Module 6 Exercise 1 - Adding a Custom DTL for Post-FHIR Transform

Modifying the DTL Transformation Code

1. Open up the Patient Resource DTL: **HS.FHIR.DTL.SDA3.vR4.Patient.Patient**. This is accessible by going to the System Management Portal -> Interoperability -> Build -> Data Transformations and clicking Open

2. Click on **Save As** to copy the **HS.FHIR.DTL.SDA3.vR4.Patient.Patient** class to a new class called: **HS.Local.FHIR.DTL.SDA3.vR4.Patient.Patient**. The naming is important here as the FHIR base code will give this custom class precedence over the out-of-the-box transform.

Save Local Patient Resource DTL

3. Create a sub-transform to do the work of mapping the extension.

From the Data Transformation Builder, open **HS.FHIR.DTL.SDA3.vR4.Address.Extension** as a model of an extension mapping.

3-1. Save As: **HS.Local.FHIR.DTL.SDA3.vR4.Patient.AgeExtension**

The location of this is not as sensitive as the code you write will reference this class directly.

3-2. Click on the **Transform** tab and update the following settings:

Setting Name	Value
Source Class	HS.SDA3.Patient
Target Class	HS.FHIR.DTL.vR4.Model.Base.Extension

Save As Age Extension DTL

4. Following the existing code in the DTL as a model, modify the values to follow the same pattern, but map out the **UDS Plus Age Extension** as detailed in the **Task** section above.

Example: This image shows an example with the age (target.extension(1).valueQuanity.value) hard-coded rather than calculated. You can do that as a first step.

Code added in Age Extension DTL

Save and Compile when done.

5. Open the Transformation for **HS.Local.FHIR.DTL.SDA3.vR4.Patient.Patient**

> Be very careful when editing this code. You are keeping the Patient Resource mapping intact while adding in a call-out to the sub-transform for the Age Extension you just created.

Follow the model in the existing class to call the sub-transform. In the example, the code block was added right before the **birthTime** mapping since the **source.birthTime** value is used to calculate the age.

Code block added in Patient Resource DTL

Code Block - backend

```
<comment>
<annotation>CNR: Hardcoding age extension to try things out</annotation>
</comment>
<assign value='target.extension.Count()+1' property='extIndex' action='set' />
<assign value='##class(HS.FHIR.DTL.vR4.Model.Base.Extension).%New()'
property='extTemp' action='set' />
<assign value='aux("transformer").GetDTL(source,
"HS.Local.FHIR.DTL.SDA3.vR4.Patient.AgeExtension")' property='DTL' action='set' />
<if condition='DTL&apos;=""' >
<true>
<assign value='$classmethod(DTL,"Transform", source, .extTemp, .aux)'
property='status' action='set' />
<if condition='extTemp&apos;=""' >
<true>
<assign value='"http://fhir.org/guides/hrsa/uds-plus/StructureDefinition/uds-plus-
age-extension"' property='extTemp.url' action='set' />
<assign value='extTemp' property='target.extension' action='set' key='extIndex' />
</true>
</if>
</true>
</if>
```

Save and Compile When done.

Test The Changes

No changes to the existing Production are needed for the local DTL code changes to take effect

1. Repeat the steps you did in **Module 6 - Exercise 1** to drop an HL7 message in **iris-container/data/durable/module6-exercise1-inbound**.
2. Check in the **Message Trace** to ensure there were no errors. Review the output FHIR bundle or transaction to confirm that the update took effect.
3. Troubleshoot the mapping as needed.

Alternate Path

You can also still call the full original "standard" DTL class without having to maintain a full copy of its code in your new custom DTL. So, for instance, if you want to only add one change, you can do the following which executes the standard class and then add your change afterwards in the DTL:

```
<assign property='status'
value='##class(HS.FHIR.DTL.Util.Execute).ExecuteStandardClass($classname(),
source, .target, .aux)' />
```

1. Give this a try and test if you are feeling brave.

FHIR Validation

1. Go to the FHIR validator at FHIR Validator.

1-1. Select the **Options** menu at the top. Then select the **FHIR Version (4.0.1)** and **Implementation Guides (hl7.fhir.us.core version 3.1.0)** and select **Add**.

1-2. Search for `uds` in the Implementation Guides. Select **fhir.hrsa.uds-plus** and select the **current** version. Click **Add**.

Now you are validating against the UDS+ IG as well! Isn't FHIR fun?

FHIR Validator with UDS

2. Click back on **Validate** and paste the updated JSON into the **Enter Resource** window.

3. Select the **Validate** button at the bottom of the screen and review the errors and warnings.

4. Search through the errors. You should find a few that are specific to the `uds-plus-age-extension`.

UDS Age Extension Error

5. To experiment, you can update the uds age extension directly in the window and validate again. This is an easy way to see what the correct mapping is supposed to be.

Summary: We muscled in a change just to see it take effect and start the testing and validation cycle. There's work ahead to complete the full mapping and make sure it conforms, but hopefully this exercise has given you a good idea of how to accomplish this task using the tools available.

To Save the Production:

You can copy the contents of the **HS.Local.FHIR.DTL.SDA3.vR4.Patient.Patient** and the **HS.Local.FHIR.DTL.SDA3.vR4.Patient.AgeExtension** class to the **FHIR-UDS-TRAINING/src/FHIRDEMO** folder via Cache Studio or Visual Studio Code.