V3_IG_CQL_HQMF_R1_D4_Vol2

HL7 Version 3 Implementation Guide: Clinical Quality Language (CQL)-based Health Quality Measure Format (HQMF) Release 1, STU 4 - US Realm

HL7 Standard for Trial Use (STU)

May 2019

Volume 2 - Using QDM in CQL-based Measures

Sponsored by:
Clinical Quality Information Work Group

IMPORTANT NOTES:

HL7 licenses its standards and select IP free of charge. If you did not acquire a free license from HL7 for this document, you are not authorized to access or make any use of it. To obtain a free license, please visit http://www.HL7.org/implement/standards/index.cfm.

If you are the individual that obtained the license for this HL7 Standard, specification or other freely licensed work (in each and every instance Specified Material), the following describes the permitted uses of the Material.

A. HL7 INDIVIDUAL, STUDENT AND HEALTH PROFESSIONAL MEMBERS, who register and agree to the terms of HL7's license, are authorized, without additional charge, to read, and to use Specified Material to develop and sell products and services that implement, but do not directly incorporate, the Specified Material in whole or in part without paying license fees to HL7. INDIVIDUAL, STUDENT AND HEALTH PROFESSIONAL MEMBERS wishing to incorporate additional items of Special Material in whole or part, into products and services, or to enjoy additional authorizations granted to HL7 ORGANIZATIONAL MEMBERS as noted below, must become ORGANIZATIONAL MEMBERS of HL7.

B. HL7 ORGANIZATION MEMBERS, who register and agree to the terms of HL7's License, are authorized, without additional charge, on a perpetual (except as provided for in the full license terms governing the Material), non-exclusive and worldwide basis, the right to (a) download, copy (for internal purposes only) and share this Material with your employees and consultants for study purposes, and (b) utilize the Material for the purpose of developing, making, having made, using, marketing, importing, offering to sell or license, and selling or licensing, and to otherwise distribute, Compliant Products, in all cases subject to the conditions set forth in this Agreement and any relevant patent and other intellectual property rights of third parties (which may include members of HL7). No other license, sublicense, or other rights of any kind are granted under this Agreement.

C. NON-MEMBERS, who register and agree to the terms of HL7's IP policy for Specified Material, are authorized, without additional charge, to read and use the Specified Material for evaluating whether to implement, or in implementing, the Specified Material, and to use Specified Material to develop and sell products and services that implement, but do not directly incorporate, the Specified Material in whole or in part. NON-MEMBERS wishing to incorporate additional items of Specified Material in whole or part, into products and services, or to enjoy the additional authorizations granted to HL7 ORGANIZATIONAL MEMBERS, as noted above, must become ORGANIZATIONAL MEMBERS of HL7. Please see http://www.HL7.org/legal/ippolicy.cfm for the full license terms governing the Material.

Ownership. Licensee agrees and acknowledges that **HL7 owns** all right, title, and interest, in and to the Materials. **Licensee shall take no action contrary to, or inconsistent with,** the foregoing.

Licensee agrees and acknowledges that HL7 may not own all right, title, and interest, in and to the Materials and that the Materials may contain and/or reference intellectual property owned by third parties ("Third Party IP"). Acceptance of these License Terms does not grant Licensee any rights with respect to Third Party IP. Licensee alone is responsible for identifying and obtaining any necessary licenses or authorizations to utilize Third Party IP in connection with the Materials or otherwise. Any actions, claims or suits brought by a third party resulting from a breach of any Third Party IP right by the Licensee remains the Licensee's liability.

Following is a non-exhaustive list of third-party terminologies that may require a separate license:

Terminology	Owner/Contact
Current Procedures Terminology (CPT) code set	American Medical Association
SNOMED CT	SNOMED International
Logical Observation Identifiers Names & Codes (LOINC)	Regenstrief Institute
International Classification of Diseases (ICD) codes	World Health Organization (WHO)
NUCC Health Care Provider Taxonomy code set	American Medical Association. Please see nucc.org. AMA licensing contact: 312-464-5022 (AMA IP services)

Contents

List of Figures

List of Tables

List of Snippets

List of Conformance Requirements

1 Introduction

1.1 Purpose

The Institute of Medicine (IOM) defines quality as: “The degree to which health services for individuals and populations increase the likelihood of desired health outcomes and are consistent with current professional knowledge.” [1] For care quality to be evaluated, it must be standardized and communicated to the appropriate organizations. To that end, this Implementation Guide has been written to provide guidance for authoring electronic Clinical Quality Measures (eCQMs) utilizing the following standards:

- Quality Data Model (QDM) v5.5 [2]
- Clinical Quality Language (CQL) R1.4 [3]
- Health Quality Measures Format Release 1 Normative (HQMF R1 Normative) [4]

Although the specification is based on the 1.4 version of CQL, backwards-compatible future versions of CQL can be used as well. In addition, if necessary, the 1.2 version of CQL can be used without loss of functionality for this Implementation Guide.

