About this manual

- This manual is explained about the main specification of DT10, a basic operation method, an additionally detailed function, and the setting.
- Neither this program nor the manual are protected by the Copyright Law, and can do the reproduction, the reprint, and the modification at all.
- The content and the specification of this product might be changed without a previous notice.
- Please acknowledge that our company cannot assume all the responsibilities about the result used.
- Microsoft, Windows, and the Windows logo are registered trademarks in the United States and other countries of United States Microsoft Corporation.
- Described other company name, logo, and product name, etc. are the trademarks or registered trademarks of corresponding each company.
# Contents

1. OUTLINE OF DT10
   1. Outline
   2. Main function
   3. About the support
   4. About the license

2. SYSTEM REQUIREMENTS
   1. Operating system
   2. Computer main body
   3. Memory
   4. Hard disk
   5. Display

3. SCREEN STRUCTURE
   1. The main window
   2. Menu
     - File
     - Plan
     - Test
     - Report Analysis
     - Test Result
   3. Tool bar
     - Standard
     - Test Plan
     - Test Execution
     - Report Analysis
     - Source code window
     - Test Report

4. START OF APPLICATION
   1. Multi edition
   2. Standard edition
   3. Lite edition
# 5. CREATING AND CHANGE IN PROJECT

5.1. New creating of Project ................................................................. 20

5.1.1. Project Property ....................................................................... 22
5.1.2. Register source file ................................................................. 24
5.1.3. Specify auto generated file ....................................................... 25

5.2. Change in Project ........................................................................ 32

5.2.1. Project Setting Dialog .............................................................. 32
5.2.2. Addition of source file .............................................................. 33
5.2.3. Registration release of source file ............................................ 33

# 6. THE RULE OF AUTOMATIC INSERTION OF TEST POINT

6.1. Outline of Automatic Insertion .................................................... 34

6.2. Character code and Newline code for source code ....................... 37

6.2.1. Outline ............................................................................... 37
6.2.2. Character code and Newline code setting for auto judgement .. 39
6.2.3. Character code and Newline code setting for each Source file .. 41
6.2.4. Character code and Newline code setting for auto judgement .. 43
6.2.5. Limitation ........................................................................... 44

6.3. Rules of automatic insertion [C/CPP] .......................................... 45
6.4. Limitations of automatic insertion [C/CPP] ................................. 47
6.5. Rules of automatic insertion [Java] ............................................. 50
6.6. Limitations of automatic insertion [Java] .................................... 52
6.7. Automatic insertion example of Test Point .................................. 53

# 7. INSERTION OF THE TEST POINT

7.1. Automatic Insertion of the Test Point ......................................... 61

7.2. Manual insertion of Test Point .................................................... 64

7.2.1. Insert Test Point .................................................................... 64
7.2.2. The function of the Test Point insertion destination is defined .. 65

7.3. Delete Test Point ........................................................................ 66

7.3.1. Delete Test Point ................................................................... 66
7.3.2. Delete Test Point by the batch ............................................... 67

7.4. Convert to the Fast Test Point ..................................................... 68

7.4.1. Outline ............................................................................... 68
7.4.2. Fast Test Point conversion .................................................... 69
7.4.3. Release of the Fast Test Point ................................................ 70

7.5. Enable or Disable Test Point ....................................................... 71
7.5.1. Enable or Disable specified Test Point ................................................................. - 71 -
7.5.2. Create header file ................................................................................................. - 73 -
7.5.3. Disable all Test Points ......................................................................................... - 75 -

8.  OPERATION ON SOURCE CODE WINDOW .................................................................... - 76 -
8.1. Find ....................................................................................................................... - 77 -
8.2. Color Setting ......................................................................................................... - 78 -
8.3. Font size Change .................................................................................................... - 80 -
8.4. Source code edit .................................................................................................... - 80 -
8.5. Line Jump ............................................................................................................. - 83 -
8.6. Find Test Report .................................................................................................... - 84 -

9.  SET THE PROPERTY ..................................................................................................... - 85 -
9.1. Type of Property ..................................................................................................... - 85 -
9.2. How to open the Property Editor .......................................................................... - 86 -
9.3. Source File Property .............................................................................................. - 88 -
9.4. Module Property .................................................................................................... - 90 -
   9.4.1. Module function of the DT10 ........................................................................... - 90 -
   9.4.2. Automatic Generation of Module .................................................................... - 91 -
   9.4.3. Addition of module ......................................................................................... - 92 -
   9.4.4. Deletion of module ......................................................................................... - 94 -
   9.4.5. The function is added and excluded to the module ........................................... - 95 -
9.5. Function Property .................................................................................................. - 98 -
9.6. Step Property ......................................................................................................... - 102 -
9.7. Confirm setting Property ....................................................................................... - 106 -

10. GETTING TEST REPORT ............................................................................................ - 108 -
10.1. Set Test Report collection condition ...................................................................... - 108 -
10.2. Setting of Ethernet connection ............................................................................ - 111 -
10.3. Setting of UART connection ............................................................................... - 113 -
10.4. Setting SD connection .......................................................................................... - 115 -
10.5. Getting Test Report .............................................................................................. - 116 -
   10.5.1. “Execute program” dialog ............................................................................. - 116 -
   10.5.2. “Getting Test Report” dialog ........................................................................ - 118 -
10.6. Bookmark Function .............................................................................................. - 120 -
   10.6.1. Set the Mark in getting the report ................................................................. - 120 -
   10.6.2. Test Report .................................................................................................... - 120 -
   10.6.3. Mark View .................................................................................................... - 121 -
10.7. Ignore the collected data Mode ............................................................................ - 123 -
   10.7.1. Ignore the collected data ............................................................................... - 123 -
11. TEST REPORT ................................................................. 131
   11.1. Test Report ................................................................. 131
   11.2. Change the unit of time in the Test Report ................. 135
   11.3. Use the jump function in the Test Report .................. 136
   11.4. Trace the place of Test Point from the Test Report .......... 137
   11.5. Retrieve in the Test Report from Folder View or Module View ........................................ 138
   11.6. Copy the Test Report data to the clipboard .......... 139
   11.7. The Test Report view is seen easily and set ........ 140
       11.7.1. Filter setting ...................................................... 140
       11.7.2. Classification setting ........................................... 141
   11.8. The Test Report is retrieved ................................... 142
       11.8.1. Open the Search settings .................................... 142
       11.8.2. Search for current Report Data ............................ 144
   11.9. Test Report Trimming function ................................. 145
   11.10. The Report line is selected by order sending ........... 146

12. ANALYZE TEST REPORT .................................................. 147
   12.1. Analyze Test Report .................................................. 147
   12.2. Multiple Analysis of Test Reports ............................ 148

13. COVERAGE REPORT ........................................................ 150
   13.1. The type of the coverage which can checked by DT10 .......... 150
   13.2. C0 Coverage Report .................................................. 150
       13.2.1. Calculation method of C0 coverage ......................... 150
       13.2.2. Display C0 Coverage Report ................................ 153
       13.2.3. Fundamental motion of Coverage Report ................. 155
       13.2.4. Not Executed TP List .......................................... 157
   13.3. C1 Coverage Report .................................................. 158
       13.3.1. How to calculate C1 coverage ................................ 158
       13.3.2. Display C1 Coverage Report ................................ 161
       13.3.3. C1 Coverage Detailed Report ............................... 164

14. EXECUTION TIME REPORT ................................................ 167
   14.1. Execution Time Report ................................................ 167
   14.2. Fundamental motion of Execution Time Report .............. 169
14.3. Execution Time Detail List ................................................................. 171
14.4. Fundamental motion of Execution Time Details List ......................... 172
14.5. Execution Time graph ........................................................................ 174
14.6. Basic movement of Execution time graph ............................................ 175
14.7. Execution time histogram ................................................................... 177
14.8. Basic movement of Execution time Histogram ..................................... 178

15. PERIOD TIME REPORT ........................................................................ 180

15.1. Period Time Report ............................................................................. 180
15.2. Fundamental motion of Period Time Report ......................................... 182
15.3. Period Time Detail List ......................................................................... 184
15.4. Fundamental motion of detail list of Period Time ................................. 185
15.5. Period time graph ............................................................................... 187
15.6. Basic movement of Period time graph ............................................... 188
15.7. Period time histogram .......................................................................... 190
15.8. Basic movement of Period time histogram ......................................... 191

16. LOOP COUNT REPORT ......................................................................... 198

16.1. Setting of the Loop Count (Step Property) ........................................... 198
16.2. Loop Count Report ............................................................................. 195
16.3. Loop Count Detail List ......................................................................... 197
16.4. Fundamental motion of Loop Count Detail List ................................... 198

17. FUNCTION TRACE REPORT ................................................................. 200

17.1. Specification of analysis range .............................................................. 200
17.2. Function Trace Report ........................................................................ 202
17.3. Fundamental motion of Function Trace Report ..................................... 203
17.4. Display Trace Report from the Test Report ......................................... 206
17.5. The transition of the task is display by using Event Trigger ................... 207
17.6. Function Trace display that considers filter setting ............................. 208

18. SET THE DISPLAY FILTER TO THE TEST REPORT ........................... 209

19. LOOK AT CPU LOAD DETECTION INFORMATION IN THE PERFORMANCE MONITOR ........................................................................ 221

19.1. CPU Load detection ............................................................................ 221
  19.1.1. Mechanism of CPU Load Detection ................................................ 221
  19.1.2. Specify CPU Load Detection ........................................................... 221
  19.1.3. Insert CPU Load Measurement Point .............................................. 222
  19.1.4. Test Report ..................................................................................... 223
19.2. Performance Monitor ........................................................................... 224
20. EVENT TRACE FUNCTION ................................................................. 227

19.3. Re-analysis of CPU load measurement data .................................................. 225
19.4. The overhead is measured ........................................................................... 225
19.5. The image of a graphical monitor is copied onto the clipboard ....................... 226

20.1. Outline ................................................................................................. 227
20.2. Insert Event Trigger Point ........................................................................... 227
20.3. Making of driver function .......................................................................... 229
20.4. Test Report Data ....................................................................................... 229
20.5. Event Definition ......................................................................................... 230
20.6. Event Analysis .......................................................................................... 232
20.7. Event Trace View ....................................................................................... 234
20.8. Switch of analytical mode .......................................................................... 239
20.9. Import/Export function of Event definition .................................................. 240
20.9.1. Export function ..................................................................................... 240
20.9.2. Import function ...................................................................................... 242
20.9.3. Description rule of Event definition file .................................................. 243

21. THE SPECIFIED VALUE OF A VARIABLE IS DISPLAYED IN THE TEST REPORT 245

21.1. Outline ................................................................................................. 245
21.2. Variable Property value setting .................................................................... 247
21.3. Automatic insertion of Variable Value Point ............................................... 251
21.4. Manual insertion of Variable Value Point ....................................................... 253
21.5. Property Editor ........................................................................................ 254
21.6. Interchangeability with existing Project file .................................................. 255
21.7. Test Report ............................................................................................. 256

22. VARIABLE MONITOR FUNCTION ...................................................... 257

22.1. Outline ................................................................................................. 257
22.2. Setting of Variable Monitor ....................................................................... 257
22.3. Variable Monitor ...................................................................................... 258

23. MULTI WAVE SCOPE ........................................................................... 260

23.1. Outline ................................................................................................. 260
23.2. Collecting the display data ......................................................................... 261
23.2.1. Collecting the data of the Analog Box .................................................... 261
23.2.2. Collect the Event Trigger data, Variable Value Output data ..................... 264
23.3. Fundamental motion in Multi Wave Scope .................................................... 265
23.4. Variable value graph .................................................................................. 267
23.4.1. Display Variable value graph .................................................................. 267
23.4.2. The bit specific inside variable ................................................................ 268
23.5. Playback Function ................................................................. 269

24. VARIABLE VALUE WRITING FUNCTION .................................................. 271
   24.1. Outline........................................................................ 271
   24.2. Insert Write Test Point .................................................. 272
   24.3. Change in Property setting ............................................ 275
   24.4. Effective specification of Write Test Point ...................... 276
   24.5. Test Report ............................................................... 278

25. CREATE TEST RESULT ........................................................................ 279
   25.1. About Test Result ........................................................... 279
   25.2. Create Test Result ......................................................... 280
   25.2.1. Selection of Report Data ........................................... 280
   25.2.2. Input to the Test Result ............................................. 281
   25.2.3. Confirmation of Test Results ....................................... 286
   25.2.4. Print the Test Result .................................................. 287
   25.2.5. Save and Open the Test Result .................................... 287

26. SETTING ....................................................................................... 288
   26.1. Setting dialog .................................................................. 288
   26.1.1. Display – Document Window ....................................... 289
   26.1.2. Display – Report Data ................................................ 290
   26.1.3. Analyze .................................................................... 291
   26.1.4. Analog Box .............................................................. 292
   26.1.5. Other ........................................................................ 293
   26.2. About connection test ..................................................... 294
   26.2.1. Explanation of dialog ................................................ 294
   26.2.2. Connected test procedure ........................................ 295

27. DISPLAY THE REPORT DATA .............................................................. 298
   27.1. About the management of Report Data ......................... 298
   27.2. Structure of Profile management .................................... 298
   27.3. Renewal of Profile .......................................................... 299
   27.4. Management of Report Data .......................................... 301
   27.5. Displays Test Report ...................................................... 304
   27.6. Rollback of Project ........................................................ 306

28. PROJECT IS SHARED BY THE IMPORT FUNCTION ................................. 307
   28.1. Sharing of Import Function with another Project ............. 307
   28.2. Outline of Import Function ............................................. 307
   28.2.1. Thing that Import Function enables ............................. 307
28.2.2. About information that the Import is not done ........................................ 308
28.2.3. About the error in Import ........................................................................ 308
28.3. Import for succession of information ......................................................... 309
28.4. Import for sharing of information ............................................................. 311
28.5. Import for adjustment of information ....................................................... 314

29. SAVE THE REPORT IN CSV BY EXPORT FUNCTION .............................. 315

30. USE DTPLANNER ..................................................................................... 318

30.1. Outline of DTPlanner .............................................................................. 318
30.2. Main window ........................................................................................... 318
30.2.1. Folder hierarchical form .................................................................... 319
30.2.2. Module hierarchical form ................................................................ 322
30.3. Menu ....................................................................................................... 322
30.3.1. File ..................................................................................................... 322
30.3.2. Tool .................................................................................................... 322
30.4. Edit of Property value ............................................................................ 323
30.5. Export ....................................................................................................... 325
30.6. Setting ...................................................................................................... 329
30.6.1. Order of column display and display/non-display specification of list .. 330
30.6.2. Width specification of column ............................................................. 331
30.7. DTPlanner automatic operation close ................................................... 332

31. DTMERGE FUNCTION ............................................................................ 333

31.1. Outline of DTMerge Function ................................................................. 333
31.2. DTMerge usage example ....................................................................... 334
31.3. Operational procedure ......................................................................... 335
31.4. Output window ....................................................................................... 339

32. DTADVISOR FUNCTION ......................................................................... 340

32.1. Outline of DTAdvisor function .............................................................. 340
32.2. Usage example of DTAdvisor ............................................................... 340
32.3. Main screen ............................................................................................ 341
32.4. Menu ....................................................................................................... 344
32.4.1. File ..................................................................................................... 344
32.4.2. Edit .................................................................................................... 344
32.4.3. Report Analysis ............................................................................... 346
32.4.4. Tool .................................................................................................. 346
32.5. Step Tracer ............................................................................................. 347
32.5.1. Display and fundamental motion of Step Tracer ......................... 347
32.5.2. Dynamic Module Tree part .............................................................. 349
32.6. Check function of Test Point at execution intervals .................................................. 350
32.7. Execute situation display function of Function .......................................................... 351
   32.7.1. Execute situation display of function ................................................................. 351
   32.7.2. Setting of the Dynamic Module ....................................................................... 353
32.8. Function History List ................................................................................................. 354
32.9. Setting ......................................................................................................................... 355

33. MULTI-CORE FUNCTIONS .......................................................................................... 358
   33.1. Outline ...................................................................................................................... 358
   33.2. The data format of Driver Functions ....................................................................... 358
   33.3. Get Test Report ........................................................................................................ 359
   33.4. Display the Test Report each core .......................................................................... 360
   33.5. Performance Monitor of each core ......................................................................... 361
   33.6. Setting Filter condition the core ............................................................................. 362

34. SOURCE CODE ANALYSIS FUNCTION .................................................................... 363
   34.1. Outline ...................................................................................................................... 363
   34.2. To Specify the Local Project Folder ....................................................................... 363
   34.3. Execution of Source code Analysis ........................................................................ 365
   34.4. Limitations ................................................................................................................ 366

35. SHORTCUT KEY LIST ................................................................................................. 367
   35.1. Shortcut key on source code window ...................................................................... 367
   35.2. Shortcut key on Test Report View .......................................................................... 368
   35.3. Shortcut key on the Function Trace Report ............................................................ 368
   35.4. Shortcut key on Event Trace ................................................................................... 368
   35.5. Shortcut key on DTAdvisor ..................................................................................... 369

36. FUNCTION ADDITION ITEM LIST(VER7.01) .......................................................... 371
   36.1. Correction of already-known problem ..................................................................... 371

37. CHANGE TRACKING ...................................................................................................... 372
1. Outline of DT10

1.1. Outline

DT10 is a development supporting tool to design and to test the program for embedded.

DT10 can get trace information (Test Report) from the running target. The Test Report is analyzed by DT10, and the "Program Execution Route analysis" "Dynamic Coverage analysis" and "Processing time analysis" is generated.

To get the Test Report, you need to insert the exclusive code for DT10 (called "Test Point") into your source code in manual/automatic.

Moreover, the program by the dynamic qualification can be tested comfortably by a high performance viewer display and the Test Report writer function of DT10.

1.2. Main function

In DT10, the following functions are provided.

- **Test Point insert function**
  - Analyze the source code and insert the Test Point in auto.
  - You can insert Test Point at any point you want to insert in manual.
  - Inserted Test Point can be set to enable/disable in easy.
  - ※To use the inserted Test Point, you need to compile your environment.

- **Test Report Collection function**
  - When doing Operation verification, Collect the Test Point execution information.
  - This function can continuous collection of Test Point execution information. Collected data is saved to PC Hard Disk. When the free space of the Hard Disk is less than 1GByte, the collecting is stop in auto. (The collected file is divided in 500MByte and saved in folder per 1GByte.)

- **Test Report Analysis function**
  - Analyze the collected Test Report.
  - You can use many analyze function, such as "Coverage Report", "Execution time Report", etc.

- **Test Result Generation function**
  - You can make Test Result in auto generate from collected Test Report and analysis result.
1.3. About the support

Please Inquiry even of the following of technical content described in this manual.

Heartland.Data Co., Ltd
User support charge
URL :  http://dt10.hldc.co.jp/

1.4. About the license

When DT10 is used, it is necessary to conclude the software licensing agreement. Moreover, it is necessary to prepare the system of the processing of the language of C・C++ compiler/assembler/linker tool separately.

Please connect the hardware key to USB with PC or enable floating license when you conclude the software licensing agreement of DT10 and use it.

When you use USB hardware key, DT10 can be used if connection of a USB hardware key is checked. DT10 starts in Free edition if not connected. When two or more DT10 is used in the development team, the USB hardware key for the required number is needed. The Test Report collection function can be used by connecting DynamicTracer with PC on that.

If you have a floating license, and is confirmed that the floating license is enabled, you can use the DT10. If it is not confirmed the license is enabled, DT10 will boot with Free edition. Floating licenses are available booting at once the number of licenses you subscribe to. The PC number of installing floating license is no limit.

The number of your floating license is managed by license server. (You need connect the license server on network.)

<table>
<thead>
<tr>
<th>Attention</th>
<th>Please use the thing for USB2.0 when you use the hub when the USB hardware key is used. Moreover, please do not remove the USB hardware key while DT10 is starting.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention</td>
<td>If you booting DT10 with USB hardware key authenticates, authentication cannot change to floating license authentication. And if you booting DT10 with floating license authentication, the authentication cannot change to USB hardware key authentication.</td>
</tr>
</tbody>
</table>
2. System requirements

2.1. Operating system

- Windows XP
- Windows Vista
- Windows 7

※Supports each version 64bit and 32bit.
※Windows Installer 3.1 and Microsoft.NET Framework 2.0 or more must be installed.

(If it is not possible to set up to PC of the uninstallation. In that case, please execute Windows Update and install it.)

2.2. Computer main body

- Only PC/AT compatible
- The personal computer equipped with the processor more than the Pentium 4 2GHz corresponding is necessary.
- The system of two CPU or more is recommended.
- The USB2.0 port corresponding to the Hi-Speed forwarding should be able to be used (Do not correspond to USB 1.0).

2.3. Memory

- The mounting memory of 1GB or more is necessary.
- The mounting memory of 2GB or more is recommended.

2.4. Hard disk

- The free space of 100MB or more is necessary for the installation of DT10.
- The free space of 2GB or more is necessary for the Test Report analysis.
  Moreover, the free space of 1GB or more is separately necessary for the Test Report collection.

2.5. Display

- It is more necessary than display resolution XGA (1024 x 768 pixels).
- It recommends more than display resolution WXGA + (1440 x 900 pixels).
3. Screen structure

3.1. The main window

- **Folder View**
  - It is a window that displays the source file in the Project by the tree view.
  - It is another window and windows that can be docked.
  - The tree is displayed in order of "Folder > file > function > Step".
  - Enable/Disable Test Point with check box.

- **Module View**
  - It is a window that displays the function of each module in the Project by the tree view.
  - It is another window and windows that can be docked.
  - The tree is displayed in order of "Module > function > Step".
  - Enable/Disable Test Point with check box.

- **Property View**
  - It is a window to confirm the value set in the Property Editor etc.
  - It is another window and windows that can be docked.
Test Report

- It is window that makes to the list and displays the data that DynamicTracer acquired.
- It is other windows and windows that can be docked.
- It is in real time updated while acquiring the Test Report.
- It is possible to be a unit change of the time value, to filter, and to retrieve it.
- The item displayed from "Customize of the Test Report view" in the environmental setting to the view can be selected.

Cores' Test Report

- The window to display the showed Test Report for each core.
- It is other windows and windows that can be docked.

Coverage Report

- It is window that makes the Step covering rate of each function a list.
- It is other windows and windows that can be docked.

Execution Time Report

- It is window that makes the execution time of each function a list.
- It is other windows and windows that can be docked.
- It displays between the maximum and the mean time from among the totaled data.

Period Time Report

- The window where the time of each function of the cycle is made to the list and displayed.
- It is other windows and windows that can be docked.
- The minimum and the maximum and the mean period are displayed from among the totaled data.

Loop Count Report

- The window where details of the judgment result of the step where "Loop Count" of the step property was set are made to the list and displayed.
- It is other windows and windows that can be docked.
- The minimum and the maximum and the mean Loop Count are displayed from among the totaled data.

klocwork INSIGHT Report

- This window can display the result of analyze "klocwork INSIGHT" source code.
- It is other windows and windows that can be docked.
Function Trace Report
- It is a window that draws in transition of the function within the set range information by graphic together with the time axis.
- It is other windows and windows that can be docked.
- Within the range of the maximum and the analysis, 24 days (=2073600 sec) in 2097152 and time is upper bounds in volume of data.
- Warning goes out when analytical volume of data exceeds 1048576.

Performance Monitor
- It is a window that draws in real time in CPU load of the target.
- It is other windows and windows that can be docked.
- CPU load output of each 100 ms becomes it.

Event Trace Report
- It is a window to which it draws in event information in the Test Report data by graphic.
- It is other windows and windows that can be docked.

Variable Monitor
- It is a window to which it draws in Variable Value Point information in the Test Report data by graphic.
- It is other windows and windows that can be docked.

Test Point Check List
- The window where the Test Point that was not able to be inserted because of an optional compilation is made to the list and displayed when the Test Point is inserted automatically.
- It is possible to insert the Test Point from the list or to select it.
- It is other windows and windows that can be docked.

Multi Wave Scope
- It is a window to which it draws in the data (logic input and AD input) acquired from an Analog Box by graphic.
- It is other windows and windows that can be docked.

