

not (a ~ b)

```
define "Not Equivalent Clause 1":  
  [Condition: "test-vs"] C  
  |  
  | where not (C.clinicalStatus ~ "Condition Active")
```



```
"where": {  
  "localId": "24",  
  "locator": "24:9-24:57",  
  "type": "Not",  
  "operand": {  
    "localId": "23",  
    "locator": "24:19-24:57",  
    "type": "Equivalent",  
    "operand": [  
      {  
        "name": "ToConcept",  
        "libraryName": "FHIRHelpers",  
        "type": "FunctionRef",  
        "operand": [  
          {  
            "localId": "21",  
            "locator": "24:20-24:35",  
            "path": "clinicalStatus",  
            "scope": "C",  
            "type": "Property"  
          }  
        ]  
      },  
      {  
        "localId": "22",  
        "locator": "24:39-24:56",  
        "name": "Condition Active",  
        "type": "ConceptRef"  
      }  
    ]  
  }  
}
```

This requires two checks for full coverage

- One for **not**
- One for **a ~ b**

a !~ b

both are translated the same

```
define "Not Equivalent Clause 2":  
  [Condition: "test-vs"] C  
  |  
  | where C.clinicalStatus !~ "Condition Active"
```



```
"where": {  
  "localId": "33",  
  "locator": "28:9-28:52",  
  "type": "Not",  
  "operand": {  
    "localId": "32",  
    "locator": "28:15-28:52",  
    "type": "Equivalent",  
    "operand": [  
      {  
        "name": "ToConcept",  
        "libraryName": "FHIRHelpers",  
        "type": "FunctionRef",  
        "operand": [  
          {  
            "localId": "30",  
            "locator": "28:15-28:30",  
            "path": "clinicalStatus",  
            "scope": "C",  
            "type": "Property"  
          }  
        ]  
      },  
      {  
        "localId": "31",  
        "locator": "28:35-28:52",  
        "name": "Condition Active",  
        "type": "ConceptRef"  
      }  
    ]  
  }  
}
```

This ALSO requires those two checks for full coverage, even though that need is effectively hidden

Not Equivalent (!~)

```
BMIEncounter.class !~ "virtual"
```

```
{
  "localId": "43",
  "locator": "195:35-195:65",
  "resultTypeName": "{urn:hl7-org:elm-types:r1}Boolean",
  "type": "Not",
  "operand": {
    "localId": "42",
    "locator": "195:35-195:65",
    "resultTypeName": "{urn:hl7-org:elm-types:r1}Boolean",
    "type": "Equivalent",
    "operand": [
      {
        "localId": "40",
        "locator": "195:35-195:52",
        "resultTypeName": "{urn:hl7-org:elm-types:r1}Code",
        "name": "ToCode",
        "libraryName": "FHIRHelpers",
        "type": "FunctionRef",
        "operand": [
          {
            "path": "class",
            "scope": "BMIEncounter",
            "type": "Property"
          }
        ]
      }
    ]
  },
},
{
  "localId": "41",
  "locator": "195:57-195:65",
  "resultTypeName": "{urn:hl7-org:elm-types:r1}Code",
  "name": "virtual",
  "type": "CodeRef"
}
]
```

Not clause gets a localId of 43 that will evaluate to **true** if there is a test case where **BMIEncounter.class is NOT EQUIVALENT** to "virtual"

Equivalent clause gets a localId of 42 that will evaluate to **false** unless there is a test case where **BMIEncounter.class IS EQUIVALENT** to "virtual"

It was impossible to see this in the highlighting because the **NOT** clause was evaluating to **true** which resulted in the clause being highlighted for coverage.

In fqm-execution 1.3.3:

- Highlighting does **NOT** change
- Calculation percentage **DOES** change
 - o The Equivalent clause is not counted in the coverage percentage calculation

Not Equal (!=)

We were not seeing any issues with coverage percentage calculation for **Not Equal** because **Equal** does not receive a localId in the current version of the cql-to-elm translator:

`C.id != 'notId'`

```
"where": {
  "localId": "23",
  "locator": "24:9-24:29",
  "type": "Not",
  "operand": {
    "locator": "24:15-24:29",
    "type": "Equal",
    "operand": [
      {
        "name": "ToString",
        "libraryName": "FHIRHelpers",
        "type": "FunctionRef",
        "operand": [
          {
            "localId": "21",
            "locator": "24:15-24:18",
            "path": "id",
            "scope": "C",
            "type": "Property"
          }
        ]
      },
      {
        "localId": "22",
        "locator": "24:23-24:29",
        "valueType": "{urn:hl7-org:elm-types:r1}String",
        "value": "notId",
        "type": "Literal"
      }
    ]
  }
}
}
```

Not clause gets a localId of 23 that will evaluate to **true** if there is a test case where **C.id** is **NOT EQUAL** to “notId”

Equal clause does **NOT** get a localId; therefore, the clause is not included in coverage percentage calculation

There is currently no issue with != in fqm-execution. However, the lack of a localId on the Equal clause is an issue in the translator that we are anticipating a fix for, so **fqm-execution 1.3.3** accounts for that change whenever it happens.

In fqm-execution 1.3.3 AND with current translator:

- Highlighting does **NOT** change
- Calculation percentage does **NOT** change