

Introduction to HL7 FHIR Shorthand



Mark Kramer & Chris Moesel



Advance Setup

1. Install [Visual Studio Code](#) and the [FSH language extension](#)
2. Install SUSHI using [these directions](#)
3. Install the FHIR IG Publisher
 - a. Install a Java runtime (e.g., [Eclipse Adoptium Temurin JDK](#))
 - b. Install Ruby and Jekyll using [these OS-specific instructions](#)
 - c. Download the IG Publisher and scripts
 - a. Open a Terminal/Command Prompt window inside the IG folder
 - b. Mac/Linux run `./_updatePublisher.sh`
 - c. Windows run `_updatePublisher.bat`



Agenda

- Mutual Introductions (1:45 - 2:00)
- Background [Lecture] (2:00 - 2:30)
- Learn to FSH [Lecture] (2:30 - 3:00)
- Cookie Break (3:00 - 3:30)
- Hands-On Exercise using FSH Online (3:30 - 3:45)
- Serious FSHing [Lecture] (3:45 - 4:00)
- Hands-On Exercise using SUSHI & IG Publisher (4:10 - 4:30)
- Deeper FSH [Lecture] (4:30 - 4:45)
- Extensions Exercise (take home exercise)
- Summary and Next Steps to Mastery (4:45 - 5:00)



Introductions

- Hi! I'm Mark Kramer
 - Distinguished Chief Engineer in MITRE's Health Innovation Center
 - Background in modeling, AI, process control, systems engineering
 - Got involved in healthcare around 2009
 - Created first (or nearly first) implementation guide in 2015
 - Created FHIR Shorthand w/ Chris Moesel (first released 2020)



SOLVING PROBLEMS
FOR A SAFER WORLD®

A US non-profit operating federally-funded research and development centers, working in the public interest. MITRE brings a conflict-free perspective and a whole of government vantage point to bring innovative ideas into existence.

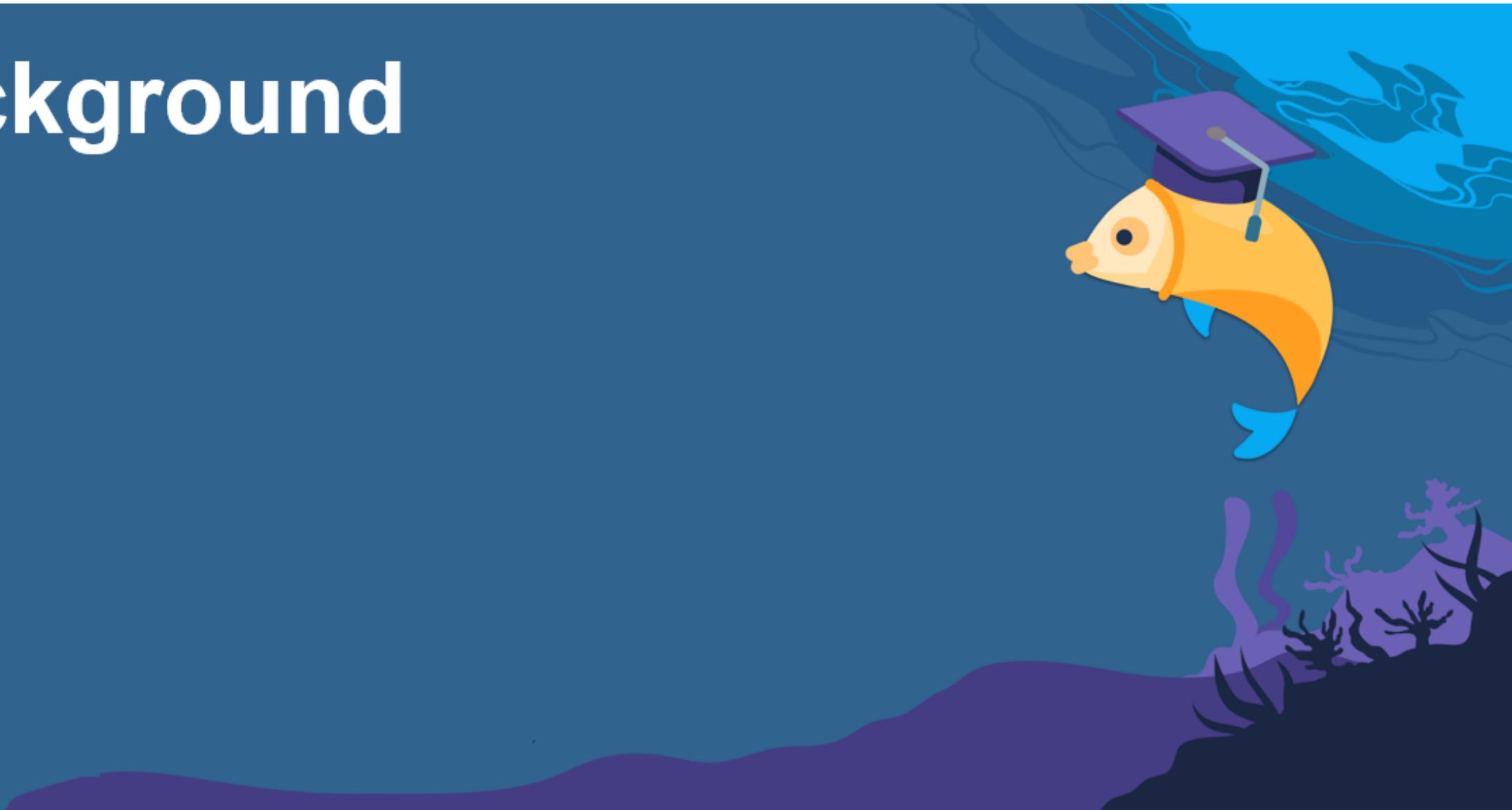


Learning Objectives

- Purpose and use of FHIR Shorthand (FSH)
- Learn the basics of the FHIR Shorthand Language
- Create an Implementation Guide (IG) with FHIR Shorthand
- Later this week (Wednesday AM)
 - Advanced HL7 FHIR Shorthand with Chris Moesel



Background

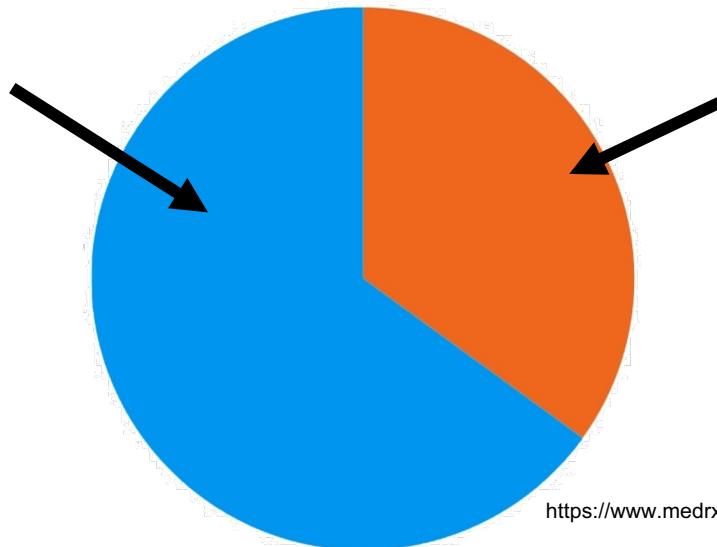


Why Create an Implementation Guide?

- FHIR is a framework, not a complete solution

Base FHIR:

- On-the-wire syntax
- API for read, write, search
- ~140 base resources
- Value sets
- Extensibility mechanism
- & more



Implementation Guide:

- Specific problem to be solved
- Actors and workflow sequences
- Profiles of base resources
- Terminology and value sets
- Additional API operations
- & more

<https://www.medrxiv.org/content/10.1101/2022.03.09.22272163v1.full>



What makes a good IG?