Mark View
- It is a list of the Mark displayed in the Test Report.
- The Mark is divided into three or less depending on the Mark Type.
- It is other windows and windows that can be docked.
■ Document View
  • It is a window where the source code window and the Report window are displayed.
  • It is revocable whether to make to the document display with the tab or to display the multi document in the environmental setting.

■ Source code window
  • It is a window where the content of the source file is displayed.
  • The position of the tab can be set by "TAB stop of the source code window" in the environmental setting.
  • It is possible to put up the line number to the source code window in the environmental setting or to select it.
  • An insertion deletion of the Test Point and an easy source code can be edited.

■ Report window
  • It is a window where the content of the Test Report is displayed.

■ Printer View
  • It is a form view that displays the printed content.
  • It draws in the print image of the Report based on the analytical result of the content and the Test Report input by setting of the Project and setting the Report.
3.2. Menu

3.2.1. File

■ New
  - The Project is newly created.

■ Open
  - An existing Project is opened.

■ Save
  - The Project is saved in the superscription.

■ Save as
  - It saves giving the Project another name.

■ Setting
  - A set dialog of the Project that is opening now is opened.

■ Import
  - The dialog that opens the file is displayed.
  - Property information and the source code set by the selected Project can be shared by a present Project.

■ Export
  - Various Reports are preserved by the CSV form.

■ Rollback
  - The content included in a past Profile is replaced with the latest test environment.

■ Close Report Data
  - Past Report Data is closed, and it returns to the latest environment.

■ Exit
  - DT10 is ended.
3.2.2. Plan

- **Insert Test Point Automatically**
  - The automatic insertion dialog of Test Point is displayed.
  - When the time to be inserted in the file of the Test Point uninsertion automatically and the Test Point are inserted again, it uses it.
  - It is possible to select it by the Folder View at the file selection.

- **Delete Test Point**
  - The deletion dialog of the Test Point is displayed.
  - The Test Point is used to do the batch deletion in each source file.

- **Check Test Point**
  - The check dialog of the Test Point automatic operation insertion cancellation point is displayed.
  - The automatic insertion investigation of the Test Point is executed.
  - The extension is a file of "*.c" "*.cpp" to the object.

- **Fast Test Point Convert**
  - The Test Point which is included specified source file is converted Fast Test Point.
  - The Fast Test Point function can be used when DT10 connect with GPIO/SPI connection.

- **Property Editor**
  - The dialog of Test Property Editor is displayed.
  - The Source File Property, the Module Property, the Function Property, and the Step Property can be edited.

- **CPU Load Detection**
  - A switch of CPU load detecting function valid/invalid is done.

- **Update Test Point setting**
  - The update dialog to set the Test Point is displayed.
  - When the Test Point is edited, the dialog can be displayed.

- **Disable all Test Points**
  - A switch of all the Test Points enabled/disabled can do ON/OFF of an individual Test Point as it is.
  - The extension is a file of "*.c" "*.cpp" to the object.

3.2.3. Test

- **Set collecting condition of Test Report**
  - The Test Report collection condition setting dialog is displayed.
  - It can set saving of the conditions precedent and the Test Report of the Test Report collection ahead.
3.2.4. Report Analysis

■ Show Test Report...
  • The selection dialog of Test Report is displayed.
  • It uses it to display the acquired Report Data.

■ Analyze Test Report
  • The acquired Test Report is analyzed.
  • It is necessary to do this analysis to display the Report.

■ Multiple Analysis of Test Reports
  • Two or more Test Reports in the same profile are added up, and analyzed.
  • The result that does the addition analysis is output to the Coverage Report, the Execution Time Report, the Period Time Report, and Loop Count Report.

■ Coverage Report
  • The Coverage Report is displayed.
  • When the Report Data is not analyzed, the Report Data is analyzed at the same time.
  • Even if the command of "Coverage Report" is executed afterwards, the result of the analysis last time is displayed if "Analysis of the Report Data" is done once.
Execution Time Report

- The Execution Time Report is displayed.
- When the Report Data is not analyzed, the Report Data is analyzed at the same time.
- Even if the command of "Execution Time Report" is executed afterwards, the result of the analysis last time is displayed if "Analysis of the Report Data" is done once.

Period Time Report

- The Period Time Report is displayed.
- When the Report Data is not analyzed, the Report Data is analyzed at the same time.
- Even if the command of "Period Time Report" is executed afterwards the result of the analysis last time is displayed if "Analysis of the Report Data" is done once.

Loop Count Report

- The Loop Count Report is displayed.
- When the Report Data is not analyzed, the Report Data is analyzed at the same time. Even if the command of "Loop Count Report" is executed afterwards, the result of the analysis last time is displayed if "Analysis of the Report Data" is done once.

Function Trace Report

- The Function Trace View display range specification dialog to display the Function Trace is displayed.
- The Function Trace Report displays the execution route of the program and the execution time graphically.

Performance Monitor

- CPU load measurement data is analyzed again, and the Performance Monitor is displayed.
- The performance monitor displays CPU load information graphically.

Multi Wave Scope

- Multi Wave Scope is displayed.
- It uses to draw in the data acquired from an Analog Box graphical.

Event Trace

- Event Trace Report is displayed.
- It uses to confirm the transition of the task etc.

Variable Monitor

- Variable Monitor is displayed.
- It uses to confirm the change of the variable value.
■ Display the Test Report of each core
  - It is displayed as “Cores’ Test Report” each core.
  - Display “Core0 Test Report List” to “Core3 Test Report List” arranged in Test Reports “Core number”.

■ Set the View filter of Test Report...
  - A detailed, set dialog of the Test Report display is displayed.
  - A variety of display filters of the Test Report can be set.

3.2.5. Test Result

■ Create
  - The print view of the Test Result is displayed.

■ Open
  - An existing Test Result is opened.

■ Save
  - The made Test Result is preserved.

■ Printer
  - The Test Result is printed.

■ Printer preview
  - The print preview of the Test Result is displayed.

3.2.6. Tool

■ Setting
  - An environmental, set dialog is displayed.

■ View
  ▼ Tool bar
  - The ON/OFF switch of the display of each window can be done.
  ▼ Docking window
  - The ON/OFF switch of the display of each Tool bar can be done.
  ▼ Status bar
  - The ON/OFF switch of the display of the status bar under the main window can be done.
  ▼ Application view
  - Design of Window can be changed.
■ DTPlanner is opened
  • DTPlanner is opened.

■ DTAdvisor is excited
  • DTAdvisor is executed.

■ Analyze source code by klocwork INSIGHT
  • DT10 execute "klocwork INSIGHT" source code analysis function.

### 3.2.7. Help

■ Version information DT10
  • Version information is displayed.

■ User’s Manual
  • Open the User's Manual

■ Hardware Manual
  • Open the Hardware Manual.
3.3. Tool bar

3.3.1. Standard

■ Whole image

■ Meaning of each part

A new Project is Created.

An existing Project is opened.

The Project is saved.

Setting dialog is opened.

Program information, the version, and the copyright are displayed.

The selected range is cut out.

The selected range is copied.

The content of the clipboard is put.

3.3.2. Test Plan

■ Whole Image

■ Meaning of each part

The Test Property Editor is opened.

"Update Test Point setting" dialog opens.
3.3.3. Test Execution

■ Whole Image

■ Meaning of each part

✔️ : The collection condition in the Test Report is set.

▶️ : The acquisition of the Test Report starts.

3.3.4. Report Analysis

■ Whole Image

■ Meaning of each part

🗂️ : An existing Test Report is displayed.

✔️ : The Coverage Report is displayed.

▶️ : The Execution Time Report is displayed.

🗂️ : The Period Time Report is displayed.

✔️ : The Function Trace Report is displayed.

🗂️ : The Performance Monitor is displayed.

🗂️ : The Multi Wave Scope is displayed.

🗂️ : The Event Trace Report is displayed.

🗂️ : The Variable Monitor is displayed.

When the Variable Monitor is not set, the Variable Monitor setting dialog is displayed.

✔️ : The display filter of the Test Report is set.
3.3.5. Source code window

■ Whole image

■ Meaning of each part

- It retrieves on the source code window.

- The classification setting of the source code window is done.

- The font is enlarged.

- The font is reduced.

- "Specification at the jump destination" dialog is opened.

- It jumps to the previous Function for the Function that has been selected now.

- It jumps to the next Function for the Function that has been selected now.

3.3.6. Test Report

■ Whole Image

■ Meaning of each part

- "Jump to the specified Report number" dialog is displayed.
4. Start of application

4.1. Multi edition

In DT10, it fixes when the edition of Standard/Lite/Free is started and the application is operated.

Please confirm the start method corresponding to the edition used.

<table>
<thead>
<tr>
<th>Attention</th>
<th>After DT10 is started, it is not possible to change to another edition while starting. Please note it.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention</td>
<td>If you booting DT10 with USB hardware key authenticates, authentication cannot change to floating license authentication. And if you booting DT10 with floating license authentication, the authentication cannot change to USB hardware key authentication.</td>
</tr>
<tr>
<td>Attention</td>
<td>When both the Standard key and Lite keys are pierced among the hardware keys, it starts by Free edition.</td>
</tr>
</tbody>
</table>

4.1.1. Standard edition

■ Start of Standard edition

The USB hardware key for the Standard edition is connected with PC or for floating licenses after confirming that the Standard edition license is enabled, "DT10.exe" is executed from the installation folder, and the application of DT10 is started.

■ Function of Standard edition

All functions of DT10 can be used.

4.1.2. Lite edition

■ Start of Lite edition

The USB hardware key for the Lite edition is connected with PC, for floating licenses after confirming that the Lite edition license is enabled, "DT10.exe" is executed from the installation folder, and the application of Dt10 is started.

■ Functional restriction of Lite edition

- Cannot use function below.
  - Execution Time/Period Time graph display function
  - Execution Time/Period Time histogram display function
4.1.3. Free edition

- **Start of Free edition**
  When you use floating license or when USB hardware key is not connected PC, execute DT10.exe from install folder, DT10 is booting "Free edition".

- **Functional restriction of Free edition**
  - In "Free edition", you can use "make DT Project", "get Test Report", "Display Test Report".
  - You cannot use following features.
    - Test Report analysis function
    - Test Report
    - Set the View filter of Test Report function
    - Test Result function
    - DTPlanner function
    - DTAdvisor function
    - The context menu on the Test Report commands other than specifying of unit change. (Function Trace/Filter setting/Classification setting/Trimming/Copy/Copy with header)
    - Real-Time Coverage function
    - Collecting Test Report by UART connection or Ethernet connection without Dynamic Tracer. (Generation of DT project is available.)

  - The Project made by "Standard/Lite edition" can open by "Free edition".
  - If you use following feature, Configuration is refreshed in auto.
    - Filtering functions
      You cannot use Filtering function in Free edition.
      If you open the project in Free edition, [no filter] is enabled in automatically.
    - Real-Time Coverage function
      You cannot use Real-Time Coverage in Free edition. If you open the project in Free edition, the checkbox [Display the Real-Time Coverage] in [Set collecting condition of Test Report] dialog is disabled in automatically.

  - Following sub tool cannot execute.
    - DTConverter
    - DTDiff
4.2. Start dialog

To start DT10, execute "DT10.exe" from Start menu or install folder.

<<Example>> How to start with Start menu.

Start dialog like below is displayed. Select process you want to do.

- **New**
  - The Project creating wizard is displayed.

- **Open**
  - An existing Project is opened.

- **Exit**
  - DT10 is ended.
5. Creating and change in Project

5.1. New creating of Project

To use DT10, generation of DT10 project file (called DT project) is needed. The way of generate DT project is to displayed "Project Generate Wizard" in below two ways.


- Select [File] -> [New...].

Project Generate Wizard is displayed.
Project Generate Wizard is consist by next three windows, "Project Property", "Register Source File" and "Specify auto generated file". Set at each window. You can see each windows detail description at [5.1.1. Project Property] [5.1.2. Register source file] and [5.1.3. Specify auto generated file].

When you click [OK] button at "Specify auto generated file", DT project is generated, and Test Point's automatic insertion is executed.

If you click [The Test Point automatic operation insertion is not executed] button, the DT project without Test Point Automatic insertion is generated. Please execute Test Point insertion at after DT project is generated.
5.1.1. Project Property

It is a dialog that sets the attribute of the Project.

The item which described "(Must)" must configure.

■ DT Project name (Must)
  - The name of the Project is specified.

■ Preservation place of DT Project (Must)
  - The place where the DT10 Project is preserved is specified.
  - In "Specify auto generated file" window, the position that the Project file of which the file name was the Project name specified for the preservation place of the DT Project is made by clicking the completion button or "Automatic Test Pint insertion is not executed" button.
  ※ The Project file name is a Project name. It becomes the form of rprj.

■ Root folder of test target (Must)
  - The root folder of the Project is specified. The place of the registered source file is specified usually.
  - Please set for the source file group that registers to exist most in the high ranks the root folder.
■ Tracer connection (Must)
  • The method of connecting the target and DynamicTracer is specified.
    Async Bus : It is an asynchronous bus connection method.
    Async Bus (function call) : The driver is an asynchronous bus connection to the access for the necessity.
    GPIO : It is a connection by the GPIO(4bit) bus output method.
    SPI : It is a connection by the SPI output bus method.
    Ethernet : It is a connection by the Ethernet.
    Ethernet (without Tracer) : It is Ethernet connection without "DynamicTracer". When connect in Ethernet connection without "DynamicTracer", PC timer is used for timestamp.
    UART : It is a connection by the UART.
    UART (without Tracer) : It is UART connection without "DynamicTracer". When connect in UART connection without "DynamicTracer", PC timer is used for timestamp.
    SD : It is a connection with SD.

■ klocwork INSIGHT local project folder
  • If you want to use statically analysis function klocwork INSIGHT from DT10, you need specify folder for local project of klocwork INSIGHT.
  • If you want to execute statically analysis function, you need to introduce "klocwork INSIGHT".

■ Comment
  • It is a free comment filling in column. Please fill it in arbitrarily.

■ Member list
  • The member of the Project is specified. Please fill it in arbitrarily.

<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>The root folder recommends setting it so that the source file group that registers may exist most in the high ranks.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>To collect Test Report from Ethernet connection without DynamicTracer or To collect Test Report from UART connection without DynamicTracer, please use DT10 of Standard Edition. (※Free Edition and Lite Edition can generating project, setting connection for those connection type.)</td>
</tr>
</tbody>
</table>
5.1.2. Register source file

It is a dialog that specifies the source file registered in the Project. The extension can be added and only the file of ".c" ".cpp" ".java" is added.

If the extension is not displayed, it is not possible to register. Please note it. Moreover, the registered source file is displayed by the gray color.

If you have not specified source file to register, project generation is only execute. After project generating, you can register source file for use to DT10.

- Add
  - When the registered source file is selected from the file list of upper right and "Add" button is clicked, the source file registered in the Project is added.
  - It is possible to specify it even by the unit of the file and the unit of the folder.

- Exc (Exclusion)
  - When the source file to be excluded from the list of the source file of the test object in the lower side is selected and "Exc" button is clicked, the registered source file is excluded from the Project.

- Up
  - The folder in the upstairs layer of the folder being displayed now is displayed.
5.1.3. Specify auto generated file

It is a dialog that variously sets it when the Test Point is inserted in the registered source file.

The item which described "(Must)" must configure.

**Header file name**
- The character string added to the head of the made header file is specified. Default is "DT_". Please be arbitrary and change.
- Please make the column of the character string a blank when you want to make it to the header file name of the source file and the same name.
- The extension is a file of "*.c" "*.cpp" to a set object.

**Folder to save header file**
- The storage place of the header file is specified. Please be arbitrary and change.
- The extension is a file of "*.c" "*.cpp" to a set object.
### Base address

- At the asynchronous bus connection, the base address of the address output on the bus is specified. Please specify it according to the target environment used.
  
  (Ex) Please specify the base address for 【0x80000010】 when the initial address of the address space of the Flash memory is 0x80000000.

- In connected methods other than the asynchronous bus connection, default can be used as it is.

- Please specify it by the hexadecimal number value.

- In DT10, 0x10 increases and allocates the address from this base address at each file. When the address is written as 0x000XXXXY, the use space becomes XXXX part. (57343 files or less can be registered.) Y part becomes a reservation area that the application uses, and cannot be used. (It cannot be a change from the application program.)

- Please do not specify it to use the value that becomes basic as an address for CPU load detection.
  
  (example) When 0x80000000 is a base address, 0x80000000 is allotted for CPU load detection. Therefore, 0x80000010 is specified for a specified value.

- The specification of the base address recommends specification at the delimitation position in every 1MB like specified value~0xZZZ00010 (Z is an arbitrary value). The number of files for which the above-mentioned values can be used most increases because it becomes possible to use the file to 0xZZZFFFF0. Please note that the number of files that can be used decreases at the specification of the fraction value.

### Type definition file

- Please describe the type of the variable of the target environment dependence to the text file, and specify passing the file.

- When the Test Point is inserted automatically, the character string specified with this file is judged to be a type definition.
  
  (example) UINT8 and INT32, etc.

- Refer to the file specified in the Source File Property when it is set in the Source File Property.

### Invalid code definition file

- Please describe the character string of the part where it doesn’t want to insert the Test Point to the text file, and specify passing the file.

- Using it is convenient for the nullified source file etc. an optional compilation of the comment on the header part of the function it
  
  (example) #ifdef DOC, etc.

- Refer to the file specified in the Source File Property when it is set in the Source File Property.
■ Type name of 32bit type without sign

- The type name of 32bit type without the sign in the target environment is specified.
- The extension is a file of "*.c" "*.cpp" to a set object.
  (Ex) 32bitCPU : unsigned int
  16bitCPU : unsigned long

■ Specify byte order of output Variables

- Byte order specification by evaluate of variable value is selected.
- Please select Lo/Hi (Little Endian) or Hi/Lo (Big Endian).

■ Insert Variable Value Point of arguments of each Function in Automatic Test Point insertion.

- When the Test Point is automatic inserting, the Variable Value Point to the argument is inserted automatically or selects it.
- When the check box is turned on, the Variable Value Point concerning the argument is inserted.

■ Setting the Character code

- You can configure the character code for source file which register to DT Project.
- In usual, this setting is configured to auto. DT10 judging character code in auto.
- If auto judge of DT10 is failed, auto insertion of Test Point cannot execute. In this case, you need set character code in manual.
- If the character code is configured in source file property, source file property's setting is overridden.
- You can see detail description in 【6.2. Character code and Newline code for source code】.

■ Setting the Newline code

- You can configure the newline code for source file which register to DT Project.
- In usual, this setting is configured to auto. DT10 judging newline code in auto.
- If auto judge of DT10 is failed, auto insertion of Test Point cannot execute. In this case, you need set newline code in manual.
- If the newline code is configured in source file property, source file property's setting is overridden.
- You can see detail description in 【6.2. Character code and Newline code for source code】.

■ Module information is generated automatically

- When automatic of the Test Point inserting it, the folder passing of the source file of the object and the module of the same name are made when the check box is turned on.

■ The number of each hierarchy is added to the module name

- When the module is made when automatic of the Test Point inserting it when the check box is turned on, the number of each hierarchy is added to the module name.
- When "Information on the module is generated automatically" item is only turning ON, and becomes effective.
Insert `#include` for header files to the next line which includes specified string.

- A specific character string is specified. Then, the character string is retrieved, and the Include sentence is inserted under the following line of the line found first.
- It is convenient to use this function when there is a file etc. that should do include by the first line of the file.
- If the check is put on "Search strings on the inside of comments." the retrieval object becomes only in the comment.
  Moreover, if the check is not put, the retrieval object becomes only outside of the comment.
- Please put the check on "Case sensitive." when you identify capital letters and small letters of the alphabet.
- The extension is a file of "*.c" "*.cpp" to a set object.
■ Change Insertion setting

- Click the "Change Insertion setting" button, and you can see following dialog, and you can change insertion setting in this dialog.
- If you want to change the insertion setting, please change it with your self-responsibility.

![Change Insertion setting dialog]

▼ Correspond to the minor error of automatic Test Point insertion.
- You can set the display of the error of automatic insertion.
- It recommended setting to [The safety level is maximum] which is warns for all errors.
- In item of [Custom], you can select the error which is gives warning.
- If select to [All warning is disregarded], the warning of the minor error is not be displayed.

▼ Corresponding to line feed code coexistence file

When you want to insert Test Point to the file which some newline codes is used in mix, you can select to "Processing is discontinued as an error" or "Compulsorily the line feed code is converted and it continues".

▼ Insertion location of FuncIn Test Point

When check the “Always insert FuncIn Test Point to top of function” checkbox, the FuncIn Test Point is inserted just below of the brace ("{") which is judged to the function start by DT10. Even if there is preprocessor directive to the insertion point of Test Point, FuncIn's insertion is execute without cancel.
- Please set it with fully check of the following limitation.

```c
void FuncA(int arg_a, int arg_b)
{
    _DtTestPoint(_DtFunc_FuncA, _DtStep_0)
    UINT16 i;
}
```

When this setting is ON, Test Point is inserted to under "{".
◆ Limitations of [Always insert FuncIn Test Point to top of function]

- If you insert FuncIn in this setting, Test Point is inserted at first line of the function (before than variable declaration).
  Therefore, by the compiler, compile may be failed.
  Please use this in the target's development environment (C compiler) which corresponds to the "ISO/IEC 9899:1999 Programing Language-C" (C99). ("ISO/IEC 9899:1999 Programing Language-C" is not an error even though the executable code is written before the variable declaration part.).

- In case you insert FuncIn, the indent of Test Point may be broken.

- In the function which FuncIn and FuncOut is inserted as single Test Point when this setting is OFF (such a function that is inserted the FuncIn&FuncOut Step), when this setting turned ON, FuncIn and FuncOut is inserted as separate Test Point.

- If user add the string between the top of function and the existing FuncIn and execute add insertion of Test Point, the result of add insertion of Test Point is different depending on whether this setting is enable or disable.

**Memo** The setting of [Insert FuncIn with ignore the FuncIn's cancel point] and the setting of [Always insert FuncIn Test Point to top of function].

In [Automatic insertion setting], user can set [Insert FuncIn with ignore the FuncIn's cancel point] item by [Correspond to the minor error of automatic Test Point insertion]. This setting insert FuncIn to under the preprocessor directive even if there are preprocessor directive to the point for insertion of FuncIn.

On the other hand, the setting [Always insert the FuncIn Test Point in the first line of function] is insert the FuncIn to just below the brace ("{") of which is judged to the function's top.

![FuncIn inserted under "#if".](image1)

![FuncIn inserted under "{").](image2)
■ Return
  • It returns to the previous dialog.

■ Completion
  • The new creating wizard of the Project is ended, and the automatic Test Point insertion is executed.
  • The edit and the superscription preservation are executed for the source file when this button is selected, and the Project file is preserved.

■ Automatic Test Point insertion is not executed
  • The new making wizard of the Project is ended without executing the automatic insertion of the Test Point.
  • The source file is registered in the Project though the Test Point is not inserted automatically.
  • Using it to insert the Test Point only by hand power is convenient.
  • When this button is selected, the Project file is preserved.

---

Attention

To output information of execution Test Point, you need to insert driver function 
"_TP_BusOut()" in your target environment.
If the driver function is not build in your environment, you need to make the driver.
To make driver, refer to the sample source code in install folder ([sample] -> [driver_samples]) or refer to attached tool [DTDriverWizard].
5.2. Change in Project

5.2.1. Project Setting Dialog

The setting of DT project can be changed after generate the DT project file.

To select [File] -> [Project Setting…] from menu, and the Project Setting dialog is displayed.

In this dialog, you can change/check setting of the "Project new creating wizard" dialog.

(You cannot change next item. [Preservation place of DT10] [Root folder of test target] [Base address])

Configure item in [Project Setting] dialog and configuration is enabled when you click [OK] button in the dialog.

But the new setting is not reflected in the source file which already registered. The new setting is applied from your next Test Point insertion.
5.2.2. Addition of source file

You can register the source file to DT project in next step.

1. Select the source file which you want to register at the [Register source file] tab, and click the [Add] button.

2. Click [OK] button.
   When [OK] is selected, automatic Test Point insertion is executed to the added source file. Moreover, the edit and the superscription preservation are executed for the added source file.
   Please select [Automatic Test Point insertion is not executed] when you do not want to insert the Test Point automatically.

