

Table: resultVer.TestRanges Table Info Fields Maps

Field Name	Datatype	Column #	Required	Unique
ID	%Library.BigInt	1	Yes	Yes
TestCode	%Library.Integer	2	No	No
MinRange	%Library.Double	3	No	No
MaxRange	%Library.Double	4	No	No

Paso 2: Necesitamos crear una función personalizada para obtener el rango de referencia, para lo que creé la función `GetReferenceRange ()` `ClassMethod` en la clase de utilidad. Ten en cuenta que esta clase debe extenderse desde `EnsRule.FunctionSet`

```
// Get reference ranges from TestRanges table against test code
Debug this method
ClassMethod GetReferenceRange(TestCode As %String) As %String
{
    &sql(SELECT MinRange, MaxRange into :minRange,:maxRange
    FROM resultVer.TestRanges where TestCode = :TestCode)
    set minRange = $DECIMAL(minRange,3)
    set maxRange = $DECIMAL(maxRange,3)

    Set refRange = minRange _ "-" _ maxRange
    //Set refRange = "Test"
    Return refRange
}
```

Necesitamos crear otra función para establecer el aviso de valor anormal (Abnormal Flag), para lo que creé la función `SetAbnormalFlag ()` `ClassMethod` en la clase de utilidad

```
// Get Abnormal flag based on Test Code and Result
Debug this method
ClassMethod SetAbnormalFlag(TestCode As %String, Result As %Decimal) As %String
{
    &sql(SELECT MinRange, MaxRange into :minRange,:maxRange
    FROM resultVer.TestRanges where TestCode = :TestCode)
    //Set Flag to L in case of Low Range
    if Result < minRange
    {
        Return "L"
    }
    //Set Flag to H in case of High Range
    elseif Result > maxRange
    {
        Return "H"
    }
    //Return empty strign in case of normal result
    Return ""
}
```

Paso 3: Ahora usaremos estas funciones en Transformación de datos (UpdateReferenceRangesDTL). Necesitamos usar foreach loop contra los segmentos repetidos:

#	Action	Condition	Property	Value	Key / Transform
1	for each ▼		source.{PIDgrpgrp() }		k1
2	for each ▼		source.{PIDgrpgrp(k1).ORCgrp() }		k2
3	for each ▼		source.{PIDgrpgrp(k1).ORCgrp(k2).OBXgrp() }		k3
4	set		target.{PIDgrpgrp(k1).ORCgrp(k2).OBXgrp(k3)...	##class(resultVer.Utility).GetReferenceRang...	""
5	set		target.{PIDgrpgrp(k1).ORCgrp(k2).OBXgrp(k3)...	##class(resultVer.Utility).SetAbnormalFlag(...	""
6	endeach				
7	endeach				

Action:

Property:

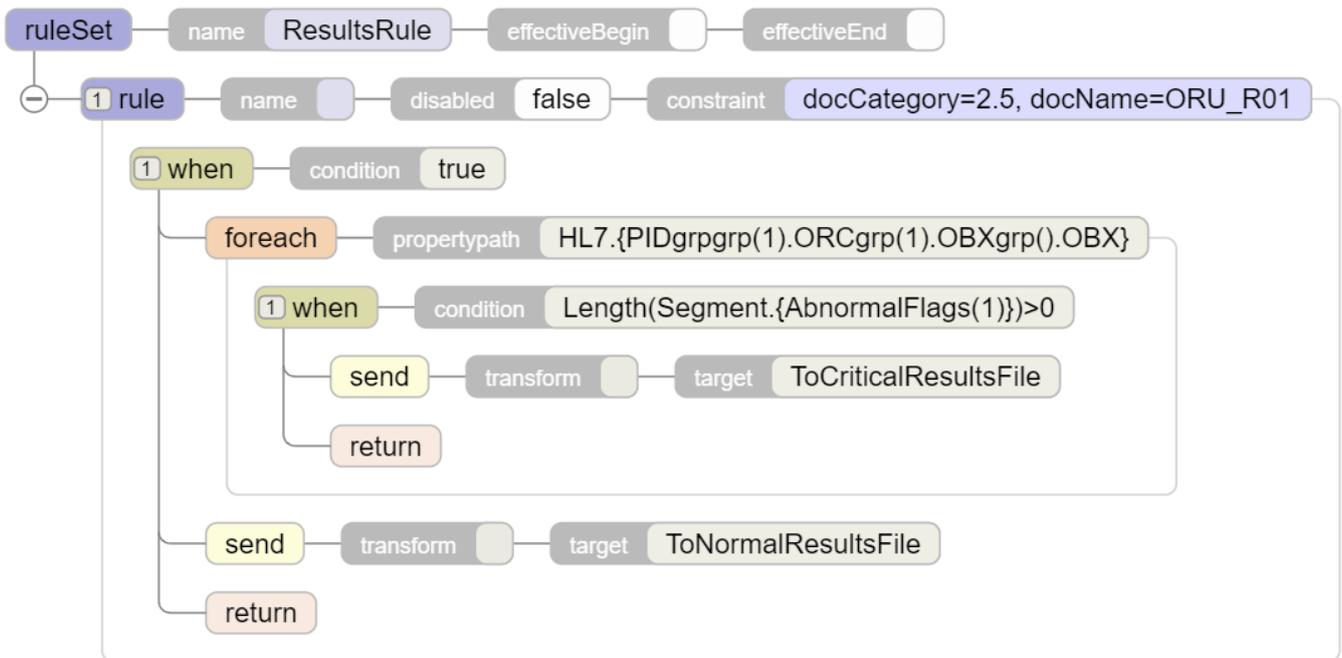
Property whose value will be set. Double-clicking on a target property in the diagram will place that property in this field.