- An IG is an agreement among implementers on how to solve a specific problem
 - If it works and enough people agree, it isn't wrong
- The effectiveness depends on multiple factors:
 - The clarity of the definition of the problem the IG attempts to solves
 - Working with the right community of future users
 - Definition of the data elements and workflow to support the use cases
 - Suitable selection of terminology
 - Balancing constraints with implementation burden

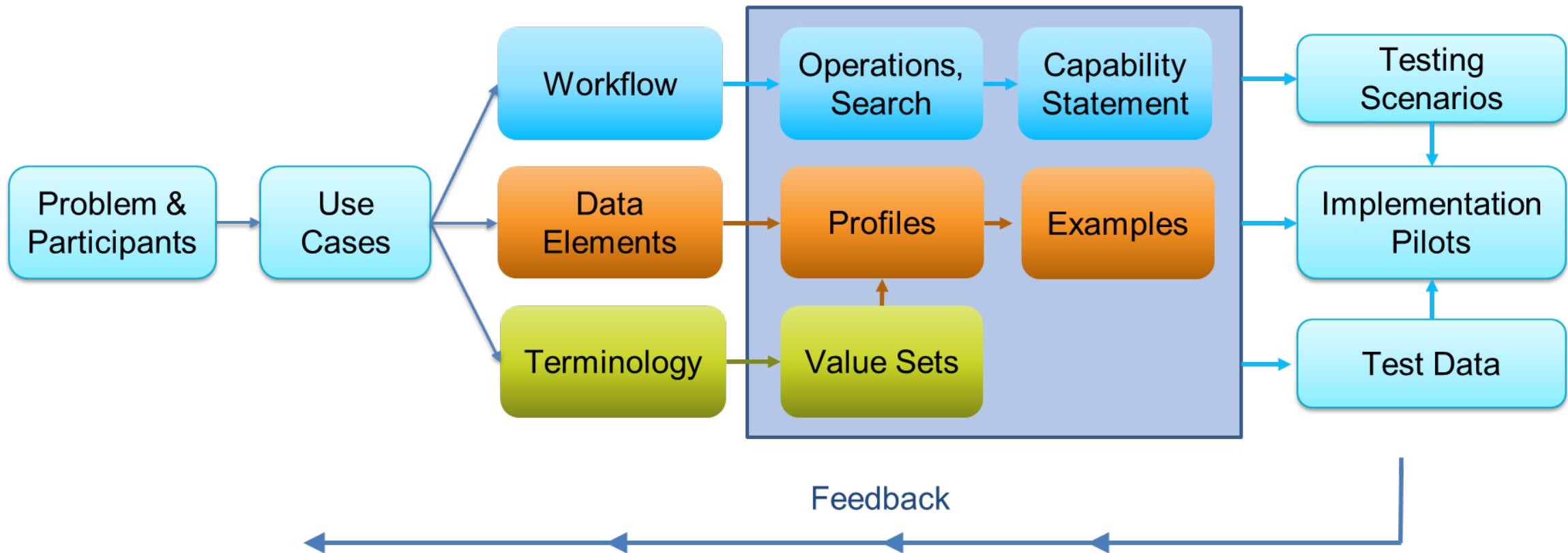


FHIR Solution Map

Scoping & Conceptual Design

Implementation Guide Creation

Validation &
Implementation



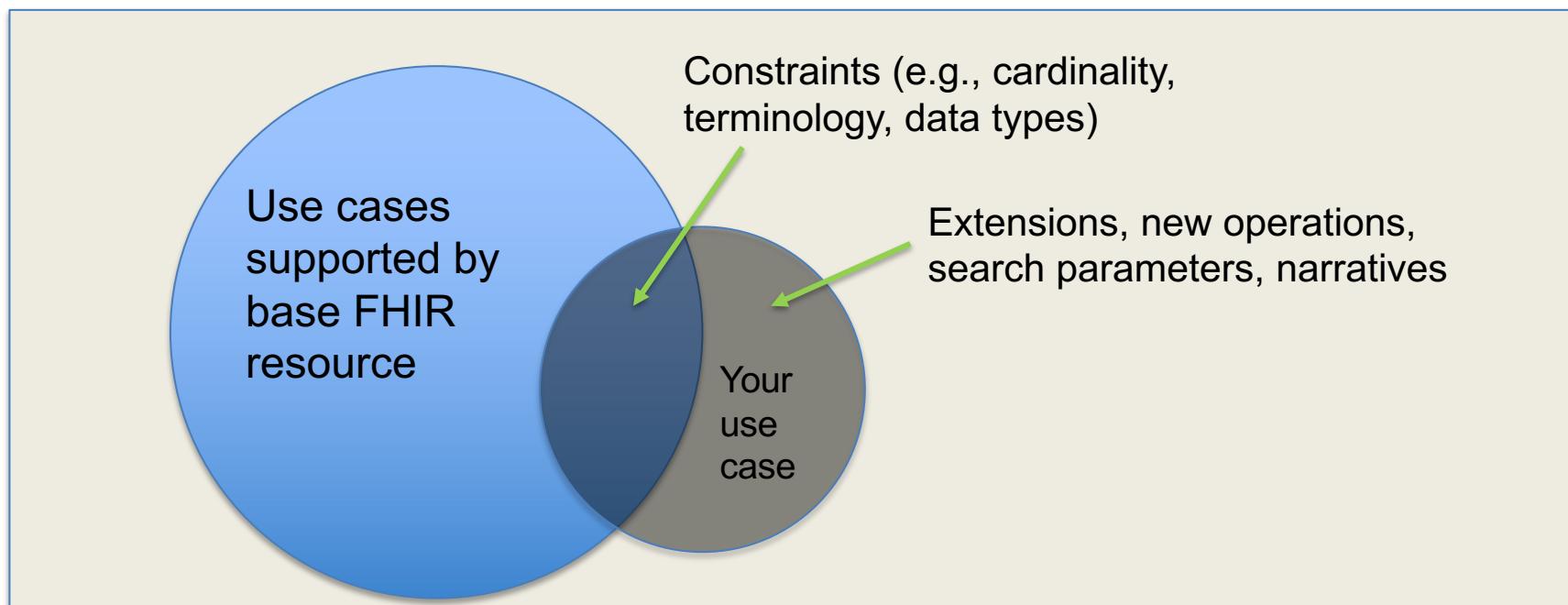
Types of IGs (according to Grahame Grieve)

- National Base IG
 - Describes how national healthcare regulations apply in the context of FHIR.
 - Examples: [US Core](#), [AU Base](#)
- Domain of Knowledge IG
 - Describes how to represent a clinical or business concept, without defining an API.
 - Examples: [International Patient Summary](#), [mCODE](#)
- Community of Implementation
 - An agreement on how data are exchanged by a group of actors (i.e., contains an API).
 - Examples: [Da Vinci Health Record Exchange](#), [National Healthcare Directory Exchange](#)
- Product-Specific IG
 - e.g., EPIC on FHIR



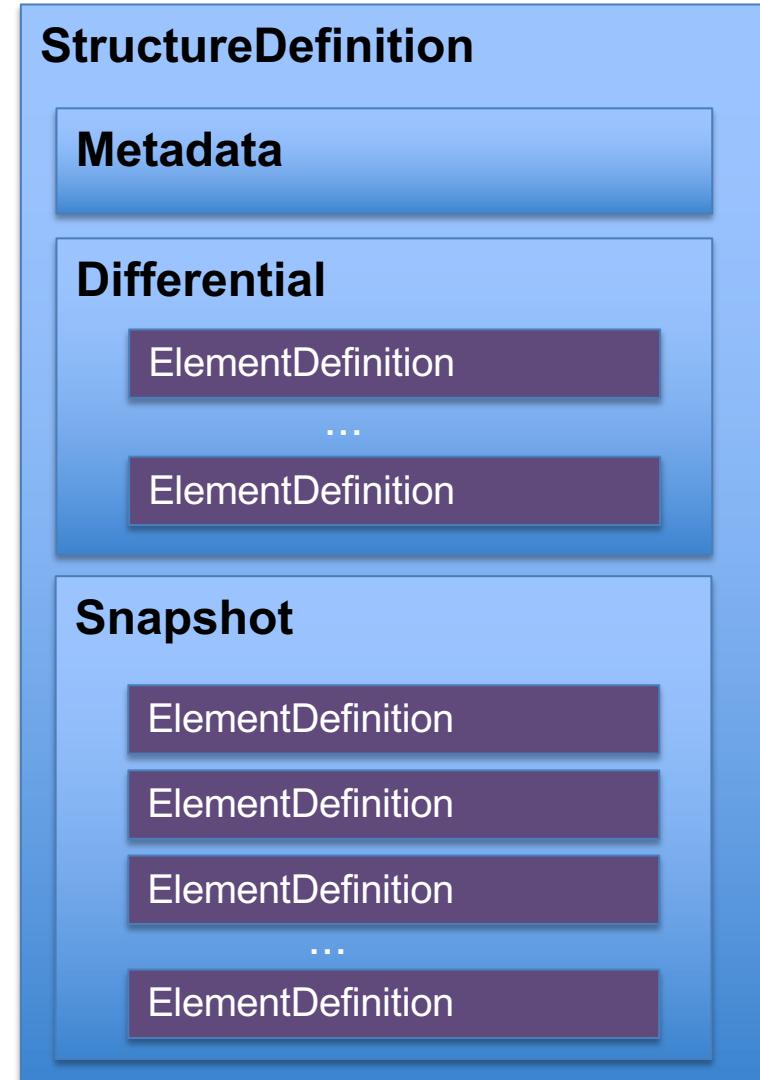
What is a Profile?

