PictMaster User Manual

July 10, 2012 Edition 1.0　Version 5.5.1

Iwatsu System & Software Co., Ltd.

About This Manual

1. If you wish to use PictMaster as soon as possible

If you wish to use PictMaster as soon as possible but you will return for more information later, please read at least Chapters 0, 3, and 4 before you use the software. PictMaster functionality is limited to the basic one by the default settings. To use PictMaster efficiently, reading the entire manual is recommended after you have tried it.

2. An example of results generation

The example table of the combination results of generation and the number of combinations have been created with the minimizing the number of test cases feature of PictMaster. The minimizing the number of test cases feature does not ensure reproducibility of the combination of generation results. Therefore, if the user tries to minimize the number of test cases with the same model, it may generate a different output from the generation results shown in this manual.

3. Before using PictMaster at work

PictMaster is a tool that generates a test case for several types of tests, including a combination test. It is necessary to learn the basic technology of test to understand the workings of this tool.

Many books describing test technology have been published. You might wish to read some of these books to extend your knowledge of basic test technology before working with PictMaster.

The technique of a combination test is one of a number of test techniques. Other test techniques may be appropriate rather than the technique of a combination test. Use the technique that meets the nature of the test.

To execute an effective combination test, parameter types for the combination are most important. The selection of the parameters needs to be considered.

PictMaster Conditions of Use

Users that are granted permission to use PictMaster will be expected to comply with the following conditions of use:

1. PictMaster (referred to as "this software") is free software, it can be freely used. However, the Software Copyright is held by Iwatsu System & Software Co., Ltd.
2. The user can generate and distribute derived results based on this software (original generation results) according to Open Software License V 2.0. However, as long as you use PICT, selling the derived results is prohibited under the Microsoft license terms. The distribution of bundled PICT is also prohibited.
3. Selling this software is strictly prohibited.
4. The user will not alter or remove the software copyright notice (Copyright © Iwatsu System & Software Co., Ltd.), except from derived results based on this software and distributed.
5. We assume no responsibility whatsoever for any damage resulting from the use of this software.
6. The PictMaster Usage Regulations are subject to change without notice.

Table of Contents

0. Installing PictMaster

[Preparation for Using PictMaster]

(1) You can obtain **PICT** at

http://blogs.msdn.com/b/nagasatish/archive/2006/11/30/pairwise-testing-pict-tool.aspx

Download and install in advance. You will need to install it in the following default installation folder.

C:\Program Files, or if 64-bit OS then C:\Program Files (x 86)

(2) Excel 2000 or later.

[How to Install]

(1) Open the PictMaster.zip compressed file and then save PictMaster.xls in any folder on your PC. It can be saved on a server. If the file is saved on a server where Mapped Network Drive is not available, a generated test case file *a.xls* or model file (the file to be input to PICT) *a.txt* will be created in the PICT folder.

(2) Excel Setup.

Excel 2007 or previous versions, select Tools -> Options -> Security ->
Macro -> Security Level and specify "Middle."
The following steps are for **Excel 2007:**
Click the Office button -> Excel Options -> Trust Center -> Trust Center Settings... -> Trusted locations -> Add new location ... -> Browse to specify the folder to save PictMaster, and then click OK.
You can specify subfolders as well.

The installation is finished.

[About Excel version]

The Excel file for PictMaster can be used from Excel 2000 to Excel 2010. The file format is Excel 1997–2003 Workbook. When using the later version of Excel 2007, you should use Excel in compatibility mode. **If you do not intend to use Excel 2003 or earlier version,** convert the file format to Excel macro-enabled workbook in Excel 2007 or later version to reduce the file size by half. Click the "Microsoft Office" button (Round button in the upper left) and select "Convert" and the file type as Excel macro-enabled workbook before saving. The file format is converted.

1. Introduction

PictMaster is Excel-based free software that improved **PICT** (Pairwise Independent Combinatorial Testing Tool) making it easier and more sophisticated. PICT is free Microsoft software generating combination test cases that use the Pairwise method (also called "All-Pairs method").

The purpose of publishing PictMaster is that many people can easily use the very superior combination testing tool, PICT, in Excel. PictMaster is provided as completely free software and as an open-source tool.

PictMaster was developed with the goal that users can use many functions implemented in operations, which are simplified as much as possible. In particular, the constraints table (the table to specify the exclusion of combinations) is not only easy to use, but also you can show many of the complicated relationships between constraints in easily understood visual expressions. The constraints table may be the most unique feature in the variety of PictMaster's features.

2. Mechanism of PictMaster

PICT itself is an application based on CUI (Character User Interface) to run on the command prompt. Now, most people are unfamiliar with the command prompt. There may be a lot of users who do not want to use PICT working in command prompt.

If you have created a test specification in Excel, generating a combination test case in Excel would be very useful. The Excel workbook, **PictMaster,** has realized this useful mechanism. PictMaster overlays the CUI-based PICT with an Excel GUI (Graphical User Interface) based shell. The diagram in **Figure 2-1** shows this.
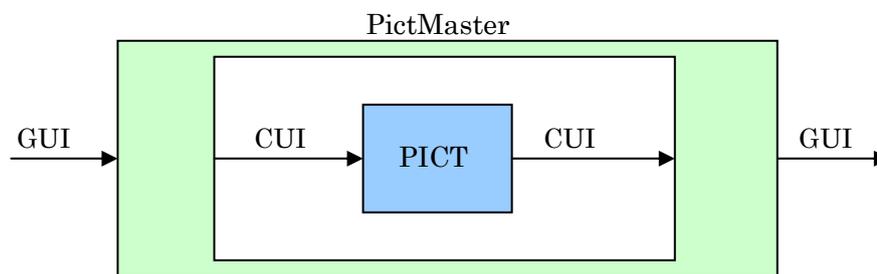
Figure 2-1 PictMaster diagram

As shown in **Figure 2-1**, PICT is completely invisible to the user. You can do all of your work based in the GUI.

PictMaster works with the following four applications:

(1) Excel VBA
(2) Command Prompt
(3) Batch File
(4) PICT

VBA (Visual Basic for Application) is a programming language (Visual Basic) for Excel. PictMaster controls the various GUIs of Excel, which uses VBA and creates model files and batch files, starts the command prompt, and runs the batch file. Also, PictMaster will change the order of generated results depending on the user's specification and draw the borders.

Batch file for a command prompt runs nkf and PICT.

PICT fulfills the roles of model files parser and combination generation engine.

3. Using PictMaster

  PictMaster performs with Excel 2000 or later. We have checked operation with Windows 7, Windows XP, and Windows 2000. In order to use PictMaster, install the following software:

(1) PICT itself
(2) Excel 2000 or later version

  PICT can be downloaded from the following website:
http://msdn.microsoft.com/en-us/testing/bb 980925.aspx

  An example of the **PictMaster** default screen image is shown in **Figure 3-1**
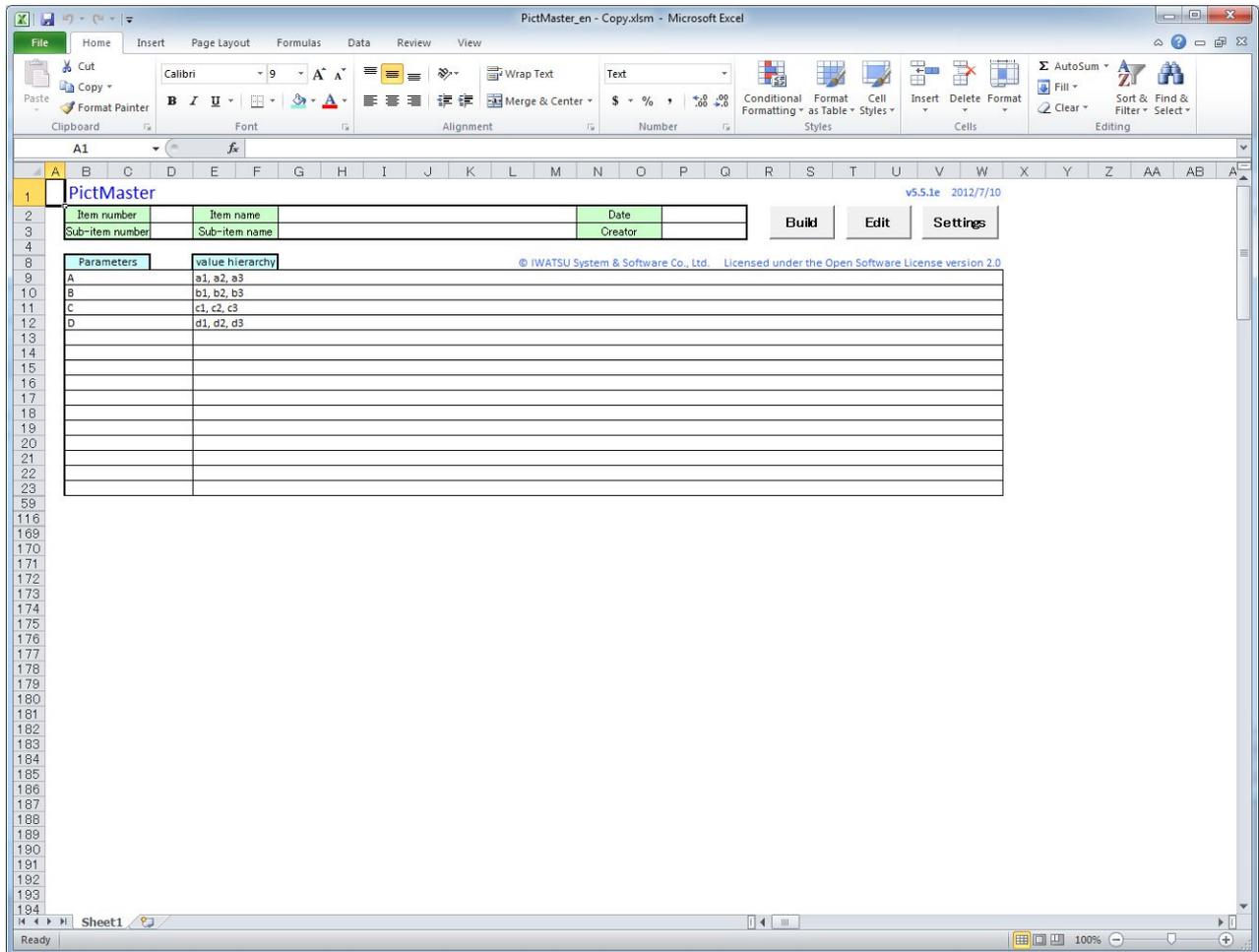
Figure 3-1 PictMaster default screen image

PictMaster comprises the following components:

Row 1 to 7: Blank cells

The blank cells are for the user to layout. You can determine the layout such as item number, sub-item number, creation date, and creator for ease of use. Rows 5 to 7 are set as hidden, display the rows from the Format menu if you wish to use them. The default blank cells layout is shown in **Figure 3-2**.


Figure 3-2 Default blank cells layout

Row 9 to 58: Parameters and value hierarchy

Enter parameters and value hierarchy separated by commas (,). The rows for parameters and value hierarchy are fixed at 50 rows. By default, the rows after 16 are hidden. In the value hierarchy row, you can enter a maximum of 50 values. Fill in the row **without leaving an empty cell.** The value can be added as an alias symbol (|), an invalid value symbol (~) or weighting specified (n).

Please note that you cannot delete or insert rows. Instead, clear contents or copy and paste rows. If the source row and the target row overlap, the row cannot be pasted correctly. Keep space open so that the target rows are not overlapped. When right clicking column A, **the shortcut edit menu is displayed**, you can move, insert, or delete rows and Undo. An example row of parameters and value hierarchy is shown in **Figure 3-3**.

Figure 3-3 Example of parameters and value hierarchy columns

* Do not use a name beginning with the characters "ID" in the first row (row 9) of parameter column.

Row 61 to 62: Sub-Models

By default, the rows are hidden and entry is invalid. The sub-model column is displayed by specifying **"Use sub-models"** in the settings form below, then the entry will be valid. You can enter any number of sub-models. An example of sub-models column is shown in **Figure 3-4**.

The details of sub-models are described in Chapter 3.5.

