CVG IS...

Combinatorial Variation Generator allows you to reduce a large, unmanageable set of test-case inputs to a much smaller set that is likely to reveal bugs in the system under test.The tool is based on the Testapi which provides a generic API for combinatorial variation generation.

Here is other list of projects that CVG make extensive use of:

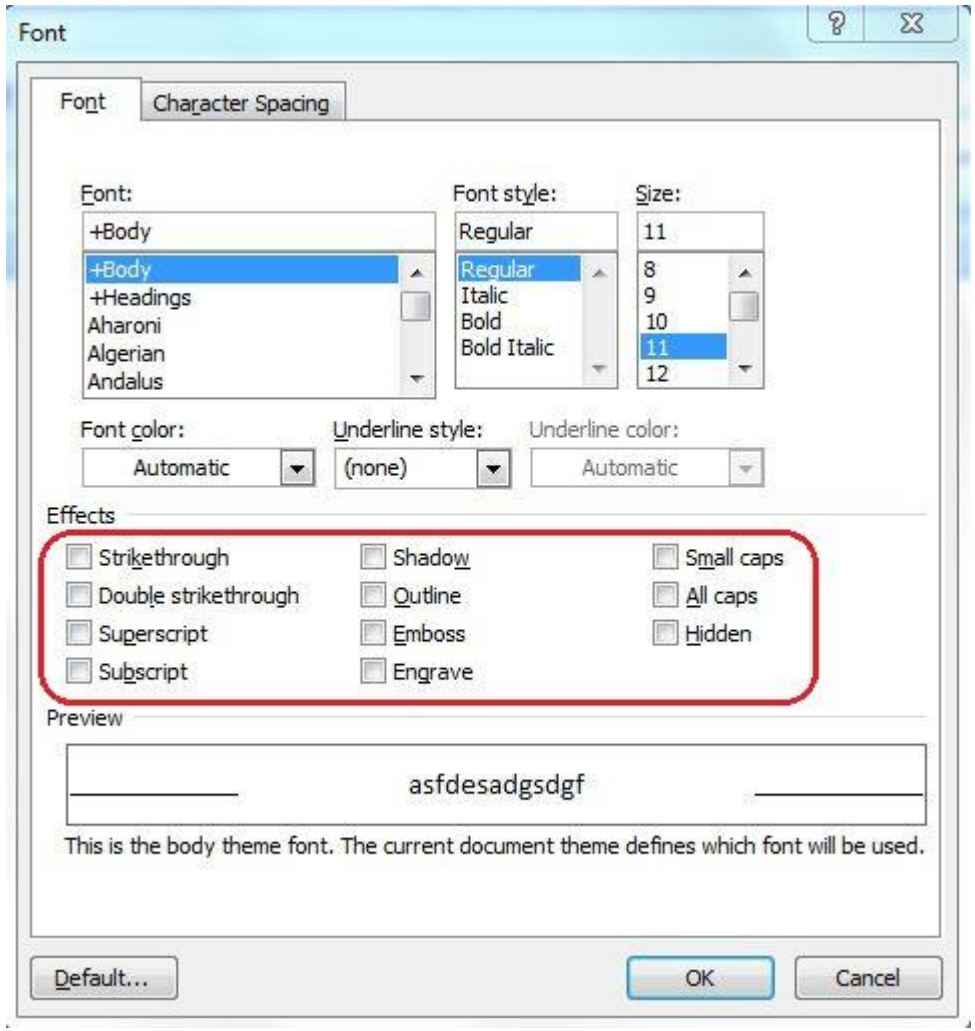
- Excel Data Reader
- ExcelPackage
- WPF customizable window

CVG is developed in WPF and it needs .Net 3.5 support.

Get Started

What is combinatorial testing? and how to Testing Efficiently with All-Pairs?

For example: How to test the effects in Font option of the MS Word application?


Font
Font
Character Spacing
Font:
+Body
+Body
+Headings
Aharoni
Algerian
Andalus
Font style:
Regular
Regular
Italic
Bold
Bold Italic
Size:
11
8
9
10
11
12
Font color:
Automatic
Underline style:
(none)
Underline color:
Automatic
Effects
Strikethrough
Double strikethrough
Superscript
Subscript
Shadow
Outline
Emboss
Engrave
Small caps
All caps
Hidden
Preview
asfdesadgsdgf
This is the body theme font. The current document theme defines which font will be used.
Default...
OK
Cancel

