

Node Visibility

GELLO code will be used to display an **ELEMENT** node or not. GELLO for node visibility can also be applied for **CLUSTERS** or data groups. This is important for:

[1 Apple archetype example](#)

- managing computer screen real estate
- developing 'smart forms' that show and hide nodes depending on value(s) entered elsewhere in the archetype tree
- excluding non-visible nodes from the resulting HL7 message

Apple archetype example

Lets go back to the apple archetype. The example here will add GELLO node visibility to 'Total Cost'. When values have been entered for the higher nodes 'Cost' and 'Number', and the **isCalculated** attribute GELLO has calculated a value , we may decide its fair enough to make the node visible. Another example where node visibility is useful has been in the pathology domain, where calculation of certain values need only be done and displayed where certain other fields have been entered, eg the calculation of electronic glomerular filtration rate (eGFR).

Open CEN1-Apple.v1.xml again. Set the **VisibleAtStart** metadata tag for 'Total Cost' on the RHS to false. We will use the **isVisible** attribute.

Here's a walk through of the GELLO required:

1. Usual beginning - already done:

```
Context CEN_Apple_v1::ArchetypeRoot
```

2. See if 'Cost' and 'Number' have values. This time we'll just retrieve them as the general type /class Observation:

```
Let cost: Observation = template.Apple.Cost
Let number: Observation = template.Apple.Number
```

3. Set up the test:

```
let VisA:Boolean =
    if cost.oclIsDefined()
    then
        If cost.value.oclaType(PQ).value > 0.0
        then
            true
        else
            false
        endif
    else
        unknown
    endif
```

```
let VisB:Boolean =  
    if number.oclIsDefined()  
    then  
        If number.value.oclAsType(Real) > 0.0  
        then  
            true  
        else  
            false  
        endif  
    else  
        unknown  
    endif
```

4. Run the test and return the result (as a boolean):

```
Let result : Boolean =  
    If VisA = true and VisB = true then true  
    else false  
endif
```

```
result
```

5. Save your work