Figure 3-4 Example of sub-models rows

Row 66 to 95: Constraints Table

By default, the rows are hidden and entry is invalid. The constraints column is displayed by specifying **"Use constraints table"** in the settings form below and then the entry will be valid. The constraints table is used to exclude combinations that cannot be combined (referred to as "constraints"). Enter the constraints with a table format—50 rows are available. By default, the rows after 16 are hidden. When editing this column, please keep in mind that there are notes like the parameters and value hierarchy rows. An example of a constraints table is shown in **Figure 3-5**.

Constraints table			
Parameters	Constraint 1	Constraint 2	Constraint 3
A	a1	a2, a3	
B	b1, b2	b3, b4	
C			c1
D			d2
E			e2

Figure 3-5 Example of constraints table

How to enter data in the constraints table is described in .

Row 119 to 168: Expected results table

By default, the rows are hidden and entry is invalid. The expected results column is displayed by specifying **"Use expected results table"** in the settings form below and then the entry will be valid. The expected result table requires the specification of possible result outcomes (referred to as results) depending on the combination. Fifty rows are available. By default, the rows after 16 are hidden. If you use the expected results table, the results you expect are automatically assigned to the results column after each test case is generated. An example of an expected results table is shown in **Figure 3-6**.

Expected results table			
Result	Parameter 1	Parameter 2	Parameter 3
aaa	a1	b1	
aaa		b3, b4	c1
bbb	♯a1		♯c1
ccc			

Figure 3-6 Example of expected results table

How to enter data in the expected results table is described in .

Row 2 to 3: "Build," "Edit" and "Settings" button

In the default layout, three buttons that are shown in **Figure 3-7** are on the right in the second and third row.

Figure 3-7 Three buttons

3. 1 "Build" button

When you click this button, after you have entered the necessary parameters and values hierarchy, test cases will be generated as the book name "a.xls." The criteria to generate a test case are specified on the Settings form by clicking the "**Settings**" button. You can display the n-way coverage (combination coverage proportion of n-parameter interactions) for the generated test cases by clicking this button.

3. 2 "Edit" button

You can edit the generated test case for sorting values or drawing the borders with specified conditions. When you click the "Edit" button, the example form **Figure 3-8** will appear.
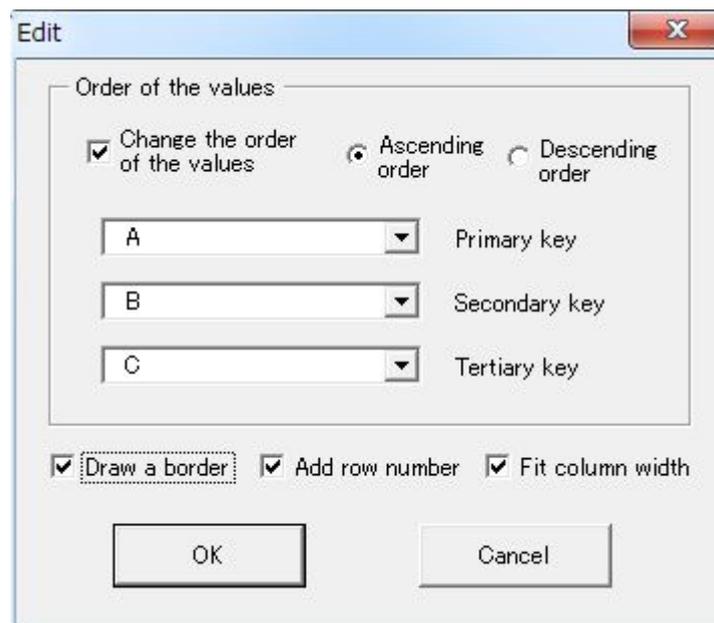
Figure 3-8 Form displayed when clicking the "Edit" button.

The top three parameters are chosen as the keys for sorting, then all items, such as Draw a border, Add row number, and Fit column width, are selected. If the row number has already been added, "Add row number" will not be available. The spreadsheet can be edited by the user. The above processes are executed by clicking the "OK" button. An example of the edited generation results by the "Edit" button is shown in **Table 3-1.**

Table 3-1 Example of the edited test cases

No.	A	B	C	D	E
1	a1	b1	c2	d1	e2
2	a1	b2	c1	d2	e2
3	a1	b2	c1	d1	e3
4	a1	b3	c2	d1	e1
5	a1	b4	c1	d1	e3
6	a2	b1	c1	d2	e3
7	a2	b2	c2	d1	e2
8	a2	b3	c1	d2	e1
9	a2	b3	c1	d2	e3
10	a2	b4	c2	d2	e2
11	a3	b1	c2	d1	e2
12	a3	b2	c2	d2	e2
13	a3	b3	c2	d2	e2
14	a3	b4	c1	d2	e1
15	a3	b4	c2	d1	e1

3. 3 "Settings" button

When you click the "Settings" button, the example form **Figure 3-9** will appear.
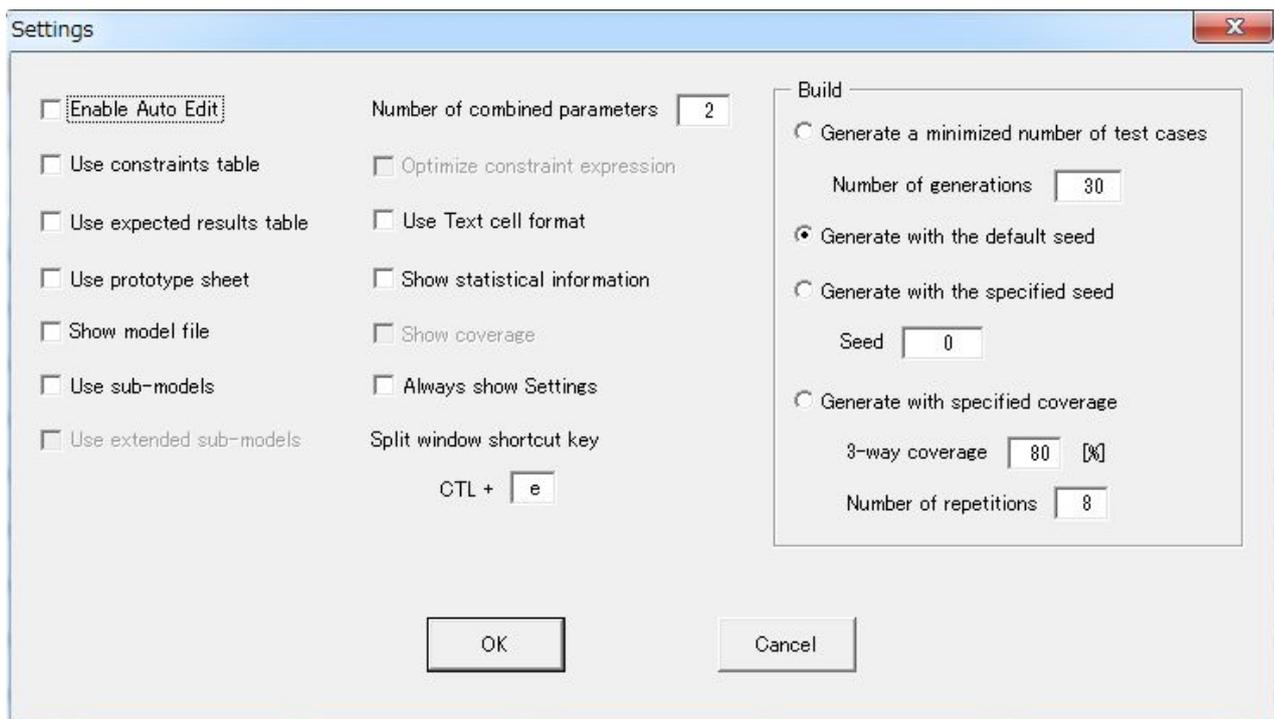
Figure 3-9 Form displayed when clicking the "Settings" button.

When you select **"Enable Auto Edit"** and generate test cases, then the generated test cases will be edited automatically. The editing conditions are the same as specified in **Figure 3-8**. If you wish to edit the test cases with different conditions, do not select this and, instead, click the "Edit" button to select the preferred condition for the test cases to generate.

When you select **"Use constraints table,"** the 50 constraints table column is displayed for the top 15 rows and specified constraints will be valid. Fifty constraints table columns are available. When you

have entered the 15th column, re-display the rest of columns, if necessary. The details are described in **Chapter 4**.

When you select **"Use expected results table,"** the 15 expected results rows are displayed and the specified conditions will be valid. Fifty expected results table columns are available. When you have entered the 15th column, re-display the rest of columns, if necessary. The details are described in **Chapter 5**.

When you select **"Use Prototype sheet"** to generate test cases, use the **next sheet on the right as the prototype sheet**. The details are described in **Chapter 3.3.1**.
An error message is displayed if there is no sheet on the right or even if the prototype sheet on the right has a different format. If the first digit of a cell in the prototype sheet has **a dollar sign ($)**, it is treated as a **comment row**. In the prototype sheet, fill in the row without leaving any empty cells. Enter the names of the parameters in the first row except for the comments row. If there is an empty row, the rest of the rows are ignored.

When you select **"Show model file"** to generate test cases, the test cases will be generated and then PictMaster creates the model file based on the parameters column, value hierarchy column and constraints table. PictMaster will transmit the model file a.txt to PICT and the file is shown in Notepad format.

When you select **"Use sub-models,"** the sub-models column is displayed and the specified sub-models will be valid. Also, the "Use extended sub-models" checkbox will be enabled. The details are described in **Chapter 3.5**.

When you select **"Use extended sub-models,"** the sub-models column is displayed and the specified extended sub-models will be valid. This checkbox is enabled when "Use sub-models" has been selected. The details are described in **Chapter 3.6**.

When you select **"Show statistical information";** the frequency of generation; the minimized, maximized, or initial number of test cases; the minimum seed value; and the elapsed time for the test cases generation will be displayed after the number of test cases has minimized completely. An example is shown in **Figure 3-10**. You can display the information for the one-time test case generation that specifies the default seed.
The initial value represents the default generation results of PICT. The minimum seed value represents the seed that has generated the minimized number of test cases.
When generating test cases with random conditions, a maximum 10 percent of variations may occur in the number of generated test cases. Often, the number of test cases can be reduced around 5 percent by running the minimized number of test cases. You can increase the probability of reducing the number of test cases if you increase the frequency of generation, usually 30 generations may be enough. If you wish to reduce the number of test case generations, specify the frequency of generation by approximately 100 to 300, and then perform the minimized number of test cases. The "Show coverage" checkbox will be enabled when this item has been selected.
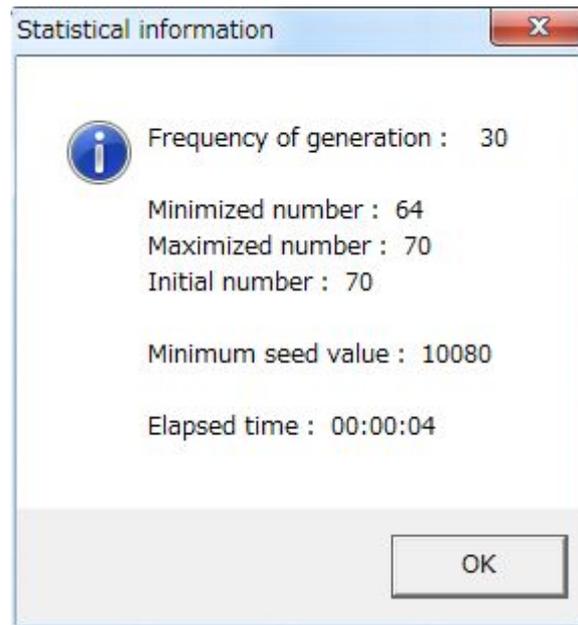
Figure 3-10 Example of statistical information window

When you select **"Show coverage,"** n-way coverage (combination coverage proportion of n-parameter interactions) and t-way coverage, which have been created during the test cases generation, will be displayed as **Figure 3-11**. The value n is defined as the specified value in the "Number of combined parameters." The value t equals the value of $n + 1$. The coverage will not be displayed if more than five are specified in the "Number of combined parameters."