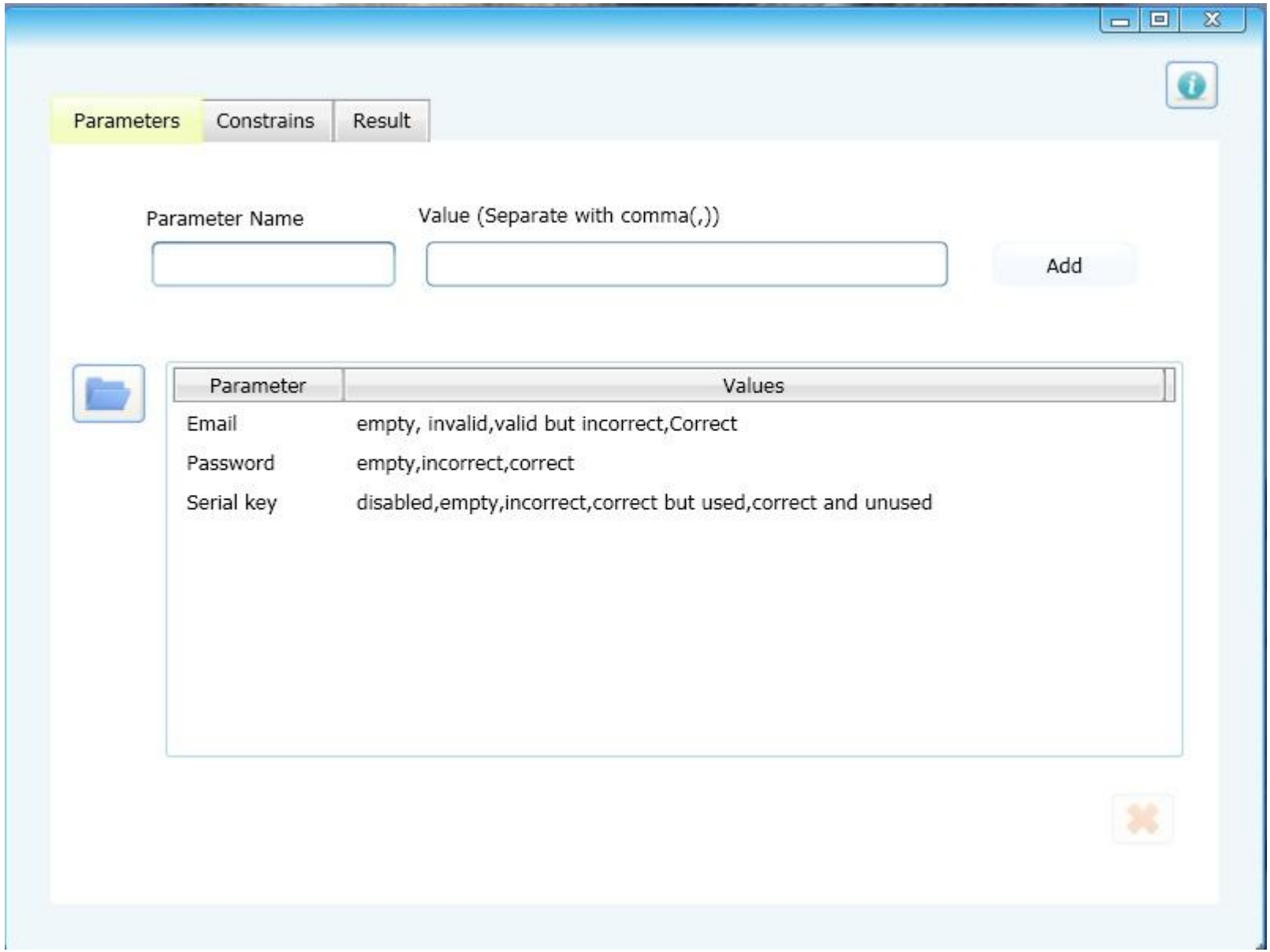
- There are 11 effects, each can be on or off
- There are 2x2x2x2x2x2x2x2x2x2x2 = 2048 test cases for all combinations (all possible inputs), A astronomical number for testing.
- OK, Let's look at all 2-way interactions (all pairwise), the only 8 cases covers all pairwise(Most field faults were caused by either incorrect single values or by an interaction of pairs of values.).

