



**HL7 Arden Syntax Implementation Guide:**  
**Arden Syntax on FHIR, Edition 1**

**Release 1**

September 2024

**HL7 STU Ballot**

**Sponsored by:**  
**HL7 Arden Syntax Work Group**

Copyright © 2024 Health Level Seven International ® ALL RIGHTS RESERVED. The reproduction of this material in any form is strictly forbidden without the written permission of the publisher. HL7 and Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. Pat & TM Off.

Use of this material is governed by HL7's [IP Compliance Policy](#).

# **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 <a href="https://www.ama-assn.org/practice-management/cpt-licensing">https://www.ama-assn.org/practice-management/cpt-licensing</a>
SNOMED CT®	SNOMED CT® International; <a href="http://www.snomed.org/snomed-ct/get-snomed-ct">http://www.snomed.org/snomed-ct/get-snomed-ct</a> or <a href="mailto:info@ihtsdo.org">info@ihtsdo.org</a>
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 <a href="http://www.nucc.org">www.nucc.org</a> . AMA licensing contact: 312-464-5022 (AMA IP services)

## Obtaining a CPT Sublicense from HL7

Contact [hg@hl7.org](mailto:hg@hl7.org) about how to obtain a sublicense from HL7 for non-production use of CPT for (i) the development and publication of value sets, profiles, and other artifacts as part of the HL7 Implementation Guides, (ii) as part of defined VSAC value sets, and (iii) to support HL7's terminology services within the Territory.

## Flow Down Clauses for CPT Sublicense from HL7

CPT content is copyrighted by the American Medical Association and CPT is a registered trademark of the AMA.

HL7, as a party to a license agreement with the AMA, is authorized to grant user a limited, non-exclusive, non-transferable, non-sublicensable license for user to use CPT content for (i) the development and publication of value sets, profiles, and other artifacts as part of the HL7 Implementation Guides, (ii) as part of defined VSAC value sets, and (iii) to support HL7's terminology services within the Territory, each of which shall be considered a non-production use. The sublicense granted hereunder shall automatically terminate upon termination of the agreement between HL7 and AMA, unless prior written consent of AMA is obtained.

The provision of updated CPT content is dependent on a continuing contractual relationship between HL7 and the AMA.

User acknowledge a separate license agreement shall be required, and shall govern any proposed use, including any distribution of CPT content for any other purposes not expressly permitted under this Agreement, and the terms of such agreement will govern such use (e.g., a separate license agreement shall govern production use and commercial purposes). AMA reserves the right to accept or reject licenses based on AMA's evaluation of the proposed use of the CPT content.

User acknowledge that User's development and commercialization of CPT-informed works developed with reference to Licensed Products may only be implemented in the Territory.

User is prohibited from making CPT content publicly available, creating derivative works (including translating), transferring, selling, leasing, licensing, or otherwise making available to any unauthorized party the CPT content, or a copy or portion of CPT content to any unauthorized party, including a subsidiary, affiliate, or other legal entity, however designated, for any purpose whatsoever except as expressly permitted under a separate agreement.

User expressly acknowledges and agrees to the extent permitted by applicable law, use of CPT content is at User's sole risk and CPT content is provided "as is" without warranty of any kind. The AMA does not directly or indirectly practice medicine or dispense medical services. Fee schedules, relative value units, conversion factors and/or related components are not assigned by the AMA, are not part of CPT, and the AMA is not recommending their use. CPT content herein does not replace the AMA's Current Procedural Terminology book or other appropriate coding authority. The coding information contained in CPT content should be used only as a guide.

U.S. Government End Users. CPT is commercial technical data, which was developed exclusively at private expense by the American Medical Association (AMA), 330 North Wabash Avenue, Chicago, Illinois 60611. This agreement does not grant the Federal Government a direct license to use CPT based on FAR 52.227-14 (Data Rights - General) and DFARS 252.227-7015 (Technical Data - Commercial Items).

User expressly consents to the release of its name to the AMA.

1.	Purpose .....	4
2.	Notes and Disclaimer .....	6
3.	Introduction .....	7
3.1	Short History of Arden Syntax and Arden Syntax on FHIR .....	7
3.2	Arden Syntax Compared to CQL .....	9
4.	Arden Syntax on FHIR .....	10
4.1	Core operations .....	10
4.1.1	Resource Type – Observation.....	10
4.1.2	Resource Type – Encounter.....	10
4.1.3	Resource Type – Condition.....	11
4.1.4	Resource Type – Patient.....	12
4.1.5	Simply Read an Observation from a Repository .....	13
4.1.6	Read the Most Recent Observation Resource from a Repository.....	14
4.1.7	Map a Read Observation Resource to a Custom Arden Syntax Object.....	15
4.1.8	Use a Filter When Reading an Observation Resource .....	16
4.1.9	Multiple Filters of Different Fields When Reading a Resource (AND operator).....	18
4.1.10	Complex Filter Field – ValueSet Object Filter Value .....	19
4.1.11	Complex Filter Field: Link to Another Resource .....	21
5.	Annex 1: FHIR Resource Object Declarations .....	21
5.1	Implementing a FHIR Resource as Pre-defined Arden Syntax Object.....	22
5.1.1	Mapping FHIR to Arden Syntax Datatypes .....	23
5.2	Arden Syntax Object Declarations of FHIR Resources .....	24
5.2.1	Example: Patient Resource Object Declaration .....	24
6.	Outlook.....	31
7.	References.....	32
8.	F.A.Q.....	33

# 1. Purpose

The Arden Syntax for Medical Logic Systems is a structured, executable formalism for the explicit representation of scientific, clinical, administrative, and other knowledge used in clinical decision support (CDS) systems. As such, it functions as a programming language for such systems, allowing knowledge authors and clinical domain experts to implement knowledge-based interventions such as alerts, informational notices, and reminders, order sets, turnaround forms and the like in order to realize their treatment quality improvement, clinical, administrative, and public health objectives. Expressed in a way that resembles English-language syntax, the Arden Syntax also facilitates validation of knowledge bases by domain experts. In light of this utility, a number of vendors of clinical information systems and decision support systems have incorporated Arden Syntax into their products, leading to its adoption and use at numerous sites worldwide.

Despite the relative ease of use and functionality of the Arden Syntax, both novel and experienced users have questions regarding how best to use the syntax with HL7 FHIR as standard data model. Examples include FHIR-based data objects, accessing FHIR resources with complex filters and incorporating highly structured clinical terminology. Any user consulting this implementation guide will find extensive examples on the use of FHIR as data source in medical logic modules.

The purpose of this implementation guide is to help answer these questions by providing, in addition to a summary of FHIR in Arden Syntax itself, ideas and examples regarding how Arden Syntax may be used in these different situations. This guide is not intended to be exhaustive in this regard, but it is meant to provide guidance on how to use the Arden Syntax to solve real-world challenges related to the implementation of CDS. Further, while the summary of the Arden Syntax features presented herein provides the important highlights of this key standard, readers are directed to the actual Arden Syntax specifications for a complete definition of the language.