To compute the n-way coverage, the combination is generated once with n-parameter interactions. To compute the t-way coverage, the combination is generated once with t-parameter interactions. Please note that the result from a large model may take some time to appear.
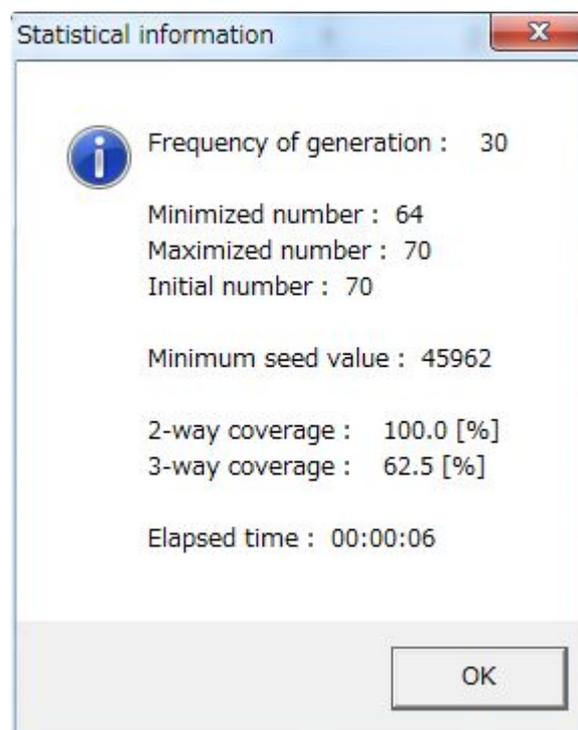
Figure 3-11 Example of coverage window

This checkbox is enabled when "Show statistical information" has been selected.

Specify a number between 1 and 50 in the **"Number of combined parameters"** box. By clicking the "Build" button, the combination with the specified parameters number will be generated. Please note that it may take quite a long time to generate the test cases if you specify a larger number.

When you select **"Optimize constraint expression"** to generate test cases, PictMaster will optimize the constraint expressions that are generated from the constraints table. If the parameter value is large and it is determined that it will take a long time to generate test cases due to the inappropriate constraints, the format of constraints will be optimized in the program to complete the generation in less time. The details are described in Chapter 3.3.2. This checkbox is enabled when "Use constraints table" has been selected.

When you select **"Use Text cell format"** to generate test cases, the results appear as text in each cell. This enables the value to be output without zero suppression, if it begins with 0 (001, 010, etc.). However, the value is treated as text, and the generation results will be sorted by not only the value but also the text length.

When you select **"Always show Settings,"** the key settings in the form will always be displayed in the upper right corner of the screen. You do not need to open the settings form many times for confirmation with this feature. The font color, size, and emboldening of the display settings can be customized.

If you enter a single letter or number in the **"Split window shortcut key"** box, the PictMaster window will be split into two parts by pressing the Control+ the letter or number you entered. The window is arranged one above the other, the lower window will be split between the parameter column and the constraints columns. This function is intended to enter many constraints in the constraints table. This is useful for the expected results table as well. The details are described in Chapter 6.3.

When you select **"Generate a minimized number of test cases,"** PICT will run with random conditions (seed) to generate test cases and output the test case results for the minimized number of test cases. Enter the number of test cases generation in **"Number of generations."** You can enter a number between 2 and 999. The default number is 30.

PICT generates the test cases by using a unique seed (default is 0) internally. The number of combinations can be a slightly different depending on the seed.

While minimizing the number of test cases, the progress bar, which is shown in **Figure 3-12** as an example, is displayed.
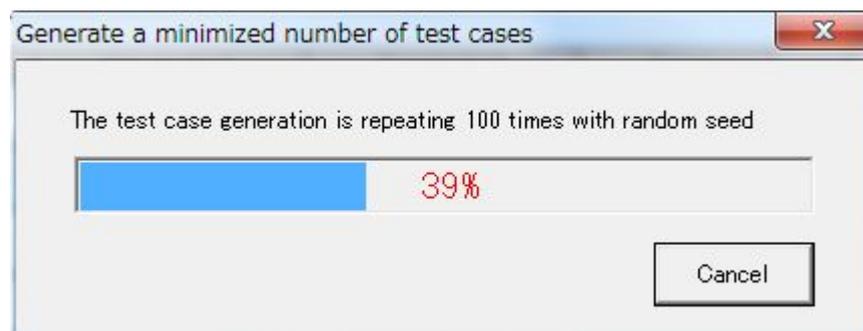
Figure 3-12 Example of the Progress Bar while minimizing the number of test cases

When you select **"Generate with the default seed,"** PICT will run with the default seed (0) to generate test cases.

When you select **"Generate with the specified seed,"** PICT will run with the seed that you entered in the "**Seed**" box to generate test cases. You can enter a number between 0 and 65535. The seed for the least number of test cases will be set in the "Seed" box after the minimum number of test cases has been found.

When you select **"Generate with specified coverage,"** the test case that ensures the desired t-way coverage will be output as the generated results. The value t equals $n + 1$ (n specified in "Number of combined parameters" in the settings form). Enter the preferred value of t-way coverage in the **"(t)-way coverage"** box. You can specify a value between 30% and 95%. To output the test cases that have ensured the desired test coverage, you will need to repeat several generations or change the test case. Specify the number of repetitions in "**Number of repetitions."** You can specify the number of repetitions between 3 and 10. The efficient use of this function is described in Chapter 3.7.

This feature can be used when the coverage of "Number of combined parameters" is between 1 and 5. This feature takes advantage of the functionality of PictMaster prototype sheet. If "Use prototype sheet" is selected in the settings form, "Generate with specified coverage" cannot be specified.

If you already assigned the weights to values, you may not be able to obtain the preferred coverage due to the functionality of the prototype sheet.

The Excel worksheet number will be modified to the higher number because this feature uses the prototype sheet functionality. For example, if you run this function three times, four Excel worksheets have been created. In this case, the new worksheet will be titled Sheet 5. If you wish to reset the worksheet number, save and restart PictMaster. The worksheet titles will be reset.

[Important Reminder]

If you specify a low value for the preferred t-way coverage, the coverage may be set much higher than the desired value. This behavior can occur if the coverage of the test cases generated with the default seed was higher than the desired coverage. In this case, if you wish to specify the lower t-way coverage, the t-way coverage will be invalid value because the n-way coverage cannot ensure 100 percent.

You can identify whether this behavior has occurred by checking the coverage of the generation results when you have specified "Generate with the default seed."

If the value has been assigned a weight, you may not obtain the desired coverage. This describes the limitations that occur under the function "Generate with specified coverage" using the prototype sheet function.

The valid test case results cannot be generated when invalid values are included. This describes the limitations in the specification of PICT where invalid values are concentrated in the latter half of the generated test cases. The progress bar shown in **Figure 3-13** is displayed during the repetition process.
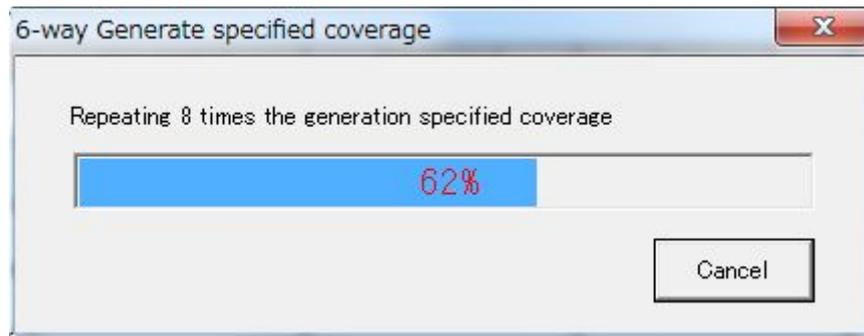
Figure 3-13 Example of the progress bar during generation with specified coverage

3. 3. 1 Using prototype sheet

There are three usages for prototype sheet.
(1) If the model previously used needs to be modified, **re-use the test case,** which was created with the previous model to create a new test case with as few changes as possible.
(2) You can **specify the important combination** that must be included in the generated test cases. The output is initialized as specified combination and then the rest of the combinations are generated.

When you select "Use prototype sheet" in the settings form to generate test cases, use the **next sheet on the right as the prototype sheet**. An error message is displayed if there is no sheet on the right or even if the prototype sheet on the right has a different format.
If the first digit of a cell in the prototype sheet has **a dollar sign ($)**, it is treated as a **comment row**. In the prototype sheet, fill in the row without leaving any empty cells.
You can enter a number of test cases between 0 and 65535 in prototype sheet. prototype sheet uses the same format (except for the row numbers) as the test case generated by PICT. Enter the names of the parameters in the first column except for the comments cell. All the same names of parameters as those in the parameters columns in PictMaster must be input. Enter the combined value hierarchy in the rest of the columns. A parameter may have no value.

[Important Reminder]
(a) If prototype sheet is including a value that the current models do not have, the combination of the parameter values will not be performed. Only the combination of the parameter value for the current model will be generated. The row containing an invalid value becomes an incomplete or partial row.
(b) If the value row in the prototype sheet violates any of the current constraints, the row will be ignored.
(c) When you use the prototype sheet, **Assigning weights to values (Chapter 3.4.3)** will not work properly. If you wish to use the generated results that were weighted as the prototype sheet, we recommend excluding the weighting from the value hierarchy columns of PictMaster to generate the test cases.
(d) Each combination in the prototype sheet is loaded by PICT. When all the combinations in n-parameter interactions have been covered in loading, PICT will stop loading the prototype sheet. The value *n* indicates the specified "Number of combined parameters" in the settings form. However, if the current model includes a value that is not entered in the prototype sheet, the whole combination of the prototype sheet will be loaded.
(e) This function cannot be used with "Generate with specified coverage" at the same time.

An example of using the prototype sheet is shown below:

Example 1: Add a new value

Figure 3-14 Prototype sheet example (1)

Figure 3-15 New model example (1)

The value b3 is added to parameter B in the new model.

Table 3-2 Test case added new values

No.	A	B	C
1	a1	b2	c2
2	a1	b1	c1
3	a3	b2	c1
4	a2	b1	c2
5	a2	b2	c1
6	a3	b1	c2
7	a3	b3	c1
8	a1	b3	c2
9	a2	b3	c1

For the new generated test cases, the data of the prototype sheet is applied and the new combinations corresponding to the added value b3 have been added. This will help you to generate tests only for the combinations for the added values when the existing test cases specification changes after testing and you need to add a new value for another test. (Note: The values in **Table 3-2** are not sorted)

Example 2: Add a new parameter

Figure 3-16 Prototype sheet example (2)

Figure 3 - 17 New model example (2)

The parameter D has been added in the new model.

Table 3-3 Test cases added new parameters

No.	A	B	C	D
1	a1	b2	c2	d3
2	a1	b1	c1	d2
3	a3	b2	c1	d1
4	a2	b1	c2	d1
5	a2	b2	c1	d2
6	a3	b1	c2	d2
7	a2	b1	c1	d3
8	a3	b1	c2	d3
9	a1	b2	c1	d1

For the new generated test case, the data of the prototype sheet is applied and the new combinations corresponding to the added parameter D have been added. (Note: The values in **Table 3-3** are not sorted)

All you need to do is just add the new parameter name to the prototype sheet, then you can generate easily the new test case that has added a new parameter.

Example 3: Specify the combination must be included

Figure 3-18 Prototype sheet example (3)

Figure 3-19 Model example

The value a1 in the prototype sheet is combined with all values of the other parameters B and C. It is not possible, even with the flexible constraints specification function of PICT, for only the particular values for parameters to be combined with the values of all other parameters. Therefore, you should **specify such special or complex combinations by using the prototype sheet.**

Table 3-4 Test case including the specified combination

No.	A	B	C
1	a1	b1	c1
2	a1	b1	c2
3	a1	b2	c1
4	a1	b2	c2
5	a2	b1	c2
6	a2	b2	c1
7	a3	b1	c1
8	a3	b2	c2

For the new generated test case, the data of the prototype sheet is applied and the new combinations corresponding to the other parameters have been added. In this example, by using prototype sheet, you can easily specify a complicated combination that cannot be specified under constraints.

3. 3. 2. Optimizing Constraint Expression

The combination generation engine PICT uses the scripting language (constraint expression) like IF statements to specify the unwanted combinations. PictMaster converts the entry in the constraints table to constraint expression for PICT to understand. As a characteristic of PICT, if you specify the constraints to parameters which have many values, the generation time of combinations may be very different depending on the specification of constraints. When you specify **"Optimize constraint expression",** PictMaster will investigate the number of parameter values that are specified with constraints. If the number is eight or more and the optimization of the constraint expression is enabled, it modifies the constraints within the program to create the constraint expression for the least elapsed time.

To see the difference in generation time depending on the constraints, refer to the model in **Figure 3-20**. The PC environment is CPU Pentium D 2.8 GHz, RAM 1 GB, OS Windows XP Professional SP 3.