- A profile is a modification of a FHIR resource
 - Both constrain and extend
- Every profile modifies a base resource or another profile, e.g.:
 - Observation → Blood Pressure
 - Observation → Laboratory Result → Hematocrit



StructureDefinition (SD) Resource

- SD is a FHIR resource used to represent all resources, datatypes, and profiles
- Much more than just "structure", the SD includes an element-by-element definition of data types, value sets, cardinality, and more
- Can be directly edited, but complex and error-prone, therefore not recommended



Parent resource or profile, publisher, version, status, etc.

Constraints and extensions

All elements (auto-generated from parent and differential)



Profile Page in IG = View of an SD

This structure is derived from [USCorePatientProfile](#)

Name	Flags	Card.	Type	Description & Constraints
Patient	S	0..*	USCorePatientProfile	Information about an individual or animal receiving health care services
deceased[x]	S	0..1	boolean, dateTime	Indicates if the individual is deceased or not

[Documentation for this format](#)

Link to definition of the base resource being profiled

Element flags

Element cardinality constraints

Type of data allowed in a given element

Parent of the profile (could be base resource or another profile)

"Short" narrative definition of an element, plus description of value set bindings and invariant constraints

Name	Flags	Card.	Type	Description & Constraints
Patient	S	0..*	USCorePatientProfile	Information about an individual or animal receiving health care services
id	Σ	0..1	string	Logical id of this artifact
meta	Σ	0..1	Meta	Metadata about the resource
implicitRules	?!	0..1	uri	A set of rules under which this content was created
language		0..1	code	Language of the resource content Binding: CommonLanguages (preferred) Max Binding: AllLanguages: A human language.



Viewing the SD



minimal Common Oncology Data Elements (mCODE) Implementation Guide
2.1.0 - Pre-STU 3 Development



Home Content by Group Conformance FHIR Artifacts Examples Release Notes More

Table of Contents > Artifacts Summary > Cancer Patient Profile

minimal Common Oncology Data Elements (mCODE) Implementation Guide, published by HL7 International Clinical Interoperability Council. This is not an authorized publication; it is the continuous build for version 2.1.0. This version is based on the current content of <https://github.com/HL7/fhir-mCODE-i>. Changes regularly. See the [Directory of published versions](#).

Content Detailed Descriptions Mappings Examples XML **JSON** TTL

18.13.1 Resource Profile: Cancer Patient Profile

Official URL: <http://hl7.org/fhir/us/mcode/StructureDefinition/mcode-cancer-patient>

Version: 2.1.0

Active as of 2022-09-06

Maturity Level: 5

Computable Name: CancerPa

A patient who has been diagnosed with or is receiving medical treatment for a malignant growth or tumor. This is the most essential profile in mCODE, since it is referenced by many other profiles. The only difference between the mCODE Patient profile and the [US Core Patient Profile](#) is that Patient.deceased is a required element in mCODE.

```
    "url": "http://hl7.org/fhir/us/mcode/StructureDefinition/mcode-cancer-patient",
    "version": "2.1.0",
    "name": "CancerPatient",
    "title": "Cancer Patient Profile",
    "status": "active",
    "date": "2022-09-06T23:19:27+00:00",
    "publisher": "HL7 International Clinical Interoperability Council",
    "contact": [
        {
            "name": "HL7 International Clinical Interoperability Council",
            "telecom": [
                {
                    "system": "url",
                    "value": "http://www.hl7.org/Special/committees/cic"
                },
                {
                    "system": "email",
                    "value": "ciclist@lists.HL7.org"
                }
            ]
        }
    ],
    "description": "A patient who has been diagnosed with or is receiving medical treatment for a malignant growth or tumor. This is the most essential profile in mCODE, since it is referenced by many other profiles. The only difference between the mCODE Patient profile and the [US Core Patient Profile](https://www.hl7.org/fhir/profiling.html#mustsupport) is that Patient.deceased is a required element in mCODE.",
    "jurisdiction": [
        {
            "coding": [
                {
                    "system": "urn:iso:std:iso:3166",
                    "code": "US",
                    "display": "United States of America"
                }
            ]
        }
    ],
    "fhirVersion": "4.0.1",
```

4375 lines long



What is FHIR Shorthand (FSH)?

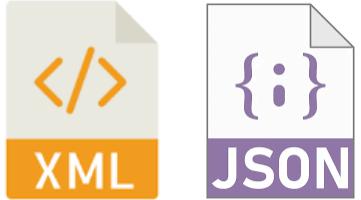
Domain-specific language (DSL) for defining FHIR artifacts

1. You author some FSH
2. Run SUSHI to create all the computable artifacts for an IG:
 - Profiles (of resources, profiles, complex datatypes, extensions)
 - Instances (e.g., examples, test data, conformance resources)
 - Terminology (value sets, code systems, codes)
3. You create the narrative portions of IG (markdown, images, etc.)
4. Run the HL7 IG Publisher to create the IG
5. Iterate until done



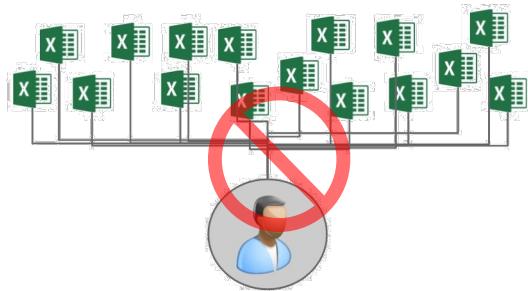
Approaches to Profiling

Direct Editing



tedious & error-prone

Spreadsheets



being phased out

Applications



Profile on Patient⁽⁵⁾: PatientNL

Properties	Narrative	Element Tree	Element Grid	XML
URL	https://example.org/fhir/StructureDefinition/PatientNL			
Resource ID				
Name	PatientNL			
Description	StructureDefinition for a Dutch Patient.			

Appeals to authors who want to be guided through the process

Declarative



MAKE ME A SANDWICH.



FSH

Sushi

Appeals to authors who prefer to be in command of the process



Why do people like and use FSH?



Concise, readable, understandable



Great set of free tools



Fast global changes via text operations



Large user community



Collaborative authoring and project management via source code control



HL7 Balloted Normative+STU standard



Error checking & application of best practices by SUSHI



Built into the FHIR Publisher



Combine with other profiling approaches



FSH-ify any existing IG (GoFSH)

Worldwide FSH Consumption is Rising



Source: Nederlands Visbureau

	Dec. 2020	Dec. 2021	May 2022	Aug. 2022
Projects ¹	40	177	294	318
Chat Subscribers ²	169	308	405	440*
SUSHI Downloads ³	25,000	82,000	109,000	122,000

