HL7 Version 3 Implementation Guide: Clinical Quality Language (CQL)-based Health Quality Measure Format (HQMF) Release 1, STU 2.1 - US Realm

September 2017

HL7 Standard for Trial Use (STU)

Volume 2 - Using QDM in CQL based Measures

Sponsored by:
Clinical Quality Information Work Group

Publication of this standard for trial use and comment has been approved by Health Level Seven International (HL7). This standard is not an accredited American National Standard. The comment period for trial use of this standard shall end 24 months from the date of publication. Suggestions for revision should be submitted at http://www.hl7.org/dstucomments/index.cfm.
Following this 24 month evaluation period, this standard, revised as necessary, will be submitted to a normative ballot in preparation for approval by ANSI as an American National Standard. Implementations of this trial use standard shall be viable throughout the normative ballot process and for up to six months after publication of the relevant normative standard.

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

Although the specification is based on the 1.1 version of CQL, backwards compatible future versions of CQL can be used as well.

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 STU 2.1 (US Realm), Standard for Trial Use*:

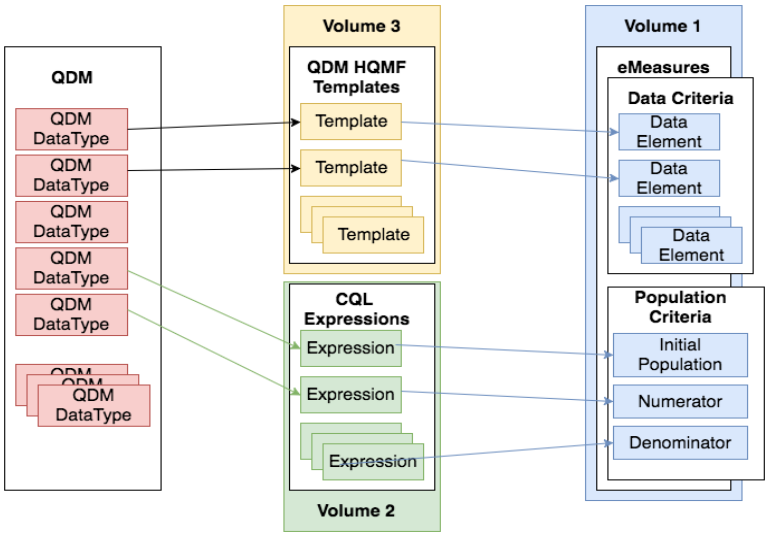

Figure 1: Relationship between QDM, CQL, eMeasure, and the volumes of this IG

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

Volume 2 : Describes how to incorporate version 5.3 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 STU 2 (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 Release 2. 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 v5.3 [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 eMeasure documents.