Parameters	Value hierarchy
A	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
B	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
C	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
D	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
E	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Figure 3-20 Example of the model for generation time comparison

The generation time of the model above that specified the following constraints is shown below. These constraints are described differently. However, the value is equivalent. The generation time is indicated as the elapsed time for the execution of thirty times minimizing the number of test cases.

(1) Constraint 1

Constraints table		
Parameters	Constraint 1	Constraint 2
A	1, 2, 3, 4, 5	
B	1, 2, 3, 4, 5	1, 2, 3, 4, 5
C		1, 2, 3, 4, 5
D		
E		

Optimize constraint expression = False -> = Generation time = 7 seconds
Optimize constraint expression = True -> = Generation time = 4 seconds

(2) Constraint 2

Constraints table		
Parameters	Constraint 1	Constraint 2
A	1, 2, 3, 4, 5	
B	#6, 7, 8, 9, 10	1, 2, 3, 4, 5
C		#6, 7, 8, 9, 10
D		
E		

Optimize constraint expression = False -> = Generation time = 3 seconds
Optimize constraint expression = True but not applied. (Already optimized)

(3) Constraint 3

Constraints table		
Parameters	Constraint 1	Constraint 2
A	#6, 7, 8, 9, 10	
B	1, 2, 3, 4, 5	#6, 7, 8, 9, 10
C		1, 2, 3, 4, 5
D		
E		

Optimize constraint expression = False -> = Generation time = 7 minutes 20 seconds
Optimize constraint expression = True -> = Generation time = 4 seconds

(4) Constraint 4

Constraints table		
Parameters	Constraint 1	Constraint 2
A	#6, 7, 8, 9, 10	
B	#6, 7, 8, 9, 10	#6, 7, 8, 9, 10
C		#6, 7, 8, 9, 10
D		
E		

Optimize constraint expression = False -> = Generation time = 7 seconds
Optimize constraint expression = True -> = Generation time = 3 seconds

Figure 3-21 Constraint expression optimization and generation time

The constraint expression optimization for constraint (3) shows the greatest effect. Furthermore, constraint (2) shows it was not optimized. The optimization is not required for this constraint. The constraint expression optimization is executed when all the following conditions are satisfied.

1. "Optimize constraint expression" is specified in the settings form.
2. The number of values that belong to the parameters for the constraint condition and constraint target must be both eight or more.
3. The number of values that are specified to be constrained is two or more.
4 The constraints apply to reverse constraints, or the target value of constraints applies to order constraints.

For the definition of constraints, refer to **Chapter 4**.

When optimization for constraints has been specified, PictMaster converts the descriptor format of the constraint column to the format of Constraint 2 in the program if there is a constraints column that satisfies the four conditions above. Therefore, the optimization may not be performed even though **"Optimize constraints"** has been specified in the Settings form. In addition, the generation time may not be reduced when the optimization of the constraints has been performed, if optimized for a part of the various constraints. The constraint for comparison with parameters is not optimized.

When the constraints optimization is executed, the different constraints from the specification of constraints table will be generated. Please note that the constraints shown in an error message and entered in the table may be different if PICT detects an error. When you specify both "Optimize constraints" and "Show statistical information," the result of optimization of the constraints table is shown in the statistical information.

3. 4 Entering Value Hierarchy

Enter the value hierarchy separated by commas (,). You can trigger the special processing by adding a particular character at the beginning of a value. You can use **pipe (|)** and **tilde (~)**. Pipe is used for **aliases** that specify multiple names for one value. Tilde is used for the **Invalid Values Test** that specifies a value that cannot be combined. By adding **the number in brackets** at the end of a value, you can **assign weights to values** so that the value appears more than other values in many combination opportunities. These characters cannot be used in the names of parameters and values.

3. 4. 1 Aliasing to specify multiple names for one value

The alias is a function that specifies multiple names for one value. You can use the alias function based on the concept of equivalence partitioning for multiple values that are not exactly but nearly the same. It is treated as one entity to generate combinations, and after the generation is completed, the value is returned to the multiple names specified by the alias.

The number of generated combinations can be reduced by using aliases. The combinatorial number that is generated by the Pairwise method is a multiple or a slightly higher value, between $V1n$: the number of values of parameter P1 that has the most combinatorial number and $V2n$: the number of values of parameter P2 that has the same combinatorial number or less. The following formula shows the approximate combinatorial generation (test cases) number represented as TCn.

$TCn = (V1n * V2n + \alpha) * m$

The alpha value is different depending on the number of values of parameter P3n, P4n... which have a lower quantity than P2. If this number is equal to P2n or almost the same, so alpha takes a relatively large value in many cases. On the other hand, if the number is smaller than P2, alpha takes a smaller value. In many cases, the alpha value becomes zero. If you choose **the parameter that has largest number of values** to apply an alias, the reduction in the number of combinations is expected to be greatest. On the other hand, applying aliases to the parameters that have a few values is not effective. When the number of constraints is 0, m becomes 1. The value of m will be gradually increased by increasing the number of constraints.

An example of the model that uses an alias is shown in **Figure 3-22**. In this example, the test is for the combination of OS type, HD capacity, and HD interface. In this model, the three Windows operating systems are treated as one entity with the alias "|" from the five values for the OS type, and the total values are treated as three. If you specified the value with an alias, use the name of the first value to describe the constraint.

Parameters	Value hierarchy
OS Type	Windows 7 | Windows XP | Windows 2000, Linux, Mac OS X
HD Capacity	250GB, 500GB, 750GB
HD interface	USB 2.0, IEEE1394, eSATA

Figure 3-22. Example of the model using an alias

The combinatorial generation results of **Figure 3-22** that used an alias are shown in **Table 3-5**, the combinatorial generation results that did not use an alias are shown in **Table 3-6**.

Table 3-5 Using an alias

No.	OS Type	HD Capacity	HD interface
1	Linux	250 GB	IEEE 1394
2	Linux	500 GB	USB 2.0
3	Linux	750 GB	eSATA
4	Mac OS X	250 GB	USB 2.0
5	Mac OS X	500 GB	eSATA
6	Mac OS X	750 GB	IEEE 1394
7	Windows 2000	250 GB	eSATA
8	Windows 7	750 GB	USB 2.0
9	Windows XP	500 GB	IEEE 1394

Table 3-6 Not using alias

No.	OS Type	HD Capacity	HD interface
1	Linux	250 GB	eSATA
2	Linux	500 GB	USB 2.0
3	Linux	750 GB	IEEE 1394
4	Mac OS X	250 GB	eSATA
5	Mac OS X	500 GB	IEEE 1394
6	Mac OS X	750 GB	USB 2.0
7	Windows 2000	250 GB	USB 2.0
8	Windows 2000	500 GB	IEEE 1394
9	Windows 2000	750 GB	eSATA
10	Windows 7	250 GB	USB 2.0
11	Windows 7	500 GB	eSATA
12	Windows 7	750 GB	IEEE 1394
13	Windows XP	250 GB	IEEE 1394
14	Windows XP	500 GB	eSATA
15	Windows XP	750 GB	USB 2.0

In this example, the number of test cases that used an alias was reduced to less than two-thirds of the number of test cases that did not use an alias. **You should use an alias if the parameter has many values and several values are equivalent.**

Within the values that were treated as one entity, only the first value can be specified in a constraints table or an expected results table. These tables are described below:

* Note

If you have many values that are to be treated as one entity by an alias, **some of them may not appear in the generation results.** This occurs if the quantities of parameter values that are not included in the alias are small. All combinations can be marked as covered even if not all alias values were set for the combinations. Originally, the alias is the value that summarizes equivalent values. There is no problem if the alias does not appear in the generation result, however, you should realize that this is possible.

3. 4. 2 Invalid Values Testing

To test valid combinations, you usually need to avoid invalid values that make the program produce errors. If the test case includes invalid values, the function may not work, and the combination of parameter interactions will not be generated properly.

This section describes the purpose of testing and includes invalid values. If one test case has more than one invalid value, it recognizes the first invalid value and the test case for the rest of the values will not be performed. The test case is incomplete. For combinations that generate program errors, you need to prevent the generation of combinations with invalid values.

PICT contains the function of Negative Testing, which can **prevent the generation of combinations with invalid values.** The invalid value can be specified by **marking the character "~" at the beginning of the value** in the value hierarchy column.

An example of the model, including invalid values, is shown in **Figure 3-23**. In this model, the test is for the combination of FAX and communications lines. The test includes a telephone as the invalid value that cannot communicate with a fax. An external IP that cannot access a fax is also included as an invalid value (in fact, you can use it without difficulty in most cases).

Figure 3-23. Example of the model including invalid values

Table 3-7 Generated test results with invalid values

No.	Sending terminal	Communications line	Receiving terminal
1	~telephone	Analog	FAX
2	~telephone	ISDN	FAX
3	FAX	~external IP	FAX
4	FAX	Analog	~telephone
5	FAX	Analog	FAX
6	FAX	ISDN	~telephone
7	FAX	ISDN	FAX

The **Table 3-7** above shows the generated results. The combination with invalid value interactions was not generated. In this example, any invalid value will be paired with all valid values. If you do not need a thorough test in constraints, you can specify each invalid value for one test case.

If the value has multiple names (i.e. alias), add the character "~" at the beginning of the first name. If the invalid value can be described in the constraints, the character "~" should not be added.

3. 4. 3 Assigning Weights to Values

The weighting feature lets you focus on testing a particular value. **When you use the weighting feature, the specified value frequently presents in the test cases, the coverage for the value will be improved**. Apply weights by adding the number in **brackets（）** to the right of the value you want to focus on in the test. An example of weighting is shown in **Figure 3-24**.

Figure 3-24 Weighting example

The numeric value must be between 2 and 10 in the brackets. The value will present more often in combinations due to the weighting of the specified numeric value. For example, if you enter c3 (3), the value c3 will appear three times more than the other values c1 and C2. However, if the generated combination is duplicated, the duplication will be removed to create the final generation results.

The following **Table 3-8** shows the generated results, including duplicated combinations, which are shown in **Figure 3-24**.

Table 3-8 Generated results including duplicated combinations

No.	A	B	C
1	a1	b1	c 3
2	a1	b1	c 3
3	a1	b1	c 3
4	a1	b2	c1
5	a1	b2	c2
6	a1	b3	c3
7	a1	b3	c3
8	a1	b4	c1
9	a1	b4	c2
10	a1	b4	c3
11	a2	b1	c1
12	a2	b2	c3
13	a2	b3	c2
14	a2	b3	c3
15	a2	b4	c3
16	a3	b1	c2
17	a3	b2	c3
18	a3	b2	c3
19	a3	b3	c1
20	a3	b4	c3

The quantity of parameter C in **Table 3-8** is C1 = 4, c 2 = 4, and c3 = 12. At this point, c3 appears three times more than other values. This is designated by the weighting. However, the combinations in the shaded cells are duplicates. The duplication happens when the combination is generated with the weighted value, which is used three times more, and the values of other parameters are less than the weighted value.

PictMaster will remove the duplication leaving one combination. Therefore, if duplicated combinations are present, the weighted value will appear less than the assigned value. The final generated results are shown in **Table 3-9**. The combination results are different from **Table 3-8** due to the different generation process.