This implementation guide was composed when Arden Syntax v3.0 was the latest approved version of the standard and further features were under development. While the examples and ideas featured here include elements of the syntax that are new to these versions and may not be present in earlier versions, substantial parts of the implementation guide also leverage the backward compatibility of Arden Syntax, allowing users of an earlier version also to make use of this implementation guide.

Finally, the reader should be aware that, while the authors have been diligent in providing useful, accurate content derived from real-world solutions already implemented in CDS systems, no guarantee of accuracy or effectiveness is made regarding the examples and other information presented in this implementation guide. After all, this guides aim is to provide guidance, so examples might have been shortened or over-simplified in cases where specific functions of Arden Syntax needed more highlighting. Any user or implementer of Arden Syntax assumes all liability regarding the use of any material contained in this guide.

The authors of this implementation guide hope that you find it a useful addition to other Health Level Seven publications related to clinical CDS in ways that allow you to make best use of the powerful and rich standard for representing CDS knowledge that is the Arden Syntax.

## 2. Notes and Disclaimer

This implementation guide is not normative. Knowledge of the Arden Syntax standard and Arden Syntax core implementation guide is a prerequisite for reading this document. For a detailed description of this standard, we would like to refer to the Arden Syntax specification available on the Health Level Seven (HL7) International's website.

## 3. Introduction

### 3.1 Short History of Arden Syntax and Arden Syntax on FHIR

Computer-based clinical decision support (CDS) has been shown to improve the quality of health care treatment and the performance of health care professionals. CDS involves delivering knowledge to decision-makers in clinical settings in order to improve the quality of decisions and the outcomes to which they lead. CDS sometimes is described in terms of the Five Rights: Delivering the right knowledge to the right person at the right time in the workflow in the right format via the right channel. Some channels are bidirectional to allow user response to be collected and documented.

In order to provide computer-based CDS, the knowledge to be delivered must be represented in digital form. In this light, CDS can be divided into two broad classes: Services that facilitate delivery of knowledge and explicit, computable representations of the knowledge itself that can be shared via transfer and reuse. In the case of a knowledge delivery service, standards facilitate communication between electronic health record (EHR) systems and other clinical software and knowledge sources, allowing connection of systems and sources from multiple vendors without having to negotiate and implement ad hoc methods for each connection. In the case of explicit knowledge encoding, standards facilitate sharing of knowledge by minimizing the changes necessary for the knowledge to be executed or used in different information systems.

The HL7 Info button standard is an example of a knowledge delivery service standard. It facilitates queries from users of EHR systems in the context of particular care activities and particular patients, providing knowledge from knowledge sources that is pertinent to these contexts. By contrast, examples of explicit knowledge encoding include the HL7 Clinical Quality Language (CQL) and Arden Syntax standards.

A knowledge representation formalism constitutes one part of an overall CDS system. Units of knowledge encoded using the formalism are stored in the knowledge base (KB), independent of but linked to the inference engine or event monitor that executes units of the KB in combination with patient data to produce tailored, context-specific knowledge-based interventions that then can be delivered to the appropriate recipient such as a clinician, patient, or administrator.

A prominent example of a knowledge formalism for encoding units of knowledge in the KB is the Arden Syntax for Medical Logic Systems. This is a computable language for encoding medical knowledge. It was previously adopted as a standard by the American Society for Testing and Materials (ASTM) as document E 1460, under subcommittee E31.15 Health Knowledge Representation. Adopted in 1992, it became Arden Syntax version 1.0.

Arden Syntax uses medical logic modules (MLMs) as units of knowledge representation. Each of these MLMs contains sufficient knowledge to make a single medical decision. MLMs have been used to generate clinical reminders and alerts, interpretations, diagnoses and therapeutic advice, screening for clinical research, quality assurance functions, and administrative support. Using a



computer program called an event monitor, MLMs run automatically, generating advice where and when it is needed.

Beginning in 1998, sponsorship of this standard was moved to HL7 International. Maintenance and further development of the standard is now overseen by the HL7's Arden Syntax Work Group. The Arden Syntax version 2.0 was formally adopted by HL7 and the American National Standards Institute (ANSI) in August 1999. Since then the standard has evolved, including the addition of new features and functionalities responding to the needs of users and vendors. This process eventually led to version 2.10, which was adopted by HL7 and certified by ANSI in May 2014.

Despite this robust development, version 2.10 still lacked a standard data model. Recognizing the heterogeneity of clinical databases that would be referenced by MLMs in order to obtain data for processing logic and making decisions, the original developers of the Arden Syntax elected not to include a standard data model. Instead, a construct was provided – the curly braces, named for the characters used to delimit local data mappings inside an MLM – to demarcate references to local data so that, when MLMs were shared with another organization or locality, mappings specific to the recipient could replace the original mappings. However, this increased the complexity of knowledge sharing, as the mappings would have to be rewritten every time a MLM was shared. This challenge, while not unique to the Arden Syntax as it affects all representations of computable clinical knowledge, nevertheless became known as the “curly braces problem” because of the way it was addressed in the Arden Syntax.

HL7 International collectively began to recognize the importance of a standard data model for representing data in its main messaging standard. The initial effort in this regard was the Reference Information Model (RIM), a complex, comprehensive model meant to facilitate standard representation of almost any data in the domain of health care. Ultimately, dissatisfaction with its complexity and relative lack of flexibility led to the effective abandonment of the RIM and to the development of the Fast Healthcare Interoperability Resources (FHIR). In FHIR, a data object - such as a person, an observation, or an order along with its dependent attributes – is represented as a “resource”. Each resource defines the component data elements, constraints on data, and data relationships that together make up an exchangeable patient record. Resources may be aggregated for specific purposes using profiles.

The FHIR data model has come to permeate most other HL7 standards and use cases. Consequently, in order to address the curly braces problem in the Arden Syntax, HL7 chose to incorporate FHIR as the standard data model inside MLMs. By doing this, MLMs can be encoded using standard references to data that need not be rewritten when MLMs are shared, thereby improving knowledge sharing and facilitating dissemination of computable knowledge. Incorporating FHIR as the standard data model of Arden Syntax led to the creation of version 3.0 of the Arden Syntax, which was approved as an ANSI standard for trial use in 2023.

HL7 created a series of implementation guides describing how to use the Arden Syntax generally to implement CDS solutions. While in some ways a successor to this series, this particular

implementation guide focuses specifically on how to use FHIR to create references to data within Arden Syntax MLMs.