* Currently the sixth busiest non-bot channel on chat.fhir.org

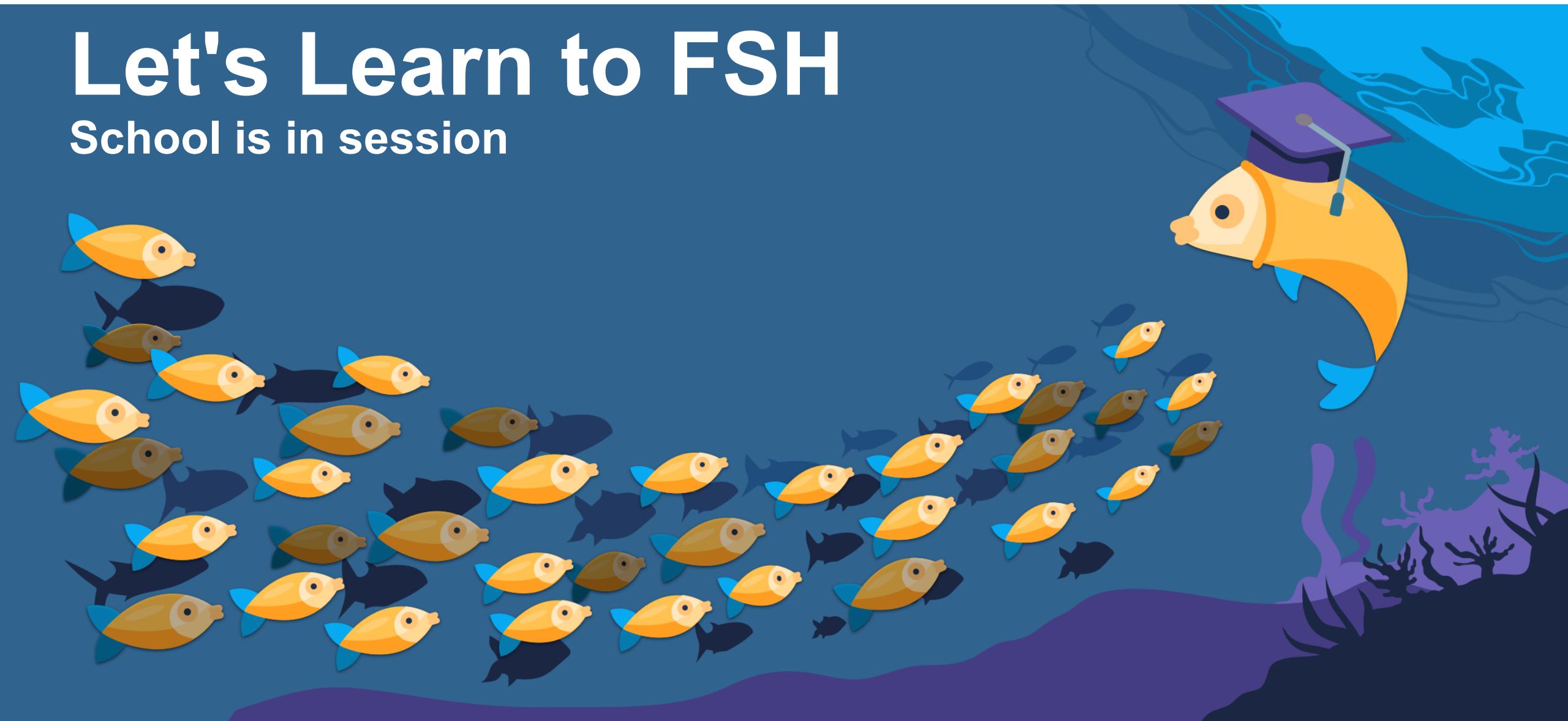
1. <https://fshschool.org/fsh-finder/>

2. <https://chat.fhir.org/#streams/subscribed>

3. <https://npm-stat.com/charts.html?package=fsh-sushi&from=2020-01-01&to=2022-08-08>

Let's Learn to FSH

School is in session



FSH Ecosystem

- [FSH Language Specification](#) -- complete syntax and examples
- [SUSHI](#) -- convert FSH into FHIR Artifacts
- [Go FSH](#) -- convert FHIR Artifacts into FSH
- [FSH School](#) -- a web site with documentation, tools, tutorials
- [FSH Online](#) -- interact with FHIR Shorthand
- [Visual Studio Code extension](#) -- Handy tools for VS Code editor



FSH Online

<https://fshschool.org> → Play with FSH

The screenshot shows the FSH Online web application. At the top, there's a navigation bar with the logo, "FSH ONLINE", and "Powered by SUSHI v2.3.0 and GoFSH v1.5.0". On the right side of the bar are "Back to School", "Save All", and "Configuration" buttons. Below the bar, there are tabs: "FSH Examples", "Convert to JSON ►", "◀ Convert to FSH", and "Save All" (which is currently selected). The main area has two sections. The left section is for "FSH" and contains a text input field with placeholder "Paste or edit FSH here...". The right section is for "FHIR JSON: Untitled" and also has a text input field with placeholder "Paste or edit single FHIR JSON artifact Create additional FHIR JSON artifact". A dropdown menu is open over the FHIR JSON section, showing options: "+ New JSON Editor", "Unknown Type", and "Untitled". At the bottom, there are "Console" and "Problems (0)" buttons, and a footer with "Source: https://fshschool.org/FSHOnline/#/", "Web Viewer Terms | Privacy & Cookies", and "Edit" buttons.

Type your
FSH here
and convert
to FHIR with
SUSHI

Or paste a
FHIR item
here and
convert to
FSH with
GoFSH
(covered in
advanced
course)



FSH Language Reference

<http://hl7.org/fhir/uv/shorthand/reference.html>

<http://hl7.org/fhir/uv/shorthand/FSHQuickReference.pdf>

The screenshot shows the FHIR Shorthand reference page. At the top is the HL7 International logo and the FHIR Shorthand logo. Below is a navigation bar with links for Home, Overview, Language Reference, and More. A yellow box contains a note: "This page is part of the FHIR Shorthand (v2.0.0: N1 - Mixed Normative and STU) based on FHIR R4. This is the current published version. For a full list of available versions, see the [Directory of published versions](#)." The main content area is titled "3 Language Reference". It includes a note about trial use, a detailed description of the chapter's purpose, and a note about key terms like MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL. A sidebar on the right lists various FSH topics.

The screenshot shows the FHIR Shorthand 1.2 Quick Reference document. It features a "KEY to Expression Syntax" table and several tables for "Creating Items" under different declaration types (Declaration, Alias, CodeSystem, Extension, Instance, Invariant, Logical or Resource, Mapping, Profile, RuleSet, ValueSet). Each table includes columns for Keywords and Applicable Rules. To the right is a "Rule Syntax" table with many rows of rules. At the bottom are sections for "More Information" with QR codes and links to FSH Specification, FSH Chat, FSH School, and HL7 Project Page.



A Complete FHIR Profile

<https://bit.ly/dd-fsh-1>

```
// This is a comment

// (1) Declaration
Profile: MyPatientProfile

// (2) Keywords
Parent: Patient      // Must specify a parent resource or profile
Description: "Example Patient"

// (3) Rules
* name 1..2
* name.given and name.family MS
* maritalStatus from http://hl7.org/fhir/ValueSet/marital-status
* deceased[x] only boolean
```



Things to Note:

- FSH items have three parts: Declaration, Keywords, and Rules
- Declaration and Keywords are followed by colon (:)

Profile: MyPatientProfile

Parent: Patient

- FSH rules begin with * symbol:

* deceased[x] only boolean

- Nested paths are expressed with periods (.)

name // top level

name.given // second level

name.period.start // third level



More Things to Note:

- Comments

```
// single-line comment
```

```
/* multi-
line
comment */
```

- White space

- Spacing doesn't matter except for indented rules (later) and in "strings"
- Empty lines are OK
- Break up long lines anywhere there is a space



Cardinality Rule

- Used to set the min and max occurrence of an element:
 - * `<element> {min}..{max} // give min, max, or both`
- Examples:
 - * `name 1..2 // change lower and upper bound`
 - * `name 1.. // change lower bound only`
 - * `name ..2 // change upper bound only`



Flag Rule

- Used to set *must support* and other information about an element:
 - * <element(s)> {flag(s)}where {flag} is MS, SU, ?!, TU, N, D
(i.e., ,Must Support, SUmmary, modifier, Trial Use, Normative, Draft)
- Examples:
 - * onsetDate MS SU
 - * identifier and identifier.system and identifier.value MS
 - * subject 1..1 MS // combined cardinality and flag rule



Binding Rule ("from")

- Used to associate a value set with a coded element in a profile:
 - * <bindable> from {ValueSet} {strength}
- where {strength} is required, extensible, preferred, example
 - If strength is not given, binding is assumed to be required
- Examples:
 - * telecom.system from <http://hl7.org/fhir/ValueSet/contact-point-system> (required)
 - * code from CancerConditionVS (extensible)
 - * gender from <http://hl7.org/fhir/ValueSet/administrative-gende>