Note that HQMF releases have typically been referred to by their STU version, so HQMF R2.1 was referring to the STU version (2.1), not the full release version, which is still 1. Now that HQMF has been released as a normative specification, the STU version is dropped. Except where noted specifically, references to HQMF in this guide are to the normative release 1 version.

Except where noted, material from the above specifications is not reproduced here.

1.2 Structure of this Guide

Three volumes comprise this *HL7 Version 3 Implementation Guide: Clinical Quality Language (CQL)-based Health Quality Measure Format (HQMF), Release 1 STU4 (US Realm), Standard for Trial Use*:

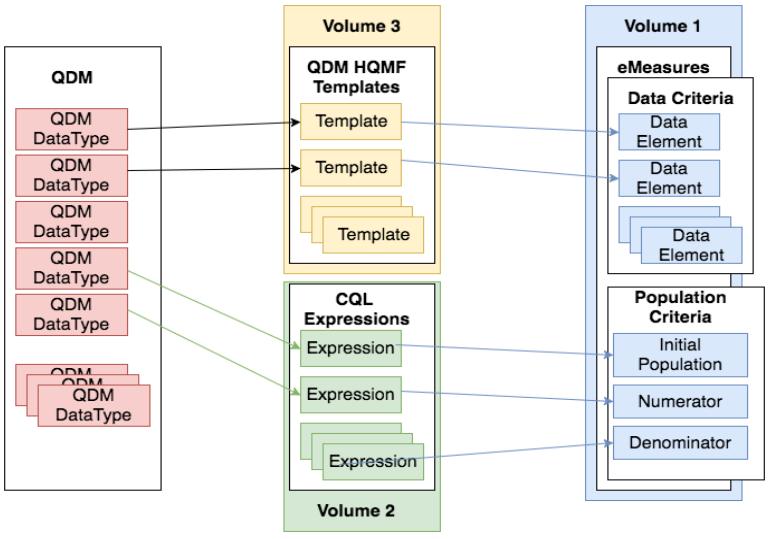

Figure 1: Relationship between QDM, CQL, eCQM, and the volumes of this IG

Volume 1 : Provides narrative introduction, background material, and conformance requirements for representing CQL-based eCQMs in HQMF.

Volume 2 : Describes how to incorporate version 5.5 of the Quality Data Model into a CQL-based eCQM in accordance with accepted formatting and usage conventions.

Volume 3 : Contains the HQMF templates for QDM data elements, necessary for constructing QDM+CQL-based HQMF measures.

1.3 Structure of this Volume

In this section, we present an outline of this volume of this *HL7 Version 3 Implementation Guide: Clinical Quality Language (CQL)-based Health Quality Measure Format (HQMF), Release 1 STU4 (US Realm), Standard for Trial Use*.

This volume is divided into 3 chapters:

Chapter 1 provides an introduction to this IG and provides more information about QDM and QRDA (related standards).

Chapter 2 provides conformance requirements for any CQL document intended to be used in an HQMF document. This chapter follows the structure of a CQL library (library-line, using-line, valueset-line, definitions).

Chapter 3 describes the connection between QRDA and HQMF.

1.4 Scope

This IG is a conformance profile, as described in the "Refinement and Localization" [6] section of the HL7 Version 3 Interoperability Standards. The base standard for this IG is the HL7 Health Quality Measures Format Normative Release 1. This IG does not describe every aspect of HQMF. Rather, it defines constraints on the base HQMF used in a CQL-based HQMF document in the US Realm. Additional optional HQMF elements, not included here, can be included and the result will be compliant with the specifications in this guide.

1.5 Conventions

The keywords SHALL, SHALL NOT, SHOULD, SHOULD NOT, MAY and NEED NOT in this document are to be interpreted as described in the HL7 Version 3 Publishing Facilitator's Guide.

- **SHALL**: an absolute requirement for the particular element. Where a SHALL constraint is applied to an XML element, that element must be present in an instance but may have an exceptional value (i.e., may have a nullFlavor), unless explicitly precluded. Where a SHALL constraint is applied to an XML attribute, that attribute must be present and must contain a conformant value.
- **SHALL NOT**: an absolute prohibition against inclusion
- **SHOULD/SHOULD NOT**: best practice or recommendation. There may be valid reasons to ignore an item, but the full implications must be understood and carefully weighed before choosing a different course

- **MAY/NEED NOT**: truly optional; can be included or omitted as the author decides with no implications

1.6 Background

1.6.1 Quality Data Model

QDM [2] is a model of information that allows quality measure developers to describe clearly and unambiguously the data required to calculate performance measures. It also allows EHR and other clinical electronic system vendors to unambiguously interpret the data and clearly locate the data required. QDM is intended to enable automation of the quality measurement process, avoiding the need for abstraction of existing information or attestation of actions that have already occurred. The templates supplied in volume 3 of this IG represent QDM v5.5 [2].

From HL7's perspective, the QDM is a domain analysis model that defines concepts recurring across quality measures. Figure 2 illustrates components of the QDM relevant to understanding how that model guides the construction of HQMF templates in eCQM documents.

