

# User Manual

## Introduction

This manual describes how the content validation test tool must be used.

To run a custom made content validation two steps needs to be done.

Create a valid content validation with the IHE-RO Content Validation Creator, run the generated test tool with DVT.

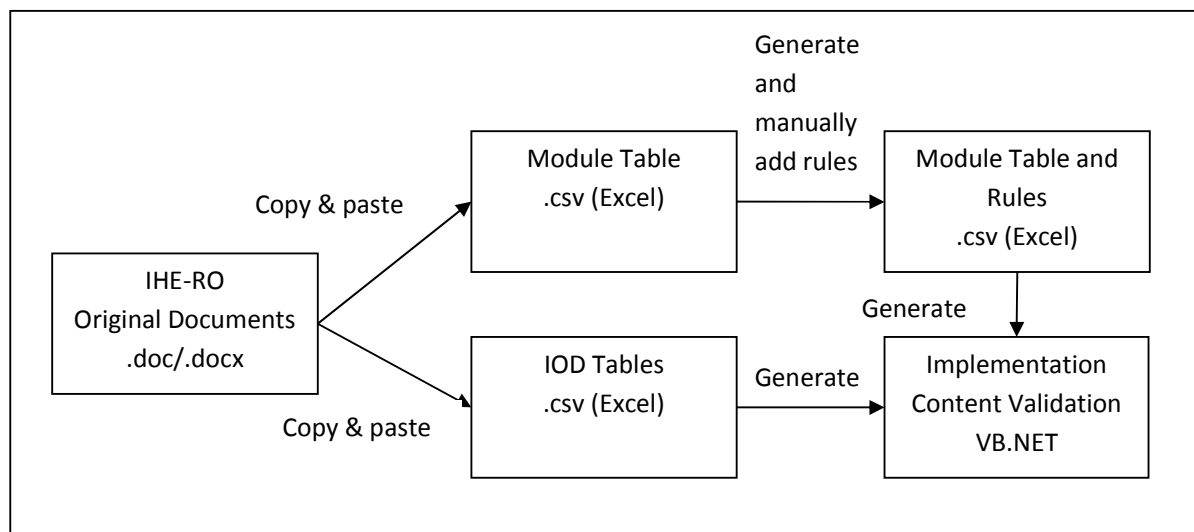
Both steps are explained in this document.

## Current Version

Version 1.0.0.

## Data Flow

The diagram below summarizes the data flow. See section “Work Flow” for more detailed information about the different steps.



## User Interface

The screenshot shows the 'IHE-RO Content Validation Creator' application window. It has a title bar with the application name. The main area is divided into several sections:

- Generate module file and manually add rules**: This section contains two options for generating module files. The first option is 'Single Module table, each module has his own file (Test data -> Profile directory + module table (.csv))', with a text box containing 'TPIC\TPIC - RT Image Module table.csv' and buttons for 'Browse', 'Open', and 'Gener'. The second option is 'Single Module table and rules: rules must be inserted in this file (.csv)', with a text box containing 'TPIC\TPIC - RT Image Module table and rules.csv' and an 'Open' button.
- Common**: This section contains a 'Test data directory:' label, a text box with a long file path, a 'Seperator' dropdown menu set to 'TPIC', and a 'Profile name' text box.
- IOD with modules overview**: This section contains a label 'Modules overview table, profile directory name + modules table file name (.csv)', a text box with 'TPIC\TPIC - IOD table.csv', and 'Browse' and 'Open' buttons.
- VB.NET**: This section contains a 'Generate' button.
- Run DVT**: This section contains a 'Run DVT' button and the text 'Run the Content Validation Test Tool in DVT'.

- “Single Module table” as default the TPIC – RT Image Module table csv file is selected, with the browse button an different file can be selected, with the open button the file can be opened and with the generate button the Single Module table and rules file will be generated, in this file the rules need to be placed.  
In the “Installation folder\RulesGenerator\Test data” directory there are three directories (BRTO, TPIC and TDIC) with some modules, if another module is needed this has to be made by hand  
(See “Work Flow” section how to make a Module table).

The directory name corresponds with the profile name, if a new profile must be made make a new directory and make a module table file for each module that has extra IHE restrictions. If a new Module table is made, browse to this file and click the Generate button, the Module table and rules file will be generated.

Open this file and add the rules in the F column.

- “Single Module table and rules” as default the TPIC – RT Image Module table and rules csv file is selected, with the open button this file can be opened.
- “Test data directory” – the path of the test data directory of the Test Tool Custom Content Validator.
- “Modules overview table” - the IOD table of the profile, each profile must have this file. Use column F for the name of the test data directory that must be used for the specific module.
- “VB.NET” - “Generate” button: generates VB.NET code using the “Modules overview table (IOD table) (.csv)” file as source. For each module that is R or M and has a corresponding Module table and rules file in the right directory the rules will be generated.
- “Run DVT” – Open the Content Validation Test Tool in DVT.

