

Decision Table Creator

Main features

- Define conditions with more than 2 states
- Define don't care entries in conditions
- Customizable code generator
- Calculation of possible combinations and coverage
- Creation of remaining testcases for 100% coverage (limited to 1000 combinations)

The screenshot shows the 'Decision Table creator - NewProject' window. It has a menu bar with 'File', 'Edit', 'Export', and 'About'. Below the menu is a text area for 'Decision table description' with the placeholder text 'This is the description of this decision table'. The main area contains two tables. The first table, 'Conditions', has 7 columns: 'Conditions', 'TC1', 'TC2', 'TC3', 'TC4', 'TC5', and 'TC6'. It contains 3 rows of conditions. The second table, 'Actions', has 7 columns: 'Actions', 'TC1', 'TC2', 'TC3', 'TC4', 'TC5', and 'TC6'. It contains 4 rows of actions. At the bottom, there are three input fields: 'Possible combinations' (18), 'Covered test cases' (18), and 'Coverage' (100,00%).

Conditions	TC1	TC2	TC3	TC4	TC5	TC6
Basic Printer status	Printer is printing	Printer is not printing	Printer is not printing	Printer is not printing	Printer is not printing	Printer is not printing
Green LED	DC	LED is off	LED is flashing	LED is on	LED is on	LED is on
Paper	DC	DC	DC	Blank sheet is ejected	Print quality is bad	Nothing is ejected

Actions	TC1	TC2	TC3	TC4	TC5	TC6
Power		Check connection				
Ink cartridge			Check filling level	Check filling level	Check filling level	
PC connection						Check connection
Paper			Check paper filling level		Check for clean (unused) paper	Check paper filling level

Possible combinations: 18
Covered test cases: 18
Coverage: 100,00%

Menu description

File - New

Create a new project

File - Open

Open an existing project

File - Save / Save as

Save the current project

File - Create sample project

Create the printer troubleshooting sample

File - Create missing testcases

Create remaining testcases for 100% coverage

File - Open ReadMe

Opens the ReadMe.pdf - pdf viewer has to be installed

File - Exit

Exit tool

Edit - Append test case / Delete test case

Add and delete test case

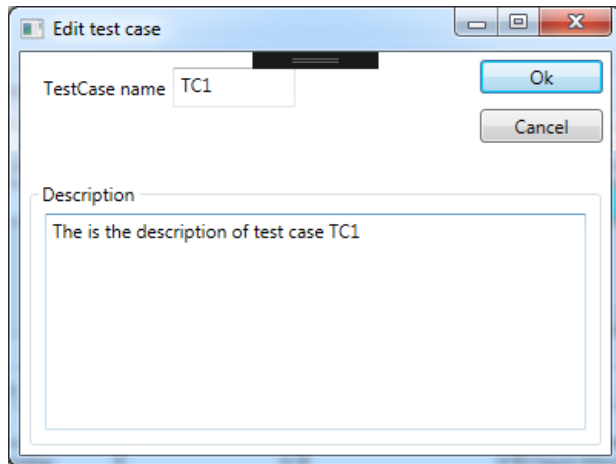
Edit - Delete most right test case

Deletes the most right test case

Edit - Edit test case description

Opens a dialog to edit the test case description

Edit test case dialog



Conditions

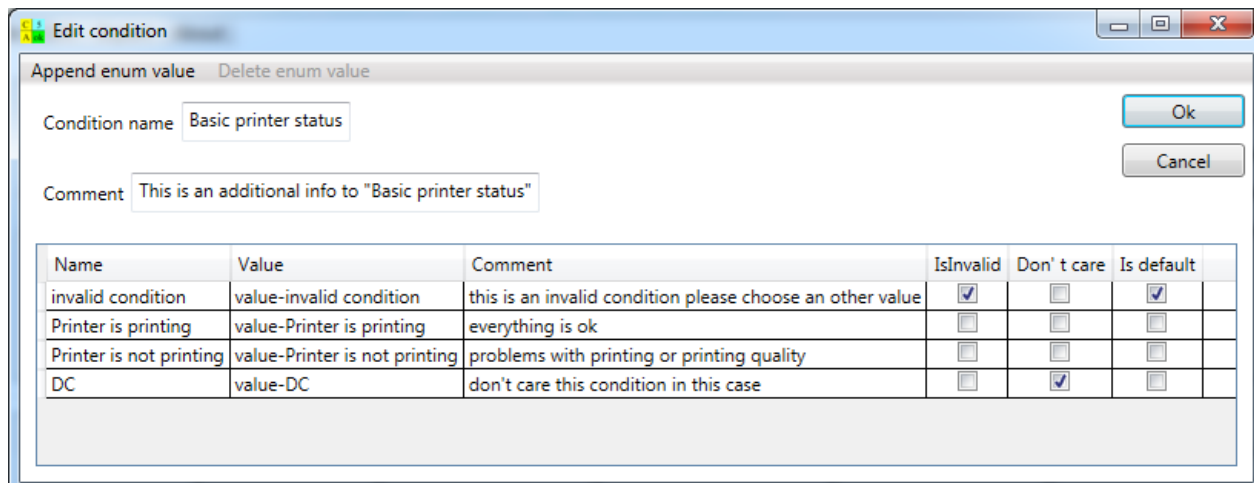
Edit - Edit condition / Edit action (or double click on condition or action name)

edit condition or action

Edit - Append condition / Insert condition

Add or insert new condition - the edit condition dialog box appears

Edit condition dialog



Name	Value	Comment	IsInvalid	Don't care	Is default	
invalid condition	value-invalid condition	this is an invalid condition please choose an other value	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Printer is printing	value-Printer is printing	everything is ok	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Printer is not printing	value-Printer is not printing	problems with printing or printing quality	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DC	value-DC	don't care this condition in this case	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Condition name

The name of the condition. This name will be displayed in the main window in the conditions column.