3. When the source file is added, the item of each hierarchy of the tree is displayed in the alphabetical order of the folder name and the source file name in the folder view module view.

4. Please select [File] -> [Save] of the menu, and save the Project after adding the source file.

5.2.3. Registration release of source file

You can cancel registration of the source file from DT project in next step.

1. Please select the source file to want to release the registration from “Register source file” dialog, and click “Exclude” button.

2. Registration of the source file selected from the Project is released by selecting [OK] or [Automatic Test Point insertion is not executed].

3. Please select [File] -> [Save] of the menu, and save the Project after releasing the registration of the source file.

Attention

The deletion and the header file of the Test Point are deleted without the user notification when the registration release of the source file is done from the Project.
6. The rule of automatic insertion of Test Point

6.1. Outline of Automatic Insertion

In DT10, the Test Point is inserted in the following positions automatically.

The Test Point insertion position becomes the function entrance, the function exit, if (include else etc.), and for, while, do, switch, catch, and throw.

Moreover, the header file that registers the macro code for the Test Point insertion is generated automatically, and # include is done.

When inserting it automatically, the Test Point is inserted without basically reformating source code.

However, when there is no space for the line that inserts the Test Point in the Test Point insertion position, the space of the amount is secured and inserted.

Moreover, please note that the source code cannot be completely returned to former shape even if the Test Point is deleted for the above-mentioned.

It automatically reformate the source code to execute the automatic insertion normally when an automatic insertion in non-reforming source code fails.

The rule of the source code plastic operation is as follows.

- { changes line when the definition of the function and the structure, etc. are declared.
- The dummy argument of the function is put on the same line as a basis.
- In the condition branching such as if, { doesn't change line.
- The width of the indent assumes four, and uses the tab.
- Space is inserted ahead of ( back and ).
- The case label does one indent.
- Puts on the same line of while of do{} while {}.
- else is put on the following line of }.
- A changes line and undergoes plastic operation on the back of { and ; like if(1){;} in the block scope in all cases.
- The label is treated as well as case::.
Actual insertion example

```
#include "out.h" /* For DynamicTracer-TestPoint */
#include "sample.h"

#define MAX_CNT (100)

static unsigned char g_cnt;
static unsigned char g_max = MAX_CNT;

static void func_a(int arg_a);

void func_a(int arg_a)
{
  short cnt_a;
  short cnt_b = 10;
  int cnt_c, cnt_d;

  __DtTestPoint(__DtFunc_func_a, __DtStep_0)/FuncInK/
  cnt_c = 10;
  cnt_a = 1, cnt_d = 20;

  for(cnt_a = 0; cnt_a < cnt_b; cnt_a++)
  {
    __DtTestPoint(__DtFunc_func_a, __DtStep_1)/For/
    cnt_c++;
  }

  if(arg_a == cnt_c)
  {
    int temp = 20;
    __DtTestPoint(__DtFunc_func_a, __DtStep_2)/If/
    cnt_c += temp;
  }
```

To make it plainly as a concrete example, the comment ("/* if*/ etc.) that adheres behind the Test Point in the above-mentioned insertion example has been described. Please note no description in an actual code.

Attention
It inserts in the do loop.

It inserts it on the function exit.

The line is judged to be a function exit immediately before return when there is return.

It doesn't insert it for do {} while in while in the back.

At the switch divergence, inserts in for each divergence.

When the insertion such as switch and insertion on return come in succession, insertion with two attributes is done.

It buries on the function exit.

The line of it becomes a function exit is judged to be a function exit immediately before.

When the case sentence comes in succession
It inserts it in the place from which the codes other than the label have come out.
6.2. Character code and Newline code for source code

6.2.1. Outline

In DT10, the character code and the newline code used in source code are judged automatically when source code analysis is executing (such as automatic insertion of Test Point).

DT10 can use the following Character code / Newline code:

<table>
<thead>
<tr>
<th>Character code</th>
<th>Newline code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift-JIS</td>
<td>Windows type(CR+LF)</td>
</tr>
<tr>
<td>EUC-JP</td>
<td>MAC type(CR)</td>
</tr>
<tr>
<td>UTF-8(with BOM / without BOM)</td>
<td>Linux/Unix type(LF)</td>
</tr>
<tr>
<td>UTF-16LE(Little Endian) (with BOM / without BOM)</td>
<td></td>
</tr>
<tr>
<td>UTF-16BE(Big Endian) (with BOM / without BOM)</td>
<td></td>
</tr>
</tbody>
</table>

**<<Auto judgement rule of Character code>>**

- The one that there is only ASCII on the source code is considered to be a character code of Shift-JIS (UTF16 is excluded).
- IBM extended character (0xFA40~0xFC4B) of Shift-JIS is recognized Shift-JIS.

**<<Auto judgement rule of Newline code>>**

- The most used Newline codes in the text are assumed to be a Newline code of the text.
- The text without Newline code is considered that its Newline code is Windows type(CR+LF).

In case of that the auto judgement of character code or newline code is failed, User can set those codes directly.

After auto judgment of character code and newline code is executed, user can change each source files character code and newline code on Source file property.

In case that the source codes auto judgement of character code or newline code is failed, user can try next step.

User set the character code and newline code on the "new Project making" Wizard or the "Specification of the automatic generation file" Dialog in advance.

And, in case of that there are some source code which cannot judged character code or newline code in auto judgement, or in case of that there are some candidate of the character code or newline code, the user setting is selected for source code analysis.
**Sample ①** In case of failed to auto judgement of character code when automatic insertion is executed, user can set character code on Source file property.

<table>
<thead>
<tr>
<th>Result of auto-detect</th>
<th>Auto insertion</th>
<th>Source file property</th>
<th>Auto insertion</th>
</tr>
</thead>
<tbody>
<tr>
<td>File A: UTF-8</td>
<td>File A: OK</td>
<td>File B: Shift-JIS</td>
<td>File B: OK</td>
</tr>
<tr>
<td>File B: failed</td>
<td>File B: failed</td>
<td>File C: UTF-8</td>
<td>File C: OK</td>
</tr>
</tbody>
</table>

Even if you failed to auto judgement of character code, you can set character code for each source file!

**Sample ②** If you know that there is file which fail to auto judgement, User can set character code on Project Setting in advance.

<table>
<thead>
<tr>
<th>Result of auto-detect</th>
<th>Auto insertion</th>
<th>Source file property</th>
</tr>
</thead>
<tbody>
<tr>
<td>File A: UTF-8</td>
<td>File A: OK</td>
<td>File B: Shift-JIS</td>
</tr>
<tr>
<td>File B: failed</td>
<td>File B: OK as UTF-8</td>
<td>File C: OK</td>
</tr>
</tbody>
</table>

If you set character code in advance, you can execute auto insertion of Test Point even if auto judgment is failed!

You can set the character code for each source code after execution of auto insertion!
6.2.2. Character code and Newline code setting for auto judgement

When user make new DT project, user set the character code and newline code for execute of auto judgement.

It can setting on the item of “Specification of the automatic generation file Dialog” (the third page of “new making wizard”), the [Character code setting] and the [Newline code setting].

If user selected [Auto], DT10 execute the auto judgement.

When the auto judgement is failed, and if you set character code/newline code in advance, DT10 can continue to analyze the source code with your advance setting of character code.
Character code Setting

- User can set the character code for source file which register to DT project.
- User can select next character code. Auto, ASCII, EUC-JP, Shift-JIS, UTF-8, UTF-16LE(Little Endian) and UTF-16BE(Big Endian)
- If user select the [Auto], DT10 judging the character code in auto. When auto judgement failed, it is error.
- In result of auto judgement, if there are some candidates of character code, the analysis is executed with next character code (in the order of next priority).
  (High-priority) UTF-8 > Shift-JIS > EUC-JP (Low-priority)
- If user set the character code to other than [Auto], and when auto judgement is failed, or when the result of the auto judgement of character code is different from that of user setting, the analysis is executed with character code which set by user.

with BOM

- Set the presence or absence of BOM(Byte Order Mark).
- This checkbox is enabled if UTF-8, UTF-16LE or UTF-16BE is selected as character code.

```
<table>
<thead>
<tr>
<th>Character code Settings</th>
<th>UTF-8</th>
<th>BOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCII</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shift-JIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EUC-JP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTF-8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTF-16LE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTF-16BE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Newline code Setting

- User can set the newline code for source file which register to DT project.
- User can select next newline code. Auto, CR+LF, LF and CR.
- If user select the [Auto], DT10 judging the newline code in auto. When auto judgement failed, it is error.
- If user set the newline code to other than [Auto], and when auto judgement is failed, or when the result of the auto judgement of newline code is different from that of user setting, the analysis is executed with newline code which set by user.

```
<table>
<thead>
<tr>
<th>Newline code Settings</th>
<th>Auto</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td></td>
</tr>
<tr>
<td>CR+LF</td>
<td></td>
</tr>
<tr>
<td>LF</td>
<td></td>
</tr>
<tr>
<td>CR</td>
<td></td>
</tr>
</tbody>
</table>
```
6.2.3. Character code and Newline code setting for each Source file

After execute the auto judgement of character code and newline code, User can set those for each source file.

User can set it on the [Character code setting] and [Newline code setting] (the item of “Source file property”).

“Source file property” is displayed with next some ways.

- Select the [Plan] → [Property editor] from Menu.
- Select the source file in folder view and right click it and select the [Open Property Editor...].

When the character code or newline code is set by source file property, the "source file property’s setting have priority, even if the character code and newline code is set by Project setting.

![Test Property Editor](image_url)
■ Character code setting

- User can set the character code of source file which selected in Property editor.
- User can select next character code. Auto(***) , ASCII, EUC-JP, Shift-JIS, UTF-8, UTF-16LE(Little Endian) and UTF-16BE(Big Endian)
- Auto(***) set the character code using the result of the auto judgement of character code. (** is judged character code in auto)
- If "Project setting" set the character code also, Source file property's setting has priority.

▼ with BOM

- Set the presence or absence of BOM(Byte Order Mark).
- This checkbox is enabled if UTF-8, UTF-16LE or UTF-16BE is selected as character code.
- If the character code setting is set in Auto(***), the set character code is result of auto judgement, so it cannot change. When the result of auto judgement is "with BOM", this checkbox is checked and displayed with gray out.

■ Newline code setting

- User can set the newline code of source file which selected in Property editor.
- User can select next newline code. Auto(***) , CR+LF, LF and CR.
- Auto(***) set the character code using the result of the auto judgement of newline code. (** is judged newline code in auto)
- If "Project setting" set the newline code also, Source file property's setting has priority.
6.2.4. Character code and Newline code setting for auto judgement

The character code and newline code set by Source file property is can check on Property view.

Select the source file from folder view and check the [Character code], [BOM] and [Newline code].

(※Property View can only check character code, BOM, and so on. To change those, Use the Property Editor.)

When source code is opened, the active file character code/newline code is displayed at right side of status bar.
6.2.5. Limitation

◆ If the file using character code that is different from other file's code is exists in the DT project, sometimes, DT10 cannot analyze successfully.

If the case such as the following example, please execute next process

After character code judgement, set file "A"s character code in directly on Source code Property and Re-execute the analyze.

<example> In case of that the file using different character code are mixed.

<table>
<thead>
<tr>
<th>Result of auto.detect</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>File A  Shift-JIS</td>
<td>File A  Shift-JIS file is analyzed as UTF-8</td>
</tr>
<tr>
<td>File B  Failed</td>
<td>File B  Analysis as UTF-8</td>
</tr>
<tr>
<td>File C  UTF-8</td>
<td>File C  Analysis as UTF-8</td>
</tr>
</tbody>
</table>

◆ If there are Source files using UTF-16 character code,

To set other than UTF-16 code to this code in Project Setting and execute auto insertion is impossible.

The file using UTF-16 is need to set its character code to [Auto] or [UTF-16]. After this setting, please execute auto insertion.

If there are Source files using other than UTF-16,

To set UTF-16 code to this source file in Project Setting and execute auto insertion is impossible.

For the file using UTF-16, you should not select other character code than UTF-16.
6.3. Rules of automatic insertion [C/CPP]

In the file whose extension is "*.c" or "*.cpp", the automatic insertion is inserted according to the following rules.

- The Test Point is inserted regardless of valid invalidity for in the compilation switch.

- The part where processing is not bundled by {} is not inserted with If, for, and while, etc.

- FuncOut in the function terminal doesn't adhere when return exists in the lowest nest in the function. Even when return is optional and tied up, it is similar.

- When the Test Point is inserted only by manual insertion without inserting it automatically, it is necessary to execute the insertion of the Include sentence separately by the document view.

- The second argument of a usual Test Point is inserted for C language and C++ by using the step number given continuously in the function like '__DtStep_x' (x is a step number).

- It doesn't insert to the function defined in the class.

- Throw is embedded under the former line. In that case, when another embeds has been generated, embed continuously.

- As for throw in the try space, the Test Point only of the throw attribute is inserted. (Test Point of the throw&FuncOut attribute) is inserted in throw (There is try space in the function call origin) that is not the try space.

- The function name uses class member's one minimum (The class names etc. of the succession origin are not used for the function name).

- If it is a same name even in case of the member function of another class in the same file, it processes it as a same name function.

- The Step number is given continuously by the same name function when there is a same name function in the source file.
The argument extraction is done based on the following rules.

・As for the extraction of the type name and the argument name, A is disregarded. The affixing character in [] is disregarded.
・The number of "*" and "[]" is totaled when the argument of the multiple indirection reference.
・The argument to which the variable identifier is omitted is assumed to be an extraction off the subject.
・The argument is not extracted when a separated function.
・Only when the dummy argument is declared as a pointer variable when the function pointer is used for the argument, it is extracted.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Extractive/Impossibility</th>
<th>Description example</th>
</tr>
</thead>
<tbody>
<tr>
<td>The dummy argument is declared as a pointer variable.</td>
<td>○</td>
<td>(void *)pFunc</td>
</tr>
<tr>
<td>The dummy argument is declared as a variable pointer.</td>
<td>×</td>
<td>int (* pFunc)(int iArg) int (* pFunc)(int)</td>
</tr>
</tbody>
</table>

・When the pre-processor instruction appears in the dummy argument declaration section, the argument is not extracted. When the pre-processor instruction doesn't appear while extracting the following argument, it is extracted.

<table>
<thead>
<tr>
<th>Example of function</th>
<th>Variable</th>
<th>Extractive/Impossibility</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>int Func( int aaa , #if 1 intbbb , #else , intccc , #endif int ddd )</td>
<td>aaa</td>
<td>○</td>
<td>(Beginning), the type, the variable, and the end (comma) become complete.</td>
</tr>
<tr>
<td></td>
<td>bb</td>
<td>×</td>
<td># instruction exists between (Beginning) and types.</td>
</tr>
<tr>
<td></td>
<td>ccc</td>
<td>○</td>
<td>(Beginning), the type, the variable, and the end (comma) become complete.</td>
</tr>
<tr>
<td></td>
<td>ddd</td>
<td>×</td>
<td># instruction exists between (Beginning) and types.</td>
</tr>
</tbody>
</table>

The argument embedded is done based on the following rules.

・When FuncIn of the function is embedded, the extracted argument is continuously embedded only by the argument.
・The number of output sizes by an automatic burial makes it do the variable identifier in sizeof and output.
・The output is assumed to be level 1 for the variable of the multiple indirection reference.
6.4. Limitations of automatic insertion【C/CPP】

In the file whose extension is "*.c" or "*.cpp", there are some limitations for the automatic insertion.

- The Test Point is not inserted when there is a pre-processor (# if, # else, # define, and # include, etc.) at the position in which the Test Point is inserted.

However, the Test Point not inserted can be listed by doing "Automatic insertion check of the Test Point ", the Test Point of the list be selected, and the Test Point not inserted automatically be inserted in DT10.

Automatic insertion check of the Test Point is operationed from "Check on the Test Point auto insert cancel point" dialog that displayed to select [Check Test Point ] of the Plan menu, or [Check Test Point ] of pop-up menu of Folder View, or insert Test Point automatically to the source have been registered already.

When the Test Point is inspected from "Check on the Test Point automatic operation insertion cancellation point" dialog, it checks it as a state that the argument embedded is done without fail. It follows the argument burial setting at that time simultaneously with the automatic insertion when doing.

"Test Point automatic operation insertion cancellation inspection result" list is displayed when there is a Test Point not inserted.
In the list, red for the Test Point of the FuncIn attribute. It displays to the Test Point of the Arg attribute in yellow.

“Test Point auto insert cancel inspection result” list displays the listed Test Point for each source. (* The displayed list can be switched by the source file name displayed in the combo box under the title)
The inserted Test Point is selected from among the list, the pop-up menu is displayed on the source code window, and when [Insert Cancel Step] is selected, the selected Test Point is inserted.
(* At this time, when the source displayed in selected Test Point and window is different or the function and the burial line of the selected Test Point are different, etc. the cancellation Step cannot be inserted. )

```c
void FileUnitTrackChange( void )
{
    #ifdef EASY_DO
    #else
    UI
    #endif
    if
    {
        Insert Test Point
        Insert Variable Value Point
        Insert CPU Load Measurement Point
        Insert Event Trigger Point
    Insert Cancel Step
    } /* ()*/
}
```

After the Test Point that exists in the cancellation list is inserted in the source code window, it is written by “Step type” item of the Test Point as Inserted, and is displayed in the gray in the list.
Moreover, when edits (TAB is excluded) other than the cancellation point embedded are done on the source code window, information in the source is annulled by the automatic operation.

**Memo** About the limitation of insertion of the Cancel Point when the FuncIn is not inserted.
You cannot insert the Cancel Point which other than FuncIn's Cancel Point to the function which is not inserted FuncIn. Not sure that the Cancel Point you want to insert belongs what function.
So, when you insert the Cancel Point, you should insert the Cancel Point after insert the FuncIn.

- A correct code formatting is not guaranteed where the following content exists.
  - Compile error
  - Function that omits return value and argument (The constructor destructor is excluded).
  - Alternative token (alternative mark)
  - Tri graph (three character mark)
  - Macro character string

- There is a thing that the embedded point shifts behind an original part when the macro exists.

- A normal embed is not warrantable for the source code where the function that omits the return value and the dummy argument exists.

- A normal burial is not warrantable for the source code where the function written by the separate style (method of the description that the global variable is declared between the function name and the text) exists.
Even if the argument can be extracted to the function where FuncIn is not embedded, the Test Point is not embedded.

The argument cannot be embedded automatically at an additional embed.
6.5. Rules of automatic insertion [Java]

In the file whose extension is "*.java", the automatic insertion is inserted according to the following rules.

- The part where processing is not bundled by {} is not inserted with If, for, and while, etc.

- FuncOut in the function terminal doesn't adhere when return exists in the lowest nest in the function.

- The step number is given for Java continuously in the function.
  As for the second argument of a usual Test Point, the sequential number in the source file is allotted.

- Throw is embedded under the former line. In that case, when another embeds has been generated, embed continuously.

- As for throw in the try space, the Test Point only of the throw attribute is inserted. (Test Point of the throw&FuncOut attribute) is inserted in throw (There is try space in the function call origin) that is not the try space.

- The function name uses class member's one minimum (The class names etc. of the succession origin are not used for the function name).

- If it is a same name even in case of the member function of another class in the same file, it processes it as a same name function.

- The step number is given continuously by the same name function when there is a same name function in the source file.

- It insert only under the source file because there is no pre-processor instruction in Java. The header file is not generated. The Test Point calls the driver directly.

- Only the method of writing in the class is made the object of the insertion in Java.

- It doesn’t insert to the limited part class.

- It doesn’t insert to an anonymous class.

- Because try·finally·synchronized is not a divergence, it is doesn't insert.

- Because interface doesn't have the method, it is excluded from the insertion.
The argument extraction is done based on the following rules.

- As for the extraction of the type name and the argument name, A is disregarded. The affixing character in [] is disregarded.
- The argument to which the variable identifier is omitted is assumed to be an extraction off the subject.

The argument embedded is done based on the following rules.

- When FuncIn of the function is embedded, the extracted argument is continuously embedded only by the argument.
- The Test Point of the argument inserted in an automatic insertion is set directly as a variable of the reference.

At this time, the output is assumed to be level 1.
6.6. Limitations of automatic insertion[Java]

In the file whose extension is "*.java", there are some limitations for the automatic insertion.

- A correct code formatting is not guaranteed where the following content exists.
  - Compile error
  - Function that omits return value and argument (The constructor destructor is excluded).
  - Alternative token (alternative mark)
  - Tri graph (three character mark)
  - Macro character string

- There is a thing that the embedded point shifts behind an original part when the macro exists.

- A normal embed is not warrantable for the source code where the function that omits the return value and the dummy argument exists.

- Even if the argument can be extracted to the function where FuncIn is not embedded, the Test Point is not embedded.

- The argument cannot be embedded automatically at an additional embed.

- When the argument is an array when the argument is inserted automatically for the source of Java, the multidimensional array is excluded from the insertion.
  An automatic insertion is done only as for one dimension array.

- It detects as an error when Japanese and '$' are used for the variable identifier by the source file of Java and an automatic insertion of the source file is not done.
6.7. Automatic insertion example of Test Point

#include "out.h" /* For DynamicTracer-TestPoint */

// 関数定義 -----------------------------
void func_b(int arg_b)
{
    short cnt_a;
    short cnt_b = 10;
    int cnt_c, cnt_d;

    DtTestPoint(__DtFunc_func_b, __DtStep_0)/#FuncIn/
    { [a]
        cnt_a = 10; [a]
        cnt_a = 1, cnt_d = 20; [a]
    } [a]

    for(cnt_a = 0; cnt_a < cnt_b; cnt_a++) {
        #ifdefined ENABLE_DR_DISABLE
        cnt_a++; [a]
        #else
        char a = cnt_a; [a]
        ext_func_b(a); [a]
        #endif [a]
    } [a]

    if(arg_a == cnt_c) {
        int temp = 20; [a]
        __DtTestPoint(__DtFunc_func_b, __DtStep_1)/#fun/
        { [a]
            cnt_c += temp; [a]
        } [a]
    }
}

// Actual insertion example①

It doesn't insert it in a free block definition.

It doesn't insert it when there is a pre-processor directive (# ~ instruction) even if it is an insertion position.
If the insertion position enters in optional for the compilation, it is possible to insert it. To the last, case where preprocessor directive exists for target point is the limitations.

Case and default for switch

Such a pattern can be a burial because it makes it to the burial target.

The insertion of switch is canceled by #endif.

The insertion of return is canceled by #endif. Insertion cancellation of return is limited to the morrow in the instruction of #if/#else/#elf/#endif.
Even if the internal block is empty for if, for, while, do, and switch, insertion is executed.

When function exit is inserted
When return is processed by the above line, it doesn't insert.

For if, for, and while
When the block is not defined by {}, it doesn't insert it.