## Work Flow

The following steps need to be performed to be able to generate VB.NET code:

- In the “Installation folder\RulesGenerator\Test data” directory use a folder for each profile, make a Module table file for each module that you want to use.
- Open the “Module table (.csv)” file. Select all fields and change the category from General to Text.
- Copy-and-paste (from the original IHE-RO document), using paste special/unicode text, the different module tables and module section headers. See “Installation folder\RulesGenerator\Test data” for an example what parts to copy.
- Press the “Generate” button.
- Manually add rules to the “Module table and rules (.csv)” file.
- For each profile make a Modules overview table.
- Copy-and-paste (from the original IHE-RO document) the IOD table and IOD section headers. See “IHE-RO Content Validation Creator\Test data” for an example what parts to copy.
- Use column F for the name of the test data directory that must be used for the specific module.
- Select the right Modules overview table.
- Press the “VB.NET” - “Generate” button.
- Press the “Run DVT” – button, the Content Validation Test Tool will be started with DVT.

The “Installation folder\RulesGenerator\Test data” directory already contains examples of the .csv files.

## Rules

Rules that may be used (using VB.NET syntax):

- **AttributeHasValueRule**  
If attribute is present it must contain one or more values  
`AttributeHasValueRule()`
- **AttributeIsRequiredRule**  
Attribute is present with or without values  
`AttributeIsRequiredRule()`
- **AttributeIsRequiredHasValueRule**  
Attribute is present with a value  
`AttributeIsRequiredHasValueRule()`
- **AttributeNotPresentRule**  
Attribute is not present  
`AttributeNotPresentRule()`
- **AttributeValueRule**  
Attribute is present with given value  
`AttributeValueRule("0")`
- **AttributeValueListRule**  
Attribute is present and contains one of the given values  
`AttributeValueListRule(New String() {"1", "2", "3"})`
- **AttributeConditionalHasValueRule**  
Attribute is present and contains one or more values if conditional attribute exists with given valueE.g. Condition for "Shall have a value if value 3 of Image Type (0008,0008) is PORTAL" →  
`AttributeConditionalHasValueRule("0x00080008", 2, "PORTAL")`
- **AttributeConditionalIsRequiredRule**  
Attribute is present if conditional attribute exists with given valueE.g. Condition for "Required if Value 3 of Image Type (0008,0008) is PORTAL" →  
`AttributeConditionalIsRequiredRule("0x00080008", 2, "PORTAL")`
- **AttributeConditionalNotPresentRule**  
Attribute is not present if conditional attribute exists with given valueE.g. Condition for "Not present if Value 3 of Image Type (0008,0008) is PORTAL" →  
`AttributeConditionalNotPresentRule("0x00080008", 2, "PORTAL")`
- **AttributeConditionalValueRule**  
Attribute is present with given value if conditional attribute exists with given valueE.g. Condition for "Should contain value 42 if value of Image Type (0008,0008) is PORTAL" →  
`AttributeConditionalValueRule("42", "0x00080008", "PORTAL")`

- AttributeConditionalWithPositionsValueRule**  
 Attribute is present with given value on given position if conditional attribute exists with given value on given position. E.g. Condition for "Should contain value 42 if Value 3 of Image Type (0008,0008) is PORTAL" →  
 AttributeConditionalWithPositionValueRule(0 "42", "0x00080008", 2 "PORTAL")
- AttributeConditionalNotValueNotPresentRule**  
 Attribute is not present if conditional attribute exists and has not the given value.  
 E.g. Condition for "shall not be present, if value 3 of image type (0008,0008) is not simulator."  
 " →  
 AttributeConditionalNotValueNotPresentRule("0x00080008", 2, "simulator")
- AttributeNumberOfSequenceItemsRule**  
 If the sequence item attribute must have a specific number of items.  
 E.g. shall have only a single item in the sequence  
 AttributeNumberOfSequenceItemsRule(1)

