

8.3. Constrained Replacement Slots

Overview

The value that can be used to replace a slot in an expression template may be constrained by an expression constraint, value list constraint or range constraint (depending on the type of replacement slot). The following types of slot constraint may be used:

- Expression constraints may be used to constrain replacement slots of type **id** and **scg**,
- Value list constraints may be used to constrain replacement slots of type **tok**, **str**, **int**, **dec** and **bool**.
- Range constraints may be used to constrain replacement slots of type **int** and **dec**.

The following sections provide examples of each of these types of replacement constraints.

Expression Constraints

Expression constraints may be used to constrain the values permitted to replace a slot of type **id** or **scg** in an expression template.

For example, the slot in the following expression template can only be replaced by a concept reference that is a descendant or self of **Anatomical or acquired body structure**.

```
71388002 |Procedure| :  
{ 260686004 |Method| = 312251004 |Computed tomography imaging action| ,  
  405813007 |Procedure site - Direct| =  
    [[+id (<< 442083009 |Anatomical or acquired body structure| )]] }
```

Because the concept **Shoulder region structure** is a descendant or self of **Anatomical or acquired body structure**, it may be used to replace the slot in the above expression template, resulting in the following expression.

```
71388002 |Procedure| :  
{ 260686004 |Method| = 312251004 |Computed tomography imaging action| ,  
  405813007 |Procedure site - Direct| = 16982005 |Shoulder region structure| }
```

However, the concept **Nonspecific site** may not be used as a replacement (because it is not a descendant or self of **Anatomical or acquired body structure**).

Similarly, in the following expression template, the slot can be replaced by any expression that is a descendant of **Anatomical or acquired body structure**.

```
71388002 |Procedure| :  
{ 260686004 |Method| = 312251004 |Computed tomography imaging action| ,  
  405813007 |Procedure site - Direct| =  
    [[+scg (<< 442083009 |Anatomical or acquired body structure| )]] }
```

Value List Constraints

Value list constraints can be used to constrain the possible values that may replace slots of type **tok**, **str**, **int**, **dec** and **bool**.

For example, the following expression template uses a value list constraint to specify the possible definitionStatus tokens that may be used. The slot in this expression template may be replaced with either the token "<<<" or the token "===".

```
[[+tok (<<< ===)]] 281647001 |Adverse reaction (disorder)| :  
246075003 |Causative agent (attribute)| = [[+id]]
```

If the causative agent is assigned the value **Amoxicillin** then the following two expressions can be generated (depending on the token selected to replace the first slot).

```
<<< 281647001 |Adverse reaction (disorder)| : 246075003 |Causative agent (attribute)| = 372687004 |Amoxicillin|
```

```
=== 281647001 |Adverse reaction (disorder)| : 246075003 |Causative agent (attribute)| = 372687004 |Amoxicillin|
```

The expression template below uses a value list constraint to constrain the value of the slot to the string "PANADOL", the string "TYLENOL", or the string "HERRON".

```
322236009 |Paracetamol 500mg oral tablet| : 774167006 |Product name (product name)| = [[+str ("PANADOL" "TYLENOL" "HERRON")]]
```

Depending on the value used to replace the slot, any of the following three expressions may be generated.

```
322236009 |Paracetamol 500mg oral tablet| : 774167006 |Product name (product name)| = "PANADOL"
```

```
322236009 |Paracetamol 500mg oral tablet| : 774167006 |Product name (product name)| = "TYLENOL"
```

```
322236009 |Paracetamol 500mg oral tablet| : 774167006 |Product name (product name)| = "HERRON"
```

In the next example, a value constraint is used to constrain a replacement slot of type **int**. In this case, the slot may be replaced by the integer values 10, 20 or 30.

```
417720003 |Zinc 25mg oral capsule| :  
{ 1142142004 |Has pack size (attribute)| = [[+int (#10 #20 #30)]],  
  774163005 |Has pack size unit (attribute)| = 428641000 |Capsule| }
```

The possible expressions that may be generated by filling the slot are:

```
417720003 |Zinc 25mg oral capsule| :  
{ 1142142004 |Has pack size (attribute)| = #10,  
  774163005 |Has pack size unit (attribute)| = 428641000 |Capsule| }
```

```
417720003 |Zinc 25mg oral capsule| :  
{ 1142142004 |Has pack size (attribute)| = #20,  
  774163005 |Has pack size unit (attribute)| = 428641000 |Capsule| }
```

```
417720003 |Zinc 25mg oral capsule| :  
{ 1142142004 |Has pack size (attribute)| = #30,  
  774163005 |Has pack size unit (attribute)| = 428641000 |Capsule| }
```

Range Constraints

Range constraints can be used by slots of type **int** or **dec** to constrain the permitted replacement values. Range constraints may specify the minimum permitted value, the maximum permitted value, or both. By default, ranges are inclusive (that is, the range includes the stated minimum and maximum values). However exclusive ranges (in which the stated minimum or maximum is not itself allows as a value) may be specified.

For example, the following expression template allows any pack size to be used that is between 20 and 30 capsules (inclusive).

```
417720003 |Zinc 25mg oral capsule| :  
{ 1142142004 |Has pack size (attribute)| = [[+int (#20..#30)]],  
  774163005 |Has pack size unit (attribute)| = 428641000 |Capsule| }
```

To specify exclusive minimum values, a greater than sign ('>') is placed before the minimum value, and to specify exclusive maximum values, a less than sign ('<') is placed before the maximum value. For example, the following expression template allows any pack size that is between 20 and 30 capsules (exclusive). This range constraint is equivalent to an inclusive range of 21..29.

```
417720003 |Zinc 25mg oral capsule| :
{ 1142142004 |Has pack size (attribute)| = [[+int (>#20..<#30)]],
  774163005 |Has pack size unit (attribute)| = 428641000 |Capsule| }
```

Multiple ranges may be specified in a single range constraint. When multiple ranges are specified, these should be interpreted as alternative ranges. For example, the expression template below permits any pack size between 10 and 20 capsules (inclusive), or between 30 and 40 capsules (inclusive). This template does not, for example, permit a pack size between 21 and 29 capsules.

```
417720003 |Zinc 25mg oral capsule| :
{ 1142142004 |Has pack size (attribute)| = [[+int (#10..#20 #30..#40)]],
  774163005 |Has pack size unit (attribute)| = 428641000 |Capsule| }
```

When a minimum value is required with no maximum, it is possible to omit the maximum value in a range constraint. For example, the expression template below permits any pack size greater than or equal to 20 capsules.

```
417720003 |Zinc 25mg oral capsule| :
{ 1142142004 |Has pack size (attribute)| = [[+int (#20..)]],
  774163005 |Has pack size unit (attribute)| = 428641000 |Capsule| }
```

Similarly, it is also possible to state a maximum without a minimum constraint. For example, the expression template below permits any pack size less than or equal to 20 capsules. It should be noted that without a concept model rule to ensure that all pack sizes are greater than zero, this template will, by default, allow a pack size of zero or a negative pack size to be defined.

```
417720003 |Zinc 25mg oral capsule| :
{ 1142142004 |Has pack size (attribute)| = [[+int (..#20)]],
  774163005 |Has pack size unit (attribute)| = 428641000 |Capsule| }
```