```c
void func_a(int args_a)
{
    __DtTestPoint(__DtFunc.func_a, __DtStep.0)/FuncIn/
    for (cnt_a = 0; cnt_a < cnt_b; cnt_a++) {
        __DtTestPoint(__DtFunc.func_a, __DtStep.1)/for/
    }
    if (args_a == cnt_c) {
        __DtTestPoint(__DtFunc.func_a, __DtStep.2)/if/
    }
    while (cnt_c < 200) {
        __DtTestPoint(__DtFunc.func_a, __DtStep.3)/while/
    }
    do {
        __DtTestPoint(__DtFunc.func_a, __DtStep.4)/do/
    } while (cnt_c < 210);
    switch (args_a) {
        case 0:
            __DtTestPoint(__DtFunc.func_a, __DtStep.5)/switch/
            break;
    }
    for (cnt_a = 0; cnt_a < cnt_b; cnt_a++)
    if (args_a == cnt_c)
    while (cnt_c < 200)
    __DtTestPoint(__DtFunc.func_a, __DtStep.6)/FuncOut/
    return;
}
```
特別な目的のあるコンパイルオプションブロック等で、解析不可能となるコードがあった場合には、それ以上解析ができなくなりますため、「無効コード」としてこのブロック内を無効にして解析させてください。

forなどの内部ブロックが；
It becomes impossible to
analyze any more when there is
a code that becomes impossible
to analyze in the compilation
option block etc. with a special
purpose it. Therefore, please let
me invalidate as "Invalid code"
in this block and analyze it.

The internal block such as for;
It is possible even to hold it
inserts it in the former line as
processing of one

When the block scope is not
made from {}, the plastic
operation is not correctly done.

Within the range of the effect, of
"Invalid code", even
corresponding # endif is all. It is
assumed that it is invalid
including # else side even if
there is # else on the way.
It doesn't insert it at the previous state of return because it is given priority not to insert it when there is no {} from the insertion of return.

It doesn't insert it in the final edge when there is already return in the passing (Nest level equal with level that entered function. In block scope without condition branching).

To insert it regardless of optional effective invalidity for the compilation, it doesn't insert it in the final edge in consideration of above return.
《実際の挿入例⑥》

```c
int func_d(int d)
{
    int a;
    _DtTestPoint(__DtFunc_func_d, __DtStep_0)/*.FuncIn*/
    [a
    int b;
    ]

    if (a == 1) {
        int c;
        _DtTestPoint(__DtFunc_func_d, __DtStep_1)/*.If*/
        [c
         int e;
         _DtTestPoint(__DtFunc_func_d, __DtStep_2)/*.FuncOut*/
         return 1;
         ]
    }
}

    switch (d) {
    case 2:
        int f = 2;
        _DtTestPoint(__DtFunc_func_d, __DtStep_3)/*.Switch*/
        [f
         int g;
         _DtTestPoint(__DtFunc_func_d, __DtStep_4)/*.FuncOut*/
         return (2);
         ]
    case 4:
    case 6:
        int h = 8;
        _DtTestPoint(__DtFunc_func_d, __DtStep_5)/*.Switch*/
        [h
         return (3);
         ]
    case 4:
    case 6:
        int i = 4;
        _DtTestPoint(__DtFunc_func_d, __DtStep_6)/*.Switch*/
        [i
         int j;
         _DtTestPoint(__DtFunc_func_d, __DtStep_7)/*.FuncOut*/
         return (4);
         ]
    }
    _DtTestPoint(__DtFunc_func_d, __DtStep_0)/*.FuncOut*/
    return (5);
}
```

- It is fixed by thinking the block scope to be a mass of processing as the insertion point.
- It is fixed by thinking the block scope to be a mass of processing as the insertion point.
- It is fixed by thinking the block scope to be a mass of processing as the insertion point.
- To permit as block scope of the divergence, {} immediately after case and default is not inserted. (Treat as well as { of if(){)
- It thinks as free block scope if it is not immediately after case in the case divergence {}. It is fixed by thinking the block scope to be a mass of processing as the insertion point case.
《Actual insertion example⑦》

```c
#include "out.h" /* For DynamicTracer-TestPoint */

// 通常関数  */
int Func_A( int a )
{
    _DtTestPoint(__DtFunc Func_A, __DtStep_0)/*FuncIn+Out*/
}

// void型関数 */
void Func_B( void )
{
    _DtTestPoint(__DtFunc Func_B, __DtStep_0)/*FuncIn+Out*/
}

// 戻り値・仮引数が省略された関数 */
Func_C( )
{
}
```

The return value and the dummy argument are done and the analysis is done normally as for void.

When the return value and the dummy argument are omitted, the analysis is not normally done.
Actual insertion example

```c
/* 分離形式(古い規格)の関数 */
int Func_D ( s1, s2, len )
int &s1;
int &s2;
unsigned int len;
{
  __DtTestPoint(__DtFunc_Func_D, __DtStep_0)/#FuncInOut#/
  return 0;
}

#define AAA  int aaa
#define BBB  int bbb;
#define CCC(x) x = 1;
#define DDD(x) x = 2;
#define EEE a = 0;
#define FFF bbb = aaa;

/* マクロ使用 */
int Func_E ( int a )
{
  __DtTestPoint(__DtFunc_Func_E, __DtStep_0)/#FuncIn#/
  AAA;
  DDD(a) if ( a == 1 ) {
    BBB a = 3;
    __DtTestPoint(__ DtFunc_Func_E, __DtStep_1)/#if#/
    DDD(a);
    return 1;
  }
  else {
    __DtTestPoint(__DtFunc_Func_E, __DtStep_2)/#if#/
    DDD(a);
    do {
      FFF EEE;
      __DtTestPoint(__DtFunc_Func_E, __DtStep_3)/#do#/
    } while ( 0 );
    __DtTestPoint(__DtFunc_Func_E, __DtStep_4)/#FuncOut#/
    return 2;
  }
  CCC(a);
  __DtTestPoint(__DtFunc_Func_E, __DtStep_5)/#FuncOut#/
  return 3;
}
```

There is no guarantee that the function of separate form is analyzed normally.

Even if the content is a declaration of the variable etc., because it doesn't do the macro generation, the analysis is done as a usual token.

Both the plastic operation and insertion become outside the guarantee for the macro to which the semicolon is not attached, too. It is taken care of as a usual token by the form judged to be the most appropriate.
7. Insertion of the Test Point

7.1. Automatic Insertion of the Test Point

DT10 can insert Test Point to function entrance, function exit, and each branch of the function. Automatic insertion of Test Point can execute when add the source file to DT project or execute next step.

Select to [Plan]→[Insert Test Point automatically] in menu, and [Insert Test Point automatically] dialog is displayed. In this dialog you can check the source file's checkbox you want to insert Test Point. After check the checkbox, click the [OK] button, and Automatic insertion of Test Point is executed.

■ All ON
  - Turn ON the all checkboxes.

■ All OFF
  - Turn OFF the all checkboxes.

■ Remove all existing Test Points before insert.
  - To insert Test Point after cleared all information, Please put the check in this check box.
  - When turn OFF the this checkbox, The Test Point information (Include the Property information) that inserted automatically or manually are taken over, and execute added insertion of Test Point to insufficient place. This insertion is called "Added Insertion".
  - If no Test Point was inserted in Source code, the "added insertion" is treated as "new insertion". The auto insertion for register the source file is become either of [Import], [Automatic Test Point insertion is not executed (Only registration)] or [Automatic insertion after the Test Point is deleted once].

■ Auto inserting Variable Value TP to Argument of each Function
  - To insert the Test Point of the argument attribute, turn ON this checkbox and execute automatic insertion.
Attention when inserting additionally

Only one Test Point a function of FuncIn can be inserted. Because FuncIn enters the upper part of the test concerned point from the position in which FuncIn that was added the code to the upper part of the Test Point of FuncIn/FuncIn+FuncOut by former code at an additional embed or added by a manual embed should be originally inserted by an automatic embed when it is the lower side (executable when FuncIn when FuncIn doesn't exist in the function is deleted), the FuncIn attribute of a point concerned is released.

\[
\text{[FuncIn]} \quad \Rightarrow \quad \text{[Free]}
\]

\[
\text{[FuncIn+FuncOut]} \quad \Rightarrow \quad \text{[FuncOut]}
\]

<Ex>

```c
UINT16 bcd2hex( UINT16 bcd )
{
    UINT16 temp = Bcd2Hex[bcd & 0xff];
    bcd = bcd; /* Insert New Line! */
    __DtTestPoint(__DtFunc_bcd2hex, __DtStep_0)
    if ( bcd > 0xff ) {
        __DtTestPoint(__DtFunc_bcd2hex, __DtStep_1)
        temp += ( bcd >> 12 & 0x00ff ) * 1000;
        temp += ( bcd >> 8 & 0x00ff ) * 100;
    }
    __DtTestPoint(__DtFunc_bcd2hex, __DtStep_2)
    return ( temp );
}
```

- After additional insertion

```c
UINT16 bcd2hex( UINT16 bcd )
{
    UINT16 temp = Bcd2Hex[bcd & 0xff];
    __DtTestPoint(__DtFunc_bcd2hex, __DtStep_0)
    bcd = bcd; /* Insert New Line! */
    __DtTestPoint(__DtFunc_bcd2hex, __DtStep_1)
    if ( bcd > 0xff ) {
        __DtTestPoint(__DtFunc_bcd2hex, __DtStep_2)
        temp += ( bcd >> 12 & 0x00ff ) * 1000;
        temp += ( bcd >> 8 & 0x00ff ) * 100;
    }
    __DtTestPoint(__DtFunc_bcd2hex, __DtStep_3)
    return ( temp );
}
```
To insert the Test Point to only one source file, you can open [Insert Test Point automatically] dialog in Folder view and this dialog is useful to find the source file.

Select the source file from Folder view, and right-click, and select [Automatic Test Point insertion] in displayed menu, and you can see [Automatic Test Point Insertion] dialog with the view of selected file.
7.2. Manual insertion of Test Point

7.2.1. Insert Test Point

In DT10, the Test Point can be inserted in user's wherever you like manually.

When you right-click on the source code window, the menu will be displayed. Please select Test Point from the menu.

The type of the Test Point can be selected from among the following.

<table>
<thead>
<tr>
<th>Type of the Test Point</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Step</td>
<td>It is the Test Point to output the trace information.</td>
</tr>
<tr>
<td>Function entrance (FuncIn)</td>
<td>It is the Test Point to output the trace information and it is inserted to entrance of function. Only one FuncIn Test Point can be inserted in each function.</td>
</tr>
<tr>
<td>Function exit (FuncOut)</td>
<td>It is the Test Point to output the trace information and it is inserted to exit of function.</td>
</tr>
<tr>
<td>Variable Value Point</td>
<td>It is the Test Point to output the variable value. You can see detail description at [21. The specified value of a variable is displayed in the Test Report].</td>
</tr>
<tr>
<td>CPU Load Measurement Point</td>
<td>It is the Test Point to measure the CPU load. You can see detail description at [19. Look at CPU load detection information in the Performance Monitor].</td>
</tr>
<tr>
<td>Event Trigger Point</td>
<td>It is the Test Point to use the event trace function. You can see detail description at [20. Event Trace Function].</td>
</tr>
<tr>
<td>Automatic operation insertion cancellation step</td>
<td>It is the test Point which is canceled insertion when there is preprocessor at the insertion point of the Test Point. You can see detail description at [6.4. Limitations of automatic insertion[C/CPP]].</td>
</tr>
<tr>
<td>Write Test Point</td>
<td>It is the Test Point to use the variable value writing function. You can see detail description at [24. Variable value writing function].</td>
</tr>
</tbody>
</table>

Attention

When the manual operation of the Test Point is inserted, it is necessary to execute the insertion of the Include sentence from the pop-up menu of the source code window separately when the Test Point is not inserted automatically.

It becomes impossible to operate the target correctly according to the insertion position of the Test Point. It becomes outside the guarantee for that case, and note the insertion position, please when you insert the manual operation.

In manual insertion of the Test Point, the function to reformat source code may be out of date. Please insert manually after editing the source code in an external editor when you want to insert the Test Point in a necessary source code reformatting (The example: part etc. without the space for the line that inserts the Test Point) manually.
**Memo**  About the type of Normal Step.

A Normal Step inserts the Test Point of free. The Test Point of this free comes to be able to select types of Steps FuncIn, if, and for etc. by clicking "Type" of the Property View of the main window.

![Property View](image)

### 7.2.2. The function of the Test Point insertion destination is defined

The belonging function of the Test Point when the manual insertion with FuncIn Test Point and FuncOut Test Point is judged in DT10.

It is judged that the insertion place of the Test Point in the application of DT10 is vague, and is displayed the following dialogs when inserting without inserting the Test Point automatically by manual operation insertion, and inserting the Test Point in the function in the state that the Test Point of FuncIn has not been inserted by manual operation insertion.

![Specify function to be inserted Test Point](image)

The function is specified at the insertion destination of the Test Point when "The function definition is added" is selected, "Function name" is filled in, and the "OK" button is clicked. Whether to select the function of the correspondence from the list box of the dialog.
7.3. Delete Test Point

7.3.1. Delete Test Point

When the Test Point is selected on the source code window and [Delete Test Point] of the pop-up menu is clicked, the selected Test Point is deleted. When [Delete Test Point] is selected, writing in the source file, the update of the header file, and the preservation of the Project file are done.

**Memo**  
Remove ineffective function

The function from which any Test Point has not been inserted in the function is said, "Ineffective function". When this invalid function is deleted from the DT Project, a pertinent function is selected on the module view / folder view and [Remove ineffective function] of the pop-up menu displayed by right-clicking is selected.
### 7.3.2. Delete Test Point by the batch

In DT10, the Test Point in the specified source file can be deleted by the batch.

“Delete Test Point” dialog is displayed by selecting [Plan]→[Delete Test Point] of the menu, or selecting Test Point deleted and clicking [Delete Test Point] from the pop-up menu in Folder View.

※ The Include sentence of the header file that DT10 generates (# include “xxxx.h” /* line of For DynamicTracer-TestPoint */) is deleted.

---

**All ON**

- All the check boxes are turned on.

**All OFF**

- All the check boxes are turned off.

**Test Point information on the Project is not deleted.**

- Only the macro statement of the Test Point can be deleted from the selected source file when this check box is turned ON, and the Test Point is deleted, and Test Point information be left for the DT Project.

- It is convenient for the version management tool to delete the Test Point, to put the check in “Test Point information on the Project is not deleted” at the check-in etc., and to delete the Test Point.

- After the check is put in “Test Point information on the Project is not deleted.”, and the Test Point is deleted, the file of the DT Project of the correspondence is opened, and DTMerge starts when reading the Project file is generated, and the Test Point is inserted in the source code by displaying the source from folder view / module view onto the source code window based on information on the Test Point that exists in the DT Project.
7.4. Convert to the Fast Test Point

7.4.1. Outline

Inserted Test Point can convert to Fast Test Point which reduced the Overhead.

By using the Fast Test Point, to check the point which is difficult to run the Test Point due to the Overhead is possible.

To use this function, insertion of the output function of the Fast Test Point is needed. For this function, please refer to sample in Install Folder in [sample] -> [driver_samples].

The Fast Test Point is using following format.

**<<C/CPP File>>**

<table>
<thead>
<tr>
<th>Test Point</th>
<th>__DtTestPointFast( data, bit_length )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Value Point</td>
<td>__DtTestPointValueFast( data, bit_length, value, size )</td>
</tr>
<tr>
<td>Event Trigger Point</td>
<td>__DtTestPointEventTriggerFast( Eventdata, bit_length )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>data</th>
<th>Sequential number of the Fast Test Point in the Source file</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit_length</td>
<td>Fast Test Point can insert up to 1024(10bit). &quot;bit_length&quot; show you the bit number of the Fast Test Point</td>
</tr>
<tr>
<td>value</td>
<td>variable name</td>
</tr>
<tr>
<td>size</td>
<td>output byte size of variable</td>
</tr>
<tr>
<td>Eventdata</td>
<td>Value of the event data. User set variable name and direct value.</td>
</tr>
</tbody>
</table>

**<<Java File>>**

<table>
<thead>
<tr>
<th>Test Point</th>
<th>DtTestPointDriver.DtTestPointFast( data, bit_length )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Value Point</td>
<td>DtTestPointDriver.DtTestPointValueFast( data, bit_length, value, size )</td>
</tr>
<tr>
<td>Event Trigger Point</td>
<td>DtTestPointDriver.DtTestPointEventTriggerFast( Eventdata )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>data</th>
<th>Sequential number of the Fast Test Point in the Source file</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit_length</td>
<td>Fast Test Point can insert up to 1024(10bit). &quot;bit_length&quot; show you the bit number of the Fast Test Point</td>
</tr>
<tr>
<td>value</td>
<td>variable name</td>
</tr>
<tr>
<td>size</td>
<td>Size of array of the output variable</td>
</tr>
<tr>
<td>Eventdata</td>
<td>Value of the event data. User set variable name and direct value.</td>
</tr>
</tbody>
</table>

※You can see following detail description about Variable Value Output Point and Event Trigger Point.

Variable Value Output Point→【21. The specified value of a variable is displayed in the Test Report】

Event Trigger Point→【20. Event Trace Function】

※CPU Load Measurement Point and Write Test Point is not correspond to Fast Test Point
**Attention**

Fast Test Point conversion function can use when connect to target with GPIO/SPI connection.

**Attention**

Fast Test Point is not have address data due to reducing the overhead. And It identify the Test Point by using 10bit data in maximum. So, user can insert Fast Test Point to only one source file which registered DT project.

Fast Test Point can insert up to 1024(10bit). Amount of Test Point Data is changed by data value's effective bit length. So, insertion to top of Source file makes Fast Test Point Faster.

**Attention**

Fast Test Point can insert up to 1024(10bit). When Fast Test Point conversion is executed against the source file that have 1025 or more Test Point, the Test Point 1025 or later is become Normal Test Point.

### 7.4.2. Fast Test Point conversion

You can see [Fast Test Point convert...] dialog when user select to [Plan]→[Fast Test Point convert...].

Check the source file's checkbox which you want to convert to Fast Test Point in this dialog, and click [OK] button.
Specified Source file's Test Point convert to Fast Test Point.

In Folder view or Module view, The source step which converted Fast Test Point is displayed with [Fast].

7.4.3. Release of the Fast Test Point

[Fast Test Point convert...] dialog can convert Normal Test point from Fast Test Point.

Click the [All are canceled] button or turn OFF all checkbox, and click [OK] button. Then, Fast Test Point converts to Normal Test Point.
7.5. Enable or Disable Test Point

7.5.1. Enable or Disable specified Test Point

In DT10, only a certain point can acquire the Test Report by switching the check box that exists in the Folder View Module View, and switching valid and the invalidity of the Test Point.

In the item of the Step, the Test Point becomes valid if the check is put in an individual check box, and when the check is removed, the Test Point becomes invalid. In items other than the Step, the batch specification by the Test Point type can be done. When the type that changes by the dialog is selected, the Test Point of the type can be changed by the batch because Test Point type specification dialog is displayed when the check boxes other than the Step are left-clicked.

When the header file is created, the change of the check box is decided.

The mark of the state of the check box is expressed by the following three states

<table>
<thead>
<tr>
<th>Check box</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>Hereafter, it is valid in all items.</td>
</tr>
<tr>
<td>☐</td>
<td>Hereafter, it is invalid in all items.</td>
</tr>
<tr>
<td>☐</td>
<td>Valid and invalidity exist together in the following items.</td>
</tr>
</tbody>
</table>
It is also possible to change valid and the invalidity of all Test Points by the batch. It right-clicks on the Folder View or Module View, and the pop-up menu is displayed. The check on all the check boxes comes off when [All Test Points OFF] that invalidates all the Test Points because of the pop-up menu is selected, and the check adheres to all the check boxes when [All Test Points ON] that makes all the Test Points valid is selected.

**Attention**

When the header file is created, the change of the check box is decided. The header file is not updated only by switching the check box, and the Test Report that reflects a valid, invalid change of the Test Point cannot be acquired. Please note it.
7.5.2. Create header file

Source (header) file is not updated only by switching the check box of the Test Point. When the check box is switched, it is necessary to renew an effective Test Point invalidly.

When [Test Plan] -> [Update Test Point setting] of the menu is clicked, "Update Test Point setting" dialog is displayed.
The header file name and the preservation place are set by this dialog, the button of [Execution of update] is clicked.

![Update Test Point settings dialog]

- **Header file name**
  - The character string added to the head of the header file is specified. Default is "DT_".
  - Please make the column of the character string a blank when you want to make it to the same header file name as the source file.

- **Folder to save header file**
  - The storage place of the header file is specified.

- **Execution of update**
  - The header file source file is updated when clicking on a button and valid and the invalid of the Test Point are decided. Test Point.
At the file of C/CPP

When the state of the Test Point is updated, the header file is made again.

Before the update

```c
#define Dt_DtPoint_UserMain_DStep_0 /#FuncIn#/  TP_BusOut( DBaseAddress, 0x0080 );
#define Dt_DtPoint_UserMain_DStep_1 /#switch#/  TP_BusOut( DBaseAddress, 0x0085 );
#define Dt_DtPoint_UserMain_DStep_2 /#switch#/  TP_BusOut( DBaseAddress, 0x008F );
#define Dt_DtPoint_UserMain_DStep_3 /#switch#/  TP_BusOut( DBaseAddress, 0x0060 );
```

After the update

```c
#define Dt_DtPoint_UserMain_DStep_0 /#FuncIn#/  
#define Dt_DtPoint_UserMain_DStep_1 /#switch#/  TP_BusOut( DBaseAddress, 0x0085 );
#define Dt_DtPoint_UserMain_DStep_2 /#switch#/  TP_BusOut( DBaseAddress, 0x0060 );
#define Dt_DtPoint_UserMain_DStep_3 /#switch#/  TP_BusOut( DBaseAddress, 0x0061 );
```

Test Point invalidity

At the file of Java

When the state of the Test Point is updated, Comment out does the Test Point.