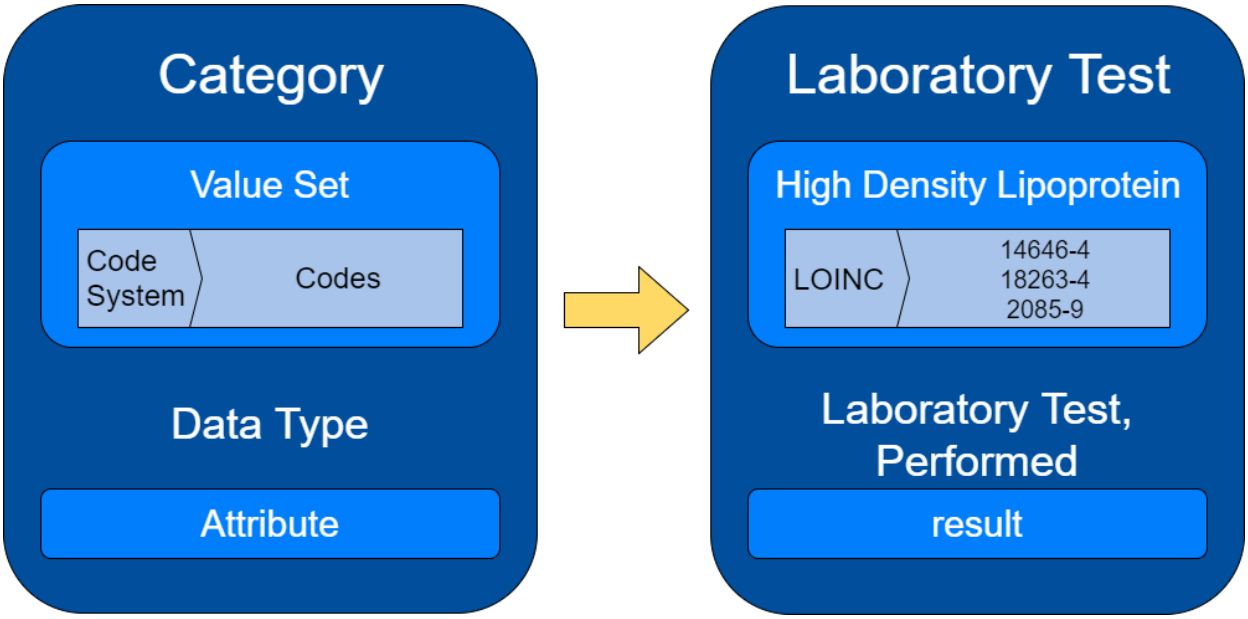

Figure 2: QDM element structure [2]

A QDM element is composed of a category, the data type in which that category is expected to be used, and a value set or direct reference code that the data type is associated with. QDM attributes include data type-specific attributes and data flow attributes, which provide additional structure to describe the QDM element. Figure 2 shows an example of a "Laboratory Test, Performed: High Density Lipoprotein" QDM element where the QDM data type, "Laboratory Test, Performed" is bound to a value set containing LOINC codes for "High Density Lipoprotein".

1.6.2 Relationship to Quality Reporting Document Architecture

The HL7 Clinical Document Architecture (CDA) Quality Reporting Document Architecture (QRDA) [5] is a standard for reporting the results of eCQMs. QRDA also specifies a method for referencing the eCQM(s) for which the quality report contains results. The HL7 CDA QRDA Category I (an individual patient-level quality report) is a CDA R2 standard.

1.6.3 QDM-HQMF Templates and QRDA Templates

Each QDM-HQMF template is a reusable building block used to form a distinct data criterion within a measure or across different measures; it has an identifier that uniquely identifies that template. For example, 2.16.840.1.113883.10.20.28.3.111 is the template identifier for the Family History QDM-HQMF template. A QDM-HQMF template specifies data criteria of a measurable data element.

An eCQM specification for a particular measure is constructed based on the QDM-HQMF templates. As a result, an implementer should be able to dynamically generate a QRDA Category I quality report based on the eCQM specification. For example, if an eCQM has a data criterion of "Family History", then an implementer should report the data using the corresponding Family History QRDA template. To maximize such automatic conversion between the QDM-HQMF templates and their corresponding QRDA templates, the QDM-HQMF templates are designed to align with their QRDA templates counterpart.

Volume 3 of this IG contains a complete list of QDM-HQMF templates based on QDM 5.5. To support the ongoing need for performance measures, new QDM data types may need to be added to the QDM, and corresponding new QDM-HQMF templates will then be developed and added to the QDM-HQMF template library.

2 CQL Basics

2.1 Libraries

A CQL artifact is referred to as a library. Any CQL library referenced as an expression document within HQMF must contain a library declaration line as the first line of the library.