FHIR Binding Strengths

- Binding strength indicates the conformance expectation associated with the value set:
 - **required**: the value MUST come from the value set provided
 - **extensible**: the value MUST come from the value set provided UNLESS the value set is missing the concept
 - **preferred**: the value MAY come from the value set provided (any value is acceptable)
 - **example**: any value is accepted
- Algorithms have trouble with **extensible**; usually, any value is accepted
- Required is the only binding with "teeth"



Type Rule ("only")

- Used to restrict data types in a profile:

- * <element> only {datatype(s)}

- Examples:

- * valueQuantity only SimpleQuantity

- * onset[x] only dateTime

- * onset[x] only dateTime or Period

- * performer only Reference(Practitioner)

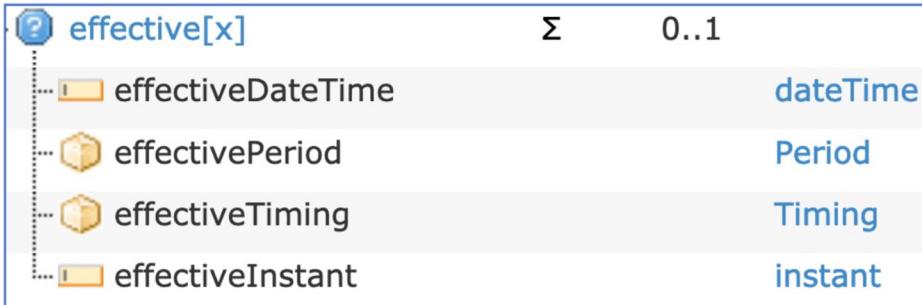
- * performer only Reference(PrimaryCarePhysician or EmergencyRoomPhysician)



Examples

Constrain Choice to a Subset

Example using FHIR R4 Observation.effective[x]



* `effective[x]` only `dateTime` or `Period`



Constrain Data Type to a Profile

Example using FHIR R4 Patient.address



* `address` only `MassachusettsAddress`



Identify the Rules Types in Sample Profile:

```
// (3) Rules  
  
* name 1..2 // ??? rule  
  
* name.given and name.family MS // ??? rule  
  
* maritalStatus from http://hl7.org/fhir/ValueSet/marital-status // ??? rule  
  
* deceased[x] only boolean // ??? rule
```



Identify the Rules in our Sample Profile:

```
// (3) Rules  
  
* name 1..2 // cardinality rule  
  
* name.given and name.family MS // flag rule (MS = Must Support)  
  
* maritalStatus from http://hl7.org/fhir/ValueSet/marital-status // binding rule  
  
* deceased[x] only boolean // type rule
```



Create a FHIR Instance

<https://bit.ly/fsh-2>

```
Instance: JaneDoe
InstanceOf: Patient // could also be a profile
Description: "A sample patient instance"
* active = true
* name[0].given[0] = "Jane"
* name[0].family = "Doe"
* gender = #female
* birthDate = "1965-01-01"
* address.line[0] = "123 Main St"
* address.city = "Anytown"
* address.postalCode = "12345"
* address.country = "US"
* maritalStatus = http://terminology.hl7.org/CodeSystem/v3-MaritalStatus#L "Legally Separated"
* generalPractitioner = Reference(DrSurtinLee)
```



Things to Note:

- Declaration and Keyword:
 - Instance:
 - InstanceOf: (required, either profile or resource)
- New type of rule: Assignment ("=")
 - Used to set a required value in a profile, or a value in an instance
 - * <element> = {value}
 - Examples:
 - * active = true
 - * gender = #female
 - * birthDate = "1965-01-01"
 - * address.line = "123 Main St"



Things to Note:

- Codes are indicated with # symbol:

#female // code system inferred from FHIR definition of gender

- Codings with explicit code system, code, and display text:

<http://terminology.hl7.org/CodeSystem/v3-MaritalStatus#L> "Legally Separated"

- Array elements indicated with square brackets, starting at 0:

- * name[0].given[0] = "Jane"
 - * name[0].family = "Doe"

- Reference to another instance using its FSH name:

Reference(DrSurtinLee)



Common Mistake



- Bind elements to **value sets** ("from")
- Assign individual codes from **code systems** ("=")

```
// IN A PROFILE, BIND TO A VALUE SET  
  
* maritalStatus from http://hl7.org/fhir/ValueSet/marital-status
```

```
// INDIVIDUAL CODES COME FROM CODE SYSTEMS (e.g., ICD-10, LOINC, SNOMED-CT)  
  
* maritalStatus = http://terminology.hl7.org/CodeSystem/v3-MaritalStatus#L  
"Legally Separated"
```

Tricks for Better Readability

- Indentation for shortening paths (optional)
 - Indent exactly two spaces to "inherit" a path from preceding rule

```
* address.line[0] = "123 Main St"  
* address.city = "Anytown"  
* address.postalCode = "12345"  
* address.country = "US"
```



```
* address  
  * line = "123 Main St"  
  * city = "Anytown"  
  * postalCode = "12345"  
  * country = "US"
```

- Remove unneeded array indexes
 - [0] can be omitted when populating the first value of a repeated element

Aliases

- Aliases can be defined for any URL
- Appear as independent statements (typically before other items)
- CAPS and \$ are conventions, not requirements

```
Alias: SCT = http://snomed.info/sct
```

```
Alias: $MS = http://terminology.hl7.org/CodeSystem/v3-MaritalStatus
```

```
* maritalStatus = http://terminology.hl7.org/CodeSystem/v3-MaritalStatus #L "Legally Separated"
```



can rewrite as

```
* maritalStatus = $MS#L "Legally Separated"
```



Re-writing the previous instance:

```
Alias: $MS = http://terminology.hl7.org/CodeSystem/v3-MaritalStatus
```

```
Instance: JaneDoe
```

```
InstanceOf: Patient
```

```
Description: "A sample patient instance"
```

```
* active = true
```

```
* name
```

```
  * given = "Jane" // refers to name[0].given[0]
```

```
  * family = "Doe" // refers to name[0].family
```

```
* gender = #female
```

```
* birthDate = "1965-01-01"
```

```
* address
```

```
  * line = "123 Main St"
```

```
  * city = "Anytown"
```

```
  * postalCode = "12345"
```

```
  * country = "US"
```

```
* maritalStatus = $MS#L "Legally Separated"
```

```
* generalPractitioner = Reference(DrSurtinLee)
```



Define a Simple Value Set

<https://bit.ly/fsh-4>

Alias: SCT = <http://snomed.info/sct>

ValueSet: ProcedureIntentVS

Id: procedure-intent-vs

Title: "Procedure Intent Value Set"

Description: "The purpose of a procedure."

* include SCT#373808002 "Curative - procedure intent"

* include SCT#363676003 "Palliative - procedure intent"

* include SCT#399707004 "Supportive - procedure intent"



Things to Note:

- Declaration
 - ValueSet: // UpperCamelCase (by convention)
- Keywords:
 - Id: // conventionally, lower-case-dash-separated
 - Title: // a "Human Readable Name"
 - Description: // "Several descriptive sentences."
- "include" is optional and can be omitted:
 - * include SCT#373808002 "Curative - procedure intent same as:
 - * SCT#373808002 "Curative - procedure intent



Intensional Value Sets

- In addition to including code one at a time, there are various ways to add groups of codes, e.g., all concepts from code system(s) or value set(s)
- You can also exclude codes and value sets

```
Alias:    SCT = http://snomed.info/sct
```

```
ValueSet: TherapeuticIntentVS
```

```
Id: therapeutic-intent-vs
```

```
Title: "Therapeutic Intent Value Set"
```

```
Description: "The purpose of a procedure."
```

```
* include codes from system SCT where concept is-a #262202000 "Therapeutic intent"
```

```
* exclude SCT#421974008 "Adjunct - intent"
```



COOKIE BREAK!



