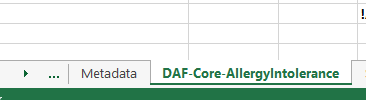
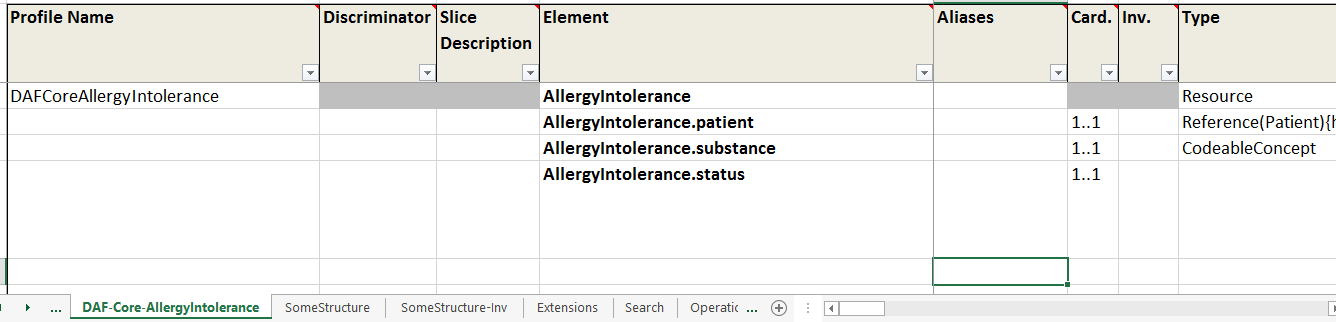
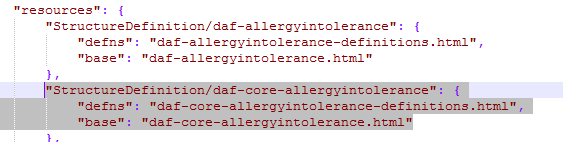
**Creation of new DAF Templates**

1. Update DAF to STU3 Names
2. Copy profile spreadsheet
3. Update name to include core
   1. allergyintolerance-daf-profile-spreadsheet.xml
   2. allergyintolerance-daf-core-profile-spreadsheet.xml
4. Edit new spreadsheet metadata tab (almost every field!)

|  |  |
| --- | --- |
| **Name** | **Value** |
| **id** | **allergyintolerance-daf-core** |
| **name** | **U.S. Data Access Framework (DAF) Core AllergyIntolerance Profile** |
| **display** | DAF-Core-AllergyIntolerance |
| **author.name** | **Health Level Seven International (FHIR - Infrastructure)** |
| **author.reference** | <http://www.healthit.gov> |
| **code** |  |
| **description** | **Defines constraints and extensions on the AllergyIntolerance resource for the minimal set of data to query and retrieve allergy information.** |
| **status** | **Draft** |
| **date** | **2016-08-01** |
| **published.structure** | DAF-Core-AllergyIntolerance |
| **version** | 0.01 |
| **extension.uri** | <http://hl7.org/fhir/daf/StructureDefinition/allergyintolerance-daf-core> |
| **introduction** |  |
| **publication.status** | STU3 |
| **notes** |  |

1. Update tab name
   1. profile name,
   2. and update elements to match Argonaut IG. Remove unused elements and bindings!



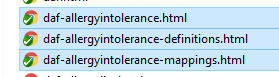
1. Locate and open daf2/daf.json \*\*\*NOTE: 1) each time you update this file need to stop and restart the publisher. 2) Put the DAF-CORE first so the valueset binding works (see below)\*\*\*\*\*
   1. Add new profile to spreadsheets\*\*\*\*
      1. 
   2. Add new structure
      1. 
2. Locate and open daf2/resources/daf.xml
   1. Add new profile



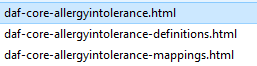
1. Add new page



1. Locate and open /daf2/pages
   1. Copy pages



* 1. Add core to title



1. Locate and open /daf2/pages/daf-core-allergyintolerance.html -

NOTE these can be autogenerated by Mapping tooling based upon the “daf-core-profiles.xlsx”

Using Title, code and example columns.

---

example: [example id]

---

* 1. Change

{% include profile-template.html %}

to

{% include core-profile-template.html %}

1. Create Intro /simple summary file ((Use argo content here using Pandoc(http://pandoc.org/try/) to convert from Wikimedia to Git markdown )
   1. Save as \_include/daf-core-[profile]-intro.md (eg daf-core-allergyintolerance-intro.md)
   2. Use the Allergies Markdown as a template template (need to work on this some more)

This profile sets minimum expectations for use of the [Type] resource to record [content] associated with a patient within the DAF FHIR IG. It identifies which core elements, extensions, vocabularies and value sets must be Supported by clients and servers. For the definition of Supported please refer to DAF FHIR IG. The data elements identified by the profile are based on ONC 2015 Edition Common Clinical Data Set(CCDS).

##### Mandatory Data Elements and Terminology

The following data-elements are mandatory (i.e data MUST be present). These are presented below in a simple human-readable explanation. Profile specific guidance and an [example](#example) are provided as well. The [\*\*Formal Profile Definition\*\*](#profile) below provides the formal summary, definitions, and terminology requirements.

\*\*Each [Type] must have:\*\*

1. a patient

2. [content]

3. a status of the allergy.

\*\*Profile specific implementation guidance:\*\*

\*

\* Additional elements from [DAF [Type] Profile](daf-[type].html) may be present.