Conformance Requirement 1 (Library Declaration):

- The library declaration line **SHALL** contain a version number.
- The library version number **SHALL** follow the convention :

```
<major>.<minor>.<patch>
```

2.1.1 Library Versioning

This IG recommends an approach to versioning libraries used within CQL-Based HQMF Measures to help track and manage dependencies. The approach recommended herein is based on the Apache APR Versioning Scheme [11].

Because CQL libraries can contain both public and private components, there are three main types of changes that can be made to a library. First, a library can be changed in a way that would alter the public use of its components. Second, a library can be changed by adding new components or functionality but without changing existing components. And third, a library can be changed in a way that does not impact the public use of its components at all, but only corrects or improves the originally intended functionality.

By exposing version numbers that identify all three types of changes, libraries can be versioned in a way that makes clear when a change will impact usage, versus when a change can potentially be safely incorporated as an update. The first type of change will be referred to as a "major" change, and will require incrementing of the "major version number". The second type of change will be referred to as a "minor" change, and will only require incrementing of the "minor version number". And finally, the third type of change will be referred to as a "patch", and will only require incrementing the "patch version number".

Version numbers for CQL libraries can then be represented as:

```
<major>.<minor>.<patch>
```

For example:

```
library CMS146 version '1.0.0'
```

This would indicate the first major version of the CMS146 library. A minor change could be released by incrementing the minor version:

```
library CMS146 version '1.1.0'
```

And a major change could be released by incrementing the major version, and resetting the minor version: Minor changes are expected to retain backwards-compatibility, but may introduce new features and functionality, while patch changes are expected to retain forward and backwards-compatibility, and may only be used to fix issues.

```
1 library EXM146 version '4.0.0'
```

Snippet 1: Library line from `EXM146v4_CQL.cql` the fourth major version.

2.1.2 Nested Libraries

CQL allows libraries to re-use logic already defined in other libraries. This is accomplished by utilizing the **`include`** line as in Snippet 2.

```
12 includes Common version '2.0.0' called Common
```

Snippet 2: Nested library within `EXM146v4_CQL.cql`

The set of all CQL libraries used to define an HQMF measure must adhere to Conformance Requirement 2.

Conformance Requirement 2 (Nested Libraries):

- CQL libraries **SHALL** be structured such that all `components` of the `populationCriteriaSection` will only explicitly reference a single library.
- CQL libraries SHALL use a `called` clause for all included libraries to establish a local identifier for the included library.
- Included libraries SHOULD use consistent local identifiers across usage (i.e. the same local identifier should be used for a given shared library whenever it is included).

Because of this conformance statement, the primary library for a measure can always be determined by looking at the library referenced by the initial population criteria for the measure.

2.1.3 Library Namespaces

CQL allows libraries to define a namespace that can be used to organize libraries across different groups of users. Within a namespace library names are required to be unique, but across namespaces, the same library name may be reused. For example, OrganizationA and OrganizationB can both define a library named Common, so long as they use different namespaces. For example, consider the following library declaration:

```
12  library CMS.Common version '2.0.0'
```

This example declares a library named Common in the CMS namespace. Per the CQL specification, the namespace for a library is included in the ELM, along with a URI that provides a globally unique, stable identifier for the namespace. For example, the URI for the CMS namespace would be https://ecqi.healthit.gov/ecqm/measures

Note that this is a URI that may or may not correspond to a reachable web address (a URL). The important aspect is not the addressability, but the uniqueness, ensuring that library name collisions cannot occur.

Conformance Requirement 2.A (Library Namespaces):

- CQL libraries **SHOULD** use namespaces.
- When a namespace is not used, the library **SHALL** be considered part of a "public" global namespace for the purposes of resolution within a given environment.

In addition, because the namespace of a library is part of the text, changing the namespace of a library requires a new version, just like any other change to the text of the library. However, because a change to the namespace is not a material change to the measure itself, changing the namespace does not require a different version-independent identifier to be used for the measure.

2.2 Data Model

CQL can be used with any data model. However, within the context of HQMF any referenced CQL library must identify a data model and that data model must be `QDM`.

Conformance Requirement 3 (CQL Data Model):

- All CQL expressions used directly or indirectly within a measure **SHALL** reference a single data model consistent with the measure's HQMF templates.
- Data Model declarations **SHALL** include a version declaration.
- All CQL expressions used directly or indirectly within a measure **SHALL** be defined using the Patient context. The Unfiltered context **SHALL NOT** be used.

For example:

```
10  using QDM version '5.0.2'
```

Snippet 3: Data Model line from `EXM146v4 CQL.cql`

2.3 Code Systems

Conformance Requirement 4 describes how to specify a code system within a CQL library.

Conformance Requirement 4 (Code System Specification):
Within CQL the identifier of any code system reference **SHALL** be specified using a URN for the code system.

For example:

```
5  codesystem "SNOMED-CT": 'urn:oid:2.16.840.1.113883.6.96'
6     version 'urn:hl7:version:201609'
```

Snippet 4: codesystem definition line from `Terminology CQL.cql`.