## Conditional rules options

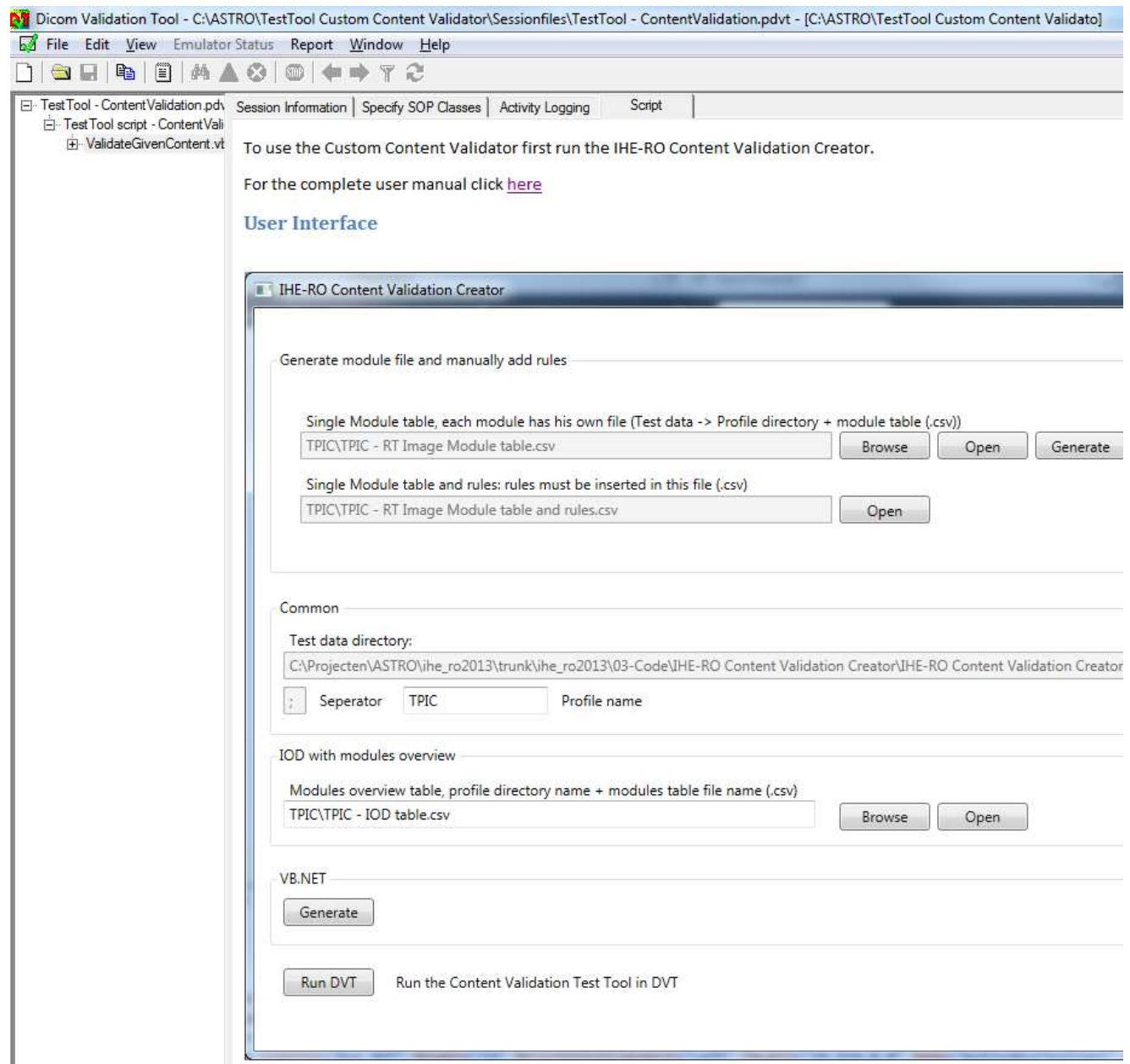
For the conditional rules there are 3 options to reference the conditional attribute tag.

"0x00080008" -> tag on highest level will be used by the rule.

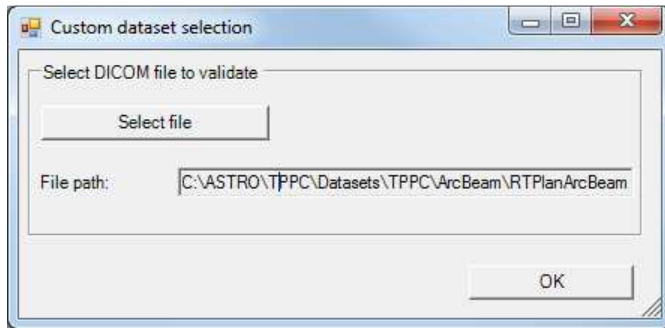
"/0x00080008" -> tag in the same sequence item as the attribute to test will be used for validation.

"../0x00080008" -> tag in one sequence item back as the attribute to test will be used for validation.

## Run validateGivenContent scenario



Double click the ValidateGivenContent.vbs.  
The dialog as shown below will popup.



Select the DICOM file that must be validated.

Two validations will be done, a DICOM validation against the definition files and the validation of the rules.

The Summary\_003\_ValidateGivenContent\_vbs\_res.xml will only show the rules that has gone wrong. In the Detail\_003\_ValidateGivenContent\_vbs\_res.xml both validations will be shown.