relationships can be found in the [Request](file:///C:\temp\request.html), [Event](file:///C:\temp\event.html) and [Definition](file:///C:\temp\definition.html) logical models.

**Request Resource Pattern**

Requests are resources that represent the proposal, plan or order for an activity to occur. A [Request pattern](file:///C:\temp\request.html) defines the common elements typically present on all request resources.

The amount of information needed for a Request to be actionable can vary by circumstance. Some request instances may not be "fully specified" - additional information from protocol, patient preference and/or professional decision-making may be necessary before the authorized action can occur. For example, a MedicationOrder might be specified without indicating a strength or route in situations where the pharmacy (or even nursing information) has the authority to determine those parameters. A VisionPrescription might not be actionable until frames have been chosen and the necessary measurements of the patient's face have been taken to allow the lenses to be positioned appropriately within the frames.

All requests with an intent of "order" authorize something. Whether what is authorized is sufficient to be immediately actionable depends on who is fulfilling the order and the context in which the fulfillment request is made. The determination of whether a given "request" is actionable may be made by the systems involved or the humans being asked to act.

As well, the existence of a "Request" instance doesn't necessarily imply that fufillment will be requested immediately - or even ever. The decision to request fulfillment may be delegated to the patient or to down-stream practitioners. Such fufilling practitioners may need to capture additional information prior to executing the fufillment.

**Event Resource Pattern**

Events are resources that represent the ongoing or completed execution of some activity or observation. For example, a clinical procedure, a financial transaction, the recording of a diagnosis, etc. An [Event pattern](file:///C:\temp\event.html) defines the common elements typically present on all event resources.

**Definition Resource Pattern**

Definitions are resources that represent activities that could be performed in a time and subject-independent manner such as a protocol, order set, clinical guideline, etc. A [Definition pattern](file:///C:\temp\definition.html) defines the common elements typically present on all event resources.

**Open Issues**

**STU Notes:**

* It is possible to replace some portions of the [MessageHeader](file:///C:\temp\messageheader.html) with a reference to the [Task](file:///C:\temp\task.html) resource. Doing so would mean consistency in how asynchronous requests are represented using REST and messaging. However, it introduces an additional layer of complexity and formality into the messaging paradigm that may be unwelcome, particularly for those systems that do not currently foresee a need to support both RESTful and messaging invocations of workflow
* The [OperationDefinition](file:///C:\temp\operationdefinition.html) resource could be used to define types of tasks and the sets of parameters that are allowed to go with them. Is this an appropriate use of the OperationDefinition resource?
* The [SupplyRequest](file:///C:\temp\supplyrequest.html), [DeviceRequest](file:///C:\temp\devicerequest.html) and [VisionPrescription](file:///C:\temp\visionprescription.html) resources have a significant degree of overlap. Should they remain distinct resources?

**xml.html**

**XML Representation of Resources**

|  |  |  |
| --- | --- | --- |
| Work Group | [Maturity Level](file:///C:\temp\resource.html#maturity): 5 | [Ballot Status](file:///C:\temp\help.html#status): [STU 3](file:///C:\temp\history.html#pubs) |

The XML representation for a resource is described using this format:

<**name** xmlns="http://hl7.org/fhir" (attrA="value")>

<**nameA**><!--  **1..1** type description of content --><nameA>

<**nameB[x]**><!-- 0..1 type1|type1 description --></nameB>

<**nameC**> <!-- **1..\*** -->

<**nameD**><!-- 1..1 type>Relevant elements --></nameD>

</nameC>

<name>

Using this format:

* To build a valid XML instance of a resource, simply replace the contents of the elements and attributes with valid content as described by the cardinality, type rules and content description found in the comment in each element
* Resource and Element names are case-sensitive (though duplicates that differ only in case are never defined)
* Elements must always appear in the order documented
* When an element is allowed to repeat, the elements are ordered, and the technical infrastructure needs to be able to access the items in the right order (see also [Cardinality Rules](file:///C:\temp\conformance-rules.html#cardinality) for a further description of elements with cardinality > 1)
* A few properties are represented as attributes: values of primitive types in a value attribute, extension URLs in the url attribute on an extension, and the id property on elements (but not on resources, where the resource id is an element)
* Any of the XML elements may have an id attribute to serve as [the target of an internal reference](file:///C:\temp\references.html#id). The id attribute is not shown in this format
* FHIR elements are always in the namespace <http://hl7.org/fhir>. This is usually specified as the default namespace on the root element. The only other namespace that occurs in FHIR resources is the XHTML namespace ([XHTML is found in most resources](file:///C:\temp\narrative.html))
* Infrastructural elements that are common to all resources are not shown in the xml representation. These must appear prior to any other defined child elements in the following order:
  + First, the elements from the [base resource](file:///C:\temp\resource.html), in order
  + Second, the elements from the [domain resource](file:///C:\temp\domainresource.html), in order
* FHIR elements are never empty. If an element is present in the resource, it SHALL have either a value attribute, child elements as defined for its type, or 1 or more [extensions](file:///C:\temp\extensibility.html)
* Attributes can never be empty. Either they are absent, or they are present with at least one character of non-whitespace content
* Implementers SHOULD trim leading and trailing whitespace before writing and SHOULD trim leading and trailing whitespace when reading attribute values
* The lock icon () denotes that an element defines or is affected by [additional rules](file:///C:\temp\conformance-rules.html#constraints) that control its presence and/or content
* XML comments, processing instructions and formatting are not part of the contents of a resource
* There SHALL be no DTD references in FHIR resources (because of [XXE security exploit](https://en.wikipedia.org/wiki/XML_external_entity_attack))
* XML resources SHALL be exchanged using UTF-8 encoding. Specifying the character encoding using a <?xml encoding="UTF-8" ?> processing instruction is optional but recommended
* Other processing instructions SHOULD not be included, and SHALL NOT be required in order to properly understand and/or present the data or narrative of the resource. Applications MAY preserve processing instructions when handling resources, but are not required to do so
* The MIME-type for this format is application/fhir+xml.

**XML Schema and Schematron**

This specification provides schema definitions for all of the content models it describes.

The base schema is called "[fhir-base.xsd](file:///C:\temp\fhir-base.xsd)" and defines all of the datatypes and base infrastructure types. In addition, there is a schema for each resource and a common schema [fhir-all.xsd](file:///C:\temp\fhir-all.xsd) that includes all the resource schemas. For schema processors that do not like circular includes, there is [a single schema](file:///C:\temp\fhir-single.xsd) that contains everything.

In addition to the w3c schema files, this specification also provides Schematron files that enforce the various constraints defined for the datatypes and resources. These are packaged as files for each resource.

XML that is exchanged SHALL be valid against the w3c schema and Schematron, though being valid against the schema and Schematron is not sufficient to be a conformant instance: this specification makes several rules that cannot be checked by either mechanism. Operational systems may choose to use schema tools to check validation, but are not required to do so. Exchanged content SHALL NOT specify the schema or even contain the schema instance namespace in the resource itself.

Given the way [extensions](file:///C:\temp\extensibility.html) work, applications reading XML resources will never encounter unknown elements. However once an application starts trading with other appplications that conform to later versions of this specification, unknown XML elements may be encountered. Applications MAY choose to ignore unknown elements in order to foster forwards compatibility in this regard, but may also choose not to - which would be the normal behavior for schema generated applications. Applications declare their behavior with regard to unknown elements using [CapabilityStatement.acceptUnknown](file:///C:\temp\capabilitystatement-definitions.html#CapabilityStatement.acceptUnknown).

**Code Generation Schema**

In addition to the validation schema, this specification provides a set of schema suitable for code generation. These schema describe the same XML syntax, but apply less validation in order to create a schema that works with more code generation tooling.