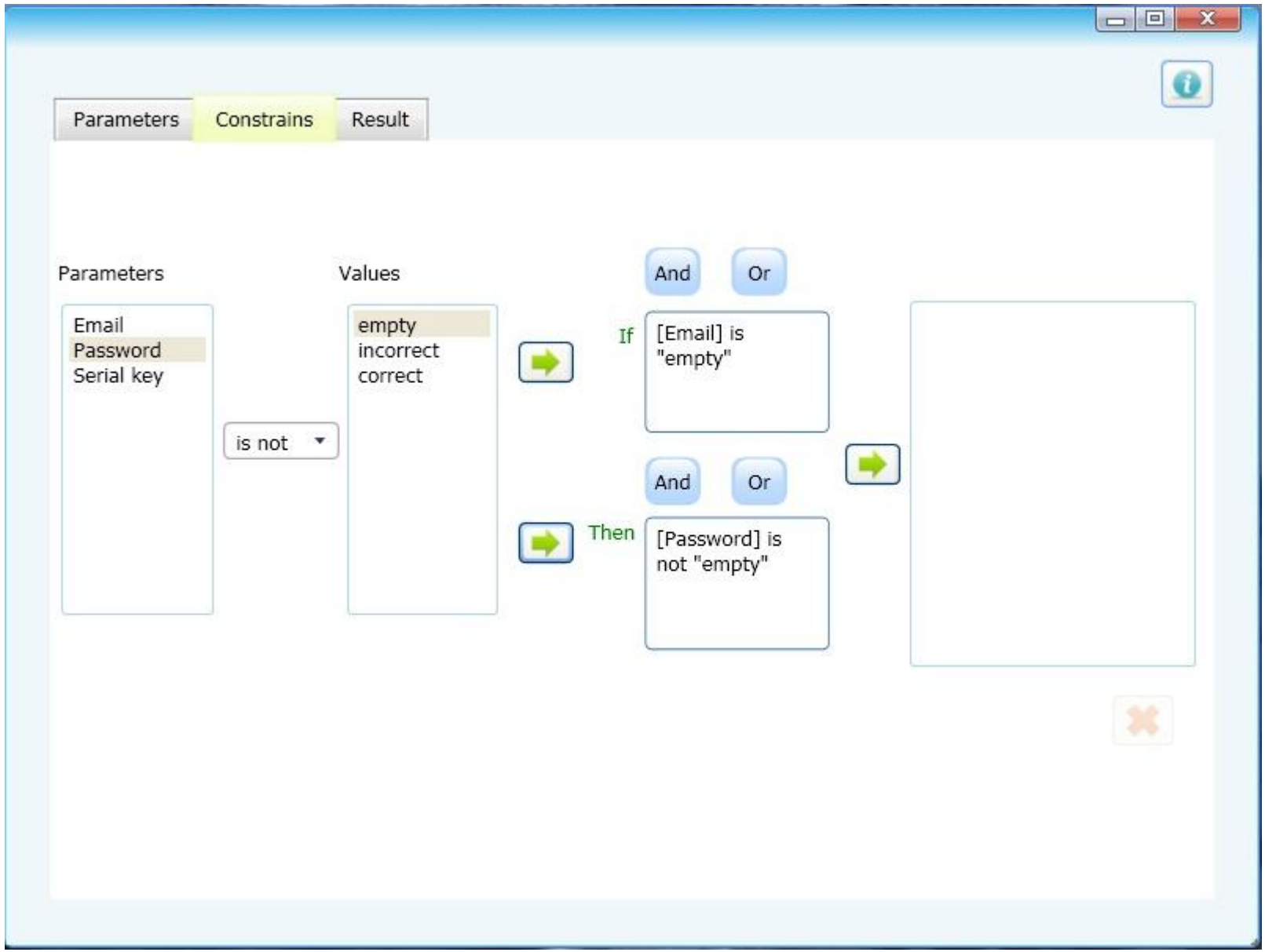
ID	Strikethrough	Double Strikethrough	Superscript	Subscript	Shadow	Outline	Emboss	Engrave	Small caps	All caps	Hidd
1	on	on	on	on	on	on	on	on	on	on	on
2	on	off	off	off	off	off	off	off	off	off	off
3	off	on	on	off	on	off	on	off	on	off	off
4	off	on	off	on	off	on	off	on	off	on	on
5	off	off	on	on	off	off	on	on	on	off	on
6	off	off	on	off	on	on	off	on	off	on	off
7	off	on	off	on	on	on	on	off	off	off	off
8	off	off	off	off	off	off	off	off	on	on	on
Only 8 test cases for All pairwise(2-way) combinations											