### 3.2 Arden Syntax Compared to CQL

Like the Arden Syntax, the Clinical Quality Language (CQL) is an HL7 standard for computable, shareable knowledge representation in CDS systems. Both use FHIR as a standard data model. While this may seem redundant functionality, HL7 maintains two distinct CDS knowledge representation standards because each facilitates distinct use cases. CQL is an expression language only, representing just the logic and data for making decisions in a CDS system, with its formative use cases focused on representation of electronic clinical quality measures (eCQMs) in a standard way. The intent is that CQL expressions would be embedded in other representations or contexts in order to implement a complete CDS solution. By contrast, an Arden Syntax MLM, beyond logical expressions and data mappings in its Knowledge category, also contains additional information in its Maintenance, Library and Resources categories that, together, represent a complete CDS representation that can make and document one or more decisions.

## 4. Arden Syntax on FHIR

### 4.1 Core operations

This section is aimed at providing simple and deconstructed examples to all the new core functionalities that have been introduced for handling FHIR resources in Arden Syntax.

Generally, FHIR resources are expected to be handled with in the data slot, as part of READ AS operations. READ AS operations build the core of obtaining a FHIR resource from a remote repository, which is then mapped to an Arden Syntax Object. That Arden Syntax Object can be handled downstream inside the MLM like any other Arden Syntax Object.

#### 4.1.1 Resource Type – Observation

Building a plain new Arden Syntax Object from the pre-defined Observation type, which is mandatorily provided by the Arden Syntax compiler/interpreter and must not be defined inside an MLM. This mirrors the [FHIR Observation](#) resource in structure and data types, however there are no enforced cardinalities. Remember to adhere to cardinalities if that object will be sent to a FHIR repository.

#### MLM

```
data:
    LET my_new_observation BE NEW Observation
      WITH [subject := NEW Reference WITH [type := "Patient",
                                           identifier := "PAT_ID_001"],
           effectiveDateTime := 2024-01-15 + 14 hours + 59 minutes,
           valueBoolean := TRUE];

;;
logic: CONCLUDE true; ;;
action: RETURN my_new_observation; ;;
```

#### Return object (result in JSON)

```
{
  "resourceType": "Observation",
  "subject": {
    "type": "Patient",
    "identifier": {
      "value": "PAT_ID_001" }},
  "effectiveDateTime": "2024-01-15T14:59:00",
  "valueBoolean": true
}
```

#### 4.1.2 Resource Type – Encounter

Building a plain new Arden Syntax Object from the pre-defined Encounter type, which is mandatorily provided by the Arden Syntax compiler/interpreter and must not be defined inside an MLM. This mirrors the [FHIR Encounter](#) resource in structure and data types, however, there are no enforced cardinalities. Remember to adhere to cardinalities if that object will be sent to a FHIR repository.

## MLM

```
data:
    LET encounter_start BE 2024-01-16T18:00:00;
    LET my_new_encounter BE NEW Encounter
        WITH [subject := NEW Reference WITH [reference := "patient/ PAT_ID_001"],
            status := "finished",
            period := NEW Period WITH [start := encounter_start,
                end := encounter_start + 49 minutes ],
            type := NEW CodeableConcept WITH [
                coding := ( NEW Coding WITH [
                    system := "http://snomed.info/sct",
                    code := "17621005",
                    display := "normal"],
                NEW Coding WITH [
                    system := "http://snomed.info/sct",
                    code := "270427003",
                    display := "Patient-initiated encounter"])]];
        ;;
logic: CONCLUDE true; ;;
action: RETURN my_new_encounter; ;;
```

## Return object (result in JSON)

```
{
  "resourceType": "Encounter",
  "status": "finished",
  "subject": {
    "reference": "patient/ PAT_ID_001",
    "period": {
      "start": 2024-01-16T18:00:00,
      "end": 2024-01-16T18:49:00},
    "type": [
      {
        "coding": [
          {
            "system": "http://snomed.info/sct",
            "code": "17621005",
            "display": "normal"
          },
          {
            "system": "http://snomed.info/sct",
            "code": "270427003",
            "display": "Patient-initiated encounter"
          }
        ]
      }
    ]
  }
}
```

### 4.1.3 Resource Type – Condition

Building a plain new Arden Syntax Object from the pre-defined Condition type, which is mandatorily provided by the Arden Syntax compiler/interpreter and must not be defined inside an MLM. This mirrors the [FHIR Condition](#) resource in structure and data types, however there are no enforced cardinalities. Remember to adhere to cardinalities if that object will be sent to a FHIR repository.

## MLM

```
data:
  LET my_new_condition BE NEW Condition
    WITH [clinicalStatus := NEW CodeableConcept WITH [coding := NEW Coding WITH [
      system := "http://terminology.hl7.org/CodeSystem/condition-clinical",
      code := "active"]],
      verificationStatus := NEW CodeableConcept WITH [coding := NEW Coding WITH [
      system := "http://terminology.hl7.org/CodeSystem/condition-ver-status",
      code := "confirmed"]],
      code := NEW CodeableConcept WITH [coding := NEW Coding WITH [
      system := "http://snomed.info/sct",
      code := "10001005",
      display := "Bacterial sepsis"]],
      onsetDateTime := 2024-01-19,
      recordedDate := 2024-01-20,
      asserter := NEW Reference WITH [reference := "Practitioner/Clinician007"]];
  ;;
logic: CONCLUDE true; ;;
action: RETURN my_new_condition; ;;
```

## Return object (result in JSON)

```
{
  "resourceType": "Condition",
  "clinicalStatus": {
    "coding": [
      {
        "system": "http://terminology.hl7.org/CodeSystem/condition-clinical",
        "code": "active"
      }
    ]
  },
  "verificationStatus": {
    "coding": [
      {
        "system": "http://terminology.hl7.org/CodeSystem/condition-ver-status",
        "code": "confirmed"
      }
    ]
  },
  "code": {
    "coding": [
      {
        "system": "http://snomed.info/sct",
        "code": "10001005",
        "display": "Bacterial Sepsis"
      }
    ]
  },
  "onsetDateTime": "2024-01-19",
  "recordedDateTime": "2024-01-20",
  "asserter": {
    "reference": "Practitioner/Clinician007"
  }
}
```

### 4.1.4 Resource Type – Patient

Building a plain new Arden Syntax Object from the pre-defined Patient type, which is mandatorily provided by the Arden compiler/interpreter and must not be defined inside an MLM. This mirrors the [FHIR Patient](#) resource in structure and data types, however there are no enforced cardinalities. Remember to adhere to cardinalities if that object will be sent to a FHIR repository.