Summary: Common Declarations and Keywords

Declarations	Data Type
CodeSystem:	name
Extension:	name
Instance:	id
Profile:	name
ValueSet:	name

Keywords	Applies to...	Purpose	Data Type
Description:	Most item types	Human-readable description	string or markdown
Id:	Most items, not instances	Item identifier	id
InstanceOf:	Instances	Instantiated profile or resource	name, id, or url
Parent:	Profiles (rarely extensions)	Base definition	name, id, or url
Title:	Most item types	Human-readable name	string



Summary: Common FSH Rules

Type	Syntax	Examples
Assignment	* <element> = {value}	<pre>* code = https://loinc.org#69548-6 * status = #active * onsetDateTime = "2019-04-02"</pre>
Binding	* <bindable> from {ValueSet} ({strength})	<pre>* gender from http://hl7.org/fhir/ValueSet/administrative-gender * telecom.system from http://hl7.org/fhir/ValueSet/contact-point-system (required) * code from CancerConditionVS (extensible)</pre>
Cardinality	* <element> {min}..{max} // specify min, max, or both	<pre>* subject 1..1 * category 1.. // only set lower bound</pre>
Flag	* <element(s)> {flag(s)}	<pre>* onsetDate MS * communication MS SU TU ?! * identifier and identifier.system and identifier.value and name and name.family MS * subject 1..1 MS // combined cardinality and flag</pre>
Type	* <element> only {datatype(s)}	<pre>* valueQuantity only SimpleQuantity * onset[x] only dateTime * onset[x] only Period or Range * performer only Reference(Practitioner) * performer only Reference(PrimaryCarePhysician or EmergencyRoomPhysician) // profiles</pre>



Exercise

- Create a profile representing **ease of respiration**
 - Based on Observation resource
 - Category "exam" (see <http://hl7.org/fhir/ValueSet/observation-category>)
 - Identifying code: SNOMED-CT 248546008 | Ease of respiration (observable entity) |
 - Potential answers (SNOMED-CT):
 - 276888009 | Normal spontaneous respiration (finding) |
 - 248549001 | Labored breathing (finding) |
 - 271825005 | Respiratory distress (finding) |



Bonus Points

- Create two FHIR instances:
 - Patient Colin Robinson, age 57, male
 - Currently experiencing labored breathing



Save Your Work

create a permanent link (content can be restored or shared) or download to your computer

The screenshot shows the FSH Online interface. On the left, there's a sidebar with 'FSH Examples' and a 'Back to School' link. The main area has two tabs: 'Convert to JSON' (highlighted with a red arrow) and 'Convert to FSH'. Below these tabs is a toolbar with icons for copy, save, and delete. The left pane displays FHIR JSON code for 'BodyStructureWithLaterality', and the right pane shows the corresponding FSH code. A sidebar on the right lists profiles, extensions, instances, and a selected item 'JaneDoe'. At the bottom, there's a 'SUSHI RESULTS' table with four columns: Profiles (1), Extensions (1), Logicals (0), and Resources (0). The bottom left corner shows the number '4'.



Serious FSHing



Krabbenkutter Ivonne Pellworm P5242390jm.JPG CC BY-SA 2.0 de

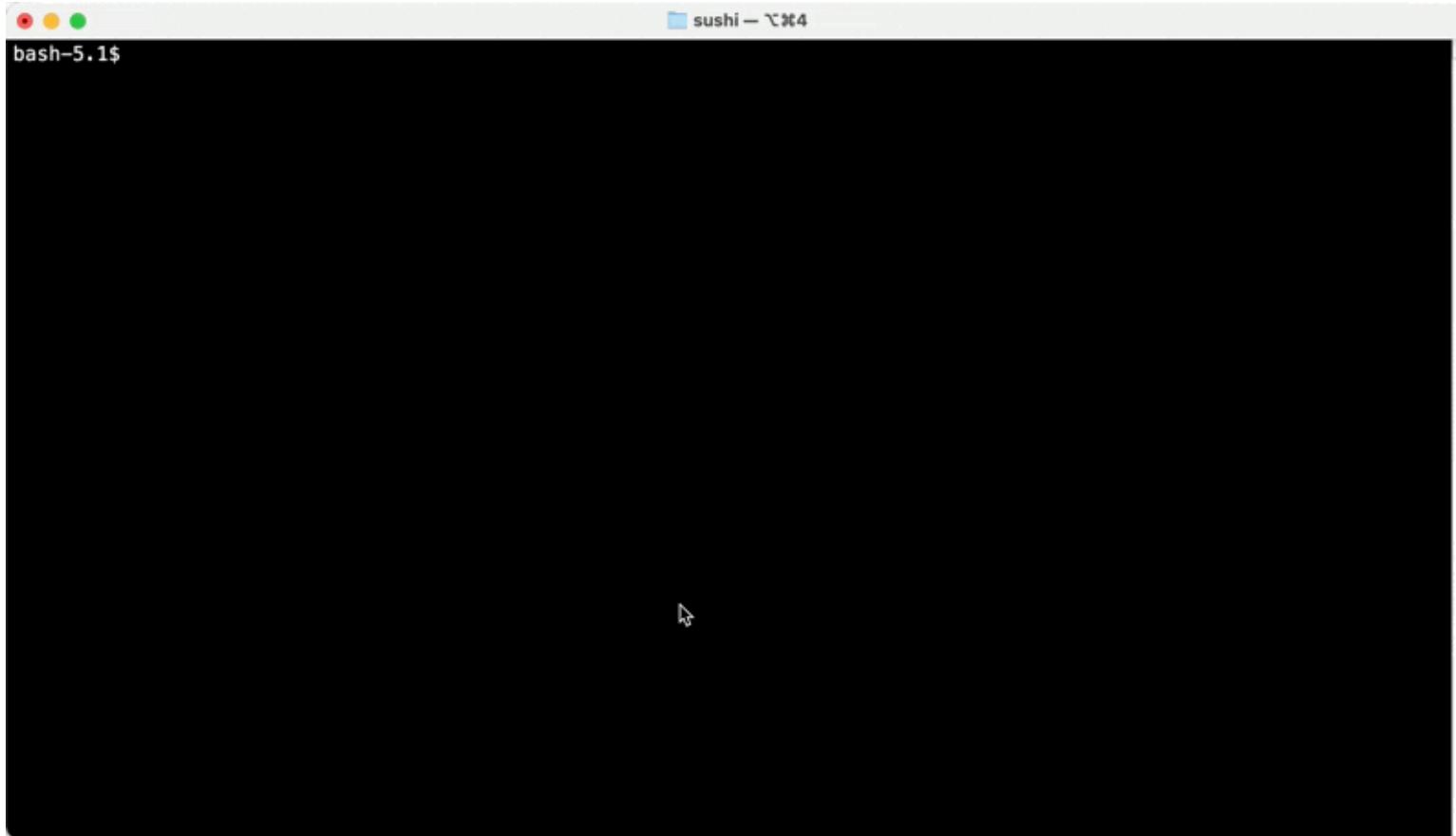
VS Code, SUSHI & FHIR IG Publisher

1. Install [Visual Studio Code](#) and the [FSH language extension](#)
2. Install SUSHI using [these directions](#)
3. Install FHIR IG Publisher
 - a. Install a Java runtime (e.g., [Eclipse Adoptium Temurin JDK](#))
 - b. Install Ruby and Jekyll using [these OS-specific instructions](#)
 - c. Download the IG Publisher and scripts
 - a. Open a Terminal/Command Prompt window inside the IG folder
 - b. Mac/Linux run `./_updatePublisher.sh`
 - c. Windows run `_updatePublisher.bat`



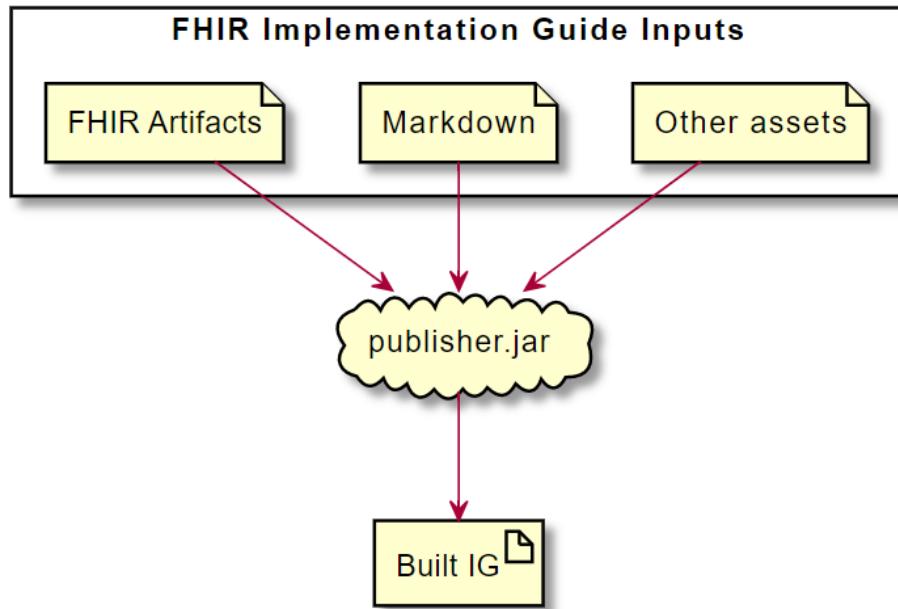
Starting a new IG Project

- Decide on the name, id, and canonical for your IG
- Open a command window
- Navigate to the directory where you want your IG
- Type command:
`sushi --init`
- Follow the prompts