The local identifier for the codesystem (`"SNOMED-CT"` in this case) should include the friendly name of the code system and optionally, an indication of the version, separated with a colon.

Version information for code systems is not required to be included in eCQMs; terminology versioning information may be specified externally. However, if versioning information is included, it must be done in accordance with the conformance requirements specified in this IG.

The URN for the version part of the identifier uses the namespace `version` to indicate that this is the logical identifier of a version.

In addition, based on request for comment (RFC) 3406 [8] for formal or informal URNs, there is an expectation of uniqueness for the namespace. The `hl7` prefix provides this unique namespace.

2.4 Value Sets

Conformance Requirement 5 describes how to specify a valueset within a CQL library.

Conformance Requirement 5 (Value Set Specification):
Within CQL the identifier of any value set reference **SHALL** be specified using a URN for the value set.

For example:

```
14  valueset "Acute Pharyngitis": 'urn:oid:2.16.840.1.113883.3.464.1003.102.12.1011'
```

Snippet 5: Valueset reference from `EXM146v4 CQL.cql`

The local identifier for the value set within CQL should be the same as the name of the value set in the Value Set Authority Center (VSAC) [9]. However, because the name of the value set is not guaranteed to be unique, it is possible to reference multiple value sets with the same name, but different identifiers. When this happens in a CQL library, the local identifier should be the name of the value set with a qualifying suffix to preserve the value set name as a human-readable artifact, but still, allow unique reference within the CQL library.

For example:

```
14  valueset "Acute Pharyngitis (1)":
'urn:oid:2.16.840.1.113883.3.464.1003.102.12.1011.1'
15  valueset "Acute Pharyngitis (2)":
'urn:oid:2.16.840.1.113883.3.464.1003.102.12.1011.2'
```

Version information for value sets is not required to be included in eCQMs; terminology versioning information may be specified externally. However, if versioning information is included, it must be done in accordance with the conformance requirements specified in this IG. Note that because the VSAC supports different approaches to retrieving the expansion of a valueset through its Sharing Value Sets (SVS) API [10]. For the purposes of this guidance, two approaches are described: 1) by version, and 2) by profile.

2.4.1 By Version

Conformance Requirement 6 describes how to retrieve an expansion of a value set by version.

Conformance Requirement 6 (Value Set Specification By Version):
When retrieving the expansion of a value set by version, the version identifier attribute **SHALL** be a URN defining the version.

For example: As with code systems, the version namespace is used to indicate that the identifier is a

version.

```
7  valueset "Encounter Inpatient SNOMEDCT Value Set":
8     'urn:oid:2.16.840.1.113883.3.666.7.307' version 'urn:hl7:version:20160929'
```

Snippet 6: valueset definition from `Terminology CQL.cql`.

2.4.2 By Profile

When retrieving expansions by profile, the version identifier attribute conforms to Conformance Requirement 7.

> **Conformance Requirement 7 (Value Set Version Specification By Profile):**
> When retrieving the expansion of a value set by profile, the version identifier attribute **SHALL** be a URN defining the profile.

For example:

```
valueset "Face-to-Face Interaction":
'urn:oid:2.16.840.1.113883.3.464.1004.101.12.1048'
   version 'urn:hl7:profile:MU2%20Update%202016-04-01'
```

Here, the profile namespace is used to indicate that the identifier is a profile.

2.4.3 Representation in HQMF

The HQMF XML representation of valueset declarations is discussed in Volume 1 Chapter 3 of this IG.

2.5 Codes

When direct reference codes are represented within CQL, the logical identifier is not recommended to be a URN. Instead, the logical identifier is the code from the code system.

> **Conformance Requirement 8 (Direct Reference Codes):**
> When direct reference codes are represented within CQL, the logical identifier:
>
> - **SHALL NOT** be a URN.
> - **SHALL** be a code from the code system.

```
9  code "Venous foot pump, device (physical object)": '442023007' from "SNOMED-CT"
```

Snippet 7: code definition from `Terminology CQL.cql`.

Note that for direct reference code usage, the local identifier (in Snippet 7 the local identifier is `"Venous foot pump, device (physical object)"`) should be the same as the description of the code within the terminology in order to avoid conflicting with any usage or license agreements

with the referenced terminologies, but can be different to allow for potential naming conflicts, as well as simplification of longer names when appropriate.

2.5.1 Representation in HQMF

When direct reference codes are used within CQL-Based HQMF measures, they will be represented in the HQMF HTML (Human-readable) as:

```
"Assessment, Performed: Assessment of breastfeeding"
   using "Assessment of breastfeeding SNOMED-CT Code (709261005)"
```

The HQMF XML representation of code declarations is discussed in Volume 1 Chapter 3 of this IG.

2.6 Concepts

In addition to codes, CQL supports a concept construct, which is defined as a set of codes that are all semantically equivalent. There is no direct counterpart within HQMF currently, and it is not clear how or when this construct would be used within measure development. As such, concepts are not recommended for use and should be avoided in favor of the other terminological constructs.