**MLM**

```

data:
    LET my_new_patient BE NEW Patient
        WITH [identifier := "PAT_007",
            active := true,
            name := NEW HumanName WITH [
                use := "official",
                family := "Doe",
                given := ("John", "Jim")]];
    ;;
logic: CONCLUDE true; ;;
action: RETURN my_new_patient; ;;

```

**Return object (result in JSON)**

```

{
  "resourceType": "Patient",
  "identifier": "PAT_007",
  "active": true,
  "name": [
    {
      "use": "official",
      "family": "Doe",
      "given": [ "John", "Jim" ]
    }
  ]
}

```

**4.1.5 Simply Read an Observation from a Repository**

Use the READ AS <fhir-object-type> statement to access a repository and retrieve a list of resources of a specific object type.

This READ AS statement takes different parameters from the classic one. Originally the classic READ AS statement takes a custom object type and performs implementer-defined logic (curly braces { }) to retrieve the data, mapped to the custom object.

If you want to perform a FHIR resource call you must use one of the pre-defined FHIR object types (from the core Arden standard the resources Patient, Observation, Encounter and Condition are supported, any implementer can of course expand that list).

The Arden Syntax engine will target a FHIR repository in a way determined by the implementer. For example, some members of the Arden Syntax Workgroup define independent configurations outside of an MLM to be used for any FHIR read requests, e.g.:

```

/*E.g. if an observation with id X is located at example.arden.org/fhir/R4B/observation/X */
/*     Therefore, the base url of that FHIR repository is example.arden.org/fhir/R4B */

```

The following request will fetch all Observation type resources from the FHIR server. It will also populate all the data fields in each resource. Such a request is not advisable in realistic situations, because of the mass of data returned. Limit the resources fetched from the FHIR server by using the WHERE clause filters. Limit the populated object fields by using custom Arden object mapping. Both methods are explained further down this guide.

**MLM**

```
data: LET my_read_observations BE READ AS Observation;
;;
logic: CONCLUDE true; ;;
action: RETURN my_read_observations; ;;
```

### Return object (result in JSON)

```
[
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/OBS_001",
      "value": "OBS_001" },
    "subject": {
      "type": "Patient",
      "reference": "patient/PAT_ID_001" },
    "effectiveDateTime": "2024-01-15T14:59:00",
    "valueQuantity": {
      "value": 120.7,
      "unit": "mmHg" }
  },
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/OBS_002",
      "value": "OBS_002" },
    "subject": {
      "type": "Patient",
      "reference": "patient/PAT_ID_001" },
    "effectiveDateTime": "2023-06-30T00:00:00",
    "valueTime": "14:00:01"
  },
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/600744391",
      "value": "600744391" },
    "subject": {
      "type": "Practitioner",
      "reference": "practitioner/AnotherPerson" },
    "valueString": "I observed something"
  }
]
```

### 4.1.6 Read the Most Recent Observation Resource from a Repository

Similar to reading a simple FHIR resource, any READ AS request can be limited to read only the most recent resource by using the LATEST keyword. In Arden the recency is always referred to the clinically relevant time (primary time). Consult the Arden Syntax standard document chapters 12.2 and 12.4 for more details on primary times in FHIR resource objects.

With such a limited request at most one FHIR resource can be returned (the most recent one), or NULL if no resource matches the request filter.

### MLM

```
data: LET my_read_observation BE READ AS LATEST Observation;
;;
logic: CONCLUDE true; ;;
action: RETURN my_read_observation; ;;
```

**Return object (result in JSON)**

```
{
  "resourceType": "Observation",
  "identifier": {
    "uri": "Observation/OBS_001",
    "value": "OBS_001" },
  "subject": {
    "type": "Patient",
    "reference": "patient/PAT_ID_001" },
  "effectiveDateTime": "2024-01-15T14:59:00",
  "valueQuantity": {
    "value": 120.7,
    "unit": "mmHg" }
}
```

**4.1.7 Map a Read Observation Resource to a Custom Arden Syntax Object**

At times, FHIR resources contain more data fields than needed in a specific MLM. Some of the field names themselves do not reflect the clinically relevant content computed by the MLM. For this case custom Arden object mapping can be applied to a READ AS statement and the returned FHIR resource objects. The mapping is applied to each FHIR resource fetched, should there be multiple.

When creating a mapping the stored variable will be list of or single Arden Syntax Object, whose object definition is created on-the-fly and not reusable. But only the mapped FHIR resource field values are stored in their corresponding Arden object type fields, thereby it does not matter if the value is an object, list or simple type. The fields accessed must be existing field names in pre-defined Arden Syntax Object types, but can be chained. The custom fields can be freely named and represent the fields of newly created object, whereby nesting is permitted.

**MLM**

```
data: LET my_read_observation BE READ AS LATEST Observation;
      LET custom_mapped_observation[
        id.validated_type, id.value,
        observed_quantity,
        primary_time,
        all_exam_results,
        first_concrete_exam ]
      BE READ AS LATEST Observation[
        identifier.use, identifier.value,
        valueQuantity,
        effectiveDateTime,
        code.coding.code,
        code.coding[0] ];

      ;;
logic: CONCLUDE true; ;;
action: RETURN my_read_observation, custom_mapped_observation; ;;
```

**Return object (result in JSON)**

```
[
  {
    "resourceType": "Observation",
    "identifier": {
```



```

    "uri": "Observation/OBS_001",
    "value": "OBS_001" },
  "subject": {
    "type": "Patient",
    "reference": "patient/PAT_ID_001" },
  "effectiveDateTime": "2024-01-15T14:59:00",
  "valueQuantity": {
    "value": 120.7,
    "unit": "mmHg" },
  "code": {
    "coding": [
      { "code": "5021-1", "system": "http://loinc.org" },
      { "code": "A00.0", "system": "http://hl7.org/fhir/ValueSet/icd-10" },
      { "display": "Typhoid fever", "system": "http://hl7.org/fhir/ValueSet/icd-10" }
    ]
  },
},
{
  "id": {
    "validated_type": null,
    "value": "OBS_001" },
  "observed_quantity": {
    "value": 120.7,
    "unit": "mmHg" },
  "primary_time": "2024-01-15T14:59:00",
  "all_exam_results": ["5021-1", "A00.0", null],
  "first_concrete_exam": {
    "code": "5021-1",
    "system": "http://loinc.org" }
}
]

```

#### 4.1.8 Use a Filter When Reading an Observation Resource

When requesting FHIR resources on a larger FHIR repository the fetched objects will likely contain results not relevant to your MLM. Instead of fetching the entire list of results consider putting simple field filters in the READ statements WHERE clause. The Arden Syntax compiler implementer can then leverage the better performance of the [FHIR search api](#) to decrease computation speed and fetch results that would not need extensive filtering inside the MLM, leaving more room for clinically relevant logic.

The WHERE clause references specific fields and allows for equality comparison (at the moment, no other operators are supported). Filter parameter fields are referenced as a field in the final variable (left side of the assignment operator). Parameter fields are referenced by their mapped variable field name.

