

# Fast Healthcare Interoperability Resources (FHIR) 101 Cooking with Clinical Quality Language (CQL)

Thursday, September 26, 2019 – 4-5 PM EST

**Q:** Who owns and maintains the Atom Text Editor?

**A:** Atom is a free, open source text editor so it is a community driven development effort. The CQL standards support team have been significant contributors to the language-cql package in order to incorporate CQL syntax highlighting and semantic validation.

**Q:** Is there an open source tool that does the QDM to QI-Core replacement or will this be done in the MAT FHIR instance?

**A:** There is currently no tool so manual translations are more common. There has been some exploration of automatic translation during the FHIR Connectathons but they are quite difficult to do. While there has been some effort in the direction to automate, the QDM to QI-core mapping provides a way for that to happen manually. At this time, manual translations are occurring. A QDM to QI-Core translation is also being explored by the MAT and Bonnie teams in their exploration work for FHIR instances of the tool. If you have requests or feedback on the MAT and Bonnie tools on FHIR, reach out to [sb-bonnie-help@semanticbits.com](mailto:sb-bonnie-help@semanticbits.com) or [sb-mat-help@semanticbits.com](mailto:sb-mat-help@semanticbits.com).

**Q:** Can you talk about why QI-Core is necessary, as opposed to just using the base FHIR models?

**A:** The base FHIR resources describe the structure, e.g., the specific attributes of an encounter. The base FHIR Encounter resource, the Encounter.type (type of encounter – e.g., consultation, surgical day-care, skilled nursing, rehab) only provides example concepts and any codable concept is acceptable. US Core specifies a specific value set for Encounter.type such that two entities sharing data can understand the meaning on both sides of the exchange. US Core specifies this value set using CPT codes. US Core also aligns with the ONC Common Clinical Data Set and will align with the pending US Clinical Data Standards for Interoperability (USCDI). QI-Core adopts the US Core constraints and includes some additional functionality to assure measures include needed constraints.

**Q:** Can you provide an example of the Clinical Reasoning? That slide was a little unclear.

**A:** Level 5 is the Clinical Reasoning module and contains resources used to define a measure, a measure report and clinical decision support artifacts. The measure resources specify the version, date it was changed, who changed it – it's the metadata that surrounds the quality measure. The measure report resource is another one that is particularly important to us – it provides the metadata for the exchange of information but also has a place for you to populate results from a particular measurement.

In general, the resources in the administrative and clinical sections of FHIR really focus on representation of patient data. Those resources will have a reference to a specific patient instance. The resources in Clinical Reasoning are talking about how we deal with or describe rules that apply to populations in general or specific populations that meet certain criteria.

**Q:** Do you know if the Language CQL package and FHIR data model validation capabilities are available in any VS Code extensions?

**A:** Visual Studio Code is an open source text code editor, it has been done but we're not sure if it's published open-source anywhere. Atom and VS Code use the same underlying language protocol and the language-cql plug-in uses that protocol. If it has not been published anywhere already, it should be a trivial matter to plug it into VS Code. The language-cql package source is available in the CQ Framework Repository on GitHub (<https://github.com/cqframework>).

**Q:** Do you have examples of the data exchange to show patient-level and outcome-level reports?

**A:** For examples of a FHIR measure report, both at the patient level and summary level, the data exchange for quality measures (the DEQM implementation guide <http://hl7.org/fhir/us/davinci%2Ddeqm/2019May/STU3/>) has examples to demonstrate both of those for several measures.

**Q:** FHIR model vs QUICK model – I see a lot of measures (such as HEDIS) using FHIR model. Why two FHIR-based data models? And moving forward, should I just be using QUICK?

**A:** To clarify, QI-Core provides specific constraints to help measure developers write explicit expressions that enable unambiguous information retrieval. However, it is specific to a version of FHIR (e.g., STU 3, R4, etc.). At the Workgroup Meeting, there was an entire quarter dedicated to discussing the ability to provide a higher-level logical model that is FHIR version neutral – that is the focus of QUICK. There are reasons for wanting to get to a model that the eQMs can use, similar to the way QDM is used in existing eQMs, that's independent of the varying levels of adoption of FHIR and the different versions of FHIR that we see and focuses on quality improvement and measurement use cases. If you target a specific version of FHIR, anyone on a different version needs to transform to use that measure. If you express all measures in all different versions of FHIR then you have an explosion of measures. There is a case for having a version-independent representation of just the conceptual information required for measure development and that's the driver behind the QUICK logical model. There is an agreement that there should be continued exploration as we see what that looks like to try to determine the best answer. If you don't have a single model that is used for representation of the measure, then you're introducing transformation burden on the expression side but if you do have a single one then there is transformation burden somewhere along the way from the FHIR version to the target version. However, a usable QUICK logical model is not currently available and it will not be available for expressing eQMs; therefore, QI-Core is currently the appropriate model for authoring eQMs.