Conformance Requirement 9 (Concepts):
The CQL construct, **`concept`**, **SHALL NOT** be used.

2.7 Terminology Membership Testing

The base CQL specification supports a wide variety of terminology-related functionality, including the ability to test for value set and code system membership. For Code and Concept values, membership testing is performed using the in operator. However, this operator also works with string values. For example:

```
valueset "Face-to-Face Interaction":
  'urn:oid:2.16.840.1.113883.3.464.1004.101.12.1048'
define "String Testing":
  '442023007' in "Face-to-Face Interaction"
```

Rather than using a Code, which carries system and potentially version information, the above example tests for a simple string. There are use cases where this functionality is appropriate, but in general, using this method for terminology membership testing ignores code system and results in the potential for invalid matches if code systems happen to have the same code values. To avoid this possibility, simple string-based terminology membership testing is disallowed.

Conformance Requirement 9.A (Terminology Membership Testing):
The String-valued overloads of the terminology membership testing operator (**`in`**), **SHALL NOT** be used.

2.8 Library-level Identifiers

A “library-level identifier” is any named expression, function, parameter, code system, value set, concept, or code defined in the CQL. The library name referenced in the library-line, the data model, and any referenced external library should not be considered “library-level identifiers”. Library-level identifiers ought to be given a descriptive meaningful name (avoid abbreviations) and conform to Conformance Requirement 10.

Conformance Requirement 10 (Library-level Identifiers):
Library-level identifiers referenced in the CQL:

- **SHOULD** Use quoted identifiers
- **SHOULD** Use Title Case
- **MAY** Include spaces

For example:

```
14  define function
15     "Includes Or Starts During"(Diagnosis "Diagnosis", Encounter "Encounter,
       Performed"):
16        Diagnosis.prevalencePeriod includes Encounter.relevantPeriod
17           or Diagnosis.prevalencePeriod starts during Encounter.relevantPeriod
```

Snippet 8: Function definition from `Common-2.0.0 CQL.cql`

2.9 QDM Data Type Names

This section refers only to Data Types described in the QDM specification [2]. QDM data types referenced in CQL libraries to be included in the HQMF conform to Conformance Requirement 11.

Conformance Requirement 11 (Data Type Names):
QDM data types referenced in the CQL **SHALL**:

- Use quoted identifiers
- Use PascalCase plus appropriate spacing
- Conform to Table 1

2.9.1 Negation in QDM

Two commonly used patterns for negation in quality measurement are:

- Absence of evidence for a particular event
- Documentation of an event or activity not occurring, together with a reason

For the purposes of quality measurement, when looking for documentation that a particular event did not occur, it must be documented with a reason in order to meet the intent. If a reason is not part of the intent,

then the absence of evidence pattern should be used, rather than documentation of an event not occurring.

To address the reason an action did not occur (negation rationale), the eCQM must define the event it expects to occur using appropriate terminology to identify the kind of event (using a value set or direct reference code), and then use additional criteria to indicate that the event did not occur, as well as identifying a reason.

The following examples differentiate methods to indicate (a) presence of evidence of an action, (b) absence of evidence of an action, and (c) negation rationale for not performing an action. In each case, the “action” is an administration of medication included within a value set for “Antithrombotic Therapy”.

Presence
Evidence that “Antithrombotic Therapy” (defined by a medication-specific value set) was administered:

```
define "Antithrombotic Administered":
   ["Medication, Administered": "Antithrombotic Therapy"]
```

Absence
No evidence that “Antithrombotic Therapy” medication was administered:

```
define "No Antithrombotic Administered":
   not exists (
       ["Medication, Administered": "Antithrombotic Therapy"]
   )
```

Negation Rationale
Evidence that “Antithrombotic Therapy” medication administration did not occur for an acceptable medical reason as defined by a value set referenced by the eCQM (i.e. negation rationale):

```
define "Antithrombotic Not Administered":
   ["Medication, Not Administered": "Antithrombotic Therapy"] NotAdministered
      where NotAdministered.negationRationale in "Medical Reason"
```

In this example for negation rationale, the logic looks for a member of the value set "Medical Reason" as the rationale for not administering the medication. However, underlying systems might not represent the negated action with a code from the "Antithrombotic Therapy" value set. When justifying the reason for not administering a class of medications, clinicians do not generally specify one of the medications in the class, they most often indicate avoidance of the entire class. In these cases, the value set may be used as a placeholder to indicate the medication class was not administered. Implementations processing data reported in this way should take into account that the reported data may not be returned with a single code, but rather a value set identifier, and should consider data with the appropriate value set identifier as satisfying the criteria for value set membership.

Similarly, "Procedure, Not Performed": "Cardiac Surgery" should not require specification of which cardiac surgery in a value set was not performed, but only reference any member of the class of procedures defined by the value set. The same process works for any application of negation rationale.

In addition, the above code demonstrates the use of the "Not" modifier to retrieve all EHR entries indicating a medication that was not administered to the patient due to a "Medical Reason".