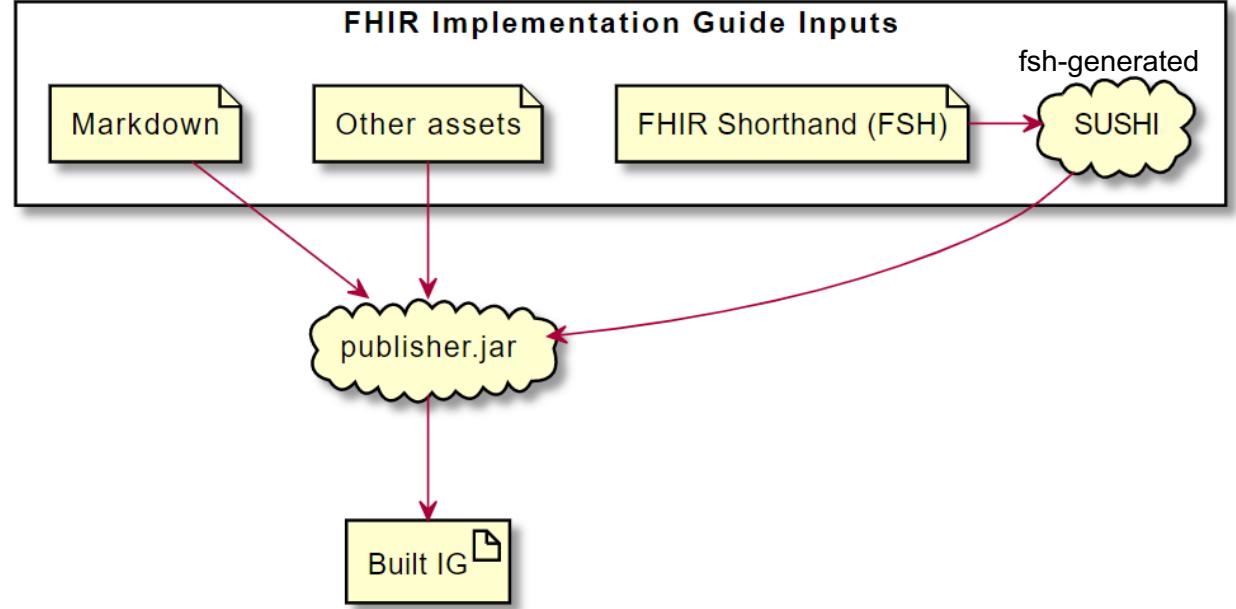


IG Creation Workflow

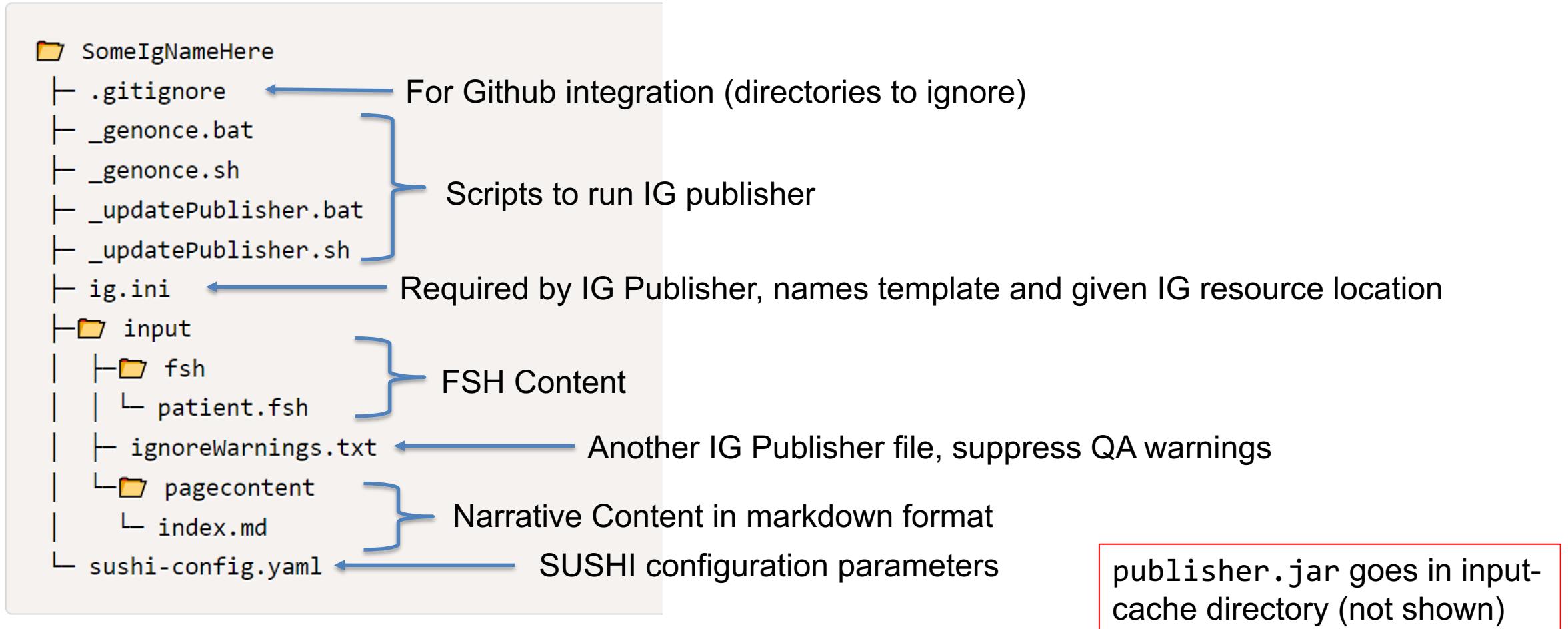
FHIR IG Publisher



FHIR IG Publisher with FSH & SUSHI



Minimal Directory Structure



`publisher.jar` goes in input-cache directory (not shown)



After Running IG Publisher:

- Generated directories:
 - **fsh-generated**: FHIR artifacts, the output of running SUSHI
 - **output**: your IG is here -- open index.html file
 - **input-cache**: stored terminology and schemas
 - **temp**: files generated during run
 - **template**: templates for building IG



FSH Files

- Text files with .fsh extension
- One project can contain multiple .fsh files
- Contains any number of items -- it doesn't matter how you divide up the content because SUSHI puts all the FSH in one big bucket
- So, whatever is easiest, e.g.:
 - One big .fsh file
 - Many small .fsh files
 - Arrange by item type, e.g., profiles in one file, value sets in another
 - Arrange by topic area



Visual Studio Code (VS Code) Editor

- FSH is pure text; you can use any text editor or your favorite integrated development environment (IDE)
- We suggest using Microsoft VS Code with the FSH extension
 - Text highlighting
 - Navigation (Go to definition)
 - Github integration
 - Snippets
 - Run Task (SUSHI inside VS Code and navigate to errors)
- See <https://github.com/standardhealth/vscode-language-fsh#readme>