Table 3-9 Final generated results removed the duplications

No.	A	B	C
1	a1	b1	c1
2	a1	b1	c3
3	a1	b2	c3
4	a1	b3	c2
5	a1	b3	c3
6	a1	b4	c1
7	a1	b4	c2
8	a2	b1	c3
9	a2	b2	c2
10	a2	b3	c1
11	a2	b3	c3
12	a2	b4	c3
13	a3	b1	c2
14	a3	b2	c1
15	a3	b3	c3
16	a3	b4	c3

In the final generated results, the duplicated four combinations were removed, and 16 combinations are shown. The quantity of parameter C in Table 3 - 9 is C1 = 4, c 2 = 4, and c3 = 8. If you minimize the number of test cases with weighted values and have specified "Show statistical information" in the settings form, the statistical information as in **Figure 3-25** will be displayed. You can find how many duplicated combinations have been removed in the "Number of removed duplications."
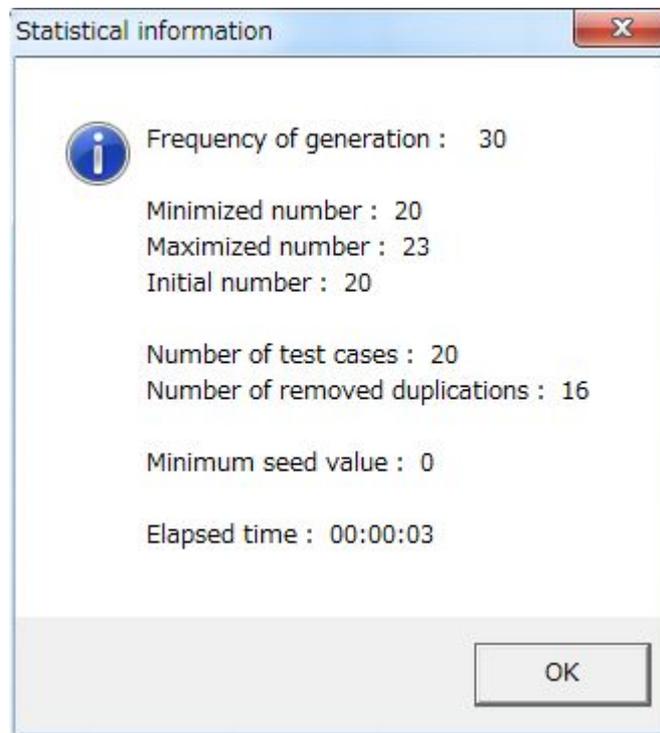
Figure 3-25 Example of statistical information of the model including weights

The accurate results that reflect the weighted values are generated when the test case has many other parameters or a large quantity of values for the parameters.

The number of values is increased due to the weights, and you can enter it up to 50. When assigning weights, the number of values must not exceed 50. The weights are not assigned to values correctly when you use weights and the prototype sheet at the same time. We recommend that weights and the prototype sheet not be used at the same time.

* The weighting feature of PICT is not available.

3. 5 Sub-Models

There are not many test cases where all parameters have the same importance. **PICT** allows you to specify the number of bundled parameters for certain parameters that need to be tested more thoroughly by using **sub-models definition**. To show the sub-models columns, select "**Use sub-models**" in the settings form. The format for the sub-models column is shown in **Figure 3-26**.

<Parameter 1>,<Parameter 2> ,..... ,<Number of combined parameters>

Figure 3-26 Format of sub-models

Enter multiple parameter names for the sub-models separated by commas. Enter the number of combined parameters at the end. This number can be between 1 and a value such as the specified number of parameters. You can enter multiple sub-models in one cell by separating them with a semicolon (;).

The multiple parameters specified in the sub-models will generate the combination of the "Number of combined parameters." For example, this number is specified as 3, the triplets are generated only for the specified multiple (3 or more) parameter interactions. After that, the other parameters that were not specified by the sub-model definition will be set for pairs. As a final result, the triplets are set with the specified multiple parameter interactions in sub-models, and the combinations with other parameters that were not specified in the sub-model are set to four parameters. After that, the other parameters that were not specified by the sub-model definition will be set for the pair with each other.

3. 6 Extended Sub-Models (weighted to parameters)

The sub-models feature supported by PICT makes test cases increase greatly, so it is not easy to use.
PictMaster allow you to generate the test cases for the combination of certain parameters that are different from the "Number of combined parameters" in the settings form. The test cases can be created only for the particularly important parameters if you specified **"Use extended sub-models"** in the settings form.
The extended sub-models are for specialized sub-models that generate certain combinations of more than two parameters that you wish to use. Compared with the usual sub-model, the increase in the number of test cases can be drastically reduced. For example, if the model has seven parameters with five values and the sub-models specify three of them for triplets, the generated test cases are as follows:

Sub-models -> 625 cases
Extended sub-models -> 129 cases

The number of generated test cases is 238 cases when you specified 3 in "Number of combined parameters" in the settings form. The number of test cases can be drastically reduced by using the extended sub-models feature. The format for the extended sub-models is shown in **Figure 3-27**.

<Parameter 1>,<Parameter 2> ,...... ,<Number of combined parameters (more than two) >

Figure 3-27 Format of extended sub-models

Enter the multiple parameter names for the extended sub-models separated by commas. Enter the number of combined parameters, which is more than two, up to the number of specified parameters in the extended sub-models. An example of the format is shown in **Figure 3-28**.

Figure 3-28 Format of extended sub-models (1)

For extended sub-models, you can enter only one sub-model in the first cell of sub-models columns. You cannot enter multiple sub-models in one cell by separating them with a semicolon (;).

If a minimized test case generation is conducted with the extended sub-models that have specified more than two parameters, the values of "Minimized number," "Maximized number," and "Initial

number" in the statistical information window are the number of combinations of the parameter interactions specified in the extended sub-models. The final number of generations is displayed in a separate item.

If the number of specified parameters and the number of combined parameters are equal, only one combination will be generated because the specified number indicates that all parameters interact with one another. So, you do not need to execute the minimized test case generation.

When you use the extended sub-models feature, the following conditions must be satisfied:

(1) More than one parameter must be specified.
(2) The number of combined parameters for extended sub-models must be greater than "Number of combined parameters" in the settings form.
(3) The number of combined parameters with extended sub-models has to be specified in the range between 3 and the specified quantity of the parameters by the extended sub-models (if the quantity of parameters is 2, specify 3).
(4) When more than two parameters are specified, "Use prototype sheet" and "Generate with specified coverage" must not be selected in the settings form.

As the **Figure 3-29** below, you can specify the "Number of combined parameters" in the settings form as 3, the quantity of parameters as 5, and the number of combined parameter interactions as 4.

Figure 3-29 Format of extended sub-models (2)

The combinations are different when two parameters are specified or when more than two parameters are specified in the extended sub-models. When two parameters are specified, these two will set the triplets with another parameter. When more than two parameters are specified, these will set the combination with each other and the specified parameter number. Also, each specified parameter will be paired with every other parameter.

The feature of extended sub-models reduces test cases significantly when a small number of parameter values are applied. On the other hand, it may not be effective for test case reduction when a large number of parameter values apply.

If more than two parameters are specified, the Excel worksheet number will be modified to the higher number because this feature uses the prototype sheet functionality.

3. 7 Generating Test Cases to Ensure the Desired Coverage

PictMaster allows you to generate test cases to ensure the desired coverage. If the number of combinatorial test cases with three parameter interactions is increased significantly, difficulties may arise. The test case, which is assigned 80 percent to three-way coverage, is able to drastically reduce the number of generated test cases.

For example, given the test cases where the model has 15 parameters and each quantity of value is 10, 8, 8, 6, 6, 6, 4, 4, 4, 4, 2, 2, 2, 2, 2, then change the three-way coverage in 5 percent increments. See the figure below:
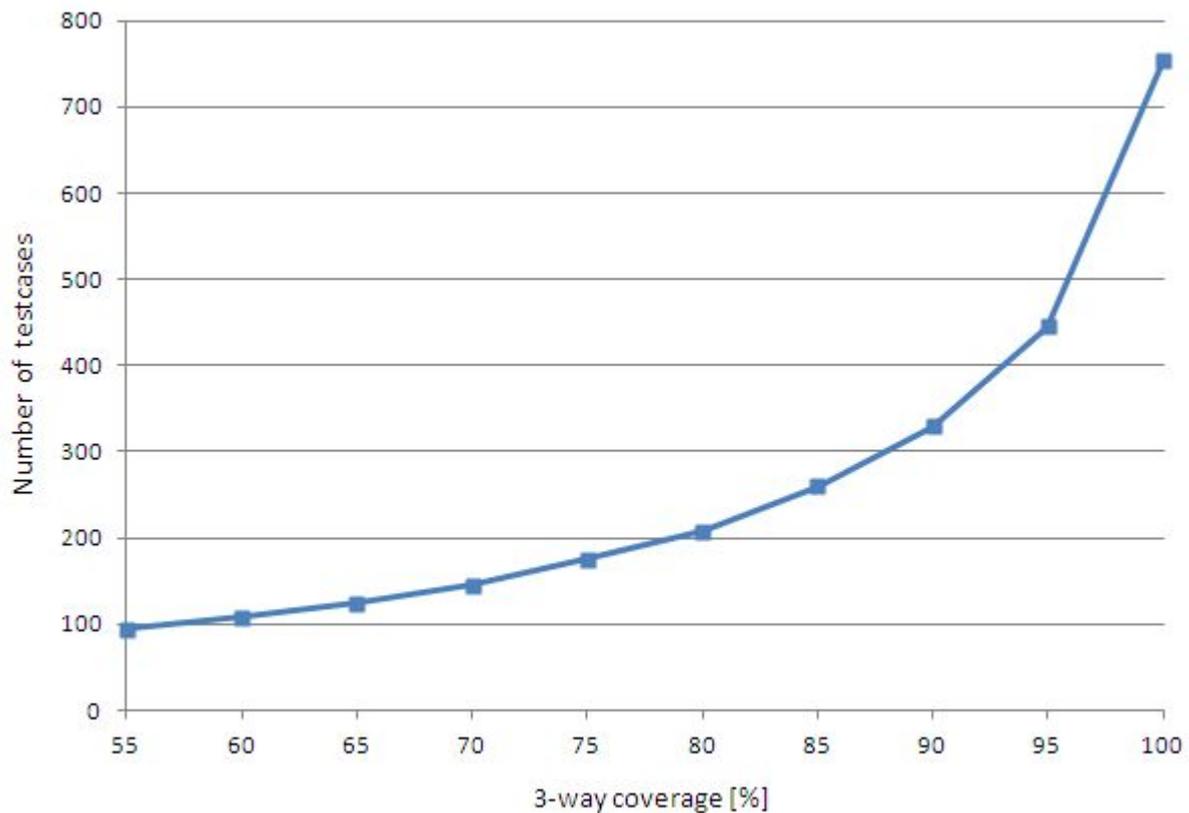
Figure 3.30 Number of test cases against changing 3-way coverage

As many as 216 test cases are generated when 80 percent of three-way coverage is applied. This output is 2.4 times the number of test cases that cover pairs and 28.6% of the number of test cases that cover the triplets. Compared to the increase in test cases, the output shows a slightly higher result for percent of three-way coverage to be covered. This can be worth executing the test for particularly important test cases.

You can specify from two-way to six-way coverage with the feature that generates the test cases to ensure the desired coverage. The t-way coverage where the value $n + 1$ (n specified in "number of combined parameters" in the settings form) is ensured. For example, when you specified 5 in "Number of combined parameters," five-way coverage ensures 100 percent and six-way coverage is ensured with the value in "Desired coverage."

4. Format of Constraints Table

4. 1 Definition of Terms for Constraints

Firstly, define the terms of constraints for the constraints table section.

Constraints indicate the unwanted combination of parameters and values. The expression that describes constraints in an IF…THEN function is defined as a **Constraint Expression**. You can exclude the unwanted combinations and leave the required combinations only.

Operators =, <> can be used for **Relational Expressions** to compare the parameters and values or parameters and parameters. PICT generates combinations based on the constraints that the constraint expression specifies.

The relations between IF and THEN are defined as **Constraint Conditions** and the relation after THEN is defined as the **Constraint Target**. The expression that uses the operator, an equal sign (=), is defined as an **Order Constraint** and the operator inequality sign (<>) is defined as a **Reverse Constraint**. The specified combinations in constraint targets are produced under the constraint conditions.

The constraint expression applied by the terms above is as follows:

IF **Constraint Conditions** THEN **Constraint Target** -> Constraint conditions are TRUE, combinations based on the **Constraints Target** will be generated.

The constraint expression above is a **Conditional Constraints Expression**. All constraint expressions that have constraint conditions are defined as a Conditional Constraints Expression. On the other hand, the constraint expression that does not have constraint conditions and just has a constraint target is defined as an **Unconditional Constraints Expression**. Only parameters are applied for an Unconditional Constraints Expression. The following is the basic unconditional constraint expression.

Parameter Operator Parameter

The **Unconditional Constraints** are **always declared** regardless of the value of the parameters.

4. 2 Constraints Table Structure

An example of a blank constraints table is shown in **Figure 4-1**.

Constraints table			
Parameters	Constraint 1	Constraint 2	Constraint 3

Figure 4-1 Blank constraints table

Figure 4-1 shows the blank table, and you can copy the parameters after column 8 in the model and then paste in the parameter columns of the constraints table. Also, you can paste the parameters automatically by clicking the OK button in the settings form. However, **if there are no constraints, copying the parameter columns is not necessary.** Each constraint 1, 2, or 3 has one constraint expression. Any values in the parameter columns of the model can be entered in the constraint columns.