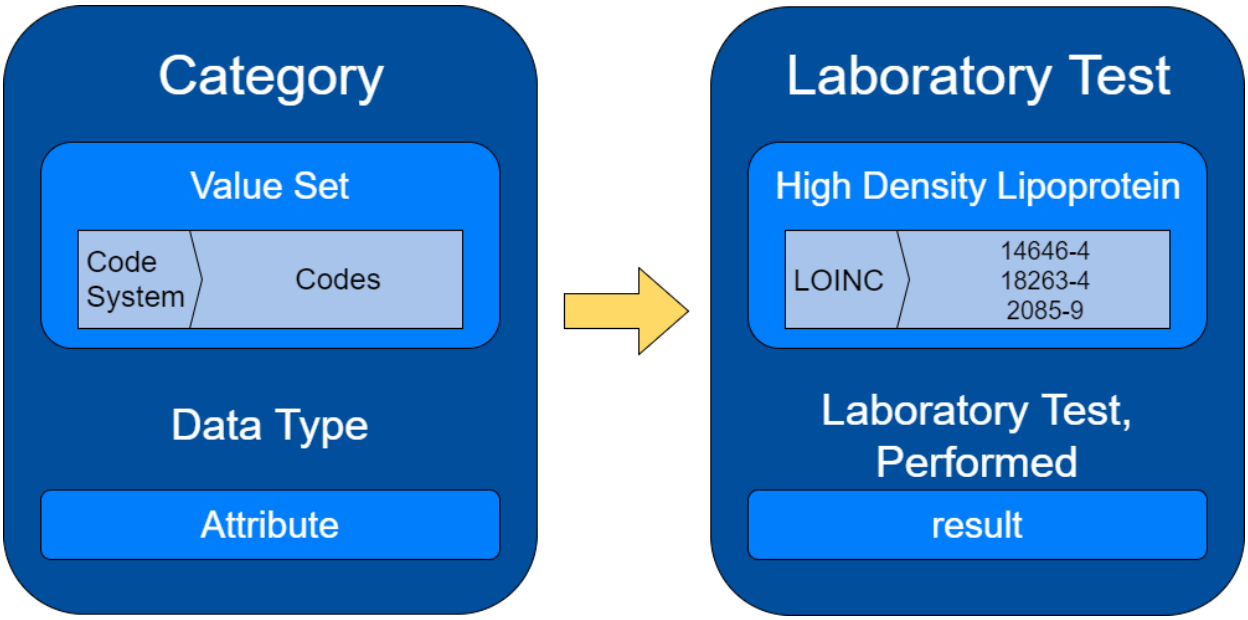

Figure 2: QDM element structure [2]

A QDM element is composed of a category, the data type in which that category is expected to be used, and a value set 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 eMeasures. QRDA also specifies a method for referencing the eMeasure(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 eMeasure 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 eMeasure specification. For example, if an eMeasure 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 5.3. 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 usage 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 usage of its components at all, but only corrects or improves the original 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 func-

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

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

tionality, while patch changes are expected to retain forward and backwards compatibility, and may only be used to fix issues.

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.

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.

2.2 Data Model

CQL can be used with any data model. However, within the context of HQMF any referrenced 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.

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 an indication of the version, separated with a colon.

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 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": '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)": '2.16.840.1.113883.3.464.1003.102.12.1011.1'
15 valueset "Acute Pharyngitis (2)": '2.16.840.1.113883.3.464.1003.102.12.1011.2'
```

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.

```
valueset "Encounter Inpatient SNOMEDCT Value Set":
   '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 are discussed in Volume 1 Chapter 3 of this IG.

2.5 Codes

When direct referenced 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 Referenced Codes):
When direct referenced 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 referenced 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 referenced 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 are 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 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`

2.8 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

2.8.1 Negation in QDM

Negation is resolved in QDM and corresponding CQL libraries utilizing QDM through a `"Not"` modifier in the data type name as follows:

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

The above code demonstrates the use of the `"Not"` modifier to retrieve all EHR entries indication 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.

QDM Data Type Names	Negated Name
"Adverse Event"	N/A
"Allergy/Intolerance"	N/A
"Assessment, Recommended"	"Assessment, Not Recommended"
"Assessment, Performed"	"Assessment, Not Performed"
"Care Goal"	N/A
"Communication: From Patient To Provider"	"Communication: From Patient To Provider, Not Done"
"Communication: From Provider To Patient"	"Communication: From Provider To Patient, Not Done"
"Communication: From Provider To Provider"	"Communication: From Provider To Provider, Not Done"

"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, Active"	N/A
"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 Characteristic"	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.

2.9 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[§]

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
```

2.10 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" Encounter
      where Encounter.relevantPeriod during "Measurement Period"

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

3 Reporting Results

The results of eMeasures 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.

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

Measure reports generated from the eMeasures 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.

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 5.3 Annotated*. Centers of Medicare & Medicaid Services; Office of the National Coordinator for Health Information Technology, 2017. https://ecqi.healthit.gov/cql

[3] *Clinical Quality Language (CQL), DSTU R1.1*. HL7, April 2015. http://www.hl7.org/implement/standards/product_brief.cfm?product_id=400

[4] HL7, *Representation of the Health Quality Measures Format (eMeasure) 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.

[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