* 1. Use argo content here
     1. using Pandoc(http://pandoc.org/try/) to convert from Wikimedia to Git markdown
  2. Note for Jekyll need spaces after text for lists etc.
  3. *Example : \_include/*daf-core-allergyintolerance-intro.md
  4. Decisions
     1. inline with formal definitions

1. Create formal summary file
   1. Save as \_include/daf-core-[profile]-summary.md (eg daf-core-allergyintolerance-summary.md)
   2. This replaces the build generated one.
   3. Markdown example
      1. Use argo content here
      2. using Pandoc(http://pandoc.org/try/) to convert from Wikimedia to Git markdown
   4. example style from Allergy Intolerance

#### Summary of the Mandatory Requirements

1. One Identification of a substance, or a class of substances, that is considered to be responsible for the adverse reaction risk in `AllergyIntolerance.code` which has an [extensible](http://hl7-fhir.github.io/terminologies.html#extensible) binding to:

- [DAF Core Substance-Reactant for Intolerance and Negation Codes](valueset-daf-core-substance.html) Value set

2. One patient reference in `AllergyIntolerance.patient`

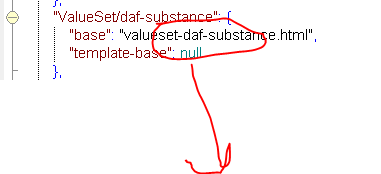
3. One status in `AllergyIntolerance.status` which has an [required](http://hl7-fhir.github.io/terminologies.html#required) binding to:

- [AllergyIntoleranceStatus](http://hl7-fhir.github.io/valueset-allergy-intolerance-status.html) Value set

1. Adding new Value Sets

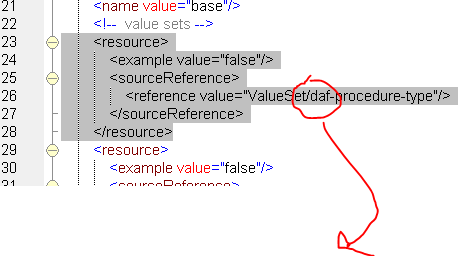
Note the daf.json file has a "specification": <http://hl7-fhir.github.io/> which should be updated to the version of the specification on which the IG is based. (not the current build as indicated here)

* 1. Link from DAF core profile home page – is part of template and will generate from the Jekyll front matter
  2. For new valuesets Create and download the ValueSet instance into “resources “ directory
     1. Can edit the description in markdown. Tool will create a narrative so I have commented out the existing text in the daf valuesets
  3. Locate and open daf2/daf.json
     1. Add structure



* 1. Locate and open daf2/resources/daf.xml
     1. Add structure

e.g.



* 1. Build will create registry of all values at the end

1. moving the existing daf ValueSets to DAF core
   1. NOTE The spreadsheet for the DAF\_Core need to be before the DAF in the json file.
   2. Make binding in DAF-Core profile spreadsheet using : “valueset-[valueset id] then refer to it in DAF profile spreadsheet using the “canonical base” http://hl7.org/fhir/daf/ValueSet/[ valueset id]

e.g

valueset-daf-ethnicity

http://hl7.org/fhir/daf/ValueSet/daf-ethnicity

Right now this does not work. otherwise it will not work!

1. Examples
   1. Name all examples “example”
      1. Instance id
      2. Put in the wrapper filesnames and front matter
      3. Add profile data for DAF-Core
   2. Use the build generated artifacts to produce three html pages for the build generated narrative connect and xml and json versions
      1. Three template files in \_include directory
         1. core-example-template.html
         2. core-example-json-template.html
         3. core-example-xml-template.html
      2. three wrapper files for each example in pages directory that provide the front matter ( these will be prepopulated in directory)
         1. names are daf-core-[profile]-example.html
         2. names are daf-core-[profile]-example-json.html
         3. names are daf-core-[profile]-example-xml.html
         4. rdf for later
   3. create examples and upload to resources directory ( will use mapper to create each of these for STU3
      1. save as daf-core-[profile]-example.xml
      2. use STU3 schema to validate
      3. Add profile tag for DAF-Core
      4. Can use Argo data for inspiration
   4. add to DAF.json

"AllergyIntolerance/example": {

"base": "allergyintolerance-example.html",

"template-base" : null

}

* 1. add to DAF.xml

<resource>

<example value="true"/>

<name value="Allergy Intolerance Example"/>

<sourceReference>

<reference value="AllergyIntolerance/example"/>

</sourceReference>

<exampleFor>

<reference value="StructureDefinition/daf-core-allergyintolerance"/>

</exampleFor>

</resource>

1. Extensions
   1. Created extension in each spreadsheet
   2. In Metadata page Make the extension.uri the something like:

|  |  |
| --- | --- |
| **extension.uri** | [http://hl7.org/fhir/daf/StructureDefinition/us-core-](http://hl7.org/fhir/daf/StructureDefinition/patient) |

* 1. Reference the extension as Extension{#code} where ‘code’ is from the 1st column in the extension tab

NOTE this is not working and other combo are not working either GG is looking into

The generated value set is also not being published either

* 1. Add the generated resources to both the DAF.xml and DAF.json files

e.g.:

"StructureDefinition/us-core-birth-sex": {

"defns": "us-core-birth-sex-definitions.html",

"base": "us-core-birth-sex.html"

},

And

<resource>

<example value="false"/>

<sourceReference>

<reference value="StructureDefinition/us-core-birth-sex"/>

</sourceReference>

</resource>