You can enter another parameter itself.

Up to 50 parameters and up to 50 constraints can be entered. Fill in the parameter columns without leaving any empty cells. PictMaster outputs the description of parameters, value hierarchy, and constraints table into the model file a.txt and transfers the data to PICT. Also, PictMaster converts the description of the constraints table for PICT to understand the constraint expression.

4. 3. Specifying Constraint Conditions and Constraints Target

The formats of constraint conditions and the constraints target are almost the same. You can use both formats of constraint conditions and constraint target, which are described in the following section.

4. 3. 1 Conditional Constraints

The background color of a constraint cell for the **Constraint Conditions must be filled with any color** except white. The value or parameter in the colored constraint cell is recognized as a constraint condition.

Do not apply background color to the constraint cell for the **Constraint Target**. If you fill in white in the cell, it is treated as No Fill. The value or parameter in the uncolored constraint cell is recognized as a constraint target.

If you wish to enter multiple values in a constraint cell, separate each value by commas (,). The reverse constraint has a **number sign (#)** at the beginning of the value and indicates that the value is to be excluded in the constraint cell. In addition, you can enter multiple values separated by commas.

To compare the values, you can add the **greater-than sign (>) or less-than sign (<)** at the beginning of the value. If you add ">" and "<" to one value of the parameter, the range of values is specified. This applies to AND conditions between two values separated by a comma (,). If the value has a numeric value and character, it is treated as a character and is compared by character code. Only a numeric is acceptable.

For specifying the value of an alias, use the name of the first value.

When you enter the parameter itself, add the **equal sign (=)** for order constraint or add an **exclamation mark (!)** for a reverse constraint. If you wish to specify multiple parameters, separate each parameter by a comma (,) and add an equal sign (=) or exclamation mark (!) to each parameter. Please note that the direct parameter entry and value entry are not the same. In addition, parameter columns and constraint columns must have at least one identical value.

The constraint cell that is neither a constraint condition nor a constraint target must be blank.

The parameter columns in the constraints table can be blank when there are no constraints.

4. 3. 1. 1 Constraints for Parameters and Values

This chapter describes how to specify the constraints when a parameter has particular values.
Figures 4-2, **4-3, and List 4-1** show the model, the constraints table, and the generated model file a.txt, respectively. The model file in **List 4** only shows the expressions.

Parameters	Value hierarchy
A	a1, a2, a3
B	b1, b2, b3
C	c1, c2, c3

Figure 4-2 Model example (1)

Figure 4-3 Example of constraints table (1)

List 4-1 Example of constraint expression (1)

```
IF ([A] = "a1")
  THEN ([B] = "b1");
IF ([A] <> "a1")
  THEN ([B] <> "b1");
```

The final generated results are shown in **Table 4-1**.

Table 4-1 Example of generated results (1)

No.	A	B	C
1	a1	b1	c1
2	a1	b1	c2
3	a1	b1	c3
4	a2	b2	c1
5	a2	b2	c3
6	a2	b3	c2
7	a3	b2	c2
8	a3	b3	c1
9	a3	b3	c3

The specification of constraints shown in **Figure 4-3** is as follows:

Constraint 1: When parameter A has a value a1, the possible combinatorial value with parameter B is b1.

Constraint 2: When parameter A has values other than a1, the possible combinatorial value with parameter B is except for b1.

The generation results in **Table 4-1** show that when parameter A has value a1, the value of the parameter is only b1 and when parameter A has values other than a1, parameter B has values except for b1. This result is according to the specification of constraints in **Figure 4-3**.

Figures 4-2, 4-4, and List 4-2 show the model, the constraints table, and the generated model file a.txt, respectively.

Figure 4-4 Example of constraints table (2)

List 4-2 Example of constraint expression (2)

```
IF ([A] = "a1" or [A] = "a2")
   THEN ([B] = "b1");
IF ([A] = "a3")
   THEN ([B] <> "b1");
```

An example of generated results is shown in **Table 4-2**.

Table 4-2 Example of generated results (2)

No.	A	B	C
1	a1	b1	c1
2	a1	b1	c2
3	a1	b1	c3
4	a2	b1	c1
5	a2	b1	c2
6	a2	b1	c3
7	a3	b2	c1
8	a3	b2	c2
9	a3	b2	c3
10	a3	b3	c1
11	a3	b3	c2
12	a3	b3	c3

Figures 4-2, 4-5, and List 4-3 show the model, the constraints table, and the generated model file a.txt, respectively.

Figure 4-5 Example of constraints table (3)

List 4-3 Example of constraint expressions (3)

```
IF ([A] = "a1" or [A] = "a2") and [B] = "b1"
    THEN [C] <> "c2" and [C] <> "c3");
IF ([A] = "a3")
    THEN ([B] <> "b1") and ([C] = "c1");
```

The final generated results are shown in **Table 4-3**.

Table 4-3 Example of generated results (3)

No.	A	B	C
1	a1	b1	c1
2	a1	b2	c3
3	a1	b3	c2
4	a2	b1	c1
5	a2	b2	c2
6	a2	b3	c3
7	a3	b2	c1
8	a3	b3	c1

In the examples above, when multiple values are entered in one constraint cell, each value normally uses the OR condition. However, when values are entered in a different row of the constraint, each value uses the AND condition. These conditions are the same when the parameter itself is entered.

If you wish to specify different parameter values with OR conditions as the **Constraint Condition**, enter the different parameter values in different constraints as the constraint condition and enter the same constraint target.

If you wish to specify different parameter values with OR conditions as the **Constraint Target**, enter the same value for the parameter in the adjacent right constraint cell as the constraint target. The constraints table for this specification is shown in **Figure 4-6**. The following **List 4-4** shows the generated constraint expression and **Table 4-4** shows the generated results. When you specify multiple constraint conditions, all constraint conditions should be identical.

Constraints table			
Parameters	Constraint 1	Constraint 2	Constraint 3
A			a1
B	b1	b3	b3
C	c1	c2	

Figure 4-6 Example of constraints table (4)

List 4-4 Example of constraint expression (4)

```
IF ([B] = "b1")
    THEN ([C] = "c1");
IF ([B] = "b3")
    THEN ([C] = "c2") or ([A] = "a1");
```

Table 4-4 Example of generated results (4)

No.	A	B	C
1	a1	b1	c1
2	a1	b2	c1
3	a1	b2	c2
4	a1	b3	c1
5	a1	b3	c3
6	a2	b1	c1
7	a2	b2	c3
8	a2	b3	c2
9	a3	b1	c1
10	a3	b2	c3
11	a3	b3	c2

In this example, the conditions for constraint 2 and constraint 3 are integrated to one constraint expression and the constraint target uses the OR condition. The value in the same constraint uses the AND condition, and the values in the different constraints use OR conditions regardless of the number of constraint targets. However, **if the cells for constraint conditions in the adjoining constraints are filled with a different color, the constraint expressions are not integrated and produce two different constraint expressions**.

4. 3. 1. 2 Constraints for Parameter Interactions

The following figures show how to specify **a parameter in the constraint column**. **Figures 4-7**, 4-8, **List 4-5, and Table 4-5** show the model, the constraints table, the generated constraint expression and the generated results, respectively.

Figure 4-7 Model example (2)

Figure 4-8 Example of constraints table (5)

List 4-5 Example of constraint expression (5)

```
IF ([A] = "a1")
   THEN ([C] <> [B]);
IF ([B] = [C])
   THEN ([A] = "a3");
```

Table 4-5 Example of generated results (5)

No.	A	B	C	D
1	a1	1	3	3
2	a1	2	1	2
3	a1	3	2	1
4	a2	1	2	2
5	a2	2	3	1
6	a2	3	1	3
7	a3	1	1	1
8	a3	2	2	2
9	a3	2	2	3
10	a3	3	3	2

When you specify parameters that are conditional upon other parameters, specify the parameters **that contain the values that you need.** In addition, the value type of parameters, characters or numbers, must be the same. You can enter multiple parameters in one cell. Add an equal sign (=) or exclamation mark (!) at the beginning of each parameter and separate each parameter by a comma (,). Each parameter uses the OR condition.

 When specifying multiple parameters that use AND conditions in the constraint cell, add an **ampersand (&)** at the beginning of the operator.
Figures 4-7, **4-9, List 4-6, and Table 4-6** show the model, the constraints table, the generated constraint expression and the generated results, respectively.

Constraints table		
Parameters	Constraint 1	Constraint 2
A	a1	#a1
B	=C, &=D	!C, !D
C		
D		

Figure 4-9 Example of constraints table (6)

List 4-6 Example of constraint expression (6)

```
IF ([B] = [C] and [B] = [D])
   THEN ([A] = "a1");
IF ([B] <> [C] or [B] <> [D])
   THEN ([A] <> "a1");
```

Table 4-6 Example of generated results (6)

No.	A	B	C	D
1	a1	1	1	1
2	a1	2	2	2
3	a1	3	3	3
4	a2	1	2	3
5	a2	2	3	1
6	a2	3	1	2
7	a3	1	3	2
8	a3	2	1	3
9	a3	3	2	1

4. 3. 2 Unconditional Constraints

Unconditional constraints do not have constraint conditions and **are always declared**. Only parameters can be entered in a constraints table. The constraints for parameter interactions are specified by entering them with an operator in a constraint cell.

You can use the operators, **equal (=)** and **exclamation mark (!)** which means not equal. Also, multiple parameters can be entered in one constraint cell separated by commas (,). In this case, each of them uses the OR condition. The second and subsequent parameters may be preceded by an **ampersand (&)** so AND conditions can be used.

Figures 4-10, 4-11, List 4-7, and Table 4-7 show the model, the constraints table of reverse constraint uses AND condition, the generated constraint expression and the generated results, respectively.

Parameters	Value hierarchy
A	a1, a2, a3
B	1, 2, 3
C	1, 2, 3
D	1, 2, 3

Figure 4-10 Model example (3)

Constraints table

Parameters	Constraint 1
A	
B	!C
C	!D
D	!B

Figure 4-11 Example of constraints table (7)

List 4-7 Example of constraint expression (7)

([B] <> [C]) and ([C] <> [D]) and ([D] <> [B]);

Table 4-7 Example of generated results (7)

No.	A	B	C	D
1	a1	1	3	2
2	a1	2	1	3
3	a1	3	2	1
4	a2	1	2	3
5	a2	2	3	1
6	a2	3	1	2
7	a3	1	3	2
8	a3	2	1	3
9	a3	3	2	1

In this example, the constraint expression is integrated. If you enter values in different constraint columns, the expressions are generated individually as an identical constraint.

Figure 4-12, List 4-8, and Table 4-8 show the constraints table of order constraint uses OR condition, the generated constraint expression and the generated results, respectively. **To specify an OR condition, add the plus sign (+) at the beginning of the value to the right of the unconditional constraint.**

Constraints table			
Parameters	Constraint 1	Constraint 2	Constraint 3
A			
B	=C		
C		+=D	
D			+=B

Figure 4-12 Example of constraints table (8)

List 4-8 Example of constraint expression (8)

([B] = [C]) or ([C] = [D]) or ([D] = [B]);

Table 4-8 Example of generated results (8)

No.	A	B	C	D
1	a1	1	1	2
2	a1	2	2	3
3	a1	3	3	1
4	a2	1	2	1
5	a2	2	3	2
6	a2	3	1	3
7	a3	1	3	3
8	a3	2	1	1
9	a3	3	2	2

For example, given multiple parameters are entered in one constraint column, **Figure 4-12, 4-13, List 4-9, and Table 4-9** show the model, the constraints table, the generated constraint expression and the

generated results, respectively.

Parameters	Value hierarchy
A	a1, a2, a3
B	1, 2, 3
C	1, 2, 3
D	1, 2, 3
E	1, 2, 3
F	1, 2, 3

Figure 4-12 Model example (4)

Constraints table

Parameters	Constraint 1
A	
B	=C, !D
C	
D	!E, &=F
E	
F	

Figure 4-13 Example of constraints table (9)

List 4-9 Example of constraint expression (9)

([B] = [C] or [B] <> [D]) and ([D] <> [E] and [D] = [F]);

Table 4-9 Example of generated results (9)

No.	A	B	C	D	E	F
1	a1	1	1	3	1	3
2	a1	1	3	2	1	2
3	a1	2	2	2	3	2
4	a1	3	2	1	2	1
5	a2	1	3	3	2	3
6	a2	2	2	3	1	3
7	a2	2	3	1	3	1
8	a2	3	1	2	3	2
9	a3	1	1	1	3	1
10	a3	1	2	2	3	2
11	a3	2	1	1	2	1
12	a3	3	3	3	1	3

4. 4 List of Usable Operators

The operators to be used for a relational expression are different from the expressions for parameters and values and the expressions for parameters and parameters. **Table 4-10** shows the operators that can be used. All operators except for # can be entered several times separated by commas (,). The condition columns represent OR or AND condition where they are to be used by multiple values or parameters in the constraint cell.