#### MLM

```

data: LET my_read_observations BE READ AS Observation;
      // GET observation?search=identifier:OBS_001
      LET filtered_observations BE READ AS Observation
        WHERE it.identifier.value = "OBS_001";
      LET filtered_mapped_observations[administrative_id, exam_result]
        BE READ AS Observation[identifier, valueTime]
        WHERE it.exam_result = 14:00:01;
      ;;

```

```
logic: CONCLUDE true; ;;
action: RETURN filtered_mapped_observations, filtered_observations, my_read_observations;
      ;;
```

## Return object (result in JSON)

```
[
  {
    "administrative_id": {
      "uri": "Observation/OBS_002",
      "value": "OBS_002" },
    "exam_result": "14:00:01"
  },
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/OBS_001",
      "value": "OBS_001" },
    "subject": {
      "type": "Patient",
      "reference": "patient/PAT_ID_001" },
    "effectiveDateTime": "2024-01-15T14:59:00",
    "valueQuantity": {
      "value": 120.7,
      "unit": "mmHg" }
  },
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/OBS_001",
      "value": "OBS_001" },
    "subject": {
      "type": "Patient",
      "reference": "patient/PAT_ID_001" },
    "effectiveDateTime": "2024-01-15T14:59:00",
    "valueQuantity": {
      "value": 120.7,
      "unit": "mmHg" }
  },
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/OBS_002",
      "value": "OBS_002" },
    "subject": {
      "type": "Patient",
      "reference": "patient/PAT_ID_001" },
    "effectiveDateTime": "2023-06-30T00:00:00",
    "valueTime": "14:00:01"
  },
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/600744391",
      "value": "600744391" },
    "subject": {
      "type": "Practitioner",
      "reference": "practitioner/AnotherPerson" },
    "valueString": "I observed something"
  }
]
```

### 4.1.9 Multiple Filters of Different Fields When Reading a Resource (AND operator)

In an ordinary WHERE clause filter parameters can be combined using AND aggregation. If you stick to the general rules laid out here than those filter parameter can be translated to a FHIR repository filter. Thereby your READ AS ... WHERE ... statement can work with optimal efficiency, including only the clinically relevant FHIR resources at the repository, saving time and computational power at the side of your MLM. Be aware that lacking support for those operations at the side of the FHIR repository can result in the filters being applied at the MLM side after fetching a generic result. You will still retrieve the result with all filters applied, but the performance can decline.

The AND operator is fully supported by the FHIR search interface and therefore any WHERE clause containing just AND operators for searchable FHIR-object fields is supported. Be aware that Arden does not safeguard you from logical errors which render the query, although syntactically correct, unusable (e.g., filter by 'status = "final" AND status = "preliminary"' - this can only return zero results as a FHIR resource cannot have two different status field values).

#### MLM

```
data: LET my_read_observations BE READ AS Observation;
      LET filtered_observations BE READ AS Observation
        WHERE it.subject.type = "Patient"
          AND it.status = "final"
          AND it.code.coding.code = "10155-0"
          AND it.code.coding.system = "http://loinc.org";
      ;;
logic: CONCLUDE true; ;;
action: RETURN filtered_observations, my_read_observations;
      ;;
```

#### Return object (result in JSON)

```
[
  {
    "resourceType": "Observation",
    "status": "final"
    "identifier": {
      "uri": "Observation/OBS_001",
      "value": "OBS_001" },
    "subject": {
      "type": "Patient",
      "reference": "patient/PAT_ID_001" },
    "effectiveDateTime": "2024-01-15T14:59:00",
    "code": { "coding": [{
      "code": "10155-0",
      "system": "http://loinc.org",
      "display": "History of allergies, reported"
    } ] },
    "valueString": "Animal Fur; Shellfish"
  },
  {
    "resourceType": "Observation",
    "status": "final"
    "identifier": {
      "uri": "Observation/OBS_001",
```

```

    "value": "OBS_001" },
  "subject": {
    "type": "Patient",
    "reference": "patient/PAT_ID_001" },
  "effectiveDateTime": "2024-01-15T14:59:00",
  "code": { "coding": [{
    "code": "10155-0",
    "system": "http://loinc.org",
    "display": "History of allergies, reported"
  }] },
  "valueString": "Animal Fur; Shellfish"
},
{
  "resourceType": "Observation",
  "status": "final"
  "identifier": {
    "uri": "Observation/OBS_002",
    "value": "OBS_002" },
  "subject": {
    "type": "Patient",
    "reference": "patient/PAT_ID_001" },
  "effectiveDateTime": "2023-06-30T00:00:00",
  "code": { "coding": [{
    "code": "10163-4",
    "system": "http://loinc.org",
    "display": "History of pregnancies"
  }] },
  "valueString": "Jan. 2020 - Sept. 2020 (birth): Twins, male"
},
{
  "resourceType": "Observation",
  "status": "preliminary"
  "identifier": {
    "uri": "Observation/OBS_002",
    "value": "OBS_003" },
  "subject": {
    "type": "Patient",
    "reference": "patient/PAT_ID_949" },
  "effectiveDateTime": "2023-06-31T00:00:00",
  "valueTime": "14:00:01"
},
{
  "resourceType": "Observation",
  "status": "final"
  "identifier": {
    "uri": "Observation/600744391",
    "value": "600744391" },
  "subject": {
    "type": "Practitioner",
    "reference": "practitioner/AnotherPerson" },
  "valueString": "I observed something"
}
]

```

#### 4.1.10 Complex Filter Field – ValueSet Object Filter Value

Most WHERE clause filters can be achieved by a simple Arden Syntax variable type – String, Number, Date, TimeOfDay, and so on. However, one clinically relevant FHIR search operator exists which is considered in Arden through a specialized comparator. A Valueset, in contrast to

simple code-system fields, can be added to a FHIR query through the use of a pre-defined ValueSet object type in an equality comparator with a Coding type FHIR object field (see example below). A query for a Valueset allows to include all FHIR resources in the result whose code-system are part of the Valueset.

When it comes to using the Valueset filter, ensure the ValueSet object to contain the base URL and expandable code. Following RESTful principles those URL and code shall be constructed to a full URL, which when subjected to a GET request returns the [FHIR ValueSet](http://fhir.loinc.org/ValueSet/LL5107-9) of this filter. e.g., system `http://fhir.loinc.org/ValueSet` and code `LL5107-9` will combine to the URL <http://fhir.loinc.org/ValueSet/LL5107-9>.

This adaptable, URL based syntax allows you to quickly access any existing web service which exposes a FHIR compliant Valueset. You can even use institution-specific Valuesets or create your own when required by the MLMs use-case.

