QUICK Logical Data Model Overview

Contents

This document provides an overview of the HL7 Quick Logical Data Model and as a guide to the artifacts provided with the model as supplemental materials. This document and the HTML documentation are informative and not to be voted on but comments and feedback are welcome.

The QUICK data model is derived from the HL7 Quality Improvement Core (QICore) Implementation Guide[1] and provides a uniform way for clinical decision support (CDS) and quality measures to refer to clinical data. In fact, the QUICK data model is generated directly from the published QICore FHIR profiles and the reference documentation is in the form of class-level documentation. QUICK is a custom view on the QICore profiles such that the QUICK model needs to be regenerated whenever significant changes occur to the QICore profiles to keep the two synchronized.

The top-level classes are mapped each to the FHIR Quality Profile that implements it (e.g. Procedure -> QICore Procedure). Each class is created from the base FHIR resource as constrained by and extended by the associated FHIR Quality Profile.

The QUICK data model can be used in Clinical Quality Language (CQL) expressions and as such some of the primitive and complex data types in QUICK use CQL data type names rather than the native FHIR data types.

The following table shows the top-level classes and associated FHIR Quality Profile name that implements it.

Logical Class name	FHIR Quality Profile
AdverseEvent	QICore-AdverseEvent
AllergyIntolerance	QICore-AllergyIntolerance
BodySite	QICore-BodySite
Communication	QICore-Communication
CommunicationRequest	QICore-CommunicationRequest
Condition	QICore-Condition
Device	QICore-Device
DeviceUseRequest	QICore-DeviceUseRequest
DeviceUseStatement	QICore-DeviceUseStatement
DiagnosticOrder	QICore-DiagnosticOrder
DiagnosticReport	QICore-DiagnosticReport
Encounter	QICore-Encounter
FamilyHistory	QICore-FamilyHistory
Flag	QICore-Flag
Goal	QICore-Goal
ImagingStudy	QICore-ImagingStudy
Immunization	QICore-Immunization

[1] Quality Improvement Core (QICore) Implementation Guide, http://hl7.org/fhir/DSTU2/qicore/qicore.html

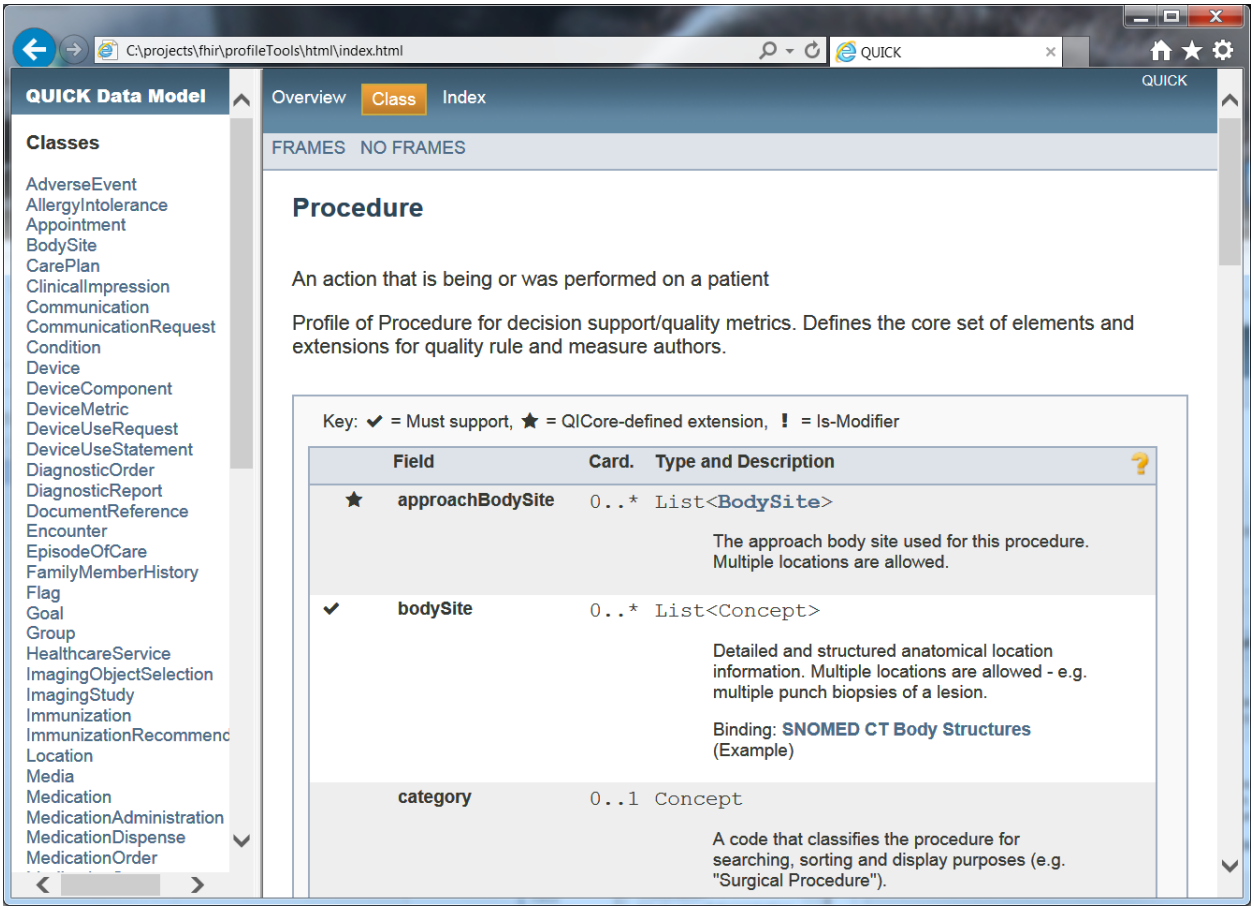
ImmunizationRecommendation	QICore-ImmunizationRecommendation
Location	QICore-Location
Medication	QICore-Medication
MedicationAdministration	QICore-MedicationAdministration
MedicationDispense	QICore-MedicationDispense
MedicationOrder	QICore-MedicationOrder
MedicationStatement	QICore-MedicationStatement
Observation	QICore-Observation
Organization	QICore-Organization
Patient	QICore-Patient
Practitioner	QICore-Practitioner
Procedure	QICore-Procedure
ProcedureRequest	QICore-ProcedureRequest
ReferralRequest	QICore-ReferralRequest
RelatedPerson	QICore-RelatedPerson
Specimen	QICore-Specimen
Substance	QICore-Substance