<div align="center">Table 4-10 Usable Operators</div>

Relational expression	Operator	Condition	Description
Parameters and values	(Not entered)	OR	Value belongs to the parameter. Same as =.
	#	AND	Values excluding the entered value. This is added at the beginning of the value and can be entered more than once.
	>, <	AND	Greater than or less than values.
Parameter and parameter	=	OR	Values of parameters are equal.
	!	OR	Values of parameters are not equal.
	&=	AND	The AND condition requires both parameter values must be equal. If you enter multiple parameters, add &! before the second and subsequent parameters.
	&!	AND	The AND condition requires both parameter values must be equal. If you enter multiple parameters, add &! before the second and subsequent parameters.
	+	(OR)	Different constraints based on the unconditional constraints use OR conditions. This is added at the beginning of the first parameter with unconditional constraints in the constraint columns on the right side.

The condition for the operator (+) that represents (OR) means conditions between the different constraints.

These operator symbols cannot be used at the beginning of the names of parameters or values.

4. 5 Editing Constraints Table

This section describes how to edit a constraints table.

If the number of rows in the constraints table has changed, the rows in the expected results table cannot be met. **Do not delete or insert a row in the constraints table** from the Excel menu. Instead, **right-click the first column of the row** you wish to delete or insert. The shortcut edit menu to add, delete, and undo will appear and select the menu item to delete or insert rows. To delete or insert constraints, **right-click the title cell of a constraint**. The shortcut edit menu to add, delete, and undo will appear and select the menu item to delete or insert constraints. The details are described in **Chapter 6.4**.

4. 6 Using Wildcards

You can use **wildcards** for the values to be entered in a constraints table. A wildcard is a special character that may be substituted for "Any of all possible characters." The " * " substitutes for any zero or more characters, " ? " substitutes for any one of the characters. For example, the name "A*" substitutes for all values that begin with "A" including just "A." "???A" substitutes for all four characters that have "A" at the end.

Wildcards can be used only for the names of values of conditional constraints and cannot be used for the numerical values in a constraints table. If the parameter value has a mixture of numbers and characters, the value type is recognized as characters.

An example of how wildcards are used and the constraint expression generated by PictMaster are shown below:

Constraints table	
Parameters	Constraint 1
A	
B	*B*
C	#C??

Figure 4-14 Example of wildcard usage

List 4-10 Constraint expression including wildcards

IF ([B] LIKE "*B*")
    THEN [C NOT LIKE "C??");

Add the appropriate characters to the name value to enter the constraints quickly by using wildcards.

5. Format of Expected Results Table
5. 1 Expected Result Table Structure

The test case results may be different from the combinatorial test cases. If you have a number of test cases, assignment of expected results (referred to as "result") is labor intensive, but the expected results table is useful to simplify this task. When repeatedly generating test cases, the time savings are significant. An example of a blank expected results table is shown in **Figure 5-1**.

Figure 5-1 Blank expected results table

In the **results column** of an expected results table, specify the possible result outcomes for the combinatorial values in each parameter column. If the results require a long entry, enter a number such as (*1) (*2) ... in the results column and describe the details in the margin of the worksheet for the generated test cases. The results column has 50 rows and 15 rows are in view by default. Unhide the rows, if necessary. To assign the names of parameters at the top of the parameter columns in the expected results table, display the settings form and then click the OK button. Fill in the results columns **without leaving any empty cells**.

In the **Parameter columns** of an expected results table, specify the combinatorial values that meet the left result column (referred to as **rules)** with the same form as the constraints table. You cannot enter the parameter itself. The parameter columns have 50 rows.

5. 2 Specifying Rules

Rules are defined in the same way as the value specification of a constraints table. Enter the combinatorial values in each parameter cell in the context of the left result column. In addition, you can enter multiple values separated by commas (,) in the cell. The reverse constraint has a number sign (#) at the beginning of the value and indicates that the value is to be excluded in the constraint cell. The fill color is not required.

Select **"Use expected results table"** in the settings form and click OK, the names of parameters in the parameter columns of the model are automatically input to the top of the parameter columns in the expected results table.

The rules must not be violated the constraints table. If one of the **results does not have any rules in the parameter row,** the contents are assigned to the results of the test cases where the results did not agree with any of the other rules.

Figures 5-2 and 5-3 show examples of a model and an expected result table, respectively.

Figure 5-2 Model example

Figure 5-3 Example of expected results table

When specifying the expected results, press Control + e (the default) to split the window, then the window displays the model and the expected results table to make it easier to enter. Four results are specified in the result columns in **Figure 5-3** above, no rules are specified in the fourth row. The final output will be assigned to the results of test cases that did not match the other three rules. You can specify the same expected results in the first two rows.

The rule uses the AND condition with parameters. The left result columns are input for the test cases only when all the entered rules are matching. Alias value names include all values. An example of the generated test cases based on the rules assigned from **Figures 5-2** to **5-3** is shown in **Table 5-1**.

Table 5-1 Generated test cases

No.	A	B	C	Results
1	a1	b1	c3	1111
2	a1	b2	c1	1111
3	a1	b3	c1	3333
4	a1	b4	c2	3333
5	a2	b1	c1	1111
6	a2	b2	c3	1111
7	a2	b3	c2	3333
8	a2	b4	c3	3333
9	a3	b1	c3	3333
10	a3	b2	c3	3333
11	a3	b3	c2	2222
12	a3	b4	c1	3333
13	a4	b1	c2	2222
14	a4	b2	c2	2222
15	a4	b3	c3	1111
16	a4	b4	c1	3333

In the generated test cases, the results columns are added to the right to ensure results that met the rules are in the expected results table. If there are dozens of test cases, this expected results table feature is useful for saving a lot of time when repeatedly generating test cases. For specifying the value of an alias, use the name of the first value.

5. 3 List of Usable Operators

    Table 5-2 shows the operators that can be used for the rules. You can enter multiple values separated by commas (,). The condition columns represent OR or AND condition where they are to be used by multiple values in one rules cell.

Table 5-2 Usable Operators

Relational expression	Operator	Condition	Description
Parameters and values	(Not entered)	OR	Values belong to the parameter. Same as =. You can enter multiple values separated by commas (,).
	#	AND	Values excluding the entered value. This is added at the beginning of the value and can be entered more than once.

5. 4 Notes for Entering

  This section reminds you to enter the expected results table.

(1) Duplicating Rules
  When defining multiple expected results, two different results for the same combination may be defined. This occurs because the rule is duplicated. In this case, PictMaster displays an error message as in **Figure 5-4** and cancels the process. To handle this error you can add a rule using "#" - reverse condition- to avoid the duplication.
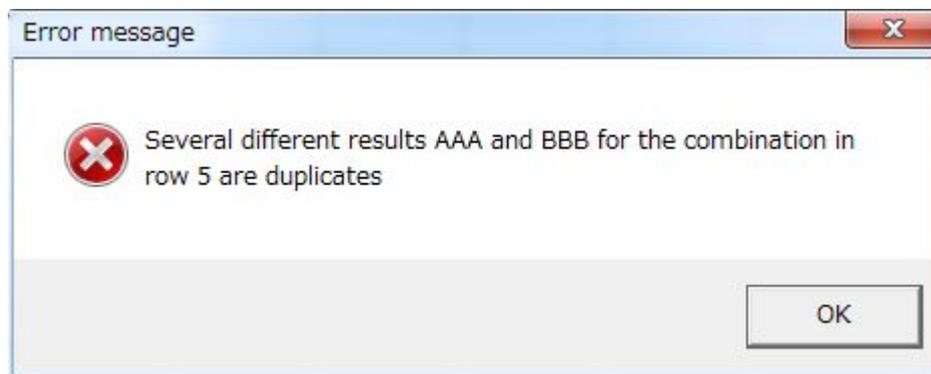

Figure 5-4 Example of the rule duplication error message

(2) Inconsistency in Rules
  There may be no conditions that match the rules. The discrepancy may happen very often if you specify the number of combinatorial parameters as 2 and more than two parameter values for the rule, or the rules that specify the combination violated the content of a constraints table. In this case, PictMaster displays an error message as in **Figure 5-5** and cancels the process.
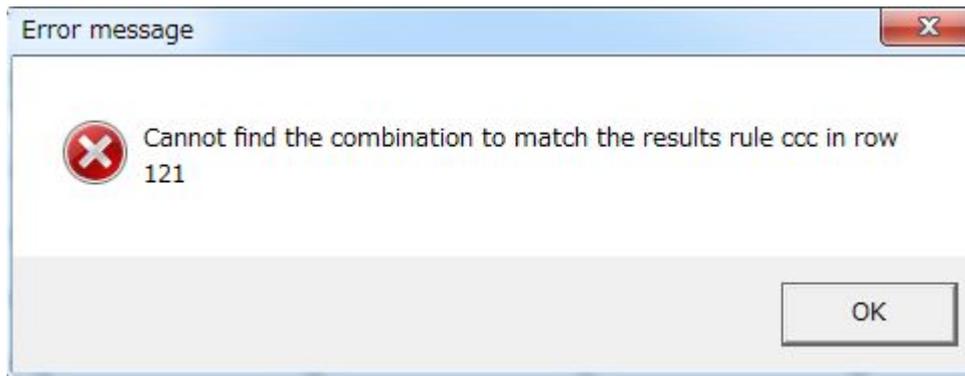
Figure 5-5 Example of the rule inconsistency error message

5. 5 Consolidating Results

As described above, the expected results table is used to show that one test case agrees with only one result. However, defining the expected result based on this procedure may be complicated depending on the test case, which has a number of parameter values that will vary the results.

For example, given the model shown in **Figure 5-6**, when the result is changed by the values of parameter A or B, you need to specify nine results (3 x 3) in the expected result table as in **Figure 5-6**. The generated results are shown in **Table 5-3**.

Parameters	Value hierarchy
A	a1, a2, a3
B	b1, b2, b3
C	c1, c2, c3
D	d1, d2, d3

Expected results table		
Result	A	B
A1B1	a1	b1
A1B2	a1	b2
A1B3	a1	b3
A2B1	a2	b1
A2B2	a2	b2
A2B3	a2	b3
A3B1	a3	b1
A3B2	a3	b2
A3B3	a3	b3

Figure 5-6 Example of model and expected results table

Table 5-3 Example of generated results

No.	A	B	C	D	Result
1	a1	b1	c2	d1	A1B1
2	a1	b2	c3	d1	A1B2
3	a1	b2	c3	d2	A1B2
4	a1	b3	c1	d3	A1B3
5	a2	b1	c1	d2	A2B1
6	a2	b2	c2	d3	A2B2
7	a2	b3	c3	d1	A2B3
8	a3	b1	c3	d3	A3B1
9	a3	b2	c1	d1	A3B2
10	a3	b3	c2	d2	A3B3

According to the previous description, in this example, the result for each combination must be entered in the expected results table. The generated results from a small quantity of values are also a small number. However, if you have a large quantity of values, the results columns will be increased significantly.

When the quantity of parameter values for one test case will be varied, you can reduce the results to be specified in the expected results table by inputting the result individually for each parameter value, but not the result for each combinatorial parameter value. The "Consolidated" contents, which met the rules, are assigned in each result and parameter column. An example of the table for consolidated results is shown in **Figure 5-7**.

Expected results table		
Result	A	B
A1	a1	
A2	a2	
A3	a3	
&B1		b1
&B2		b2
&B3		b3

Figure 5-7 Example of consolidated expected results table

The results that were generated based on the consolidated expected results table are exactly the same as **Table 5-3**. For example, given the name of a result starts with an ampersand (&), if values of test cases are different from the results defined by the rules in the expected results table, each result is "**Consolidated**" and re-created as the new result. In other words, when the content in the "Results" of generated test cases and the "Results" that are defined by the rules are different, each name of the result is "**Consolidated**" and re-created as the new name. In this example, the result is computed as 3 + 3 = 6 instead of 3 x 3 = 9. The difference will become greater when the values that have an impact on the results are increased.