## MLM

```
data: LET valueset_filter BE NEW ValueSet WITH [system := http://fhir.loinc.org/ValueSet,
                                              code := "LL5107-9" ];

      LET filtered_observations BE READ AS Observation
      WHERE it.code.coding IS IN valueset_filter;

;;
logic: CONCLUDE true; ;;
action: RETURN filtered_observations;
;;
```

## Return object (result in JSON)

```
[
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/OBS_002",
      "value": "OBS_002" },
    "code": {
      "coding": [{
        "code": "LA29437-3",
        "system": "http://loinc.org",
        "display": "Bacillus anthracis"
      } ] },
  },
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/OBS_002",
      "value": "OBS_002" },
    "code": {
      "coding": [{
        "code": "LA29440-7",
        "system": "http://loinc.org",
        "display": "Bacillus mycoides"
      } ] },
  },
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/OBS_002",
      "value": "OBS_002" },
    "code": {
      "coding": [{
```

```

    "code": "LA29441-5",
    "system": "http://loinc.org",
    "display": "Bacillus pumilus"
  ]] }
]

```

#### 4.1.11 Complex Filter Field: Link to Another Resource

When a link to another resource is used in a FHIR filter it shall comply to the standard set forth by the [FHIR IG search parameter reference](#). For an Arden Syntax WHERE clause field, a link filter shall use only the EQUALITY operator and contain at least the [logical id](#) of the resource, preferably the resource name and logical id to resolve which resource is referenced by that link in case multiple resource types are allowed on the same link.

#### MLM

```

data: LET filtered_observations BE READ AS Observation
      WHERE it.subject = "54321";
      // Preferred link filter: Include the resource name
      LET filtered_observations BE READ AS Observation
      WHERE it.subject = "Pateint/54321"
      AND it.encounter = "Encounter/00001";

;;
logic: CONCLUDE true; ;;
action: RETURN filtered_observations;
;;

```

#### Return object (result in JSON)

```

[
  {
    "resourceType": "Observation",
    "identifier": {
      "uri": "Observation/OBS_002",
      "value": "OBS_002" },
    "code": {
      "coding": [{
        "code": "LA29437-3",
        "system": "http://loinc.org",
        "display": "Bacillus anthracis"
      }] }
  ]
]

```

## 5. Annex 1: FHIR Resource Object Declarations

All Arden Syntax Objects used to represent FHIR resources are pre-defined by the implementer. All resources explicitly mentioned by the standard declaration have to be implemented in an Arden Syntax compiler/interpreter in order to achieve compliance. Those Arden Syntax Objects have to conform to the following declaration, whereas extended content (any resources implemented but not covered by the standard) should be well documented by the implementer, re-use objects declared here whenever possible (e.g., CodeableConcept, etc.) and stick to the convention for converting FHIR data types to Arden Syntax data types exemplified here.

## 5.1 Implementing a FHIR Resource as Pre-defined Arden Syntax Object

Pre-defined Arden Syntax-FHIR Objects serve the main purpose of being recognized in the READ AS <> statement and triggering the READ statement to fetch from a FHIR repository. Additionally, the pre-defined Object type reflects the FHIR resource in Arden Syntax, which is necessary for processing FHIR resources inside an MLM and potentially allow the implementer more methods tied to the FHIR resource, even in remaining curly braces statements for the WRITE, INTERFACE or EVENT statements.

While the implementer has the freedom to add more FHIR resources to the Arden Compiler general good practices should be followed. Those are developed to guide implementers through problematic areas and sufficiently define their FHIR resource usage to be included in any MLM:

- Support the whole FHIR resource as it is defined by the HL7 FHIR standard of the targeted version.
  - Arden Syntax allows, using custom mapping, to optimize the naming and structure of the Arden Syntax Object variable finally used in the MLM. In order to preserve exchangeability of MLMs and their READ statements, implementations should not differ from the FHIR standard. Keep in mind that MLM writers will consult the FHIR documentation to help them along implementing their use-cases.
  - Consult the extensive FHIR profiles if you need a FHIR resource modified for your use-case. Also, always remember that your Arden Syntax Object definition should serve every possible MLM, even for use-cases not imaginable at the point of implementation, so allow the MLM the same degree of freedom as the well-evolved FHIR standard does.
- Map simple data types to their closest Arden Syntax relative and define new nested FHIR objects where they are used in the Resources as well
  - Make sure the needed conversion between FHIR and Arden Syntax data types is as small as possible. FHIR is based on a simple set of primitive data types (numbers, dates, times, strings, ...) which have their counterparts in Arden Syntax as well. Before defining the conversion of data types consult the Standard document and Implementation Guide first to find already worked-out examples of type conversion, stick to those and establish a common practice amongst all implementations.
  - A shared strength between Arden Syntax and FHIR is the dynamic approach to Object and Array (in Arden Syntax: List) usage to structure data. Arden Syntax Object declaration can mirror any defined FHIR resource 1:1.
- Publish high-quality documentation alongside your implementation
  - It goes almost without saying, nevertheless the importance of which cannot be underestimated: Quality documentation makes or breaks an Arden Syntax implementation.

### 5.1.1 Mapping FHIR to Arden Syntax Datatypes

Mapping primitive data types from FHIR to Arden Syntax is generally not an Issue, because Arden Syntax is less restrictive than FHIR. Consult the table below for concrete mappings. Note also that Arden Syntax serves different use-cases and therefore defines data types which have no primitive counterpart in FHIR (like day-of-the-week, duration, fuzzy-set, or truth-value).

Proving more difficult is converting back from Arden Syntax to FHIR data types, which is the case when using eg. The WHERE statement in a READ query. Arden does not allow for the restrictions of different data formats. With Number and Time this is not an issue, but various restrictions on String data must be documented by the implementer and checked by the MLM writer, or incorrect formatting of the various String subtypes (uri, base64binary, code, uid, ...) will cause the FHIR repository to reject the request.

We recommend to always test possible FHIR query inputs and for the implementer to provide as much support to the MLM writer as possible. Features like real-time code highlighting or basic syntax validation might include the [documented FHIR primitive type restrictions](#), so syntactical mistakes can be avoided early-on.

Complex FHIR data types and FHIR resources are all mapped to Arden Syntax Objects. The object definition shall be pre-defined for any MLM, so any FHIR resource (such as *Patient* or *Observation*) and FHIR General-Purpose, Special Purpose and Metadata Type (such as *HumanName*, *Reference* or *ContactDetail*) can be created inside an MLM. This is driven by the ability of the user to not only READ but also create NEW FHIR resources whenever needed.