```java
public static void funcA() {
    DtTestPointDriver.DtTestPoint(0x00000010, 0);
    Flag = ON;
    //Dt-Disable   DtTestPointDriver.DtTestPoint(0x80000010, 1);
}
```

The Test Point is done in comment out, and invalidated.
7.5.3. Disable all Test Points

If [All Test Points OFF] is clicked on the pop-up menu in the Folder or Module View, all the Test Points are deleted. However, the information if the Test Point has been Enabled or Disabled before is also deleted.

In DT10, by clicking [Plan] > [Disable all Test Points] on the menu, Test Point information can be canceled temporarily without changing the Test Point information enabled or disabled. Moreover, if [Disable all Test Points] is clicked again, all the Test Point information will go back before.

(※ Please note becoming off the subject of "All the Test Points are prohibited" command when the source file is Java.)

Because the header file is created automatically at the same time as disabling all Test Points, the header file need not be created again.

The value of "#define __DtAllEnable" in the header file becomes "0" by generating the header file. (The value of "#define __DtAllEnable" at time is "1" usually) As a result, # if sentence of the next line becomes invalid, and all the Test Points can be invalidated with a valid, invalid setting left.

![Header file creation](image)
8. Operation on source code window

In the source code window, it is possible to operate as follows.

① Find
② Color Setting
③ Fontsize change
④ Source code edit
⑤ Line Jump
⑥ Find Test Report
⑦ Shortcut key

(Please see 【35.1. Shortcut key on source code window】 in detail.)

■ Tool bar

①～③ is operated by ‘Source code window toolbar (It is only recorded later as the toolbar)’ of the main screen.

④⑤ is possible from the toolbar and the pop-up menu (menu displayed by right-clicking) in the source code window.

Source code window toolbar

<table>
<thead>
<tr>
<th>①</th>
<th>②</th>
<th>③</th>
<th>⑤</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Process" /></td>
<td><img src="image" alt="Color Setting" /></td>
<td><img src="image" alt="Fontsize change" /></td>
<td><img src="image" alt="Source code edit" /></td>
</tr>
</tbody>
</table>

Standard tool bar

④

![Standard tool bar](image)
8.1. Find

It retrieves in the character string in the source code window.

The source code window to be retrieved is valid window in now (It is active in the source code window).

The button related to the find is nullified when there is no valid source code window.

The retrieval begins from the position of a present cursor. The retrieval ends if returning to the position of a present cursor. The character string is made a selective state if the specified character string is found and the retrieval ends. In this case, the retrieval does the starting point and the terminal of the file to the toggle (After it retrieves it to the file terminal, retrieve it from the file head when the retrieval is put from the position of the terminal of the file on the direction of ↓).

The character string was not found if there was no change in the selection character string and the cursor.

When the retrieved character string is found, it is registered in the combo box as a history (Up to 50). In the history, the character string retrieved recently is displayed at a high linking position.

The methods for retrieval operation are followings.

Find dialog

The Find dialog is displayed when button of the source toolbar is selected.

If the character string in the source code window has been selected, the character string is assumed to be a retrieval character string (initial value). Of course, it is possible to change to a past selection character string in the dialog, and to set the character string newly again. It is retrieved that details are optional of the retrieval of the direction of the retrieval, the case sensitivity (identify), and each word etc. specifying it.

Word only: The retrieved direction is specified.

Case-sensitive: Whether the capital letter of the alphabet and the small letter are treated in the same way when retrieving or treats separately is specified.

Find string: The delimitation of the character string judges whether to agree with the unit of the word as the word district switching off that is not following character.

- ‘a’ ~ ‘z’
- ‘A’ ~ ‘Z’
- ‘0’ ~ ‘9’
- ‘-‘, ‘#’
- 2 bite character

In retrieval for no unit of word, the position of comparison beginning is one word unit, and the comparison end position is end of line.
“Find” button: The retrieval begins when the retrieval beginning button is clicked. Also ‘Enter’ begins retrieving.

Distinction of specified capital letters and small letters and find-option by word unit are succeeded to the following retrievals.
Moreover, it is preserved when the program ends, and it is read when DT10 will start next time.

■ Find with ‘Enter’
The character string is input to the retrieval combo box of the toolbar, and the retrieval begins with ‘Enter’
Retrieval direction is ↓. When ‘Enter’ is input while pushing the shift key, retrieval direction is ↑.
This function is valid while focus in the retrieval combo box.

■ Find with UP/Down button
The character string displayed in the combo box is retrieved up direction (UP button) or down direction (Down button).

8.2. Color Setting

The color setting dialog is displayed with buttons on the tool bar.
The character string of the source code window is classified.
It classifies into the following items.

- Trace Point (Non-pass): The one that Test Point is not passed.
- Trace Point (Pass): The one that Test Point was passed (After analyzes).
- C language Reserved word: if while int #include #ifdef etc.
- C language Comment: Character string enclosed with /* and */. It is a character string since // by each line.
- C language Character string: Character string enclosed with " and ".
- Line number: Line number displayed in left end of source code window in option.
- User definition Keyword: The user can register a favorite key word.
  Three kinds of categories can be defined, and two or more character strings can be defined respectively.
- Character color: Character that doesn't apply to either the above-mentioned
- Background color: Background color of source code window

The color can be set individually. Moreover, the selection that doesn’t do the classification function either can be done.
Setting

The classification is set by the following color setting dialogs.

Color Setting

It selects setting color or not from among as follows.

- Code color
- Code color of only Trace Point
- Don't code color

Color setting of item

The color of each item is set. There are two kinds of operations.

① After the item is selected, the change button is clicked. In this case, the setting that selects two or more items is possible.

② The item is double-clicked.

Both standard color setting dialogs are displayed and the color is selected.

The set color can be made sure in the window below. It is possible to return it to an initial value with the button [Restore Default].
Register Keyword

The keyword can be registered with the key word registration button.
The key word has three categories. And separate color and an optional comparison can be set to each category.
The following key word setting dialog is displayed from key word 1 with three key word buttons.

Key word: The key word is specified. When plurals are specified, changes line and specifies.
Compare Options: It is the same meaning of the retrieval function as the option.

Color sample

The whole in a present setting status can be seen.

8.3. Font size Change

The font of the source code window grows if it clicks on a button. The font becomes small if it clicks on a button. When being start next time of the application, the font size is maintained.
A related button becomes invalid when there is no valid source code window.

8.4. Source code edit

The source code can be edited on the source code window.
When an addition of the comment on the source code and a small-scale change are done, it is convenient. (* We will recommend the work such as coding to be done basically in an external editor.)

Copy

The character string of the source code window is selected, and button on the toolbar is clicked or when "Copy" of the pop-up menu is selected, copies onto the clipboard. "Ctrl+C" on the source code window becomes similar, too.
### Cut

The character string of the source code window is selected, and button on the toolbar is clicked or when "Cur" of the pop-up menu is selected, the character under the selection is cut out. "Ctrl+X" on the source code window becomes similar, too.

The Test Point cannot be cut out.

When the object file is an inspection mode, it is not possible to cut out.

The clipboard empties empty data when executing it with nothing has been selected, and it enters, and the clipboard empties.

The color substitution of the change line by the cutout changes the color judging only by a pertinent line regardless of processing before and after.

It saves a file immediately after having cut out.

### Paste

Buttons on the toolbar are clicked or when "Paste" of the pop-up menu is selected, it puts it. "Ctrl+V" on the source code window becomes similar, too.

When the object file is an inspection mode, it is not possible to put.

It is not possible to put for the line with the Test Point.

The color substitution of the change line by putting changes the color judging only by a pertinent line regardless of processing before and after.

It saves a file immediately after having cut out

### Line edit

Every one line is edited on the source code window of DT10.

It becomes becoming the edit mode if the cursor is matched to the line to be edited, and 'Enter' is pushed, and possible the character entry compared with the source code.

(* Do not become an edit mode when the range selection, and the file are for reading alone.)

The cursor moves from the edit line or the edit mode ends if 'Enter' is pushed again.

The color substitution of the change line by the line edit changes the color judging only by a pertinent line regardless of processing before and after.

Preservation to the file is done at each end of the edit mode.
The edit is canceled by the escape key, and returns at the time of the edit.
The key operation of Ctrl+Z (Return it) and Ctrl+Y (repetition) becomes valid in the edit mode. This operation becomes invalid at the same time as ending the edit mode.

■ Key operation on source code window

Changing line is put in the cursor position with [Ctrl] + Enter key.
The word just before the cursor is deleted with [Ctrl] + BackSpace key.
The word just behind the cursor is deleted with [Ctrl] + Delete key.
Only the number of constant characters moves right in the cursor with the Tab key.
Only the number of constant characters moves left in the cursor with Shift + Tab key.

It saves a file immediately after the change of the source code.

■ About the preservation of the file

When the source code is changed on the source code window, it saves a file in every case.
It saves a file at that time when focus is moved while editing it.
However, the following messages are displayed only when the window is closed, and select whether to preserve the document, please
8.5. Line Jump

When it right-clicks on the source code window, and the [Jump] is selected, the following menus are displayed.

Jump to specified position: The following dialogs are displayed. The line number is specified in the remark, and the jump to a specific position is also possible. "Ctrl+G" on the source code or the button of the toolbar becomes similar, too.

- **To the top of the file**: It jumps to the head of the file that has been selected now.
- **To the end of the file**: It jumps to the terminal of the file that has been selected now.
- **To the top of previous function**: It jumps to the first Step of a present function.
  
  "Ctrl+K" on the source code window becomes similar, too.
  
  "Ctrl+P" on the source code or the button of the toolbar becomes similar, too.
- **To the top of present function**: It jumps to the first Step of a present function.
  
  "Ctrl+K" on the source code window becomes similar, too.
To the top of next function: It jumps to the first Step of the following function.
"Ctrl+N" on the source code or the button of the toolbar becomes similar, too.

To the previous Test Point: It jumps to the previous Test Point.
"Ctrl+U" on the source code window becomes similar, too.

To the next Test Point: It jumps to the next Test Point.
"Ctrl+D" on the source code window becomes similar, too.

8.6. Find Test Report

The list item of the Test Point that selects the Test Point on the source code window and exists in the Test Report can be retrieved. As for this retrieval, the first list item below the list item that has been selected in the Test Report to which the condition is corresponding is retrieved. (Lower retrieval)

First of all, the search strategy selects the list item of the Test Report that becomes a retrieval beginning position. And, the cursor is matched to the Test Point to be retrieved on the source code window, it right-clicks, and the pop-up menu is displayed. When the retrieval from the Test Report [Find Test Report] is clicked from the pop-up menu, the list item of the selected Test Point is displayed in the Test Report.
# 9. Set the Property

## 9.1. Type of Property

DT10 have next property [Project Property] [Module Property] [Function Property] and [Step Property]. You can see/setting each property's attribute, the nature, and so on.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Property</td>
<td>You can see basic information about DT10.</td>
</tr>
<tr>
<td></td>
<td>Project name, Location of DT project, Root path, Connection type, etc...</td>
</tr>
<tr>
<td></td>
<td>You can see detail description in [5.1.1. Project Property].</td>
</tr>
<tr>
<td>Source file Property</td>
<td>You can see full path name of source file, base address and header file's location.</td>
</tr>
<tr>
<td></td>
<td>Also you can each source code setting. (Setting of Invalid code definition file, Type definition file, Character code setting, Newline code setting, and so on.)</td>
</tr>
<tr>
<td></td>
<td>You can see detail description in [9.3. Source File Property].</td>
</tr>
<tr>
<td>Module Property</td>
<td>You can set each module which is made by DT10.</td>
</tr>
<tr>
<td></td>
<td>(Module name change, Add/Delete the function which belong the module, and so on.)</td>
</tr>
<tr>
<td></td>
<td>You can see detail description in [9.4. Module Property].</td>
</tr>
<tr>
<td>Function Property</td>
<td>You can enter the design value of function's Execution Time/Period Time.</td>
</tr>
<tr>
<td></td>
<td>Entered design value is used to compare to measured value to judge the value is equal to designed value.</td>
</tr>
<tr>
<td></td>
<td>You can see detail description in [9.5. Function Property].</td>
</tr>
<tr>
<td>Step Property</td>
<td>You can enter the design value of the step's Period Time/Loop Count.</td>
</tr>
<tr>
<td></td>
<td>Entered design value is used to compare to measured value to judge the value is equal to designed value.</td>
</tr>
<tr>
<td></td>
<td>You can see detail description in [9.6. Step Property].</td>
</tr>
</tbody>
</table>

Project Property is set by Project Setting dialog. Source file Property, Module property, Function Property, and Step Property is set by Property Editor.
9.2. How to open the Property Editor

To open Property Editor, select the [Plan] -> [Property Editor].

Following Property editor dialog is displayed. Open the list of the left-top of display, and select the property you want to display.

![Property Editor Dialog](image-url)
You can also open Property Editor from Folder view or Module view.

Select the item you want to open its Property Editor and right-click it. Then select [Open Property Editor...], you can see the property editor of selected item.
9.3. Source File Property

It is a dialog that inputs the designed value of the source file.

- Full path name
  - Full path name of the registered source file.

- Description
  - Column where Description of the source file is described.

- Base address
  - Base Address allocated to the source file. This value cannot be modified because it is allocated automatically by DT10.
Header file
- Full path of the generated header file.

Type definition file
- Please describe the type of the variable of the target environment dependence to the text file, and specify the pass to the file.
- When blank, the Type definition file specified by the Project is referred.
- Please enumerate the type definition file as one word (definition) in the file in the line.
- It treats from the head of line to line feed as a definition.
- Please not put Japanese, TAB, blank, null line, etc. the analysis is likely to fail.
- The capital letter and the small letter are judged distinguishing.

Invalid code definition file
- Please describe the character string of the part where it doesn't want to insert the Test Point to the text file, and specify passing the file.
- Refer to the invalid code definition file specified by setting the Project when making it to the blank.
- Please enumerate the invalid code definition file as one definition in the file in the line.
- It treats from the head of line to changing line as a definition.
- The head is not "#" but the line is disregarded.
- As an invalid code, # if~, #ifdef~, #if defined~, please specify it. It is not possible to analyze it for invalid codes other than them.
- Please do not put Japanese. It is likely to fail in the analysis.
- The capital letter and the small letter are judged distinguishing.
- Space between identifier and value is ignored.

Time Unit
- The unit of time in the property editor is changed.
- A revocable unit is Nano second[ns] and Micro second[us], Milli second[ms], and Second[s].
9.4. Module Property

9.4.1. Module function of the DT10

DT10 can display each module function together to using the Module view. (※Module generating is optional)

When registering to DT10 is executing, This Module can generated in automatically based on the folder structure.
You can also generate module in manually in Module view.
Generated module’s setting is changed by Module Property. (You can add/delete the function which belongs to the module, and so on.)

If you generated the Module, you can display the Module name on the Test Report.

In the “Test Result”, the result of the test is displayed each for module.
You can see detail description about the “Test Result” in [25. Create Test Result].

[Image of Test Result table and window]
9.4.2. Automatic Generation of Module

If you want to generate module, you need to check the checkbox [Generate Module in automatically]. You can see that check box on [Specification of the automatic generation file] tab in the [Project setting dialog] or in the page three of [new making wizard]. When turn ON this checkbox, the automatic generation of the module is executed at the timing of execute automatic insertion to source code.

If you check the [Automatic Generation of module Information] checkbox, the number is added to the generate module.

If you check the [Automatic Generation of module Information] checkbox or you check the [The number of each hierarchy is added to the module name] checkbox, following module is generated when execute auto-insertion to the source file.
9.4.3. Addition of module

Because the pop-up menu is displayed when right-clicking on the Module View, [Add module] is selected from the pop-up menu. Then, the properties dialog of the module opens.

When a necessary matter is filled in on the dialog, and [OK] is clicked, the module is added on the Module View.

Configuration when you add a module, which can be changed in the module property.

- **Module name**
  - It is a module name.

- **Description**
  - It is a column where the Description of the module is described.
■ Type

- It is a classification of the processing form of the module. Please use it when you examine the module classification when it designs.
- This content doesn't influence the display and the Test Report of the Test Report.
  
  **Process**: The module of the task is recommended to be classified as a process.
  **Drive**: The hardware control system and the interrupt processing system are recommended to be classified as a drive.
  **Special**: Special processing of the emergency system processing etc. is recommended to be classified as special.
  **Common**: The common processing used from two or more tasks and modules is recommended to be classified.

■ Parent module

- A parent module is displayed.

■ Processed task name (Interrupt)

- It is a column where the processed task name is described.
- This content is reflected in the task column of the Test Report.
- The dispatch part of the task and the part of interrupt can be confirmed by setting the task with the Function Trace Report.

■ Priority

- The priority of the processed task is described.
- This setting doesn't take part in the processing of DT10 for the explanation of the module to the last. Please describe arbitrarily.
9.4.4. Deletion of module

When the module is deleted, the module deleted on the module view is selected, and [Delete Module] is selected.

The following dialogs are displayed, and whether all modules below the module that specifies whether to delete only the specified module are deleted can be selected.

Yes : Only the specified module is deleted.
No : The following all modules of the module are deleted.
Cancel : Processing is discontinued.
9.4.5. The function is added and excluded to the module

It does on the Module View to add and to exclude the function to the module or it does from the Module Property.

The function is added and excluded to the module

◆ When the function is added and excluded on the Module View.

The function item to be added is selected to add the module on the Module View, and item of the module drags it.

Moreover, the function item to be excluded is selected, and it drags it from the module to a high-ranking tree when deleting it from the module.

◆ When the function is added or deleted from Module Property.

First, Module Property is opened. Module Property is opened by clicking [Open Property Editor...] of pop-up menu that is opened by right-clicking on the Module View, or by selecting [Plan] -> [Property Editor...] of the menu.

To add the module, [Addition] of "Belonging function" is clicked. Then, when the function is selected from the dialog, the function is added to the module because the selection dialog of the function is displayed.

Moreover, when the function is excluded from the module, the function to be excluded is selected by having a look at "Belonging function" and [Exclude] is clicked. Moreover, when [Exclude all] is clicked, the function is excluded from the module to exclude all functions.

When the addition of the function and the setting of exclusion etc. are completed, [OK] of the Module Property is clicked.
In the Module Property, the change in the item of "Module name", "Description", "Type ", "Processed task name", and "Priority" set when the module is made is also possible. Please see [9.4.3. Addition of module] about the explanation of each item.

■Belonging function

- It is a list of the function that belongs to this module.
- The function is added, and exclusion is also possible from this dialog.

[Add] : "Selection of the function" dialog is displayed. The function added from the dialog to the module is selected.

[Exclude] : When the function excluded from the list box of "Belonging function" is selected and the [Exclusion] button is clicked, the selected function is excluded.

[Exclude all] : All functions that belong to the module are excluded.

- In "Selection of the function" dialog, the function that has already been registered in a present module is displayed as yellow. Moreover, the function registered in another module is displayed red.
In "Selection of the function" dialog, the list can be permuted by clicking the item name of the item to be permuted in order of going up and down of each item.

■ Belonging module

- It is a list of the sub module that belongs to this module.
- The sub module is added, and exclusion is also possible from this dialog.
  
  **[Add]**: The dialog of "Selection of the module" is displayed. The module added from the dialog is selected.

  **[Exclusion]**: When the excluded module is selected from the list box of "Belonging module" and the [Exclusion] button is clicked, the selected module is excluded.

  **[Exclusion of all]**: All modules that belong to the module are excluded.

- In the dialog of "Selection of the module", the one that has already been registered in a present module is displayed in yellow.
- In the dialog of "Selection of the module", the list can be permuted by clicking the item name of the item to be permuted in order of going up and down of each item.
9.5. Function Property

It is a dialog that inputs the designed value of the function.

- **Description**
  - It is a column where the description of the function is described.

- **Attribute**
  - It is a processing attribute of the function.
    - **Normal**: Please select at normality system processing.
    - **Illegal**: Please select at abnormality system processing of error processing etc.

- **Belonging module**
  - The module that the function belongs is displayed.
Execution

- It is an execution attribute of the function.
  - Normal: It is a setting of default.
  - Unused: ‘Unused’ means that this function is not used in the target. Please select in the invalidated part by compile switch.
  - Unexecute: ‘Unexecute’ means that this function is not executed in the target. Please select when this function is never executed though it is valid on the code.
    (Ex) Exit of main() function.
  - Non-pass: It is shown not to influence the coverage rate though the substance of the function exists. Please select when the call origin is invalid etc.
  - Pending: The Test Point that sets this attribute in which it is shown not to have fixed whether to use this function with the target influences the coverage rate.
    Please distribute it to the attribute of "Normal", "Unused", "Unexecute", and "Non-pass" when the usage condition is fixed.

- When the execution attribute is changed in the function property, it is reflected in the step property in the step in which it belongs to the function.

Argument

- The change of set values such as "Reference", "Level", and "Signed" concerning the Variable Value Point of the argument (Arg attribute) is possible.
- Please see [21. The specified value of a variable is displayed in the Test Report] for details of each item.
- The Variable Value Point of the argument is inserted and cannot be deleted from the Property Editor.

Period

- Please set the period for the function regularly called. Min, Max, Typ (regulated value), and Illegal value can be set.
- It is suitable for the designed value to which an actual Period is set by "Period" when "Period" is set in DT10 can be judged.
- After the test is analyzed, the judgment result can be confirmed from the Folder View and the Module View, etc.
- The setting method puts the check in the check box of "Period", and inputs the designed value to each item.
## Event

- A specific Test Point can be specified.
- When the Test Point is specified, same step and "Variable Value Point", "CPU Load Point", "Event Trigger Point", and "Write Test Point" cannot be specified.
- Please set time until reaching this function from the specified Test Point. It can be judged, time until reaching this function is "Designed value" from the selected Test Point with DT10 weather. Min, Max, Typ (regulated value), and illegal can be set.
- After the test is analyzed, the judgment result can be confirmed from the Folder View and the Module View, etc.
- Because the selection dialog of the Test Point is displayed as for the setting method when [Add] of "Event" is clicked, the Test Point to be specified there is selected. Then, it is reflected in "Event" list, and input the designed value in the list, please.

## Execution Time

- Entrance of function the execution time can be set to each exit. Min, Max, Typ (regulated value), and illegal value can be set.
- Time from the function entry to the exit is suitable for the design when "Execution time" is set can be judged with DT10.
- After the test is analyzed, the judgment result can be confirmed from the Folder View and the Module View, etc.
- Please input the designed value to each item of "Execution time" about the setting method.

## Interface

- It sets in "Interface" tab.
- It sets to the function called from a specific module or function.
- [Select] button of the Interface items is clicked, and the module or the function to call the function by the displayed "Interface setting" dialog is selected.
- Selected module and function can be cleared with a [Clear] button.
- It becomes exclusive with the Common Interface attribute.

## Common Interface

- It sets in "Interface" tab.
- It sets to the function (function shared between modules and the task) called from two or more modules or tasks.
- The [Add] button of the Common Interface item is clicked, and the module, the function or the task of calling the function by the displayed "Common Interface setting" dialog is selected.
- The module and the task under the selection are excluded with the [Exclusion] button. All modules and tasks displayed in the list with the [Exclusion of all] button of the exclusion of all are excluded.
- It becomes exclusive with the Interface attribute.
**Time Unit**

- The unit of time in the property editor is changed.
- A revocable unit is Nano second[ns] and Micro second[us], Milli second[ms], and Second[s].
9.6. Step Property

It is a dialog that inputs the designed value of the Test Point.

- **Description**
  - It is a column where the description of the Test Point is described.

- **Attribute**
  - It is a processing attribute of the Test Point. It uses by the coverage calculation.
    - **Normal**: Please select at normality system processing.
    - **Illegal**: Please select at abnormality system processing of error processing etc.

- **Execution attribute**
  - It is an execution attribute of the Step. It uses to calculate coverage…
    - **Normal**: It is a setting of default.
    - **Unused**: Please select in the invalidated part an optional compilation.
    - **Unexecute**: On the code, please select for the part over which it never logically passes even if it is valid.
      (Ex) Exit of main() function, etc.
    - **Non-pass**: It is shown not to influence the coverage rate though substance exists. Please select when the call origin is invalid etc.
**Pending**

It is shown not to have fixed whether to use this Test Point with the target.

The Test Point that sets this attribute influences the coverage rate.

Please distribute it to the attribute of "Normal", "Unused", "Unexecute", and "Non past" when the usage condition is fixed.

- When the step and the execution attribute of the function are different, the execution attribute of the function is reset in "Normal" in the place where the execution attribute is changed in the step property.

**Period**

- Please set the Period for the Step regularly called. Min, Max, Typ (regulated value), and Illegal can be set.
- It is suitable for the designed value to which an actual period is set by "Period" when "Period" is set in DT10 can be judged.
- After the test is analyzed, the judgment result can be confirmed from the Folder View and the Module View, etc.
- The setting method puts the check in the check box of "Period", and inputs the designed value to each item.

**Event**

- A specific Test Point can be specified.
- When the Test Point is specified, same step and "Variable value output point", "CPU Load Measurement Point ", "Event Trigger Point", and "Write Test Point" cannot be specified.
- Please set time until reaching this function from the specified Test Point. It can be judged, time until reaching this Test Point is "Designed value" from the selected Test Point with DT10 weather. Min, Max, Typ (regulated value), and Illegal can be set.
- After the test is analyzed, the judgment result can be confirmed from the Folder View and the Module View, etc.
- Because the selection dialog of the Test Point is displayed as for the setting method when [Add…] of "Event" is clicked, the Test Point to be specified there is selected.

Then, it is reflected in "Event" list, and input the designed value in the list, please
Loop Count

- A specific Test Point can be specified.
- When the Test Point is specified, same step and "Variable value output point", "CPU load point", "Event Trigger Point", and "Write Test Point" cannot be specified.
- Please set how many times to execute the specified Test Point by the attainment from the specified Test Point to this Test Point. It can be judged, the Loop Count is "Designed value" until reaching this Test Point from the selected Test Point with DT10 weather. Min, Max, Typ (regulated value), and Illegal value can be set.
- It can be confirmed whether the loop frequency is correct when the Test Point in the loop is specified with the Test Point after it loops.
- After the test is analyzed, the judgment result can be confirmed from the Folder View and the Module View, etc.
- Because the selection dialog of the Test Point is displayed as for the setting method when [Add...] of "Loop Count" is clicked, the Test Point to be specified there is selected. Then, it is reflected in "Loop Count" list, and input the designed value in the list, please.

Module start

- It sets in "Interface" tab.
- It sets to the step before the module changes.
- The module and the function changing ahead are selected by "Module start setting" dialog displayed clicking a [Select] button of the module start items.
- Selected module and function can be cleared with a [Clear] button.

Module end

- It sets in "Interface" tab.
- It sets to the step after the module changes.
- The module and the function changing ahead are selected by "Module end setting" dialog displayed clicking a [Select] button of the module start items.
- Selected module and function can be cleared with a [Clear] button.

Task start

- It sets in "Interface" tab.
- It sets to the step before the task changes.
- The module and the function changing ahead are selected by "Task start setting" dialog displayed clicking a [Select] button of the task start items.
- Selected module and function can be cleared with a [Clear] button.
---

### Task end

- It sets in "Interface" tab.
- It sets to the step after the task changes.
- The module and the function changing ahead are selected by "Task end setting" dialog displayed clicking a [Select] button of the task end items.
- Selected module and function can be cleared with a [Clear] button.

### Variable

- The change of set values such as "Reference", "Level", and "Signed" concerning the variable value output point and the variable value writing point is possible.
- Please see [21. The specified value of a variable is displayed in the Test Report] for details of the
- It is also possible Variable Value Point.
- Please see [24. Variable value writing function] for details of Write Test Point.
- The Variable Value Point of the argument is inserted and cannot be deleted from the Property Editor.
- To change the setting concerning the Variable Value Point of the argument (Arg attribute) by this item. In that case, the change result is reflected in the Argument Property of the function.

### Time Unit

- The unit of time in the property editor is changed.
- A revocable unit is Nano second[ns] and Micro second[us], Milli second[ms], and Second[s].

---

**Attention**

In the combo box where the function in the upper part of the Step Property is displayed, the function is displayed in alphabetical order.
9.7. Confirm setting Property

The setting done in the Source File Property, the Module Property, the Function Property, and the Step Property can be confirmed from the Property View of the main screen. It is displayed that the source file, the function, the Step to be confirmed by the Folder View Module View, and the module are selected in the Property View.

Moreover, the NG item judgment results such as set "Period" and "Event".. is displayed in the Function Property and the Step Property after the Report Data is analyzed and “!” mark is displayed by the Folder View Module View when is.

(※The Report Data is analyzed by [Analysis of Test Report] of the menu [Report Analysis].)

The occurrence condition of NG of each item is as follows.

<table>
<thead>
<tr>
<th>Item</th>
<th>NG condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>If you get a number less than the design value, NG is.</td>
</tr>
<tr>
<td>Max</td>
<td>If the number exceeds the value obtained design, NG is.</td>
</tr>
<tr>
<td>Range</td>
<td>It is an item used for the range judgment of the value of a variable.</td>
</tr>
<tr>
<td></td>
<td>When numerical values other than the range of specification are acquired, it</td>
</tr>
<tr>
<td></td>
<td>becomes NG.</td>
</tr>
<tr>
<td>Typ</td>
<td>It is not used in judgment.</td>
</tr>
<tr>
<td>Illegal value</td>
<td>Min, Max, even if conditions are met, if the design value is set to get</td>
</tr>
<tr>
<td></td>
<td>the outlier, NG is.</td>
</tr>
</tbody>
</table>

※1 In "Max" and "Min", Number less than the design value is determined by truncation.

(Ex) "Max" to set 3us.

3001ns → NG, 2999ns → OK

※2 "Illegal value" is judged by the unit of ns of the Nano second.

(Ex) "Illegal value" to set 3us.

3000ns → NG, 3001ns → OK
When there are NG items, if NG item of Property View is double-clicked, NG item list is displayed in main window.

If the item of the NG list is double-clicked, you can jump to the appropriate point in the Test Report.

In Test Report, the error data’s “No,” item is painted green.
## 10. Getting Test Report

### 10.1. Set Test Report collection condition

When [Test] → [Test Report collection condition setting] of the menu is selected, the following dialogs are displayed.

![Test Report collection condition settings dialog](image)

- **Manual**
  - The Test Report acquisition is stopped manually.

- **Attention**
  - When the free space preservation ahead becomes 1024MB or less even when making it to the reactor manual shutdown, acquisition is stopped by the automatic operation.
Auto

- The acquisition of the Test Report can be stopped on the terms and conditions stated below automatically.

  **Time**: The Test Report acquisition is stopped at specified time.
  **Elapsed time**: The Test Report acquisition is stopped by elapsed time from Test Report acquisition execution beginning.

  If Elapsed time is set "0", the automatic stop setting becomes invalid.

  Please click "Stop" by hand Power, and stop getting Report.

- **Log size**: It stops by the data size of the acquired Test Report.

Specify Test Point to start or stop Trace at.

▼Start Point

- The Test Point that start tracing can be specified.
- The specified Test Point is begun and the trace begins in DT10 when the specified frequency is passed.

▼Stop Point

- The Test Point that stop tracing can be specified.
- The specified Test Point is stopped after the trace begins and the trace is stopped in DT10 when the specified frequency is passed.

▼Repeat (Enable multiple start points)

- Multiple start points can be made valid.
- The repetition is done until the stop frequency in which operation that the trace is begun when the specified beginning point is passed, and the trace stops when the specified stop point is passed is specified with the stop point is executed.
- It is possible to specify it when both the start point and the stop point are valid.
- The frequency of the start point becomes a gray display, and is fixed to “1”.

Preservation specification

- Specify the stored place of Profile to save Test Report Data manually.
- A Profile is set of source file of same version. Test Report corresponded to the source file, and Test Point setting. Please see 【27. Display the Report Data】 of the explanation of the Profile in detail.

Create Profile

