FHIR Spreadsheet Profile Authoring

The FHIR Build process leverages Excel spreadsheets saved in XML format. It should also work using OpenOffice, though this has not been rigorously tested. The process defined below is a short-term (for the next DSTU cycle) process for doing internal HL7 profile development. While still technically supported by the HL7 IG publisher, it is not supported in other publishing frameworks and there are Profile Tooling alternatives that are superior. IG authors are encouraged to migrate away from using spreadsheets to author profiles as support for this capability is not guaranteed in the long term.

Contents

- Decide on resource-specific or "general" profile
- Set up resource-specific profile
- Set up "general" profiles
- Specify Profile Metadata
- Profile Content Options
- Define Extensions
- Define Search Criteria
- Define structures
  - Referencing "extra" elements
  - Slicing

Steps

Decide on resource-specific or "general" profile

Resource-specific profiles are those that profile only one resource or that primarily focuses on a single resource. They will be published in the Resources tab for the resource. "General" resources show up in the list of profiles off the implementer page, but not anywhere else. (As a rule, resource-specific is better because it provides two ways to discover the content.)

Set up resource-specific profile

1. Copy the template-profile-spreadsheet.xml file from the build/source/templates directory to the same folder as the resource. The name should be "[ResourceName]-[some qualifier]-profile-spreadsheet.xml" where [some qualifier] is a short 1-2 word descriptor that makes the name unique within the folder.
2. In addition, consider copying the template-introduction.xml and template-notes.xml files from the build/source/templates directory if your profile will need to have custom HTML in front of or after the rendering of the profile itself. Most profiles should at least have an introduction. The file-names should be "[ResourceName]-[same qualifier]-[introduction|notes].xml".
3. Edit the spreadsheet of the "resource" to point to the new profile on the Profiles tab. (If it's an older resource that doesn't have a Profiles tab, or if the Profiles tab doesn't have all of the columns discussed below, you may need to add the tab or update it to align with the content specified in the build/source/templates/template-spreadsheet.xml.) Populate the columns as follows:
   a. Name: The name used to label and reference the profile
   b. Description: Description of the profile shown when it's listed in profileList.html (one sentence or so)
   c. Source: The name of the profile spreadsheet xml file
   d. Filename: The name of the profile.xml file that should be generated from the spreadsheet. This should generally be ResourceName- [same qualifier].xml
   e. Type: This should always be "spreadsheet" when using the spreadsheet method.

   Note: In the future, the Name and Description columns might be removed as they are redundant with information already found in the profile spreadsheet.

Set up "general" profiles

1. Copy the template-profile-spreadsheet.xml file from the build/source/templates directory to the /build/profiles folder. The name should be "[some qualifier]-spreadsheet.xml" where [some qualifier] is a short 1-2 word descriptor that makes the profile unique within the folder.
2. In addition, consider copying the template-introduction.xml and template-notes.xml files from the build/source/templates directory if your profile will need to have custom HTML in front of or after the rendering of the profile itself. Most profiles should at least have an introduction. The file-names should be "[same qualifier]-[introduction|notes].xml".
3. Edit the profiles section of the source/fhir.ini file to include a reference in the form [id]=[file], where:
   a. [id] is a unique identifier-like name for the profile
   b. [file] is "profiles/[same-qualifier]-spreadsheet.xml"

Specify Profile Metadata

Open the profile spreadsheet and select the Metadata tab and populate the value column as as described in FHIR Spreadsheet Authoring.
Profile Content Options

Within a given profile, you can do any of three things:

- Define extensions for resource and/or data type profiles
- Define search criteria for a particular resource
- Define structures (constraints & extensions on a particular resource or data type)

Refer to the FHIR Methodology for guidance on how to determine what should be in a single profile vs. what is best captured in separate profiles.

Define Extensions

Use the Extensions tab and populate the columns as described in FHIR Spreadsheet Authoring.

Define Search Criteria

Search criteria for extensions are defined exactly the same way they are for resources except that the search tab must be specific to a particular structure. The search tab needs to be named [StructureName]-search. The instructions for populating the search columns can be found in FHIR Spreadsheet Authoring.