The full class list is found in the panel on the left under "Classes" label. Clicking on any class name displays the detailed class definition in the right panel (e.g. Procedure) showing the complete list of fields. QICore as of the Oct 2015 balloted version, profiled 34 resources out of the 94 base resources in FHIR DSTU2. Note some of the classes in the list (e.g. SupplyRequest, VisionPrescription, etc.) do not have an associated QICore profile, but are referenced as a type in one of the profiles so base resources are included as classes if they relate to at least one of the QICore profiled resources. Following the main-class list which directly map to a base resource is a list of complex types (E.g. Address, ContactPoint, etc.) which are also represented as classes.

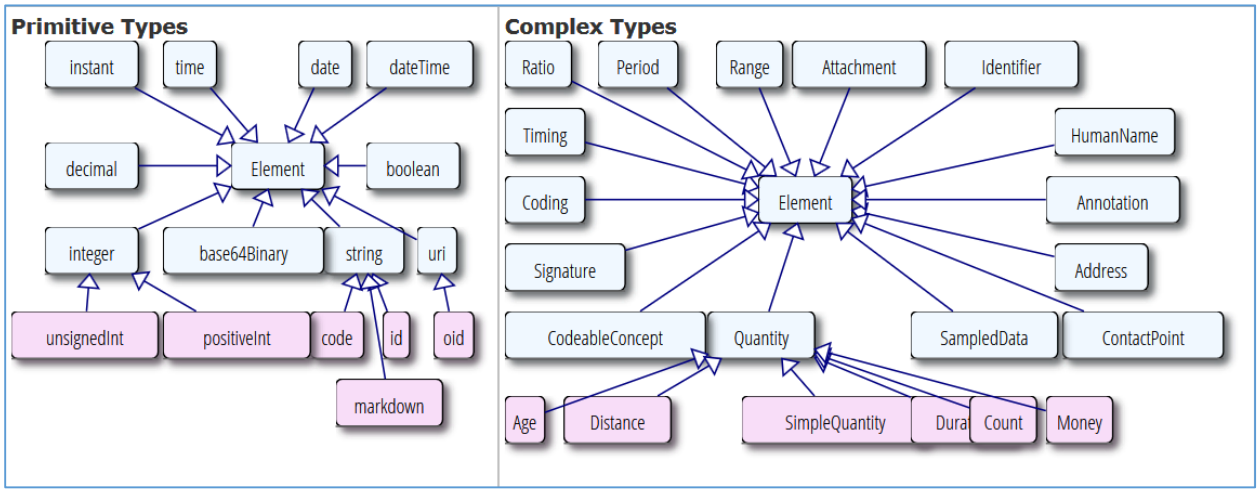
Elements and extensions are normalized and listed as fields on the class with a given type, description, and cardinality. In QUICK there is no distinction between an element and an extension - such distinctions can be found in the physical data model represented by FHIR and its implementation.

There are special flags associated with some of the elements. The element field name may be prefixed with one of 3 symbols for 1) Must support, 2) QICore-defined extension, and 3) is-modifier.

The must support fields are described in the *Quality Improvement Core (QICore) Implementation Guide* in the Must Support section. A QICore-defined extension is an extension defined in the QICore profile that extends the base resource. The is-modifier flag is a term from FHIR that indicates these elements may change the interpretation of the resource, depending on their value. Examples of modifying elements include status (in many resources), negations (e.g. Immunization.wasNotGiven), and certainty qualifications (e.g. Observation.reliability). QI implementations MUST always check the values of modifying elements.



QUICK is primarily intended for rule measure authors to reference QUICK for the data elements that appear in CQL expressions. The [FHIR data types] both simple and complex are defined in the following diagram.



For some of the primitive and complex types, there is a mapping of the FHIR data types to CQL data types. Where a CQL data type does not exist, the FHIR data type will be used.

Elements with no upper bound in cardinality (e.g. card=0..* or 1..*) are expressed with the CQL List<> notation so for example, Condition.bodySite with type=CodeableConcept and cardinality=0..* will be expressed as *List<Concept>*.

Here are the mappings of FHIR to CQL data types.

Primitive Types

FHIR	CQL
boolean	Boolean
integer	Integer
string	String
markdown	String
decimal	Decimal
uri	Uri
instant	DateTime
date	DateTime
dateTime	DateTime
time	Time
base64Binary	*n/a*

Complex Types

FHIR	CQL
Element	Any
Period	interval<DateTime>
Range	interval<Quantity>
Quantity	Quantity
CodeableConcept	Concept
Coding	Code
Attachment	*n/a*
Identifier	*n/a*
HumanName	*n/a*
Address	*n/a*
ContactPoint	*n/a*
SampledData	*n/a*
Timing	*n/a*
Ratio	*n/a*

The index is a cross reference of all fields found in all classes with links back to detailed class page with that field or extension.