- Even if the test execution is not done, the Profile is created (preservation of the Project and backup of the source file).
- It is convenient when a new Profile is necessary and when the profile folder is necessary in the use such as DTConverter etc.
Collect Data (Logic input and Analog input) from Analog Box

- Please turn on this check box when you collect data from an Analog Box

Display the Real-time Coverage

- You need to check this checkbox if you want to view "Coverage Report" during getting Test Report. If you want to know "Real-time Coverage", Please refer to [10.8. Real-time Coverage].

Execute

- Display "Execute Program" dialog.
10.2. Setting of Ethernet connection

It is necessary to do connect setting before execution of the test when “Ethernet” or “Ethernet(without Tracer)” is selected by “Tracer connection” of the attribute dialog of the Project when the Project is made.

(※ To output the trace information, the “TP_BusOut” function for Ethernet connection is needed to add to your environment. Please refer to the sample source in the installation folder.)

To do connected setting, [Ethernet connection settings] of [Test] menu is selected.

In the case of Ethernet, when [Ethernet connection settings] is selected, following Ethernet connection setting dialog is displayed. Please set the value according to each dialog.

Ethernet connection, the connection mode “UDP protocol” can reduce the overhead, so that we recommend selecting UDP connection.

If the “Target Program collect the Execution Time Data” is set off, time information of made by DynamicTracer is used in Test Report. If the “Target Program collect the Execution Time Data” is set on, time information of made by Target Program is used in Test Report. The time information made by target program can give time information which is more close to real behavior to user. (※You need to add a output function for the time information to the target program.)
Attention

"IP address" is the IP address of the Dynamic Tracer. User need to match the IP address which set in DT10 driver (DynamicTracer's IP address) and the IP address which is not used in your environment.

| Attention
---|---
If the target program is make the execution time information, user need the correspond firmware for execution mode. If you bought DynamicTracer in before than V6.11 version, you need firmware update. Please contact our support.

<table>
<thead>
<tr>
<th>URL</th>
<th><a href="http://dt10.hldc.co.jp/">http://dt10.hldc.co.jp/</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail</td>
<td><a href="mailto:user@hldc.co.jp">user@hldc.co.jp</a></td>
</tr>
</tbody>
</table>
10.3. Setting of UART connection

It is necessary to do connect setting before execution of the test when "UART" or "UART(without Tracer)" is selected by "Tracer connection" of the attribute dialog of the Project when the Project is made.

(※ To output the trace information, the "TP_BusOut" function for UART connection is needed to add to your environment. Please refer to the sample source in the installation folder.)

To do connected setting, [UART connection settings] of [Test] menu is selected.

When [UART connection settings] is selected, UART connection setting dialog is displayed. Please set the value according to each dialog.
When you connect to target with UART connection (without tracer), the following dialog is displayed.

![UART connection setting](image)

In UART connection without DynamicTracer, user needs to set [COM port]. If user has no COM port, please use the USB serial conversion cable etc. Com port number in use can check on Device Manager. [Parity], [Stop bit] and [Baud rate] need to set setting value of your environment, same as the setting of the UART connection with DynamicTracer.
10.4. Setting SD connection

It is necessary to do connect setting before execution of the test when “SD” is selected by “Tracer connection” of the attribute dialog of the Project when the Project is made.

※ To output the trace information, the “TP_BusOut” function for SD connection is needed to add to your environment. Refer to Install Folder ([sample] → [driver_samples], and please add the “TP_BusOut” function.)

To do connected setting, [Test] >> [SD connection settings] of the menu is selected.

When the [SD connection settings] is selected, the following dialogs are displayed. Please set the command for the transmission for the SD interface, and click “OK”.

The command for the transmission is an arbitrary value corresponding to the 1 byte transmitted by the driver function. Please specify it by the hexadecimal number of 0x00~0x3f.
10.5. Getting Test Report

10.5.1. “Execute program” dialog

If following dialog is displayed

Selecting [Test] -> [Execute] of the menu or [Execute] of “Test Report collection condition setting” dialog, the following dialog is displayed.

![Execute program dialog]

- **Test item name**
  - The item name of the test is filled in.
  - Test item name can be up to 2048 characters.
  - Test item name is displayed in the window title of the Test Report view. If test item name is longer than Window Title size, the name is displayed “...”.
  - This test item name is reflected in the Test Report. If the name length is longer than frame length, the name is omitted like “...”.
  - The test item name input in the past from the combo box of the test item name can be selected.
  - The histories of the test item name can be maintained up to 20.
Profile to store Test Report

- It is displayed saving of the Test Report ahead.

Stop

- The conditions precedent set by the Test Report collection condition setting dialog is displayed.

Ignore the collected data (You get the only first data passing trace point.)

- If you check this checkbox, while getting Test Report, You get the only first data passing trace point.
- You can select "Save Test Report" (When passed through specified Test Point, last Report data is saved other file) or “Restart” (When passed through specified Test Point, Restart the Ignore the collected data) as option.
- If you want to know about Ignore the collected data, read [10.7. Ignore the collected data]

Variable value rewriting

- The Write Test Point that does the value of a variable writing is set.
- The writing Test Point is set by "Write Test Point select" dialog displayed to click "Change" button.
- Please see [24. Variable value writing function] for details of the value of a variable writing.

Multi-Core Target

- If you check this checkbox, and in case that you use Multi Core CPU, You get the Test Point compatible with Multi Core.
- You can display Performance Monitor based on got Report or set a Test Report filter subject to the core.
- If you want to know about Multi-Core functions, read [33. Multi-Core functions]
10.5.2. “Getting Test Report” dialog

The acquisition of the Test Report starts when [Execution] is selected by the execution dialog of the program, and the following dialogs are displayed.

- **Stop**
  - The acquisition of the Test Report is stopped.

- **Details are concealed**
  - Details of "Getting Test Report" dialog (item following "Execution start") are made non-display.

- **Variable value rewriting**
  - The Write Test Point set by the execution dialog of the program is displayed.

- **Execution start**
  - Time for the acquisition of the Test Report to have begun is displayed.
Elapsed time

- The elapsed time after the Test Report acquisition begins is displayed.

Size of collected log data

- The data size of the Test Report that will be acquired by present is displayed.

DynamicTracer buffer use rate

- The rate of the use of the buffer of a dynamic tracer is displayed.

Bookmark

- The comment input to the mark column of the Test Report that it fills in on the comment column, and [Set] is selected is displayed.

<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Test Report file acquiring it now and the file of the same name are not done while acquiring the Test Report and the operation when the superscription is copied doesn’t become outside the guarantee, and do not do such an operation, please.</td>
</tr>
<tr>
<td>When the forced ending etc. occur while acquiring the Test Report, and the acquisition of the Report Data doesn’t end normally, only the data normally stored in the Report Data file is displayed as valid data.</td>
</tr>
<tr>
<td>Please note that the Test Report acquired with DTSimpleTracer of DT10 of ver. 3.00 and bundled ver.1.20 cannot display it by the application of DT10 of the version before ver.250.</td>
</tr>
</tbody>
</table>
10.6. Bookmark Function

10.6.1. Set the Mark in getting the report.

During get the Test Report, you can set the Mark whenever you want to, and it is commented in Test Report data. If you set the Mark at start timing of events (power on/off, function start, etc...), you can use it as marker.

The Mark set by item “Bookmark” (in dialog when getting Test Report).

Enter up character in comment area and push the “Type X Set” button. It displayed in bottom window. The Mark which was set can select by combo box in “Comment”.

Mark is 3 types. You can select type of Mark by set button. You can set the type of Mark.

10.6.2. Test Report

Set the Mark while getting the Test Report, you can see it in “Mark Type” “Mark” item of Test Report.

"Mark Type" "Mark" displayed at Test Report can edit. Double-click on the item and you entering edit mode.

You can enter up comment to "Mark". You can select one "Mark Type" in 3 types.

You can add BookMarks after Getting Test Report.

(※In case the "Mark" have no characters ,you cannot edit the "Mark Type" )

You can use it as memo while checking the Test Report.
10.6.3. Mark View

"Mark View" is displayed if "Book mark" is set in Test Report.
"Mark View" display the list of Mark displayed in Test Report.

Double-click on the Mark's line and you jump to the Test Report.
You can find marked data in easy. "Mark View" can edit "Mark" and "Mark Type", and it is reflected Test Report.

<table>
<thead>
<tr>
<th>Mark View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark</td>
</tr>
<tr>
<td>START</td>
</tr>
<tr>
<td>END</td>
</tr>
</tbody>
</table>

■ Mark
  • The comment when the bookmark is set while acquiring the Test Report is displayed.
  • The edit of the mark becomes possible by double-clicking.

■ Mark Type
  • It is the mark type displayed on the Test Report.
  • The edit of the mark type becomes possible by double-clicking.

■ No.
  • Test Report No. is displayed.

■ Elapsed time
  • Time to have set the Mark is displayed.
Right-click on the Mark View to display this popup menu.

Delete: The Mark that has been selected is deleted.

Jump: It jumps to the corresponding section of the Test Report of the Mark that has been selected.

Function trace: The Function Trace Report is displayed within the range specified that the mark of two or more lines is selected by the Mark View and "Function Trace" is selected. When "Function Trace" is executed, "Analytical range specification" dialog is displayed in the situation only by one line, and the selection line should make Report No of the specified Mark a starting point, and input Report No. of the terminal.
10.7. Ignore the collected data Mode

10.7.1. Ignore the collected data

In case that “Ignore the collected data” is enable, while getting Test Report, You get the only first data passing trace point. If you want to get C0 Coverage, “Ignore the collected data” is effective because your data file size will be small.

When getting Test Report, Check the checkbox “Ignore the collected data (You get the only first data passing trace point.” in Execute Dialog of program. And you are entering the “Ignore the collected data” Mode.
10.7.2. Test Report Save Option

If "Ignore the collected data" is enabled, and when Test Point was passed, "Test Report Save Option" saves the Report data which the Report data of range of before passed Test Point as another Report data. (The number of Maximum data is 8192.)

<<Example>> If you want to know the route to pass the specific Test Point, when you destructing the duplicate Report data.

If you set the Test Report setting that destruct passed Test Point, and when you get data in this order [01] -> [05] -> ...[During processing Test Point ]...->[02]->[03], you cannot know route to [03] Test Point.

If you enable the "Test Report Save" option, you can know the route in detail at another Report.

And you can check the route to go to processing point of interest with smaller data size.

![Diagram of Test Point composition and Test Report tables](image)
Test Report Save Option is set by “execute program” dialog. (Click to [Test] -> [Execution] at the Menu)

Check the "Test Report Save" (it is enabled if the checkbox [Ignore the collected data] is checked) and click to [Select Test Point], and you can select Test Point with displayed dialog.

After setting the options, execute getting the Test Report.

If the Test Point (specified in dialog) is passed during getting Test Report, the following message is displayed.
If the Test Point (specified in dialog) is passed through, the previous Test Report is got when passing through the specified Test Point.

The previous Test Report passed through specified Test Point.

- Test start time/test end time is same to main Report.
- Report data name displayed item "Report Data" is [[Report Data name+ branch number] of Main Report.
- [specified TP#n passage] Test item name] is displayed at "Test item name" .
  "#n" is the same as a number of “No.” in the dialog that specified Test Point.

<table>
<thead>
<tr>
<th>Attention</th>
<th>Number of Report data can be stored in separate files when you go through the TP is 8192 in maximum. If it is saved to the Report given by specified TP passage, next Report data is saved other file because the save file is changed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention</td>
<td>The data saved other file is not having Mark setting data. Mark setting during getting data is reflected to Main Test Report only.</td>
</tr>
</tbody>
</table>
10.7.3. Restart Option

Restart Option make clear the passage status of Test Point, and start destruction the data of Report Data of same Test Point if "Ignore the collected data" is enable.

<<Example>> If you want to check the passed through route when the specific operation (passed through [01]TP etc...) occur.

When the purpose of getting Test Report is to check the specified operation, specify the Test Point which starting point of the operation and enable option. Then, if specified Test Point is passed through, passage information of Test Point is reset and Duplicate check of Report data is restarted. It make you can trace the execution route after passed through specified TP with Report data.
Execute program dialog [displayed by [Execution Report] -> [Execute]) can setting "Restart Option".

Check the [Restart] and click to [Select Test Point] button and you can see the dialog. You can select the Test Point at the dialog.

Setting options and execute get the Test Report.

During getting Test Report, if specified Test Point passage occurs, you can see following message.

When Test Point getting is complete, if specified Test Point passage occurs, the passage information is reset. And you can see Test Report restart to check the same Test Point.
10.8. Real-Time Coverage

10.8.1. Real-Time Coverage

“Real-Time Coverage” is function to check C0 level Coverage while getting Test Report.

(※If you want to know about Coverage Report, read [13. Coverage Report].)

Check the [Real-time Coverage] in "Set collecting condition of Test Report" dialog and you can see Real-Time Coverage.

Coverage Report is displayed, when start getting Test Report.

The item related C0 Coverage value is refreshed per 500ms in Coverage report.
You can use following action during show the Real-Time Coverage.

- Check the C0 level Coverage in Coverage Report.
  When you check the item "C0(Effective)", it is effective to setting coloring for background color at [C0 color settings]. To display [C0 color settings], right-click on the Coverage Report, and you can see the popup menu, and select the [Coverage data settings...]. If you want to know about this detail, read [13.2.3. Fundamental motion of Coverage Report].

- You can check the passage status of Test Point at Source Code Window.
  (In default setting, Non-pass Test Point is painted green, and Executed Test Point is painted Blue.)

- To double-click on the line of Coverage Report, you can check the Not Executed TP List each Functions.
  And if you select the Test Point from Not Executed TP List, you can jump to the corresponding line.
  If you want to know more about Not Executed TP List, read [13.2.4. Not Executed TP List].

**Memo** The difference [Real-Time Coverage] and [Analysis Function during getting Report]

While analyzing the Test Report, select the [Coverage Report] from [Report Analysis Menu] and the analyzing is done and displayed Coverage Report. This Function and Real-Time Coverage have following difference.

- Refresh timing of value of Coverage Report.
  - Real-Time Coverage : Refreshing automatically per 500ms
  - Analysis Function during getting Report : Analysis menu selected timing

- The jump to Source Code Window.
  - Real-Time Coverage : Jump to appropriate point from Not Executed list.
  - Analysis Function during getting Report : Jump to top of Function is double-clicked on Coverage Report.

### 10.8.2. The limitation of Real-Time Coverage

There are following limitation to use Real-Time Coverage.

- You cannot use following analysis menu while you getting Test Report with Real-Time Coverage is enabled.
  - Coverage Report
  - Execution Time Report
  - Period Time Report
  - Loop Count Report

- While getting Test Report, the color of Source Code Window is refreshed, until the Test Report getting is complete.
  So, when scroll the Source Code Window with Test Report not analyzed, passed through Test Point is painted to the color for Not Executed Test Point sometime.
11. Test Report

11.1. Test Report

When the acquisition of the Test Report begins, the following windows are displayed.

In the test by the dynamic qualification, information can be acquired during the program execution route, order, and that time.

Is it an execution route as a design? How is the execution time in that case going? It is possible to confirm easily.

(※ The item of the Test Report is customizable in the environmental setting. Please see 【26.1. Setting dialog】 in detail.)

![Test Report window]

- **No.**
  - It is an acquisition number of the Test Report.

- **Core**
  - When you get the report from target CPU as Multi Core CPU, Display the Mark of each Core data.
    - Core 0: Blue  
    - Core 1: Green  
    - Core 2: Yellow  
    - Core 3: Red

- **Source**
  - It is a source file name to which the Test Point that passed has been described.

- **Function**
  - It is a function name to which the Test Point that passed has been described.

- **Step**
  - It is a Step number in the function of the Test Point that passed.

- **Description**
  - The content of "Description" that can be input in the Test Property of each Step Property is displayed.
    - The Step where "Description" is not input becomes an empty column.
    - Doing Double click and enters edit mode.
**Description2 (Comment)**

- When doing Test Point insertion, the comment which auto-extracted from Source code is displayed.
- The comment is extracted according to the following rule. If you want to extract the comment at Description2, you need write comment according to the rule.
  - If you find entrance of function or "{" of "if", then, hold the found "comment block", save that "comment block" to "Description2", as the comment of the next block.
  - In case of "switch", save found "comment block" to "Description2" in same setting, even if the "switch" has no "{".
  - In the range from Function entrance or "{" to the "Test Point", if there are some comment, the first comment is held.
- Doing Double click and enters edit mode.

**Memory**

- When the Variable Value Point is passed, it is displayed.

**Value of a variable**

- When the Variable Value Point is passed, it is displayed.

**Elapsed time**

- A total elapsed time until the Test Point is passed is displayed from the acquisition beginning the Test Report.
- The unit of time can switch nano second [ns], and micro second [us], millisecond [ms], and second [s].

**Difference**

- The execution time from the Test Report that passed by one ahead is displayed.
- The unit of time can switch nano second [ns] and micro second [us], millisecond [ms], and second [s].

**Module**

- The module name that the function to which the Test Report is described belongs is displayed.

**Task name**

- The task name that the function to which the Test Report is described belongs is displayed.

**Event ID**

- When the Event Trigger Point passes, the event ID is displayed.

**Event definition**

- When the Event Trigger Point passes, the event definition is displayed.
■ L-Ch1-4
  - The logic input data of each channel is displayed.

■ A-Ch1-2
  - The AD input data of each channel is displayed.

■ Mark
  - The comment when the bookmark is set while acquiring the Test Report is displayed.

■ Mark Type
  - The Mark Type is displayed.

■ Interface
  - The module name and the function name set by "Interface" of the function property are displayed.
■ Common Interface
  • The task name, the module name, and the function name set by “Common Interface” of the function property are displayed.

■ Module start
  • The module name and the function name set by “Module start” of the step property are displayed.

■ Module end
  • The module name and the function name set by “Module end” of the step property are displayed.

■ Task start
  • The task name and the function name set by “Task start” of the step property are displayed.

■ Task end
  • The task name and the function name set by “Task end” of the step property are displayed.

Attention

In an Analog Box, the voltage of about 40mV or less is recognized as all 40 mV. Please note it when you measure the voltage of less than 40mV.
11.2. Change the unit of time in the Test Report

In the Test Report, the unit of the time of "Elapsed time" and "Difference" can be changed, and displayed it. The unit that can be selected is "Nano second[ns]", "Micro second[us]", "Millisecond[ms]", and "Second[s]."

The change method right-clicks in the test, and displays the pop-up menu. "Nano second[ns]", "Micro second[us]", "Millisecond[ms]", and "Second[s]" can be selected by clicking the [Change the unit of the time value] in the pop-up menu.

( the units of time value is changed in conjunction with other report's value, such as the report also execution time. )

<table>
<thead>
<tr>
<th>Memory</th>
<th>Value of...</th>
<th>Elapsed time(ns)</th>
<th>Difference(ms)</th>
<th>Module</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2,303</td>
<td>0</td>
<td>1_1_23_Proc</td>
<td></td>
</tr>
</tbody>
</table>

Change time unit
Change the radix of variable
Change the radix of A-Ch1
Nano second[ns]
Micro second[us]
Milli second[ms]
Second[s]

It changes from the millisecond[ms] at the Nano second[ns] every time.

<table>
<thead>
<tr>
<th>Memory</th>
<th>Value of...</th>
<th>Elapsed time(ns)</th>
<th>Difference(ns)</th>
<th>Module</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2,303,507,320</td>
<td>1,720</td>
<td>1_1_23_Proc</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,303,509,040</td>
<td>1,720</td>
<td>1_1_23_Proc</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,303,613,410</td>
<td>4,370</td>
<td>1_1_23_Proc</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,303,615,130</td>
<td>1,720</td>
<td>1_1_23_Proc</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,303,616,920</td>
<td>1,790</td>
<td>1_1_23_Proc</td>
<td></td>
</tr>
</tbody>
</table>
11.3. Use the jump function in the Test Report

In DT10, it is possible to jump to the specified list item by specifying the number of "No." of the Test Report. The function is called the jump function of the Test Report.

The use of the jump function right-clicks in the Test, and displays the pop-up menu. When [Jump to the specified Report number] of the pop-up menu is clicked, the jump dialog to the specified Report number is displayed.

It jumps to the list item of the Report number specified that the Report number that jumps is specified by the dialog, and [OK] is clicked.

**Memo** Jump with the filter setting done.

Because the filter setting hangs when the line number of the Report Data is set to "Sequential number for the all data" with the filter set, and it jumps, the specified number is not occasionally found.

In that case, the message is displayed, and it jumps to the Report Data near a specified number.

Please set the line number of the Report Data to [Sequential number for the displayed data] when you want always to jump to the specified number.

The line number of the Report Data is revocable in "Setting" dialog displayed to select [Tool]→[Setting] of the menu.
11.4. Trace the place of Test Point from the Test Report

In DT10, the place of the Test Point of the selected list can be traced by double-clicking the list or selecting and clicking 'Enter' with the Test Report on the source code window.

The list is double-clicked or it selects and the "Enter" key is clicked.
11.5. Retrieve in the Test Report from Folder View or Module View

In DT10, the Test Point can be selected from the Folder View and the Module View, and the list item of the Test Point in the Test Report be retrieved.

This search is search for first list item which the conditions are. Reference position is Report data which selected by Test Report.

First of all, the search strategy selects the list item of the Test Report that becomes a retrieval beginning position. And, Test Point (STEP) of the Folder View Module View to be retrieved is selected, it right-clicks, and the pop-up menu is displayed. Click [Find previous by Test Report] or [Find next by Test Report], and you can see Test Report's list item you selected.
11.6. Copy the Test Report data to the clipboard

The data of the acquired Test Report can be used by copying to the clipboard, and putting it on Microsoft Excel etc.

First of all, the data to be copied is selected as the method. At this time plurals, to select the line, two or more lines can be selected by clicking while pushing the “Shift” key at the selection end position after the selection beginning position is clicked.

And, the pop-up menu is displayed by right-clicking, and when [Copy] of the pop-up menu is selected, it is copied onto the clipboard.

At this time, the item name can be copied together by selecting [Copy with header].

The copied data is put on spreadsheet software such as Microsoft Excel, straightened the appearance, assumed material, and can be used.
11.7. The Test Report view is seen easily and set

11.7.1. Filter setting

The character string of selected items and the value can be non-displayed on the Test Report View...

The one that it is not necessary to confirm it with the Test Report etc. are made non-display, and only necessary data can be confirmed easily.

The character string or the value of the item non-displayed is selected on the Test Report View, and the pop-up menu is displayed by right-clicking.

"[Character string or Value] is deleted from [Item name] and display it again." of the pop-up menu is selected.

The item to which the filter can be set is a source, a function, a Step, a module, a task name, L-ch1~4, and a mark.

(* When the character string or the value is not displayed in the item, the filter cannot be set.)

Please release the filter by "Detailed setting of the Test Report display" dialog.

The button of "Report analysis" value is not displayed in the Step are made non-display, and display "Detailed setting of the Test Report display" dialog.
11.7.2. Classification setting

The character string and the value of selected items can be classifying set on the Test Report View.

The data that I want to see in the main confirming running on of the overall real machine with the Test Report in classifying the data to be confirmed by the main can be firmly confirmed.

The pop-up menu displays when the character string or the value of the item that does the classification setting is selected on the Test Report View, and right-clicks.

“[The item name] of [The character string or the value] is colored” of the pop-up menu is selected.

The item to which the classification can be set is No, Core, Source, Function, Step, Memory, Variable value, Module, Task name, Event ID, Event definition, L-ch1~4, A-ch1~2, mark, mark type, and Interface.

(※ When the character string or the value is not displayed in the item, the filter cannot be set)

![Test Report View]

When the classification setting is released, classified character string and value are selected, the pop-up menu is displayed by right-clicking, and “[The item name] of [The character string or the value ] of is classifying released.” of the pop-up menu is selected.
11.8. The Test Report is retrieved

11.8.1. Open the Search settings

It is possible to retrieve in the Test Report. It right-clicks, the pop-up menu are displayed, and when [Search settings...] is selected, the following dialogs are displayed in the Test Report.

- **Retrieves by Source file name.**
  - It retrieves by the specified source file name.
  - The source file can be specified from a right combo box.

- **Retrieves by Function name**
  - It retrieves by the specified function name.
  - The function can be specified from a right combo box.

- **Retrieves by Module name**
  - It retrieves by the specified module name.
  - The module can be specified from a right combo box.
■Retrieves by Step
  • It retrieves by the specified Step.
  • The Step can be specified by [Selection of the Test Point] dialog displayed to display to select a right "Setting of the Step" button.

■Retrieves by the Mark character string
  • It retrieves by the specified Mark character string.
  • The Mark can be specified from a right combo box.

■Retrieves by the Mark Type
  • It retrieves by the specified Mark Type.
  • The Mark can be specified from a right combo box.

■Retrieval strategy
  • The search strategy can be selected “From a local point to the down”, “From a local point to the up”, “From the head to the down” and “From the end to the up”.
  • “Turn” can be selected in the option.
  • It returns to the head and the Test Report is retrieved when retrieving to the last minute when retrieving to the down when the check is put in "Turn".
    It returns at the end and it retrieves when retrieving to the head of the Test Report when retrieving to the up.

■Retrieval start
  • The retrieval of the Test Report start.
  • It is possible to retrieve in the direction with the "F3" key, and to retrieve in the opposite direction with the "Shift+F3" key.
  • After retrieves, "Retrieval of the Test Report" dialog can be opened with the "Ctrl+S" key.
  • After searching, and to enter [Ctrl+F3] key, you can search in conjunction with Folder view and Source code window.
  • Enter [Shift+Ctrl+F3] key, and you can backward search in conjunction with Folder view and Source code window.
11.8.2. Search for current Report Data

Selected Report data on Test Report can automatically entered in search dialog and searches the Report.

Select the line of Report data you want to search.

Right-click on the line to open popup-menu and select “Find this item” and you can see Search Dialog.

At this time, “Source” “Function” “Step” “Module” “Mark characters” and “Mark Type” is entered in Search Dialog at automatically. You can execute search in easy if you select the item you want search.

This command is enable only in the normal Step.

You cannot use this command if you use following Report data.

- Event Trigger
- Report Data of each byte of the variable value output data (Include the Var Step, Arg Step, and Dump Line)
- Data for Write Test Point
- CPU Load Measurement Point
- Analog Box Notify
11.9. Test Report Trimming function

The Test Report can be trimmed. It becomes possible to take out only a necessary part of the Test Report by using the Trimming function, and to use as another Test Report.

First of all to use the Trimming function, it right-clicks in the Test Report, and the pop-up menu is displayed. The [Trimming] is selected from the menu.

When the [Trimming] is selected, the following dialogs are displayed. When the preservation place and the range of the Trimming are selected by this dialog, and the [Execution] button is selected, the Test Report is trimmed.

**Preservation place**

- The place where the trimmed Report Data is preserved is specified.
- To save the location you saved trimming to Profile, perform following operation.
- Create New Profile / Show Report Data of another profile / Close Test Report of past Profile / Save / Save as / Test
Range of Trimming

- The range to trimming the Test Report is specified.
- When "Intended for all displayed data" is selected, only the data being displayed in the Test Report now is trimmed. When the filter setting is done, only the data displayed in the report data is trimmed.
- When "The start end point is specified" is selected, the start point and the end point are specified, and it trims within the range. Start and the end point can be specified by "Line number", "Time", and "Mark".

Option

- When the Report Data is trimmed, the option can be put.
  When the check box of "Project information is copied" is turned on, project information (*.pprj) is copied together. When a past Test Report is opened, the *.pprj file is a necessary file.
- If the check box of "Mark information is copied" is turned on, it becomes possible to trim information on "Mark" item of the Test Report together.
- After trimming, the profile is restructured when the check box of "After ends, the profile is restructured" is turned ON.
  Please do to the profile of the correspondence later in "Restructure Report Data List" by manual when you turn OFF this check box. The restructuring of the Report Data selects "Report Analysis “→ "Display Test Report" of menu, and clicks "Restructure Report Data List “ button of the displayed dialog.
- The test item name can be named by turning on the check box of "The test item name is named".

Execution

- The Trimming of the report data is executed.

Cancel

- Processing is discontinued.

11.10. The Report line is selected by order sending

The following line (previous line when the Shift key is pushed) that has been selected now can be selected by inputting F10 of the function key when the Test Report is displayed.
Moreover, the line that corresponds in the source code window at this time is displayed.
The execution route of the program can be traced by repeating the F10 input.
12. Analyze Test Report

12.1. Analyze Test Report

When the Test Report is acquired, the Report Data is analyzed in DT10, and much information can be used plainly.

When [Report analysis] -> [Analyze Test Report] of the menu is clicked, Test Report is analyzed.

- The judgments such as “Period”, “Event”, “Execution time”, and “Execution frequency” set for the Test Report to be analyzed in various properties are done, and the color of the Test Point that passed in the document view etc. changes from green into blue, and the passed place is displayed plainly. (Please see [9. Set the Property] about “Period”, “Event”, “Execution time”, and “Execution frequency”.)

- “Coverage Report” and “Execution Time Report” and “Period Time Report” can be displayed by using the acquired Report Data in DT10. The Report is analyzed when the analysis of the Report Data is clicked. [Analyze Test Report] is clicked, and “Coverage Report” and “Execution Time Report” and “Period Time Report” are displayed by the automatic operation. If the Test Report is analyzed and the displayed various Reports are used, the data acquired more validly can be used.

- [Analyze Test Report] is executed by the automatic operation in that case when “Coverage Report” and “Execution Time Report” and “Period Time Report” are displayed before [Analyze Test Report] is executed. Even if the command of “Coverage Report”, “Execution Time Report”, and “Period Time Report” is executed afterwards, the result of the analysis last time is displayed if “Analysis of the Report Data” is done once. Please [Report analysis]—[Analyze Test Report] of the menu when the re-analysis of the Report Data is necessary in the reason when the value of the property is changed.

- It is also possible to analyze the Report while acquiring the Test Report, and to make them display the Coverage Report, the Execution Time Report, the Period Time Report, and the Loop Count Report. The coverage rate etc. can be confirmed while testing.
12.2. Multiple Analysis of Test Reports

Two or more selected Report Data can be adding up analyzed when two or more Test Reports are in the same profile, and output to the Coverage Report, the Execution Time Report, and the Period Time Report.

When the Report Data has been switched by the test pattern by the Addition analysis of the Report Data, the Coverage rate in the entire test and Period time, etc. can be confirmed.

The analysis of the Test Report selects the [Report analysis] → [Multiple Analysis of Test Reports].

The following dialogs are displayed. The Report added up by this dialog is selected.

When the check box of the Report added up is turned on, and the [OK] button is selected, the addition analysis is done.

When the addition analysis ends, the addition Coverage Report, the addition Execution Time Report, the addition Period Time Report, and the addition Loop Count Report are displayed.

The addition Report can be saved by CSV. Please see [29. Save the Report in CSV by Export Function] in detail.

The background of folder view/module view is displayed in the gray, and it becomes a change prohibition while displaying the addition Report.
Moreover, the following commands become the use prohibitions.


The addition Report shuts the Report Data or opens another Report Data, and when the DT Project is switched, is shut.
13. Coverage Report

13.1. The type of the coverage which can checked by DT10

DT10 can check C0/C1 Coverage.

- **C0 Coverage**
  - It is the statement coverage. Coverage of instruction
  - It viewed amount of executed statement which specified for testing in program.
  - It will be 100% when all statement is executed at least once.

- **C1 Coverage**
  - It is the branch coverage. Coverage of the branch.
  - It viewed amount of executed branch in program.
  - It will be 100% when all branches are executed at least once.
  - Intensity of the C1 Coverage is higher than statement coverage.

13.2. C0 Coverage Report

13.2.1. Calculation method of C0 coverage

DT10 have two C0 coverages. [C0] coverage and [C0 (Effective)] coverage.

[C0] coverage is calculating coverage based on amount of executed Test Point inserted in the function.

[C0 (Effective)] coverage is the C0 coverage rate calculated by the Test Point enabled on the folder view/module view.

This coverage is calculated by using "Unused", "Unexecute" and "Non-pass" set in the Step property, and the data of the Test Point of "Normal/Abnormal" of the processing attribute based on the following calculating formula. You can see detail description about Step property at 9.6. Step Property.

- **C0**

  \[
  \frac{(\text{Number of normal execute} + \text{Number of illegal execute})}{(\text{TP setting number} - \text{Unused} - \text{Unexecute} - \text{Non pass})} \times 100\% 
  \]

- **C0(Effective)**

  \[
  \frac{(\text{Number of normal execute} + \text{Number of illegal execute} \times 2)}{(\text{Number of effective TP} \times 1)} \times 100\% 
  \]

  ※1 Valid TP is a Test Point that doesn't apply to each condition of Unused, Unexecute, Non-pass, and the Invalid Test Point.

  ※2 The number of execute and the number of Illegal execute are usually the numbers of molecules of the valid numbers of TP of denominators.
<<Sample ① >> How to calculate of C0 coverage. (Basic)
Calculate C0 coverage depending on the amount of executed Test Point which inserted in function.