The "Duplication in rules" error message is displayed when the results in the expected results table that match the rules and the results in the test cases are different, then the latter results do not have an ampersand (&) at the beginning.

6. Advanced Usage
6. 1 Customizing PictMaster

PictMaster is a sophisticated Excel workbook. So, you can customize it for ease of use. Rows 1 to 7 are for the user to layout. You can re-name the file name or worksheet title of PictMaster.

Also, PictMaster **allows you to add any worksheets for test cases. Select "Move or Copy" from the edit menu and the form will appear, select the "Create a copy" box and click the "OK" button.** However, if you use the Insert menu to create a new worksheet and copy and paste the PictMaster sheet to the created worksheet, it will not work properly.

We recommend that you create the PictMaster workbook for the test cases and manage the models of combinatorial test cases dispersed amongst multiple worksheets. If you have created a test specification in Excel, you can input the test specification for the test cases to be generated with PictMaster to another worksheet **so that the test specification and PictMaster are in the same workbook**. In that case, you should change the workbook name from "PictMaster" to one describing the test cases. An example of the names of the worksheet sequence is shown in **Figure 6-1**.
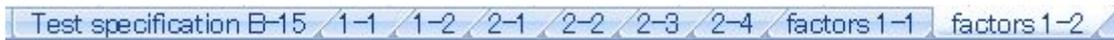
Figure 6-1 Example of worksheet name sequence

In this example, the worksheet name "Test specification B-15" represents the test name and workbook name. This is a worksheet for **Test design specification** (*1) that describes the test method, verification, and data settings for the test cases in worksheets 1-1 to 2-4. These worksheets are the **Test design specification** (*1) corresponding to each verification. In these test cases, the names having "factors" are the PictMaster worksheets used for the combinatorial test cases generated with PictMaster.

When you customize, do not change the row or column numbers of the cells that have parameters, value hierarchy, sub-models, constraints table, or expected results table. VBA will no longer to have the ability to recognize the location of such models.

*1: IEEE 829 standard documentation. The details of the test procedure are included in the test
    design specification or test case specification.

6. 2 Managing Error/Warning Messages

PictMaster has many verification options for the process of conversion from a constraints table to constraint expression. So, PICT rarely displays an error/warning message.
While an error/warning message is displaying, you can open the file e.txt, which contains the message, with Notepad. When opening the file, click the OK button of the error (warning) message and then you can investigate the mistake in parameter definitions or constraints table in PictMaster while looking at the error/warning message in Notepad. Also, when you click the OK button of the warning message, the test cases are generated. You can confirm the generated result by opening the a.xls file.

6. 2. 1 Quick Check for Conflicting Constraints

PICT displays the message shown in **Figure 6-2** very often.
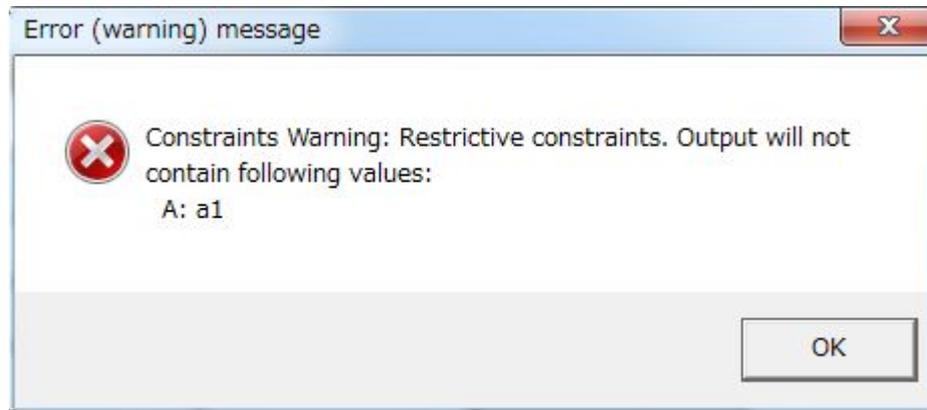
Figure 6-2 Example of PICT message

  This example shows one warning message where the combination does not contain the value a1 of parameter A. In some cases, five or six errors are indicated. The message is displayed when you incorrectly specified mutually conflicting multiple constraints. To find the wrong constraints, you should review the constraints table, which may have a conflict, such as the specified parameter values cannot be combined. When more than one error is indicated, focus on one at a time.

  The simplest constraint that produces a warning message is shown below:

Figure 6-3 Example of conflicted constraint (1)

    The value of the constraint conditions and the corresponding constraint target value are not met in this constraints table. Therefore, the value a1 does not appear in the combination. You can easily recognize the mistake in this table. However, normally a large number of constraints cause the conflict between constraint conditions and constraint targets. To define whether the **constraints are conflicting or not, you need to verify that there are two different values in the cells in the same row. The values must not be the same so the relationship will not conflict.**

6. 3 Splitting the Screen for Easy Entry of a Constraints Table

  When you have a large number of constraints, you need to scroll the screen horizontally to the right so that the parameter column is no longer visible. You may scroll the screen horizontally to the left to make sure of the parameter name. This happens when entering an expected result table. There is an easy way to avoid this hassle.

  Click the settings button to display "Split window shortcut key" and enter a single letter (default is "e") in it. Then the PictMaster window will be split into two parts by pressing Control + the letter you entered. The windows are arranged one above the other, the lower window will be split between the parameter column and the constraints columns. Specify the shortcut key to be enabled on the settings

form of each worksheet when PictMaster contains multiple worksheets.

When opening PictMaster 4.3 or later and 4.22 or a previous version at the same time, the split window shortcut key for the older version will be disabled. If you wish to use the split window shortcut key for the older version, you need to exit Excel. An example of the screen using the split window shortcut key is shown in **Figure 6-5**.
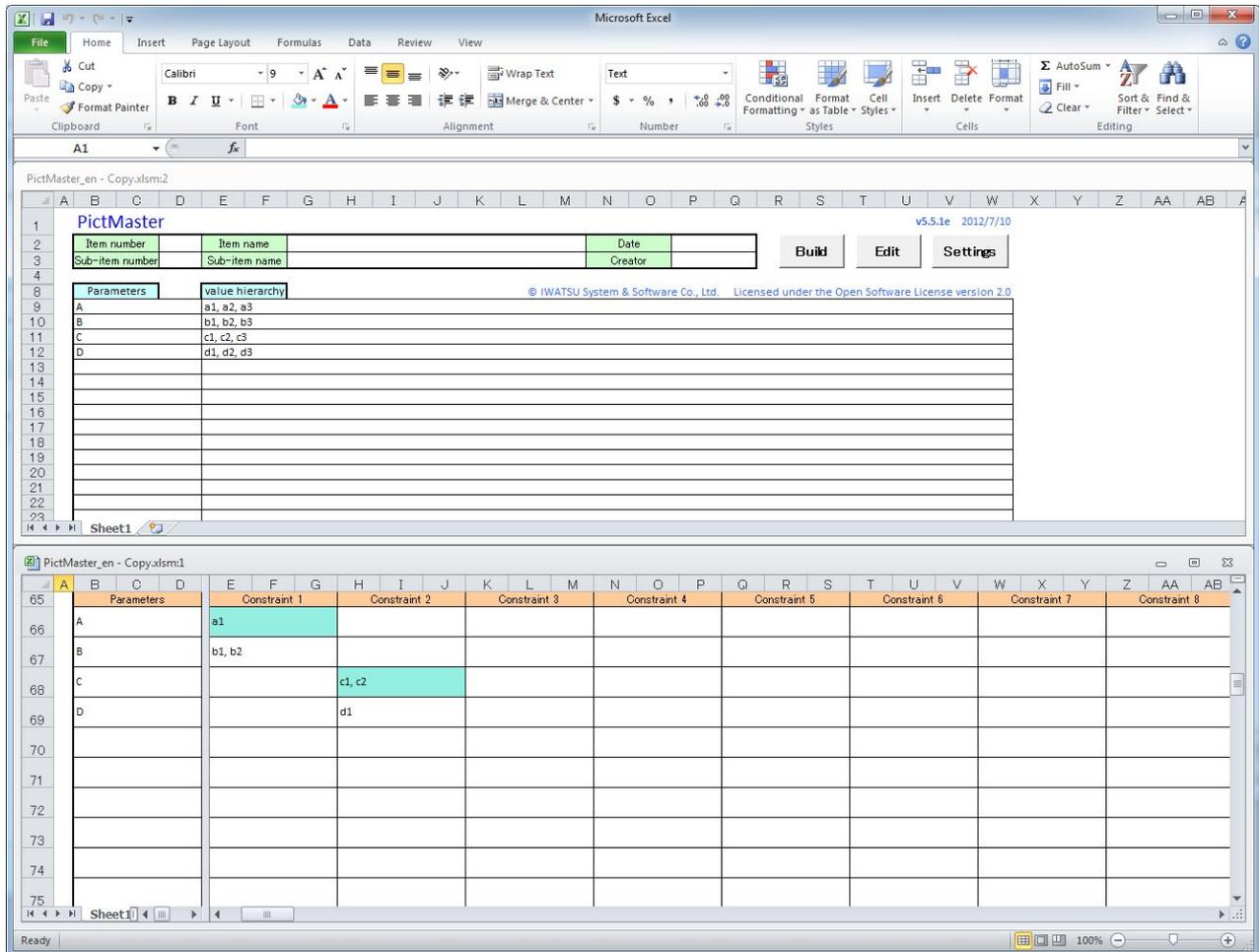

Figure 6-5 Example of split window

In this screen, the parameter column is always visible even when you scroll horizontally. This is useful when entering a constraints table that requires a number of constraints. If you wish to return to the original screen, press the Control + shortcut key. The split window may be changed if you switch to another worksheet or a workbook. You can return to the split window by pressing the Control + shortcut key several times. **When selecting another worksheet in the lower window, the selected worksheet will open with the original zoom (magnification).** When selecting another worksheet in the upper window, the selected worksheet will open with 100% zoom.

6. 4 Using the Shortcut Edit Menu

You can edit the parameters or value hierarchy columns and constraints table or an expected results table by using the shortcut edit menu.

When right clicking the first column, the shortcut edit menu that is shown in **Figure 6-6** will appear. You can shift a row up or down or insert or delete a row by clicking the shortcut menu.

Figure 6-6 Shortcut edit menu for rows

When right clicking the constraint title in the constraints table, the shortcut edit menu that is shown in **Figure 6-7** will appear. You can shift a column right or left insert or delete a column by clicking the shortcut menu.

Figure 6-7 Shortcut edit menu for columns

If you deleted a row or column, you can revert to the original state by clicking "Undo Delete" in the shortcut menu. One row or column can be edited in one click. The last row or column is not enabled for insert or delete.

Appendix A Specifications

No.	Item	Value
1	Maximum number of parameters	50
2	Maximum number of per parameter values	50
3	Maximum number of constraints in the constraints table	50
4	Maximum number of rules in the expected results table	50
5	Maximum spread of parameter values that can be specified in one row of the expected results table (*1)	500
6	Maximum number of rows in the expected results table to generate results	65535
7	Coverage for combined parameter interactions	1 to 50
8	Range of minimizing the number of test cases generation	2 to 999
9	Range of minimizing the number of test cases generation seed	0 to 65535
10	Maximum number of editable rows	65535
11	Maximum number of rows in the prototype sheet	65535
12	Maximum number of sub-models you can specify	Depending on PICT
13	Maximum number of extended sub-models you can specify	1
14	Range of number of combined parameters in extended sub-models	1 to number of parameters specified in the extended sub-models
15	Range for assigning weights to values	2 times to 10 times
16	Maximum number of rows of duplicated combinations that can be deleted	65535
17	Range for the number of combinations that you can specify coverage	2 to 6
18	Range for t-way coverage that you can specify	30 to 95 (%)
19	Range for number of times to repeat the generation of the specified coverage	3 to 10

*1: Total number of values specified in the parameters row. If the value is including an alias, the number of values specified with the alias value should be added to the count.

Appendix B Limitations
[Important Reminder]

When specifying many parameters, values, constraints, or combinations, **it may take quite a long time to generate the test cases** because of the PICT capacity. When you wish to cancel the process, start the Task Manager and select **"pict.exe"** from the process to exit from PICT.