Finding the FSH VS Code Plug-in

The screenshot shows the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar indicates the file is SD_ComorbidCondition.fsh - fhir-mCODE-ig - Visual Studio Code. On the left, the Extensions Marketplace sidebar has a search bar containing "FSH Shorthand". A red box highlights the first result, "FHIR Shorthand", which is described as FHIR Shorthand (FSH) Language Support by MITRE-Health. Below it are other extensions: "FHIR tools" by Yannick Lagger, "FHIR Converter" by Microsoft, "Kodjin FHIR Profiler" by edenlab.io, and "FSH Mapping Language" by Healex Systems. The main workspace shows the FSH code for a Comorbidity Condition resource:

```
input > fsh > SD_ComorbidCondition.fsh
  condition resource. The extension elements SHALL BE USED ONLY
  "
  20 * insert NotUsed(bodySite)
  21 * insert NotUsed(specimen)
  22 * insert NotUsed(device)
  23 * subject only Reference(CancerPatient)
  24 * subject ^definition = "The patient whose comorbidities are
  25
  26 // No Must Supports in the abstract profile
  27 /* component and component.extension MS
  28
  29
  30 Profile: ComorbiditiesElixhauser
  31 Parent: ComorbiditiesParent
  32 Id: mcode-comorbidities-elixhauser
  33 Title: "Comorbidities Elixhauser Profile"
  34 Description: "Comorbid condition checklist and optional risk
    categories as defined by the Agency for Healthcare Research and Quality"
```



Review: Four Ways to Run SUSHI

- FSH Online (for low overhead experimentation, sharing FSH)
- From the command line (typical for serious IG building)
- As a task inside VS Code (alternative to command line)
- Transparently by running the HL7 FHIR IG Publisher with /fsh directory present



Exercise

- Decide on the name, id, and canonical URL for your IG
- Initialize up a FSH project
- Create FSH file(s) using the raw FSH from the previous exercise
- Edit/view in VS Code
- Compile with SUSHI
- Add some content to the index.md file
- Produce the IG -- review the results



Let's Go Deeper



<https://www.europeanscientist.com/en/research/deep-sea-fish-living-in-near-darkness-have-evolved-incredible-vision/>

<https://www.europeanscientist.com/en/environment/the-mystery-of-the-deep-sea/>

FHIR Extensions

1. Every element includes an extension array (0..*)

Name	Flags	Card.	Type	Description & Constraints	?
Element	I		n/a	Base for all elements	
id		0..1	string	All FHIR elements must have a @value or children Unique id for inter-element referencing	
extension		0..*	Extension	Additional content defined by implementations	

2. An extension is an element (see #1)

Name	Flags	Card.	Type	Description & Constraints	?
Extension	I N		Element	Optional Extensions Element + Rule: Must have either extensions or value[x], not both	
url		1..1	uri	Elements defined in Ancestors: id, extension identifies the meaning of the extension	
value[x]		0..1	*	Value of extension	

type wildcard

Create a Simple Extension

- Use declaration "Extension:"

```
Extension: Laterality
Id: laterality-extension
Title: "Laterality Extension"
Description: "Extension for specifying the side of the body."
* value[x] 1..1
* value[x] only CodeableConcept
* value[x] from http://hl7.org/fhir/ValueSet/bodysite-laterality (required)
```

- Why no url?
 - When you use the "Extension:" keyword, the url is automatically produced from the Id and the IG's canonical URL
 - We will get into this later



Incorporating an Extension

- Every element has an extension array
- We want that array to contain our extension

New rule type: "contains":

* <extension-element> contains <extension ID> named <local-name> <cardinality>

Profile: BodyStructureWithLaterality

Parent: BodyStructure

Description: "Add the Laterality extension to BodyStructure"

* extension contains Laterality named sideOfBody 0..1

Note: You can also use an extension's canonical URL for the extension ID



Populating the Extension in an Instance

```
Instance: JaneDoeLeftEye
InstanceOf: BodyStructureWithLaterality
* patient = Reference(JaneDoe)
* location = SCT#81745001 "Structure of eye proper (body structure)"
* extension[sideOfBody].valueCodeableConcept = SCT#419161000 "Unilateral left"
```

Things to note:

- Refer to the extension as a named member of extension array
- Assignment statements must always use specific type
 - foo[x] represents a choice of data types; it is not an actual element
 - Must use: fooString, fooQuantity, fooCodeableConcept, etc.



Complex Extensions

- Use when multiple data elements group together
 - E.g., Device maintenance: (i) service type, (ii) date, (iii) performer
- Two approaches:
 - Define each sub-extension as a separate extension, then combine
 - Use if the sub-extensions are useful, separately
 - Use "local extensions"
 - Use if the sub-extension always occur together



Standalone Complex Extension Approach

```
Extension: DeviceMaintenance
Id: device-maintenance
Title: "Maintenance performed on a device"
Description: "Includes type of service performed, date, and performer"
* extension contains
  ServiceType named serviceType 1..1 and
  ServiceDate named date 1..1 and
  ServicePerformer named performer 1..1

// Repeat for each sub-extension
Extension: ServiceType
Id: service-type-extension
Title: "Service Type"
Description: "Extension for specifying type of service performed on a device."
* value[x] 1..1
* value[x] only CodeableConcept
```



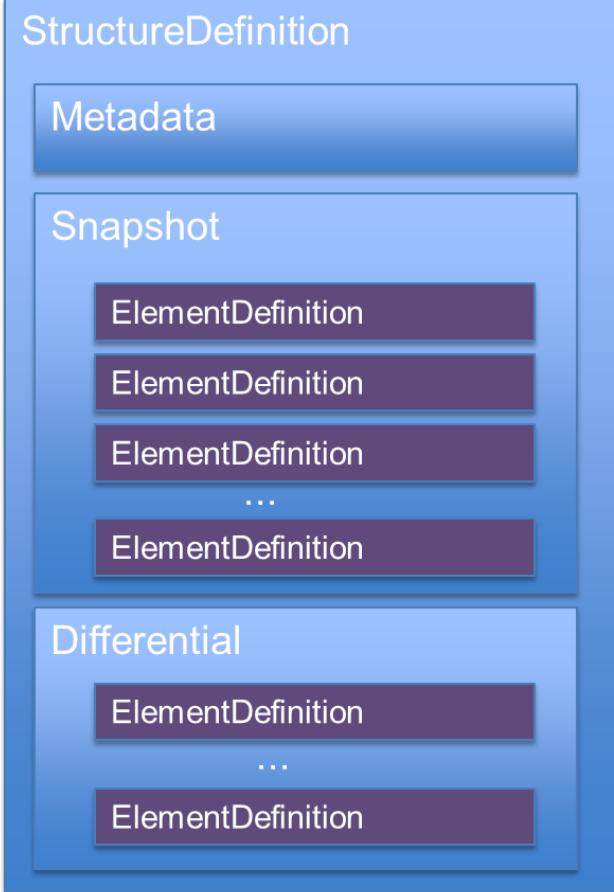
Local Extension Approach

```
Extension: DeviceMaintenance
Id: device-maintenance
Title: "Maintenance performed on a device"
Description: "Includes type of service performed, date, and performer"
* extension contains
  serviceType 1..1 and
  date 1..1 and
  performer 1..1
* extension[serviceType] only CodeableConcept
* extension[date] only date
* extension[performer] only string
```

- More compact approach
- Uses simplified form of "contains" rule



Caret Rules (^)



Elements in `StructureDefinition` that do not appear in instances (status, publisher, contact, purpose, copyright, keywords, etc.)

```
* ^copyright = "Macroserif Corporation © 2022"  
* ^keyword[0] = "foo"  
* ^keyword[1] = "bar"  
* ^publisher = "Gill Bates"  
* ^experimental = true
```

Snapshot is auto-generated

Up to now, all rules affect the differential



Extensions Exercise (take home)

- Add a respirator device profile to our IG
- Two extensions (one simple, one complex):
 - Total device hours used (0..1)
 - Maintenance events (0..*)
- Each maintenance event has three elements, all required:
 - datePerformed (date), deviceHours (integer), performer (string)



Review Solution



SUMMARY



What We've Learned

- Basic FSH language
 - Profiles, extensions, value sets
 - Did not cover code systems, logical models, invariants, mappings
 - See FSH Language Reference
(<http://hl7.org/fhir/uv/shorthand/reference.html>)
- Setting up a FSH Project
- Creating an FSH-based IG



Next Steps to Mastery



Advanced FSH (Weds AM)

- Controlling and customizing the organization of your IG
 - SUSHI configuration file (sushi-config.yaml)
- Advanced FSH language features:
 - Rule sets (simple and parameterized)
 - Soft array indexing
 - Slicing
 - Caret rules
 - Invariants
- GoFSH -- tool to convert an existing IG to FSH





So long and
thanks for all
the FSH!