Value:

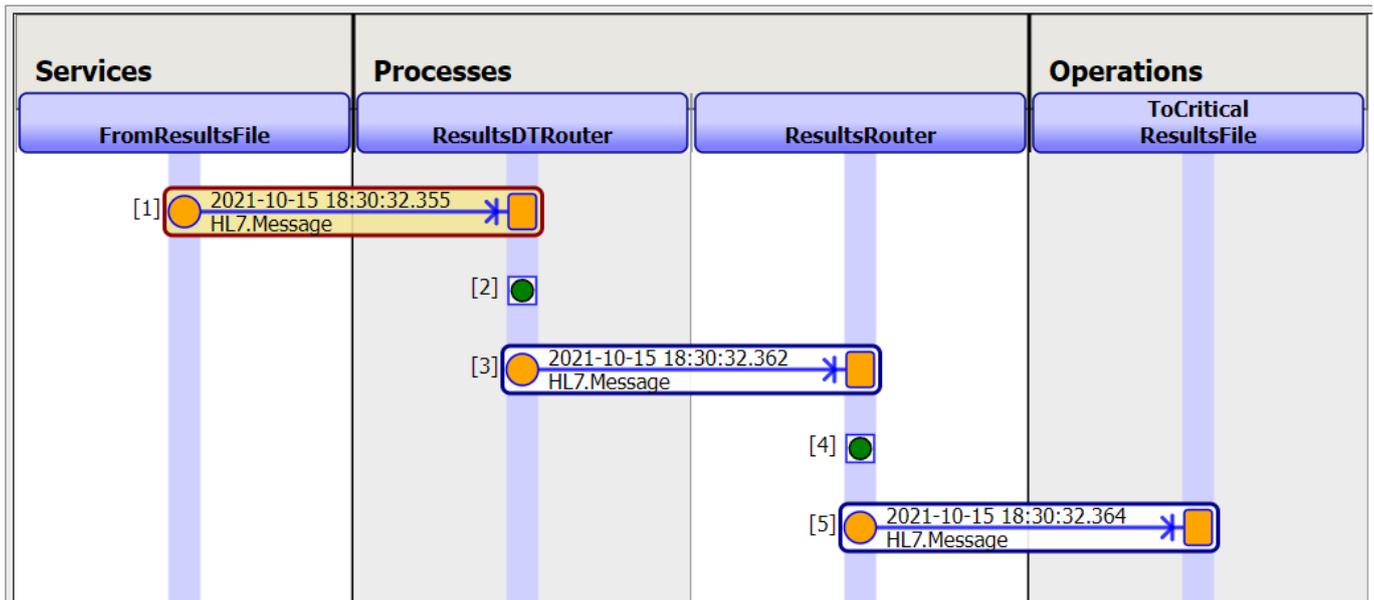
Value to assign to the property. Double-clicking on a property in the diagram will place that property in this field.

Ten en cuenta que para obtener el aviso de valor anormal (Abnormal Flag), tenemos que pasar OBX: Valor de observación (Resultado) junto con OBX: Identificador de observación (Código de prueba).

Paso 4: Transformaremos el mensaje de la regla de negocio "ResultsDTRule" y enviaremos el mensaje a la regla de negocio "ResultsRule" otra vez, para enviar el mensaje a la operación "ToCriticalResultsFile" en caso de que se establezca el aviso de valor anormal (Abnormal Flag); de lo contrario, se enviará a la operación "ToCriticalResultsFile":



Seguir el Visual Trace debajo de Visual Trace es para que el mensaje se envíe correctamente a la operación "ToCriticalResultsFile".



Gracias

[#Reglas empresariales](#) [#DTL](#) [#HL7](#) [#SQL](#) [#Ensemble](#) [#InterSystems IRIS for Health](#)
[Compruebe la aplicación relacionada en InterSystems Open Exchange](#)

URL de fuente: <https://es.community.intersystems.com/post/c%C3%B3mo-actualizar-el-rango-de-referencia-de-resultados-y-el-aviso-de-valor-anormal-usando-hl7>