```c
void main(void)
{
    DtTestPoint //Pass Initial();
    do{
        DtTestPoint //Pass
        if(Timer >= 2)
        {
            DtTestPoint //Pass
            UserMain();
            SystemTimer = 0;
        }
    }while(1);
    DtTestPoint //NotPass
}
```

In the function that inserted four Test Points, The three Test Point is executed.

\[ C0 = \frac{3}{4} \times 100\% = 75\% \]

<<Sample ② >> How to calculate of C0 coverage. (If there is a test point outside the scope of coverage by the execution attributes)
In calculating, If you set [Execution attribute] to [Unused], [Unexecute] or [Non-pass] in Step Property, the Test Point that inserted in unexecuted area is excluded from the target of coverage calculate.

```c
void main(void)
{
    DtTestPoint //Pass Initial();
    do{
        DtTestPoint //Pass
        if(Timer >= 2)
        {
            DtTestPoint //Pass
            UserMain();
            SystemTimer = 0;
        }
    }while(1);
    DtTestPoint //NotPass
}
```

The Test Point which it's execute attribute set to [Non-Pass] is excluded from calculate for coverage

\[ C0 = \frac{3}{3} \times 100\% = 100\% \]
How to calculate of C0 coverage. (If there is a test point which its checkbox of the Folder view and Module view is set to OFF.)

If you turn OFF the checkbox of Folder view/Module view, you can check the coverage rate which calculated with the Test Point which its checkbox of [C0 (Effective)] item is set ON.

In the [C0 (Effective)], Test Point which is match with following cases is excluded from calculation target.

- Execution attribute is set to [Unused], [Unexecute] or [Non-pass].
- Unchecked on Folder view/Module view checkbox.

In C0 (enabled), the Test Point check box is set OFF is excluded from the calculation.

\[
\text{C0 valid} = \frac{3}{3} \times 100\% = 100\%
\]

\[
\text{C0} = \frac{3}{4} \times 100\% = 75\%
\]
13.2.2. Display C0 Coverage Report

When [Report analysis] -> [Coverage Report] of the menu is selected, the following windows are displayed.

The coverage of each function is displayed.

In the test by the dynamic qualification, the covering rate at the statement level of the program can be measured.

In the coverage report, items related to C0 coverage is “TP setting number” to “C0(Effective)”. 

<table>
<thead>
<tr>
<th>TP setting number</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **TP setting number**
  - It is a Test Point number in the function.

- **Unused**
  - It is a Test Point number set in Property Editor (Step) as unused. When coverage is calculated, it uses.

- **Unexecuted**
  - It is a Test Point number set in Property Editor (Step) as the street prohibition. When coverage is calculated, it uses.

- **Non-pass**
  - It is a number of Test Points set in the property editor (Step) as non-street. When coverage is calculated, uses it.

- **Ineffective TP**
  - It is a number of Test Points to which the check box of the Test Point is invalid in folder view/module view.

- **Effective TP**
  - It is a number of Test Points to which unselected Unused/Unexecute/Non-pass in Step Property and the check box of the Test Point is valid in folder view/module view.
● Execution (Normal)
  - It is a Test Point number that actually passed with the Test Point set in Property Editor (Step) as the normality system.

● Execution (Illegal)
  - It is a Test Point number that actually passed with the Test Point set in Property Editor (Step) as the abnormality system.

● C0
  - It is a covering rate at the statement level of the function.
  - This coverage is calculated by using "Unused", "Unexecute" and "Non-pass" set in the Step property, and the data of the Test Point of "Normal/Abnormal" of the processing attribute based on the following calculating formula.

\[
\frac{\text{Number of normal execute} + \text{Number of illegal execute}}{\text{TP setting number} - \text{Unused} - \text{Unexecute} - \text{Non pass}} \times 100\% 
\]

● C0 (Effective)
  - It is a coverage rate intended for only the Test Point with a valid check box.
  - It is calculated based on the following formulas.

\[
\frac{\text{Number of normal execute} + \text{Number of illegal execute} \times 2}{\text{Number of effective TP} \times 1} \times 100\% 
\]

※1 Valid TP is a Test Point that doesn't apply to each condition of Unused, Unexecute, Non-pass, and the Invalid Test Point.

※2 The number of execute and the number of illegal execute are usually the numbers of molecules of the valid numbers of TP of denominators.

Memo About Executed attribute Test Point
About the DT Project created by DTDiff.
If there are Test Point get Step Pass information from other project and be added "pre-passed attribute".
"Executed attribute" is DTDiff's Internal attribute
DT10 cannot add/modify to "Executed attribute"
You delete the Test Point or you execute auto insert new Test Point, and the attribute is removed.
If you want to know "Executed attribute", Please refer to "DTDiffmanual.pdf"
13.2.3. Fundamental motion of Coverage Report

■ When the item name is clicked, it is displayed in the order of the item.

■ The pop-up menu is displayed when right-clicking on coverage.

・ Configuration Coverage Data
  • Display configuration dialog of coverage data.

  ❖ View filter setting
    The filter is set according to the coverage rate.
    Please select the item that sets the filter from "C0" "C0 (Effective)" "C1" "C1 (Effective)", specify the numerical value of the coverage rate, and click "OK".
    The filter can be put without setting the upper bound or the lower bound though a range specification possible region is from 0% to 100%.

  ❖ Configuration C0 Coverage Color
    It is color setting of background of "C0(Effective)" item by each coverage rate.
    Three color settings by C0 value are enabled.
    Enter up number to item which you enabled its checkbox, and select color by "Color select" button.
    Top checkbox have priority. So you enter up biggest number to top checkbox.
    If you checked "The function passed through the Test Points other than normal attribute" item, and if Execution attribute passed the "Unused"/"Non-pass"/"Unexecute" Test Point, the Test Point function's color will change.
• **Copy**  : The line that the Coverage Report has selected is copied onto the clipboard.
  If you want select two or more lines, two or more lines can be selected by clicking while pushing "Shift" key at the selection end position after the selection beginning position is clicked.

• **Copy with header**  : The line that the Coverage Report has selected is copied with the header.

• **Export**  :
  - "The Coverage Report is preserved by CSV" dialog is displayed.
  - The Coverage Report can be exported from this dialog.

**Batch off passed Test Points**

  : The Test Point that passed can be invalidated.
  It is convenient that it finishes measuring coverage, and the acquisition of the Test Point of the assed part uses this function when there is no necessity.
  Moreover, it can be selected whether to update the header file by the automatic operation at the same time in the following dialogs in that case.
  (※When the Test Point is invalidated, the state of the filter is not considered.)

![Dialog](image.png)

**Yes**  :
The Test Point that passed is invalidated. The header file is updated by the automatic operation at the same time.

**No**  :
The Test Point that passed is invalidated. At this time, the automatic renewal of the header file is not done.
  It will be necessary to update it later by manual.

**Cancel**  :
The invalidity of the Test Point that passed is not done.
13.2.4. Not Executed TP List

Click on C0 Coverage Report line you like, and you can see “Not Executed TP List”.

“Not Executed TP List” is the list of non-pass Test Point of each function.

※Invalid Test Point is not include in list.

If "Not Executed TP List" is displayed, while displaying Real-Time Coverage, the list is refreshed automatically.

It is jumped to the applicable line on Source Cord Window when line of “Not Executed TP List” is double-clicked.

<table>
<thead>
<tr>
<th>Not Executes TP List: PwmSwitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line number</td>
</tr>
<tr>
<td>146</td>
</tr>
<tr>
<td>148</td>
</tr>
<tr>
<td>151</td>
</tr>
</tbody>
</table>

```
void PwmSwitch(void)
{
    _DtTestPoint(_Dt, _PwmSwitch, _DtStep_0)
    if(PwmTimer >= PwmLine && FlagPwn ==0)      //時間判定
        _DtTestPoint(_DtFunc_PwmSwitch, _DtStep_1)
        if(_TILR == 0xff)
            _DtTestPoint(_DtFunc_PwmSwitch, _DtStep_2)
        _TILR = 0x70;  //ブザーが停止していれば起動
    else
```

Attention

"Not Executed TP List" is the function to check Statement Level (C0) Coverage.
While analyzing C1 Coverage, Not Executed TP List is not displayed.
13.3. C1 Coverage Report

13.3.1. How to calculate C1 coverage

C1 coverage is display the percentage of executed branch("if" "switch" and so on) in test.

For each branch, the path to the branch destination is called "Branch route".
"C1" coverage can be calculated by the number of branches that the route for the number of branches was calculated route.
[C1 (Effective)] is the C1 coverage targeting the branch route of the Test Point which its checkbox is turned ON by Folder view/Module view.
The branch route of the Test Point which its Execution attribute of the Step Property is set to [Unused], [Unexecute], and [Non-pass] is excluded from the calculate target. You can see detail description about Step Property at [9.6. Step Property].

■ C1

\[
\text{Executed branch route} \div \text{number of branch route} \times 100[\%]
\]

■ C1(Effective)

\[
\text{Executed branch route} \div (\text{number of branch route} - \text{invalid branch route} \times 1) \times 100[\%]
\]

※1 Invalid branch route is the route which it's Test Point's checkbox is turn OFF in the Folder view/Module view.

In DT10, the branch route is calculated with Test Point which inserted in the branched code.

<table>
<thead>
<tr>
<th>The rule of the calculate of branch route</th>
</tr>
</thead>
<tbody>
<tr>
<td>◆ For each branch, the Test Point inserted in branch destination source code is used for calculate the number of the branch destination. The total value of the branch of each function is used for calculate the C1 coverage.</td>
</tr>
<tr>
<td>◆ Test Point related branch is the only data which is used for calculate the branch route. The Test Point of [Free attribute], [Variable value output point], [CPU Load Measurement point], [Event Trigger point], [Write Test Point], and so on is excluded for branch route calculation target. So, user can insert those Test Point in manual with no effect on C1 coverage.</td>
</tr>
<tr>
<td>◆ When calculating the branch route of &quot;if&quot;, and when the &quot;else&quot; is unwritten in that branch, that unwritten &quot;else&quot; is considered for calculation of the branch route DT10 assumed the unwritten &quot;else&quot;, and calculate the branch route with a virtual step named [Virtual] which is generated by DT10 in internally.</td>
</tr>
<tr>
<td>◆ When calculate &quot;switch&quot; route, the calculation is assumed that the all branch of &quot;switch&quot; is written in source code.</td>
</tr>
</tbody>
</table>
If "default" is omitted, that "default" is not included in route calculation. So, if you want to include "default" to calculation, you should not omit it.

◆ In the loop branch route ("for", "while", "do" and so on), the routes that pass through the actual is used in calculation.
Virtual branch route in internally is not generated.

<<Sample ① >> Example for calculate of branch route.