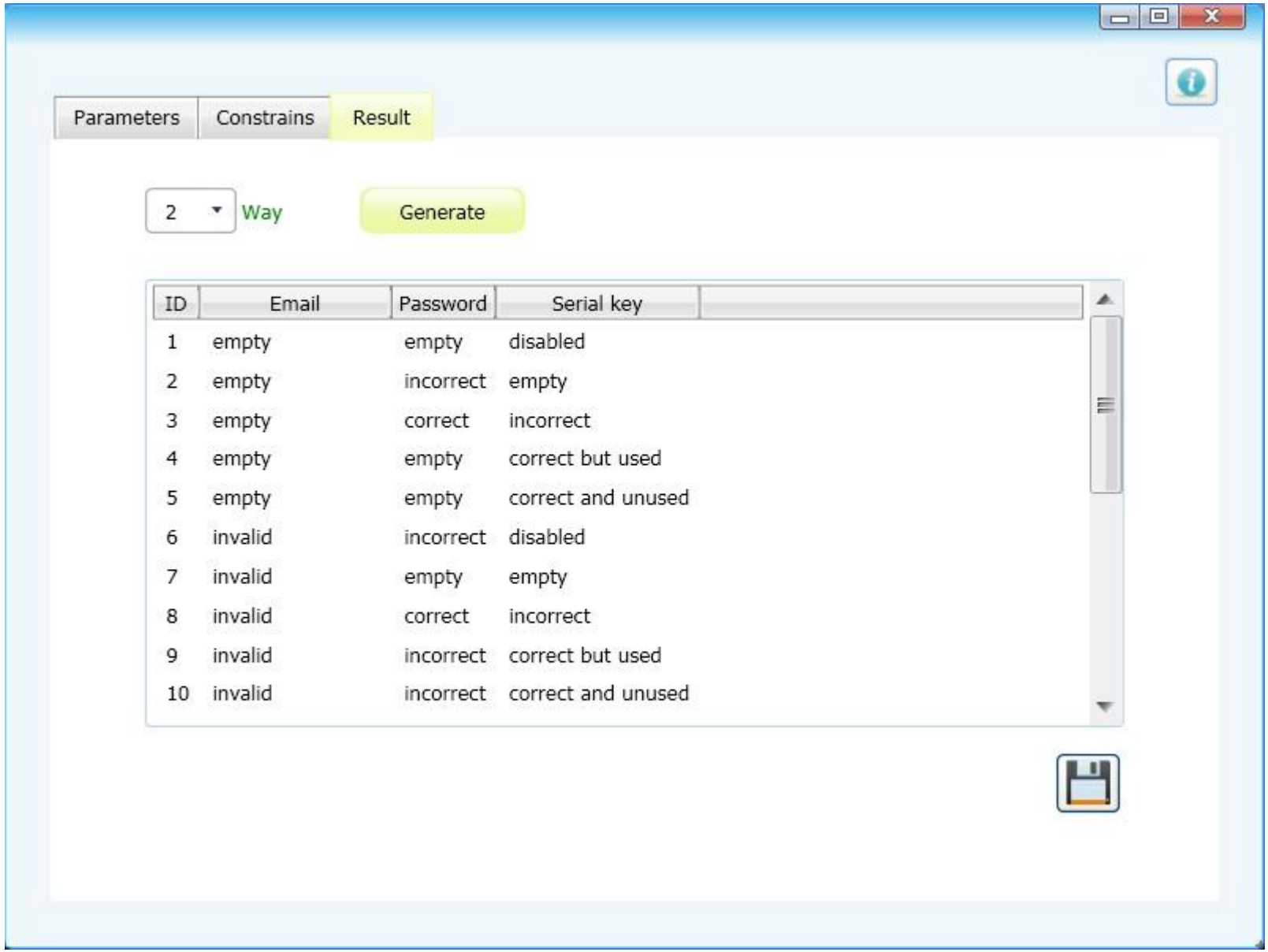
Exhaustively testing all possible inputs to any nontrivial software component is generally impossible due to the enormous number of variations. One approach to create a test suite with high coverage and a low number of variations is known as combinatorial testing. One common strategy, known as pairwise, tests a set of variations where every possible pair of parameters appears at least once.
For more combinatorial variation generation resources see Pairwise Testing

- How to use Variation Generator

Screenshots


Parameters
Constrains
Result
Parameter Name
Value (Separate with comma(,))
Add
Parameter
Values
Email
empty, invalid,valid but incorrect,Correct
Password
empty,incorrect,correct
Serial key
disabled,empty,incorrect,correct but used,correct and unused


Parameters
Constrains
Result
Parameters
Values
Email
Password
Serial key
is not
empty
incorrect
correct
And
Or
If
[Email] is "empty"
And
Or
Then
[Password] is not "empty"


Parameters
Constrains
Result
2
Way
Generate
ID
Email
Password
Serial key
1 empty empty disabled
2 empty incorrect empty
3 empty correct incorrect
4 empty empty correct but used
5 empty empty correct and unused
6 invalid incorrect disabled
7 invalid empty empty
8 invalid correct incorrect
9 invalid incorrect correct but used
10 invalid incorrect correct and unused

Contributors

Project Development:
Weifeng lu Email
Xiaohui Zhang Email

QA
Yan Liu
Zhaoqing Li
Weiping He

UX
Fuming Sun