<b>FHIR Primitives</b>	<b>Arden Syntax Primitives</b>	<b>Example (FHIR -&gt; Arden Syntax)</b>
boolean	Boolean	<i>true</i> -> <i>true</i>
integer	Number	<i>-45</i> -> <i>-45.0</i>
decimal	Number	<i>45.54</i> -> <i>45.54</i>
unsignedInt	Number	<i>45</i> -> <i>45.0</i>
positiveInt	Number	<i>45</i> -> <i>45.0</i>
instant	Time	<i>2015-02-07T13:28:17.239+02:00Z</i> -> <i>2015-02-07T13:28:17.239+02:00</i>
date	Time	<i>2015-02-07</i> -> <i>2015-02-07T00:00:00</i>
dateTime	Time	<i>2015-02-07T13:28:17.239</i> -> <i>2015-02-07T13:28:17.239+00:00</i>
time	Time-Of-Day	<i>13:28:17.239</i> -> <i>13:28:17.239</i>
string	String	<i>FHIR &amp; Arden</i>
uri	String	<i>urn:id:33394</i>
url	String	<i>ftp://serv.fhir.arden.co</i>
canonical	String	<i>urn:id:33394</i>
base64Binary	String	<i>QXJkZW4tUw==</i>
code	String	<i>valid</i>



oid	String	<code>urn:oid:1.2.3.4.5 -&gt; urn:oid:1.2.3.4.5</code>
uid	String	<code>urn:uuid:c757873d-ec9a-4326-a141-556f43239520</code>
id	String	<code>Id-01</code>
markdown	String	<code>*A*: `markdown code`</code>

FHIR Complex Types	Arden Syntax Complex Types	Example
General-Purpose, Special-Purpose, Metadata Types	Object	<code>{"system": "http://loinc.org", "code": "LL344-1"}</code>
FHIR resources	Object	<code>{"resourceType": "Observation", ...}</code>
Cardinality many (1/0 .. n)	List	<code>[{"code": "LL344-1"}, {"code": "A578"}]</code>

## 5.2 Arden Syntax Object Declarations of FHIR Resources

This section covers the FHIR resources mandated by the Arden Syntax standard: *Patient*, *Observation*, *Condition* and *Encounter* as well as the General-Purpose, Special-Purpose and Metadata Types those depend on. Any Arden Syntax Compiler or Interpreter compliant with FHIR as standard data Model in Arden Syntax must at least have those object definitions pre-defined for each MLM.

Note again that any adherence to FHIR primitive value constraints (like codes or URIs) are up to the implementers, therefore they are not part of this implementation guide.

The actual object definitions have been extracted from the FHIR R4B documentation and according to best practices reformulated as Arden object definition.

### 5.2.1 Example: Patient Resource Object Declaration

This one fully outlined object declaration of the Patient resource, core to the FHIR standard, is provided to exemplify object declarations according to the implementation practices and rules established in the Arden Syntax Standard and Arden Syntax on FHIR IG. Additional resource object declarations shall adhere to the example set here and follow rules as well as best practices established.

Patient ( <a href="#">FHIR R4B link</a> )		
<b>id</b>	<i>String</i>	Logical id of this artifact
<b>meta</b>	<i>Meta</i>	Metadata about the resource
<b>implicitRules</b>	<i>String</i>	A set of rules under which this content was created

<b>language</b>	<i>String</i>	Language of the resource content
<b>text</b>	<i>String</i>	Text summary of the resource, for human interpretation
<b>contained</b>	<i>List of any <b>Resource Object</b></i>	Contained, inline Resources
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>modifierExtension</b>	<i>List of <b>Extension</b></i>	Extensions that cannot be ignored
<b>identifier</b>	<i>List of <b>Identifier</b></i>	An identifier for this patient
<b>active</b>	<i>Boolean</i>	Whether this patient's record is in active use
<b>name</b>	<i>List of <b>HumanName</b></i>	A name associated with the patient
<b>telecom</b>	<i>List of <b>ContactPoint</b></i>	A contact detail for the individual
<b>gender</b>	<i>String</i>	Any of <i>male</i> , <i>female</i> , <i>other</i> or <i>unknown</i>
<b>birthdate</b>	<i>Time</i>	The date of birth for the individual
<b>deceasedBoolean</b>	<i>Boolean</i>	Indicates if the individual is deceased or not
<b>deceasedDateTime</b>	<i>Time</i>	
<b>address</b>	<i>List of <b>Address</b></i>	An address for the individual
<b>maritalStatus</b>	<i><b>CodeableConcept</b></i>	Marital (civil) status of a patient
<b>multipleBirthBoolean</b>	<i>Boolean</i>	Whether patient is part of a multiple birth
<b>MultipleBirthInteger</b>	<i>Number</i>	
<b>photo</b>	<i>List of <b>Attachment</b></i>	Image of the patient
<b>contact</b>	<i><b>PatientContact</b></i>	A contact party (e.g., guardian, partner, friend) for the patient
<b>communication</b>	<i><b>PatientCommunication</b></i>	A language which may be used to communicate with the patient about his or her health
<b>generalPractitioner</b>	<i><b>Reference</b> of type <b>Organization</b>, <b>Practitioner</b> or <b>PractitionerRole</b></i>	Patient's nominated primary care provider
<b>managingOrganization</b>	<i><b>Reference</b> of type <b>Organization</b></i>	Organization that is the custodian of the patient
<b>link</b>	<i><b>PatientLink</b></i>	Link to another patient that concerns the same actual person

#### PatientContact

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>relationship</b>	<i>List of <b>CodeableConcept</b></i>	The kind of relationship

<b>name</b>	<b>HumanName</b>	A name associated with the contact person
<b>telecom</b>	<i>List of ContactPoint</i>	A contact detail for the person
<b>address</b>	<b>Address</b>	Address for the contact person
<b>gender</b>	<i>String</i>	male   female   other   unknown
<b>organization</b>	<i>Reference of type Organization</i>	Organization that is associated with the contact
<b>period</b>	<b>Period</b>	The period during which this contact person or organization is valid

#### PatientCommunication

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of Extension</i>	Additional content defined by implementations
<b>language</b>	<i>CodeableConcept</i>	The language which can be used to communicate with the patient about his or her health
<b>preferred</b>	<i>Boolean</i>	Language preference indicator

#### PatientLink

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of Extension</i>	Additional content defined by implementations
<b>other</b>	<i>Reference of type Patient or RelatedPerson</i>	The other patient or related person resource that the link refers to
<b>type</b>	<i>String</i>	replaced-by   replaces   refer   seealso

#### Reference ([FHIR R4B link](#))

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of Extension</i>	Additional content defined by implementations
<b>reference</b>	<i>String</i>	Literal reference, Relative, internal or absolute URL
<b>type</b>	<i>String</i>	Type the reference refers to (e.g. "Patient")
<b>identifier</b>	<i>String</i>	Logical reference, when literal reference is not known
<b>display</b>	<i>String</i>	Text alternative for the resource