Specifically, these schemas are generated without any xsd:choice elements, for code generators that don't deal with choices well. Implementers that use these schemas will need to enforce the correct usage of the [choice elements](file:///C:\temp\formats.html#choice) without schema support.

Implementers making use of schema driven code generation tooling need to consider how to handle the [decimal](file:///C:\temp\datatypes.html#decimal) data type. The decimal data type is defined to be precision aware - that is, that implementers need to preserve the difference between "2.0" and "2.00" - this is ubiquitiously considered important in handling observed data in healthcare. Both schemas map this data type to xsd:decimal, but the base [W3C schema decimal type](http://www.w3.org/TR/xmlschema-2/#decimal) is specified not to be precision aware. Schema driven implementations vary as to how precision is handled, and implementers will need to determine how their generated code handles decimals, and consider changing the type for decimal in the schema from xsd:decimal to xsd:string. Specifically, implementers may wish to change

<xs:simpleType name="decimal-primitive">

<xs:restriction base="xs:decimal">

<xs:pattern value="-?([0]|([1-9][0-9]\*))(\.[0-9]+)?"/>

</xs:restriction>

</xs:simpleType>

to this:

<xs:simpleType name="decimal-primitive">

<xs:restriction base="xs:string">

<xs:pattern value="-?([0]|([1-9][0-9]\*))(\.[0-9]+)?"/>

</xs:restriction>

</xs:simpleType>

Alternatively, if supported, implementers may wish to use the [precisionDecimal](http://www.w3.org/TR/xsd-precisionDecimal/) from the XSD 1.1 framework.

Note that most code generation frameworks ignore the pattern restriction.

**Canonical XML**

Resources and/or Bundles may be digitally signed (see [Bundle](file:///C:\temp\bundle.html) and [Provenance](file:///C:\temp\provenance.html)).

This specification defines the following method for canonicalizing FHIR resources, when represented as xml:

* No whitespace other than single spaces in attribute values and in the xhtml in the [Narrative](file:///C:\temp\narrative.html)
* Use default namespaces for the FHIR and XHTML namespaces
* Omit all elements that have a default value, if a default value is defined
* Omit all comments
* Always use the unicode character representation for any XML entities (e.g. &#39; instead of &quot;)
* Include the XML processing instruction <?xml version="1.0" encoding="UTF-8"?>
* Using the XML canonical method [Canonical XML 1.1](http://www.w3.org/TR/xmldsig-core1/#sec-Canonical) (http://www.w3.org/2006/12/xml-c14n11)

This canonicalization method is identified by the URL http://hl7.org/fhir/canonicalization/xml. The following additional canonicalization URLS are also defined:

|  |  |
| --- | --- |
| http://hl7.org/fhir/canonicalization/xml#data | The narrative (Resource.text) is omitted prior to signing (note the deletion is at Resource.text, not Resource.text.div) |
| http://hl7.org/fhir/canonicalization/xml#static | In addition to narrative (Resource.text), the Resource.meta element is removed. This makes the signature robust as the content is moved from server to server, or workflow and access tags are added or removed |
| http://hl7.org/fhir/canonicalization/xml#narrative | The method only covers the Resource.id and Narrative is retained |
| http://hl7.org/fhir/canonicalization/xml#document | The signs everything in a Bundle, except for the Bundle.id and Bundle.metadata on the root Bundle (allows for a document to copied from server to server) |

These canonicalization methods allow system the flexibility to sign the various portions of the resource that matter for the workflow the signature serves.

Note: One consequence of signing the document is that URLs, identifiers and internal references are frozen and cannot be changed. This might be a desired feature, but it may also cripple interoperability between closed ecosystems where [re-identification](file:///C:\temp\managing.html) frequently occurs. For this reason, it is recommended that systems consider carefully the impact of any signature processes. The impact of signatures on [Document bundles](file:///C:\temp\documents.html) and their related processes is the most well understood use of digital signatures.