Define structures

Each structure represents the constraints and extensions declared for a particular use of a specified resource or data type. Each structure must be defined on a separate tab and have a unique name within the profile. On each structure tab, the second row of the tab identifies both the resource or data type being profiled as well as the "structure name". The structure name appears in cell A2 (Profile Name). The resource or data type name appears in cell B2 (Element). In addition, if a structure is intended to be based on another profile structure, this can be specified in the same row in the "Profile" column (usually H2).

The structure name should be upper camel case (no spaces and no characters other than alphabetic, numeric or dashes). The format for Profile is the base URL for the profile + "/#{StructureName}" within the referenced profile. Obviously if creating a profile based on another profile, both must refer to the same resource or data type.

All profiles are authored as "differential" profiles. This means that only those rows and elements that are different from the underlying resource or data element (or specified profile structure on that resource or data element) need to be specified. However, it may be convenient to list all data elements, cardinalities and data types from the underlying resource or profile, just to have a record of what elements are possible. (Note that these should be kept up-to-date if the underlying resource or data type is adjusted). This can be done by copying and pasting the relevant columns from the Resource or Data Type definition spreadsheet.

Most of the columns in the structure definition tab are the same as those used for defining resources. Details on the meaning of each column can be found in FHIR Spreadsheet Authoring. However, there are some additional columns and processes used when profiling structures. These are documented below. Finally, there are rules about what sort of changes can be made when profiling resources and data types. These rules can be found here.

Referencing "extra" elements

When defining a resource, the only elements that appear in the spreadsheet are those that are specific to the resource. If you reference a data type that contains elements, you don't identify the elements that appear inside the data type, only the element that references the data type. In addition, some elements that are implicitly part of a resource (narrative, extension, modifierExtension, etc.) are not mentioned in resource spreadsheets at all. They're automatically assumed to be included at the appropriate position within the resource automatically.

However, when defining profiles on resources and data types, there may be a need to constrain or extend a data element inside a referenced data type or one of the default elements (e.g., requiring that "text" be present within a CodeableConcept or asserting that narrative is 1..1 for a particular resource.

If the set of constraints or extensions on a data type are expected to be common, defining a separate structure and simply referencing it as a profile may be best. However, it is also possible to reference such elements inline. For example, to declare a constraint on the "use" element from the ContactPoint data type within Patient.telecom, the element name would be Patient.telecom.use. Similarly, it would be possible to make a constraint on Patient.narrative, even though that isn't explicitly declared in the defining resource spreadsheet.

Slicing

Slicing is the process by which multiple distinct constraints can be specified on repeating elements. For example, for the repeating Patient.telecom element, you might want to assert that you want 1..5 phone numbers 0..1 fax numbers, 0..2 email addresses and 0..0 web addresses. There are two types of slicing - slicing extensions & modifierExtension; and slicing other elements.

When slicing, it's essential that a given repetition can only correspond to a single slice and that this be verifiable. To accomplish this, when an element needs to be sliced, it's necessary to first declare the "discriminator" or set of discriminators that will allow the repetitions to be disambiguated. A discriminator is a path to an element inside the repeating element that will have a fixed value for each "slice". For example, in the Patient.telecom example above, the discriminator would be "system" because that's the element that would be fixed within each slice (phone/fax/email/url).

For most groups of slices, the discriminator for the group must be declared on the first slice in the group by specifying the relative path in the "discriminator" tab. Multiple discriminators may be specified by separating each discriminator in the list with a space.
There is one exception to the need to declare a discriminator. When slicing extensions or modifier extensions, the default discriminator is always "url" which does not need to be declared.

In addition, a profile may need to indicate whether order of the slices is important. Finally, a profile can define the "rules" for the slice, indicating whether the set of slices is considered to be complete (closed) or whether additional repetitions that don't meet any of the slice patterns are also permitted, and if so, whether those additional repetitions may appear anywhere (open), or whether they must appear after the declared slices (openAtEnd). The profile might also declare whether slices in an instance must appear in the order specified.

Using the spreadsheet mechanism, the slicing path, ordered nature and rules are all specified in a single column, separated by "|".

Finally, each "slice" must have a unique name. This name appears in the specification and allows referencing the constraints associated with a particular slice. (It also allows slicing slices in subsequent profiles.) This name is specified in the Profile Name column. It can be any string, though is generally limited to only one or two words.