**Meta ([FHIR R4B link](#))**

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>versionId</b>	<i>String</i>	Version specific identifier
<b>lastUpdated</b>	<i>Time</i>	When the resource version last changed
<b>source</b>	<i>String</i>	Identifies where the resource comes from
<b>profile</b>	<i>List of String</i>	Profiles this resource claims to conform to
<b>security</b>	<i>List of <b>Coding</b></i>	Security Labels applied to this resource
<b>tag</b>	<i>List of <b>Coding</b></i>	Tags applied to this resource

**Extension ([FHIR R4B link](#))**

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>url</b>	<i>String</i>	Identifies the meaning of the extension
<b>value[x]</b>	<i>X identified by any of <b>Base64Binary, Boolean, Canonical, Code, Date, DateTime, Decimal, Id, Instant, Integer, Markdown, Old, PositiveInt, String, Time, UnsignedInt, Uri, Url, Uuid, Address, Age, Annotation, Attachement, CodableConcept, CodeableReference, Coding, ContactPoint, Count, Distance, Duration, HumanName, Identifier, Money, Period, Quantity, Range, Ratio, RatioRange, Reference, SampledData, Signature, Timing, ContactDetail,</b></i>	Value of extension

**Contributor,  
DataRequirement,  
Expression,  
ParameterDefinition,  
RelatedArtifact,  
TriggerDefinition,  
UsageContext, Dosage**

\* *Extension can assume a multitude of types (valueBase64Binary and so on). In terms of FHIR, those types represent [primitive and other Data types](#), but not actual resources, so implementers are highly recommended to support all extension values.*

#### Identifier ([FHIR R4B link](#))

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>use</b>	<i>String</i>	usual   official   temp   secondary   old (If known)
<b>type</b>	<b>CodeableConcept</b>	Description of identifier
<b>system</b>	<i>String</i>	The namespace for the identifier value
<b>value</b>	<i>String</i>	The value that is unique
<b>period</b>	<b>Period</b>	Time period when id is/was valid for use
<b>assigner</b>	<i>Reference of type Organization</i>	Organization that issued id (may be just text)

#### HumanName ([FHIR R4B link](#))

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>use</b>	<i>String</i>	usual   official   temp   nickname   anonymous   old   maiden
<b>text</b>	<i>String</i>	Text representation of the full name
<b>family</b>	<i>String</i>	Family name
<b>given</b>	<i>List of String</i>	Given names, includes middle names
<b>prefix</b>	<i>List of String</i>	Parts that come before the name
<b>suffix</b>	<i>List of String</i>	Parts that come after the name
<b>period</b>	<b>Period</b>	Time period when name was/is in use

**ContactPoint ([FHIR R4B link](#))**

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>system</b>	<i>String</i>	phone   fax   email   pager   url   sms   other
<b>value</b>	<i>String</i>	The actual contact point details
<b>use</b>	<i>String</i>	home   work   temp   old   mobile
<b>rank</b>	<i>Number</i>	Specify preferred order of use
<b>period</b>	<b>Period</b>	Time period when the contact point was/is in use

**Address ([FHIR R4B link](#))**

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>use</b>	<i>String</i>	home   work   temp   old   billing
<b>type</b>	<i>String</i>	postal   physical   both
<b>text</b>	<i>String</i>	Text representation of the address
<b>line</b>	<i>List of String</i>	Street name, number, direction & P.O. Box etc.
<b>city</b>	<i>String</i>	Name of city, town etc.
<b>district</b>	<i>String</i>	District name (aka county)
<b>state</b>	<i>String</i>	Sub-unit of country
<b>postalCode</b>	<i>String</i>	Postal code for area
<b>country</b>	<i>String</i>	Country (e.g., can be ISO 3166 2 or 3 letter code)
<b>period</b>	<b>Period</b>	Time period when address was/is in use

**CodeableConcept ([FHIR R4B link](#))**

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>coding</b>	<i>List of <b>Coding</b></i>	Code defined by a terminology system
<b>text</b>	<i>String</i>	Plain text representation of the concept

**Coding ([FHIR R4B link](#))**

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>system</b>	<i>String</i>	Identity of the terminology system (URI)
<b>version</b>	<i>String</i>	Version of the system - if relevant
<b>code</b>	<i>String</i>	Symbol in syntax defined by the system
<b>display</b>	<i>String</i>	Representation defined by the system
<b>userSelected</b>	<i>Boolean</i>	If this coding was chosen directly by the user

#### Attachment ([FHIR R4B link](#))

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>contentType</b>	<i>String</i>	Mime type of the content, with charset etc.
<b>language</b>	<i>String</i>	Human language of the content (BCP-47)
<b>data</b>	<i>String</i>	Data inline, <b>base64ed</b>
<b>url</b>	<i>String</i>	Uri where the data can be found
<b>size</b>	<i>Number</i>	Number of bytes of content (if url provided)
<b>hash</b>	<i>String</i>	Hash of the data (sha-1, <b>base64ed</b> )
<b>title</b>	<i>String</i>	Label to display in place of the data
<b>creation</b>	<i>Time</i>	Date attachment was first created

#### Period ([FHIR R4B link](#))

<b>id</b>	<i>String</i>	Unique id for inter-element referencing
<b>extension</b>	<i>List of <b>Extension</b></i>	Additional content defined by implementations
<b>start</b>	<i>Time</i>	Starting time with inclusive boundary
<b>end</b>	<i>Time</i>	End time with inclusive boundary, if not ongoing

## 6. Outlook

A challenge in using FHIR as a standard data model in Arden Syntax is the varying level of maturity of different FHIR resources within the overall standard. Moreover, FHIR is continuing to evolve, with successive releases introducing new resources promoting others to higher levels of maturity. As FHIR evolves in this manner, it is expected that the Arden Syntax will evolve through successive releases in order to incorporate updated or new resources and thereby extend access to data represented in a standard way within MLMs.



## 7. References

- (1) *HL7 Standards - Health Level Seven Arden Syntax for Medical Logic Systems, Edition 3.0*, URL: [https://www.hl7.org/implement/standards/product\\_brief.cfm?product\\_id=639](https://www.hl7.org/implement/standards/product_brief.cfm?product_id=639)
- (2) *HL7 Standards - Health Level Seven Fast Healthcare Interoperability Resources (FHIR) v4.3.0*, URL: <http://hl7.org/fhir/R4B/index.html>
- (3) *HL7 Standards – Health Level Seven Clinical Quality Language (CQL) Release 1*, URL: <https://cql.hl7.org/>

## 8. F.A.Q.

### **This IG only covers Arden on FHIR, where can I find more guidance on Arden Syntax?**

Implementation guides target knowledge about a group of use-cases in a practical approach. Arden Syntax IGs are separated in two published documents, this one covering FHIR related use cases in Arden. To start working on Arden the first time we recommend starting with [arden-syntax-IG-R3-2019-09-16-ballot-reconciliation.pdf](#) (available from Arden's Confluence page) as well as the latest [Arden Syntax 3](#) standard publication.