Comment

It is possible to define a comment for this condition. This comment will be displayed as tooltip and the comment is available for code generation.

Condition enum items

Name

The name of the condition enum item. This name will be displayed in the test case column.

Value

The value of the condition enum item. The value is used for code generation.

Comment

It is possible to define a comment for this condition enum item. This comment will be displayed as tooltip and the comment is available for code generation.

IsInvalid

This check box mark this enum item as an invalid choice. It will be displayed with a different color. In combination with "Is default" this will be the default enum item. An enum value with IsInvalid is not included in the coverage calculation.

Don't care

This check box mark this enum item as an don't care for this test case. This is included in the coverage calculation and reduces count of needed test cases.

Is default

This check box mark this enum item as the default item. This means that this is the default value when a new test case is added. If more than one enum item have the default flag, the first default enum item becomes the default enum item. If none of the enum items have a default flag, the first enum item becomes the default enum item.

Actions

Edit - Delete action / Delete condition

Delete condition or action - the condition or action must be selected

Edit - Move up / Move down

Move condition or action one line up or down - the condition or action you want to move must be selected

Edit action dialog

Append enum value Delete enum value

Action name: Power

Comment: This is a comment for the action "Power"

Ok Cancel

Name	Value	Comment	IsInvalid	Is default
	value-	nothing to do	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Check connection	value-Check connection	Check the printer power connection	<input type="checkbox"/>	<input type="checkbox"/>

Action name

The name of the action. This name will be displayed in the main window in the actions column.

Comment

It is possible to define a comment for this action. This comment will be displayed as tooltip and the comment is available for code generation.

Action enum items

Name

The name of the action enum item. This name will be displayed in the test case column.

Value

The value of the action enum item. The value is used for code generation.

Comment

It is possible to define a comment for this action enum item. This comment will be displayed as tooltip and the comment is available for code generation.

IsInvalid

This check box marks this enum item as an invalid choice. It will be displayed with a different color. In combination with "Is default" this will be the default enum item.

Is default

This check box marks this enum item as the default item. This means that this is the default value when a new test case is added. If more than one enum item has the default flag, the first default enum item becomes the default enum item. If none of the enum items have a default flag, the first enum item becomes the default enum item.

Export - Export to clipboard

Copy the current decision table to clipboard. Use this to export the table to a word processing or spreadsheet tool.

Export - External template - Sample.file.stg

The templates for code generation are located in the directory "MyDocuments"/DecisionTableCreatorTemplates/*.stg. The first template "Sample.file.stg" is written to this directory during the first start of the tool. This is the place to store other templates. During the start of the tool a submenu entry is created for every template in this directory. A template must have the extension ".stg".

Statistics

Possible combinations

The possible combinations are calculated based on the defined conditions and enum values. This is the count of test cases you need to get a coverage of 100%.

Covered test cases

The number of test cases that are defined. In this calculation the test cases with "Don't care" are considered. This calculation is suppressed if the possible combinations exceeds 1000.

Coverage

The coverage in percent which are reached with the defined test cases. This calculation is suppressed if the possible combinations exceeds 1000.

Code generation

The code generation is based on StringTemplate <https://github.com/antlr/stringtemplate4>

Further documentation: [StringTemplate cheat sheet](#)

Template

A valid template must be stored in the template directory "MyDocuments"/DecisionTableCreatorTemplates/ and must end with .stg The root entry must be **TestCasesRoot(root) ::= "..."**

Object model

The following list of objects (interfaces) are available for code generation. The root object supports the interface ITestCasesRoot

```
public interface ITestCasesRoot
{
    ObservableCollection<ITestCase> TestCases { get; }
    ObservableCollection<IConditionObject> Conditions { get; }
    ObservableCollection<IActionObject> Actions { get; }
    string Description { get; }
}

public interface ITestCase
{
    String Name { get; }
    int DisplayIndex { get; }
    ObservableCollection<ValueObject> Conditions { get; }
    ObservableCollection<ValueObject> Actions { get; }
    string Description { get; }
}

public interface IConditionActionObject
{
    string Name { get; }
    IList<ValueObject> TestValues { get; }
    Background Background { get; }
    ObservableCollection<EnumValue> EnumValues { get; }
    string Comment { get; }
}

public interface IValueObject
{
    ObservableCollection<EnumValue> EnumValues { get; }
    Background Background { get; }
    object Value { get; }
    IConditionAction ConditionOrActionParent { get; }
}

public interface IEnumValue
```

```
{
    string Name { get; }
    String Value { get; }
    string Comment { get; }
    bool IsDefault { get; }
    bool IsInvalid { get; }
    bool DontCare { get; }
}

public interface IBackground
{
    BackgroundColor BackgroundColor { get; }
    string HtmlColor { get; }
}
```