Conformance Requirement 12 (Negation):
Negated QDM data type names **SHALL** conform to Table 1.
Negated criteria **SHALL** include a reference to negationRationale.

QDM Data Type Names	Negated Name
"Adverse Event"	N/A
"Allergy/Intolerance"	N/A
"Assessment, Order"	"Assessment, Not Ordered"
"Assessment, Recommended"	"Assessment, Not Recommended"
"Assessment, Performed"	"Assessment, Not Performed"
"Care Goal"	N/A
"Communication, Performed"	"Communication, Not Performed"
"Device, Order"	"Device, Not Ordered"
"Device, Recommended"	"Device, Not Recommended"
"Device, Applied"	"Device, Not Applied"
"Diagnosis"	N/A
"Diagnostic Study, Order"	"Diagnostic Study, Not Ordered"
"Diagnostic Study, Recommended"	"Diagnostic Study, Not Recommended"
"Diagnostic Study, Performed"	"Diagnostic Study, Not Performed"
"Encounter, Order"	"Encounter, Not Ordered"
"Encounter, Recommended"	"Encounter, Not Recommended"
"Encounter, Performed"	"Encounter, Not Performed"
"Family History"	N/A
"Immunization, Order"	"Immunization, Not Ordered"
"Immunization, Administered"	"Immunization, Not Administered"
"Intervention, Order"	"Intervention, Not Ordered"
"Intervention, Recommended"	"Intervention, Not Recommended"
"Intervention, Performed"	"Intervention, Not Performed"

"Laboratory Test, Order"	"Laboratory Test, Not Ordered"
"Laboratory Test, Recommended"	"Laboratory Test, Not Recommended"
"Laboratory Test, Performed"	"Laboratory Test, Not Performed"
"Medication, Active"	N/A
"Medication, Administered"	"Medication, Not Administered"
"Medication, Dispensed"	"Medication, Not Dispensed"
"Medication, Discharge"	"Medication, Not Discharged"
"Medication, Order"	"Medication, Not Ordered"
"Patient Care Experience"	N/A
"Patient Characteristic"	N/A
"Patient Characteristic Birthdate"	N/A
"Patient Characteristic Clinical Trial Participant"	N/A
"Patient Characteristic Ethnicity"	N/A
"Patient Characteristic Expired"	N/A
"Patient Characteristic Payer"	N/A
"Patient Characteristic Race"	N/A
"Patient Characteristic Sex"	N/A
"Provider Care Experience"	N/A
"Physical Exam, Order"	"Physical Exam, Not Ordered"
"Physical Exam, Recommended"	"Physical Exam, Not Recommended"
"Physical Exam, Performed"	"Physical Exam, Not Performed"
"Procedure, Order"	"Procedure, Not Ordered"
"Procedure, Recommended"	"Procedure, Not Recommended"
"Procedure, Performed"	"Procedure, Not Performed"
"Substance, Order"	"Substance, Not Ordered"
"Substance, Recommended"	"Substance, Not Recommended"
"Substance, Administered"	"Substance, Not Administered"
"Symptom"	N/A

Table 1: QDM Data Type names.

2.10 Attribute Names

This section refers only to attributes described in the QDM specification. All QDM attributes referenced in the CQL follow Conformance Requirement 13.

Conformance Requirement 13 (Attribute Names):
QDM attributes referenced in the CQL:

- **SHALL NOT** Use quoted identifiers
- **SHALL** Use camelCase[§]

[§]Note QDM considers Datetime to be one word when considering casing.

Examples of attributes conforming to Conformance Requirement 13 is given below. For a full list of valid of attributes per QDM datatype please refer to the QDM specification [2].

```
relevantPeriod
authorDatetime
result
```

2.11 Aliases and Argument Names

Aliases are used in CQL as local variable names to refer to sections of code. When defining a function, argument names are used to create scoped variables that refer to the function inputs. Both aliases and argument names conform to Conformance Requirement 14.

Conformance Requirement 14 (Aliases and Argument Names):
Aliases and argument names referenced in the CQL :

- **SHALL NOT** Use quoted identifiers
- **SHALL** Use PascalCase
- **SHOULD** Use descriptive names (no abbreviations)

For example:

```
define "Encounters During Measurement Period":
   "Valid Encounters" QualifyingEncounter
      where QualifyingEncounter.relevantPeriod during "Measurement Period"

define function "ED Stay Time"(Encounter "Encounter, Performed"):
   duration in minutes of Encounter.locationPeriod
```


3 Translation to ELM

Tooling exists to support translation of CQL to ELM for distribution in XML or JSON formats. As described in Volume 1, these distributions are included with eCQMs to facilitate implementation. The existing translator tooling applies to both measure and decision support development, and has several options available to make use of different data models in different environments. For measure development with Quality Data Model, the following options are recommended:

Option	Description	Recommendation
EnableAnnotations	This instructs the translator to include the source CQL as an annotation within the ELM.	This option should be used to ensure that the distributed ELM could be linked back to the source CQL.
EnableLocators	This instructs the translator to include line number and character information for each ELM node.	This option should be used to ensure that distributed ELM could be tied directly to the input source CQL.
DisableListDemotion	This instructs the translator to disallow demotion of list-valued expressions to singletons. The list demotion feature of CQL is used to enable functionality related to use within Fast Healthcare Interoperability Resources (FHIR).	This option should be used with QDM to ensure list demotion does not occur unexpectedly.
DisableListPromotion	This instructs the translator to disallow promotion of singletons to list-valued expressions. The list promotion feature of CQL is used to enable functionality related to use within Fast Healthcare Interoperability Resources (FHIR).	This option should be used with QDM to ensure list promotion does not occur unexpectedly.
DisableMethodInvocation	This instructs the translator to disallow method-style invocation. The method-style invocation feature of CQL is used to enable functionality related to use within Fast Healthcare Interoperability Resources (FHIR).	This option should be used with QDM to ensure method-style invocation cannot be used within eCQMs.
EnableDateRangeOptimization	This instructs the translator to optimize date range filters by moving them inside retrieve expressions.	This feature may be used with QDM.
EnableResultTypes	This instructs the translator to include inferred result types in the output ELM.	This feature may be used with QDM.
EnableDetailedErrors	This instructs the translator to include detailed error information. By default, the translator only reports root-cause errors.	This feature should not be used with QDM.
DisableListTraversal	This instructs the translator to disallow traversal of list-valued expressions. With QDM, disabling this feature would prevent a useful capability.	This feature should not be used with QDM.

4 Reporting Results

The results of eCQMs represented in HQMF are reported using the Quality Reporting Document Architecture (QRDA) format. The QRDA standard [5] specifies the use of an HQMF measure ID (an `id` element under the `QualityMeasureDocument` root element) and population IDs (each `id` element under each population criteria) when reporting the results of a measure.

Measure reports generated from the eCQMs conforming to this implementation guide shall conform to the QRDA specification and use the measure and population IDs when identifying results. A measure with multiple denominators would define a different ID for each result, so a measure report can identify which result belongs to which denominator.

Change Log

STU4

Each volume has a separate change log. This change log only addresses changes to this volume:

- STU4B1: Disallowed used of string overloads of CQL membership testing operators
- STU4B2: Added change log
- STU4B3: Added related-context retrieve example
- STU4B5: Added timing examples for relevantPeriod vs relevantDatetime and authorDatetime
- STU4B23,24,25: Corrected typographical and grammatical errors
- STU4B31: Added section 2.1.3 and Conformance Requirement 2.A to address library namespaces
- STU4B32: Updated Conformance Requirement 2 to address the use of called clauses and consistent local identifiers for included libraries
- STU4B35: Updated Conformance Requirement 3 to mandate use of Patient context for HQMF measures that use QDM (and disallow Unfiltered)

References

[1] *Crossing the Quality Chasm: A New Health System for the 21st Century*. Institute of Medicine, March 2001. http://www.nationalacademies.org/hmd/Reports/2001/Crossing-the-Quality-Chasm-A-New-Health-System-for-the-21st-Century.aspx

[2] Quality *Data Model, Version 5.5*. Centers of Medicare & Medicaid Services; Office of the National Coordinator for Health Information Technology, 2019. https://ecqi.healthit.gov/qdm

[3] *Clinical Quality Language (CQL), STU R1.4*. HL7, July 2019. http://www.hl7.org/implement/standards/product_brief.cfm?product_id=400

[4] HL7, *Representation of the Health Quality Measures Format (HQMF) Release 1*. HL7, June 2017. http://www.hl7.org/implement/standards/product_brief.cfm?product_id=97

[5] *HL7 Implementation Guide for CDA Release 2: Quality Reporting Document Architecture – Category I STU Release 5 (US Realm)*. HL7, ballot cycle September 2017 http://www.hl7.org/implement/standards/product_brief.cfm?product_id=35.

[6] *Refinement, Constraint and Localization, Release 2*. HL7, September 2015. http://www.hl7.org/v3ballotarchive_temp_52E32C7C-1C23-BA17-0CA99EC07A928F9D/v3ballot/html/infrastructure/conformance/conformance.html

[7] *URN Syntax*. The Internet Engineering Task Force, May 1997. https://www.ietf.org/rfc/rfc2141.txt

[8] *Uniform Resource Names (URN) Namespace Definition Mechanisms*. The Internet Engineering Task Force, October 2002. https://www.ietf.org/rfc/rfc3406.txt

[9] *Value Set Authority Center*. U.S. National Library of Medicine. https://vsac.nlm. nih.gov/

[10] *VSAC SVS API v2*. Value Set Authority Center, May 2015. https://www.nlm.nih.gov/vsac/support/usingvsac/vsacsvsapiv2.html

[11] *Versioning Numbering Concepts - The Apache Portable Runtime Project*. APR Developers, 2016. https://apr.apache.org/versioning.html