**Q:** Moving forward, is QUICK the model that should be used?

**A:** Currently, there is not enough of a tool chain to support description of an evaluation of a measure in QUICK. You can express a measure in QUICK using the CQL to ELM translator but to evaluate that, you would need a data source that speaks QUICK and we don't have that available yet.

**Q:** What is the easiest way to convert customized eCQMs created in MAT to FHIR for ingestion into PopHealth?

**A:** The conversion process we've undertaken for eCQMs has primarily been manual with review. It is a collaborative effort where measure developers will use the QDM to QI-Core mappings to replace the data elements and then we provide feedback on the CQL and we discuss the changes. We are not as familiar with PopHealth to answer this question at this time but we can follow up.

**Q:** Is conversion from CQL to ELM before or after FHIR conversion?

**A:** It is after because part of that process of converting from CQL to ELM is validating that the CQL is correct. Your CQL expressions will be referencing FHIR data types when you run it through the CQL to ELM translator. CQL is data model agnostic so the translator will work in the same fashion to convert that to ELM.

**Q:** In which reporting year's eCQMs are these changes expected to take effect, like from QDM to QI-Core, HQMF to FHIR, and QRDA to FHIR? Is there a timeline on when FHIR will be used for eCQM reporting?

**A:** CMS is exploring testing the eCQMs in FHIR so they are manually converting eCQMs to FHIR for testing purposes. They are also planning to pilot infrastructure for this. No timeline is set in stone but CMS is looking at testing and the results of testing to inform their timeline.

**Q:** Will measure reporting with FHIR be done exclusively through APIs?

**A:** CMS is looking at the use of FHIR-based APIs for burden reduction purposes and that is part of the pilot testing so there has been no formal decision for CMS' direction to move into FHIR. If anyone is interested in pilot testing FHIR data exchange with CMS, you can reach out to [Bridget.Calvert@cms.hhs.gov](mailto:Bridget.Calvert@cms.hhs.gov).

**Q:** Is there a tool like MAT that I can use to create a FHIR-based quality measure spec?

**A:** Work has started to create a MAT and Bonnie FHIR instance of the tool for general use but it will take a number of months with testing. The existing MAT and Bonnie versions that the measure developers are using with the current standard will be maintained. In addition, there will be FHIR instances of the tool and there will be an announcement when those tools are available.

**Q:** How long will the existing versions of MAT and Bonnie be supported?

**A:** The existing version of MAT and Bonnie will continue to be supported as long as CMS requires the current standards to be reported. Any changes or moves to FHIR will be dependent on the testing that will be ongoing for the next year. The results of testing and information will be shared once CMS has that information available to share.

**Q:** Is there a list of active Open Source projects we should be using, which are production ready, versus that which have been abandoned and should be avoided?

**A:** There is a Community Projects page

([https://github.com/cqframework/clinical\\_quality\\_language/wiki/Community-Projects](https://github.com/cqframework/clinical_quality_language/wiki/Community-Projects)) where we encourage people to say what they're doing there. As far as production ready, we're not sure that there is any claim on that for projects.

**Q:** Is PopHealth active? What about the execution engines from DBCG and/or cqframework?

<https://github.com/cqframework/cql-execution>

**A:** As for PopHealth, it was migrated to OSEHRA (<https://www.osehra.org/popHealth>) and appears to be actively maintained there. As for the others, those are active projects that are supported and maintained and things that we use as part of this effort and testing to make sure all the specs work and the authoring environments function as they should – CQL execution, CQL engine, and the javascript engine.

**Q:** I'm trying to get the cqf-ruler examples working in my local instance (see here:

<https://github.com/DBCG/cqf-ruler/wiki/Resource-Loading>). But, I can't access some of the example resource bundles. Can you point to some working bundles?

**A:** You can find the latest bundles here: <http://github.com/DBCG/connectathon>