```c
void FuncA( int arg_a, int arg_b, int arg_c )
{
    DtTestPoint //FuncIn
    if( arg_a == TRUE )
    {
        DtTestPoint //if Step
        if( arg_b == TRUE )
        {
            DtTestPoint //if Step
            cnt_a = 0;
        }
        //Virtual Step
    }
    else
    {
        DtTestPoint //ifStep
        cnt_a = 1;
    }
    switch( arg_c ){
    case0:
        DtTestPoint //switch Step
        cnt_b = 20;
        break;
    case1:
        DtTestPoint //switch Step
        for( int i = 0; i < 4; i++ ){
            DtTestPoint //forStep
            cnt_C += ;
        }
        break;
    default:
        DtTestPoint //switch Step
        cnt_C = 100;
    }
    DtTestPointValue //Variable Value Point
    DtTestPoint //FuncOut
}
```

The number of branch route of function "FuncA" = 8

① There are two route (true/false) for "if( arg_a == TRUE)" branch.

② There are two route (true/false) for "if( arg_b == TRUE)" branch.
If "else" is not written, the route is calculated assume that virtual step exist.

③ There are three route (case0 / case1 / default) for "switch( arg_c )" branch.
If "default" is omitted, "default" is not included route calculation. User need to write it.

④ There is one route for "for( int i = 0; i < 4; i++)" branch.
In case of calculate for loop branch, the route which will execute is used for the route calculate.

⑤ The Test points inserted manually (e.x. Variable Value Point) is removed from the route calculation
**Sample ②**  
How to calculate the C1 Coverage (Compared with the C0 coverage rate)

※In case of that all Execution attributes is set to "Normal" and no invalid Test Point is set

```c
void FuncA(int arg_a, int arg_b)
{
    DtTestPoint //FuncIn //Pass
    if (arg_a==TRUE)
    {
        DtTestPoint //if Step //Pass
        if (arg_b==TRUE)
        {
            DtTestPoint //if Step //Pass
            cnt_a = 0;
        }
        //Virtual Step //Not pass
    }
    else
    {
        DtTestPoint //if Step //Not Pass
        cnt_a = 1;
    }
    DtTestPoint //FuncOut //Pass
}
```

### C1 Coverage Rate

C1 Coverage is calculated by the number of executed branch route in the function.

In the function of sample ②, there are two branch route of the "if(arg_a==TRUE)" and two branch route of the "if(arg_b==TRUE)". There are four routes of total.

\[
C1 = \frac{2}{4} \times 100\% = 50\%
\]

### C0 Coverage Rate

C1 Coverage is calculated by the number of executed Test Point which is inserted.

\[
C0 = \frac{4}{5} \times 100\% = 80\%
\]
13.3.2. Display C1 Coverage Report

The check is first put in “Analyze C1 Coverage in Analysis of Test Report” by “Setting” dialog (Display it by “Tool” -> “Setting” of the menu) to analyze the C1 coverage, and “OK” is clicked.

![Setting dialog with check for Analyze C1 Coverage in Analysis of Test Report]

The Report is analyzed by “Analyze Test Report” or “Multiple Test Reports”.

When the check is put in “Analyze C1 Coverage in Analysis of Test Report” by Setting dialog, the following dialogs are displayed, and the source file that analyzes the C1 coverage can be selected.

![Select files to analyze C1 Coverage dialog]

When the source file is selected, and the [OK] button is clicked, the Report analysis is done.

Moreover, it is also possible not to analyze the C1 coverage with [Skip C1 Coverage] button, and to analyze the Report.
- Number of branch
  - It is a number of branch in the function.

- Number of branch route
  - It is a number of branch routes in the function.
  - The FuncIn Step inside function is branch route start position.
  - Passing the parent hierarchy / not passing the parent hierarchy are considered as the branch route.

- Invalid branch route
  - It is a number of branch routes in the function to which the check box of the Test Point is invalid in folder view/module view.

- Execution branch route
  - It is a number of branch routes that actually execute.

- C1
  - It is a covering rate at the branch level of the function.
  - This coverage is calculated by using the data of the Test Point of "Normal/Abnormal" of "Unused" set in the Step property, "Unexecute", "Non-pass", and the processing attribute based on the following calculating formula.

\[
\frac{\text{Execution branch route}}{\text{Number of branch route}} \times 100\% 
\]

- C1 (Effective)
  - It is a coverage rate intended for only the Test Point with a valid check box.

\[
\frac{\text{Execution branch route}}{\text{Number of branch route} - \text{Invalid branch route}} \times 100\%
\]
<table>
<thead>
<tr>
<th>The rule for C1 Coverage Report display.</th>
</tr>
</thead>
<tbody>
<tr>
<td>◆ As for the function judged it is illegal because the branch structure is analyzed, “×” is displayed in the item concerning the C1 coverage.</td>
</tr>
<tr>
<td>◆ Example of judgment that branch structure is illegal &gt;&gt;</td>
</tr>
<tr>
<td>・FuncIn doesn’t exist.</td>
</tr>
<tr>
<td>・“Unused” attribute is not set to an unused Test Point in the Step Property.</td>
</tr>
<tr>
<td>◆ It becomes blank for the function not to analyze the Report.</td>
</tr>
<tr>
<td>◆ “-” is displayed in the item concerning the C1 coverage when there is no branch in the function.</td>
</tr>
<tr>
<td>◆ When the transition of the Step goto and in the function of the recursive processing etc. is special, the coverage is outside the guarantee</td>
</tr>
<tr>
<td>◆ Coverage might not be able to be displayed normally when FuncOut has not been correctly inserted.</td>
</tr>
</tbody>
</table>

The menu displayed when you right-click on the Coverage Report is same with C0 coverage report.
You can see detail description about this in [13.2.3. Fundamental motion of Coverage Report].
13.3.3. C1 Coverage Detailed Report

When the line with the data of the C1 coverage is double-clicked in the Coverage Report, C1 Coverage Detailed Report of the function is displayed.

![C1 Coverage Detailed Report](image)

**List part**

- The item of the list part is as follows.

  **Index**: It is a serial number of C1 Coverage Detailed Report. It absolutely displays it by the number.

  **Report number**: It is a number of the corresponding Test Report view. It is possible to fly to the Report number of the correspondence when an arbitrary line of the list part is double-clicked.

  **Status**: It is displayed for the structure of cannot the normal performance it as "ERROR". It is displayed for the route from which the Write Test Point is executed as "Write". When happening with "Write" both of "ERROR", it is displayed as "ERROR ; Write".

![Structural tree part](image)
The pop-up menu is displayed when right-clicking on the list part.

**The Function Trace Report is displayed (automatic picking).**

The Function Trace Report from the pertinent Report number to the FuncOut is displayed. You can use this if only one data selected.

![](image)

### Structural tree part

- The branch structure is displayed in figure.
- When each Step is clicked, a pertinent Step is displayed in the source code window.
- The following meanings are in the Step type.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>The Step shown in the quadrangle in the frame border of the orange is a</td>
</tr>
<tr>
<td></td>
<td>Have [Return]</td>
<td>The Step shown in the oval in a blue frame border is a Step with the Step</td>
</tr>
<tr>
<td></td>
<td>Virtual</td>
<td>The Step shown in a dotted line quadrangle is a virtual Step disembodied</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Background color</th>
<th>Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>A blue Step is a Step that started the divergence. (The condition is the truth.)</td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>A red Step is a Step that did not start the divergence. (The condition is false.)</td>
<td></td>
</tr>
<tr>
<td>T / F</td>
<td>A yellow Step is a Step that doesn't start the divergence and has entered.</td>
<td></td>
</tr>
<tr>
<td>No Check</td>
<td>A white Step is a Step over which it did not pass.</td>
<td></td>
</tr>
<tr>
<td>Invalid</td>
<td>The check box of the Test Point of folder view/module view is an invalid Step.</td>
<td></td>
</tr>
</tbody>
</table>
• The color of the line connecting the step has following mean.

<table>
<thead>
<tr>
<th>Color of Line</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The step connected with green line is the step of same nest.</td>
</tr>
<tr>
<td></td>
<td>The step connected with red line is the step of deeper nest.</td>
</tr>
<tr>
<td></td>
<td>The step connected with blue line is the step of next branch.</td>
</tr>
</tbody>
</table>
14. Execution Time Report

14.1. Execution Time Report

When [Report Analysis] → [Execution Time Report] of the menu is selected, the following windows are displayed.

In the Execution Time Report, the execution time of each function is displayed.

(※The execution time of each function enters another function in the function, and is time including time to have dispatched the task)

It can be easily confirmed whether the maximum/minimum execution time is displayed, and there is unexpected value different from the design.

<table>
<thead>
<tr>
<th>Function</th>
<th>Source</th>
<th>Module</th>
<th>Total time(ns)</th>
<th>Min time(μs)</th>
<th>Max time(μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSelectUnitPro...</td>
<td>Process_TekTask.c</td>
<td>1_1_6_Proc...</td>
<td>1,492,028,270</td>
<td>75,110</td>
<td>414,140</td>
</tr>
<tr>
<td>ProcessSubPb</td>
<td>ProcessSubPb.c</td>
<td>1_1_10_Proc...</td>
<td>204,032,650</td>
<td>10,250</td>
<td>327,320</td>
</tr>
<tr>
<td>UnitProcessELECT</td>
<td>ProcessSubPb.c</td>
<td>1_1_10_Proc...</td>
<td>165,268,320</td>
<td>103,950</td>
<td>314,960</td>
</tr>
<tr>
<td>UnitProcessSTOP</td>
<td>ProcessSubPb.c</td>
<td>1_1_10_Proc...</td>
<td>147,920</td>
<td>2,730</td>
<td>3,740</td>
</tr>
<tr>
<td>UnitCheckMedia...</td>
<td>ProcessCommon.c</td>
<td>1_1_7_Proc...</td>
<td>46,720</td>
<td>5,600</td>
<td>6,160</td>
</tr>
<tr>
<td>ProcessMsgGet...</td>
<td>ProcessCommon.c</td>
<td>1_1_9_Proc...</td>
<td>318,320</td>
<td>25,420</td>
<td>48,410</td>
</tr>
<tr>
<td>UnitStatusFileInf</td>
<td>ProcessCommon.c</td>
<td>1_1_7_Proc...</td>
<td>9,560</td>
<td>1,580</td>
<td>1,610</td>
</tr>
<tr>
<td>UnitStatusDisIn...</td>
<td>ProcessCommon.c</td>
<td>1_1_7_Proc...</td>
<td>11,700</td>
<td>1,960</td>
<td>1,960</td>
</tr>
</tbody>
</table>

- **Function**: It is a function name.

- **Source**: It is a source file name to which the function has been described.

- **Module**: The module name that the function belongs is displayed.

- **Total time**: It is an accumulation execution time of the function.
  
  The harmony of a total time of all functions doesn’t become it because it is accumulation time of each function equal to the Test Report acquisition time.

- **Min time**: It is a minimum execution time of the function.
Minimum time [Setting – Measured]

- The difference time of the value and the actual measurement value set by "Min" of "Execution time" item of the function property editor is displayed.
- It becomes a difference with the mean value when there are two or more set values of "Execution time".

Max time

- It is the maximum execution time of the function.
- There is a possibility that abnormal operation outside some assumption has occurred when growing more than the designed value.

Maximum time [Setting – Measured]

- The difference time of the value and the actual measurement value set by "Max" of "Execution time" item of the function property editor is displayed.
- It becomes a difference with the mean value when there are two or more set values of "Execution time".

Average time

- It is an average execution time of the function.

Typ head

- The room degree of the execution time calculated from the value of Typ (regulated value) set by "Execution time" of the Function Property is displayed.
- It is calculated based on the following calculating formula.

  
  "Typ head" at Execution Time / Average time × 100[\%]

Max head

- The room degree of the execution time calculated from the value of Max set by "Execution time" of the Function Property is displayed.
- It is calculated based on the following calculating formula.

  
  "Max head" at Execution Time / Average time × 100[\%]

Pass count

- If the Report Data is analyzed by valid item for the function and the Step only, the frequency that passed for each Test Point is displayed.
- If it is function, the function agrees to the frequency that passed over the Step of the FuncIn attribute.
14.2. Fundamental motion of Execution Time Report

■ When the item name is clicked, it is displayed in the order of the item.

■ The pop-up menu is displayed when right-clicking on the Execution Time Report.

![Popup menu for Execution Time Report]

• Change unit of the time
  : Time unit of "Total time", "the minimum", "the maximum" and "Between the mean time" is changed and can be displayed.
  The unit that can be selected is "Nano second, ns", "Micro second, us", "Millisecond, ms", and "Second, s".
  The unit of time value is changed in conjunction with other report.

• Display Execution time graph
  : User can display Execution time graph.
  You can see detail description about this in [14.5, Execution Time graph].

• Display Execution time histogram
  : User can display Execution time histogram.
  You can see detail description about this in [14.7, Execution time histogram].

• Copy
  : The line that the Execution Time Report has selected is copied onto the clipboard.
  If you want to select two or more lines, two or more lines can be selected by clicking while pushing "Shift" key at the selection end position after clicking the beginning position of selection.

• Copy with header
  : The line that has been selected is copied with the header

• Export
  : "The Execution Time Report is preserved by Comma Separated Value" dialog is displayed.
  The Execution Time Report can be exported from this dialog.
- Input the measurement value as setting value

: "Batch Input to the Property from execution time list" dialog is displayed.

The actual measurement value of the function specified by this dialog is reflected in a set value of the Property.
14.3. Execution Time Detail List

Execution Time Details List of the function is displayed as well as the Period Time Report when double-clicking by an arbitrary line of the Execution Time Report.

The Execution Time Details List is a list of all execution times of the arbitrary function displayed in the Execution Time Report.

![Execution Time Details List: NoSelectUnitProcess]

**Report No.**
- It is a line number from the head of the Report Data (The filter is not applied).

**Elapsed time**
- It is passed time for the FuncOut Step.
  
  It is same value with "Elapsed time" in "FuncOut" line of Test Report.

**Execution time**
- It is an elapsed time (period) from FuncIn to FuncOut.

**Status**
- It is displayed for the route from which the Write Test Point is executed as "Write".

Please see [24. Variable value writing function] for details of Write Test Point.
14.4. Fundamental motion of Execution Time Details List

■ In an Execution Time Details List, when the item name of “Report No” or “Execution time” is clicked, it is displayed in the order of the item. (※ It might take time for sorting when there is a lot of volume of data of the list. Please note it.)

■ A pertinent line and the source in the Report Data are displayed when double-clicking an arbitrary line of the Details List.
At this time, the Report Data List displays the line number column up to the FuncOut Step specified from FuncIn of the function that has been selected in green, and expresses the Test Point executed at the Execution time.

■ The pop-up menu is displayed when right-clicking on the Execution Time Details List.

- Change unit of the time
  : The time unit of “Execution time”, “Execution time” changed and can be displayed.
  The unit that can be selected is “Nano second [ns]”, “Micro second [us]” “Millisecond [ms]”,and “Second [s]”.
  The unit of time value is changed in conjunction with other report.
- **Copy** : The line that the Execution Time Details List has selected is copied onto the clipboard.
   If you want to select two or more lines, two or more lines can be selected by clicking while pushing "Shift" key at the selection end position after clicking the beginning position of selection.

- **Copy with header** : The line that has been selected is copied with the header.

- **MIN** : Select minimum value Execution time.

- **MAX** : Select maximum value Execution time.

- **Function Trace Report**
   : Display Report data which selected line with Function Trace.
   It is "No." area of Test Report displayed with green when you double-click on Detail List.
   You can use this if only one data selected.
14.5. Execution Time graph

The time information you can check on the Execution Time Report can display as bar-graph for each functions. Right-click on Execution Time Report and select [Display Execution Time graph], and you can see Display Execution Time graph.

Select the data you want to display from list box of top.

You can select Total time, Average time, Minimum time, Maximum time, or Occupancy.
14.6. Basic movement of Execution time graph

- Maximum/Minimum value of Number axis (vertical axis) can change.

Click the button on Execution time Graph, and you can see following display.

In default, [Auto] is checked.

If you want to change the Maximum/Minimum value, check OFF the [Auto], and enter the value. Then click [OK] button and you can see graph with entered value.

- The functions names displayed on horizontal axis are lined up in the same order as Execution time Report.
  This functions names line is synchronized with Execution time Report's sort.

- Double-click the vertical axis of the functions of the Execution time Graph, and you can see that functions histogram.
  You can see detail description about Histogram in [14.7. Execution time histogram].
Right-Click on the execution time graph, and you can see pop-up menu.

- **Change unit of time**: You can display graph with change the time unit of the time axis of the graph. You can select next unit.
  - Nano Seconds [ns]
  - Micro Seconds [us]
  - Milli Seconds [ms]
  - Seconds [s]
  Changing the units of time value is synchronized with other reports.

- **Display the histogram**: Display the histogram of function which is focused with cursor.

- **Copy**: Copy the Execution time Graph to clipboard
14.7. Execution time histogram

You can check the execution time's distribution of Execution time Report to using the Histogram.
Select the function on Execution time Report, and right-click it. Then select [Display the Execution time Histogram], and you can see Execution time Histogram.

Following Execution time Histogram is displayed.
The vertical axis is the frequency, the horizontal axis is the data range.
The max value and min value is displayed at the bottom of the graph.
14.8. Basic movement of Execution time Histogram

- You can change Y axis, max / min value of data range, and resolution.

Click the button on Execution time Histogram, and you can see following dialog.

In default, [Auto] is checked.

If you want to change the Maximum/Minimum value, check OFF the [Auto], and enter the value. Then click [OK] button and you can see graph with entered value.

Resolution of the data range can select [10 split] or [20 split].

- Double-click the any vertical axis on Execution time Histogram, and you can see Execution Time Details List.
Right-click on Execution time Histogram, and you can see pop-up menu.

- Change unit of time
  - You can display graph with change the time unit of the time axis of the graph.
  - You can select next unit.
  - “Nano Seconds [ns]” “Micro Seconds[us]” “Milli Seconds [ms]” “Seconds [s]” Changing the units of time value is synchronized with other reports.

- Copy
  - Copy the Execution time Histogram to clipboard.
15. Period Time Report

15.1. Period Time Report

In the Period Time Report, the execution interval of all functions is totaled, and displays between the minimum, the maximum and the mean time by the list form in each functions. (※However, execution interval of function which not include FuncIn Step is not include.)

The validity of the operation of the function from which regular execution is expected in using the Period Time Report can be easily verified.

The following windows are displayed with [Report Analysis] → [Period Time Report] of the menu.

- **Function**
  - It is a function name.

- **Source**
  - It is a source code name where the function is included.

- **Module**
  - It is a module name where the function is included.

- **Min period**
  - It displays it among all execution intervals every time that the minimum has been selected now.
  - 0 is displayed when never executed.

- **Min period [Setting-Measured]**
  - The difference time of the value and the actual measurement value set by "Min" of "Period" item of the function property editor is displayed.

- **Max period**
  - It displays among all execution intervals every time that the maximum has been selected now.
  - 0 is displayed when never executed.
■ Max period [Setting – Measured]
  - The difference time of the value and the actual measurement value set by "Max" of "Period" item of the function property editor is displayed.

■ Average period
  - It displays every time that it has been selected between the mean time when the total at all execution intervals were divided by execution frequency -1 now.
  - 0 is displayed when never executed.

■ Typ head
  - The room degree of the period time calculated from the value of Typ (regulated value) set by "Period" of the Function Property is displayed.
  - It is calculated based on the following calculating formula.

  "Typ head" at Period Time / Average time × 100[%]

■ Pass count
  - It is an execution frequency of the function.
15.2. Fundamental motion of Period Time Report

■ When the item name is clicked, it is displayed in the order of the item.

■ The pop-up menu is displayed when right-clicking at Period Time.

| Min period(us) | Min Period(us) | Max Period(us) | Max Period(us) | Average period(us) | Typ ...
|----------------|---------------|---------------|---------------|-------------------|---------
| 70             |               | 6,277         |               | 827,706           | 0.000%  |

- Change unit of the time
  - "Total time", "the minimum", "the maximum" and "Between the mean time" is changed and can be displayed. The unit that can be selected is "Nano second, ns", "Micro second, us", "Millisecond, ms", and "Second, s".
  - The unit of time value is changed in conjunction with other report.

- Display the Period Time graph: User can display the Period Time graph.
  - You can see detail description about this in [15.5 Period time graph].

- Display the Period Time Histogram: User can display the Period Time histogram.
  - You can see detail description about this in [15.7 Period time histogram].

- Copy: The line that the Period Time Report has selected is copied onto the clipboard.
  - If you want to select two or more lines, two or more lines can be selected by clicking while pushing "Shift" key at the selection end position after clicking the beginning position of selection.

- Copy with header: The line that has been selected is copied with the header.

- Export: "The Execution Time Report is preserved by CSV" dialog is displayed.
  - The Execution Time Report can be exported from this dialog.
- Input the measurement value as setting value

  "Batch Input to the Property from period time list" dialog is displayed.

  The actual measurement value of the function specified by this dialog is reflected in a set value of the Property.
15.3. Period Time Detail List

When a mouse left button is double-clicked by an arbitrary line of the Period Time Report, Period Time Details List of the function is displayed.

A Period Time Details List is a list at all Period Time of the arbitrary function displayed in the Period Time Report.

<table>
<thead>
<tr>
<th>Report No</th>
<th>Elapsed time</th>
<th>Period (us)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>2,953</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>3,032</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>3,111</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>225</td>
<td>3,190</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>290</td>
<td>3,269</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>335</td>
<td>3,349</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>420</td>
<td>3,428</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>485</td>
<td>3,507</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>550</td>
<td>3,586</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>615</td>
<td>3,666</td>
<td>79</td>
<td></td>
</tr>
</tbody>
</table>

Report No.

- It is a line number from the head of the Report Data (The filter is not applied).

Elapsed time

- Elapsed time is passed time on the FuncIn Step.
  - It is same value with "Execution time" in "FuncIn" line of Test Report.

Period

- It is an elapsed time (period) from last FuncIn.

Status

- It is displayed for the route from which the Write Test Point is executed as "Write".
  - Please see [24. Variable value writing function] for details of Write Test Point.
15.4. Fundamental motion of detail list of Period Time

■ In a Period Time Details List, when the item name of “Report No” or “Period” is clicked, it is displayed in the order of the item. (※ It might take time for sorting when there is a lot of volume of data of the list. Please note it.)

■ A pertinent line and the source in the Report Data are displayed when double-clicking an arbitrary line of the Details List.

At this time, the Report Data list displays the line number column of the line within the range of the selected period time in green, and expresses the Test Point executed at period time.

![Image of Test Report]

■ The pop-up menu is displayed when right-clicking on a detailed list of Period Time.

• Change unit of the time

  : Time unit of “Execution time”, “Period” are changed and can be displayed.

  The unit that can be selected is “Nano second, ns”, “Micro second,us”, “Millisecond,ms”, and “Second, s”.

  The unit of time value is changed in conjunction with other report.

• Copy

  : The line that Period Time Details List has selected is copied onto the clipboard.

  If you want to select two or more lines, two or more lines can be selected by clicking while pushing “Shift” key at the selection end position after clicking the beginning position of selection.
• **Copy with header**: The line that has been selected is copied with the header.

• **MIN**: The line of the minimum period on detailed list is selected.

• **MAX**: The line of the maximum period on detailed list is selected.

• **Function Trace Report**

  Display report data which selected line with Function Trace.

  It is "No," area of Test Report displayed with green when you double-click on Detail List.

  You can use this if only one data selected.
15.5. Period time graph

The time information you can check on the Period Time Report can display as bar-graph for each functions.

Right-click on Period Time Report and select [Display Period Time graph], and you can see Display Period Time graph.

Select the data you want to display from list box of top.

You can select Total time, Average time, Minimum time, Maximum time, or Occupancy.
15.6. Basic movement of Period time graph

- Maximum/Minimum value of Number axis (vertical axis) can change.

Click the button on Period time Graph, and you can see following display.

In default, [Auto] is checked.

If you want to change the Maximum/Minimum value, check OFF the [Auto], and enter the value. Then click [OK] button and you can see graph with entered value.

- The functions names displayed on horizontal axis are lined up in the same order as Period time Report.

This functions names line is synchronized with Period time Report's sort.

- Double-click the vertical axis of the functions of the Period time Graph, and you can see that functions histogram.

You can see detail description about Histogram in [15.7. Period time histogram]
Right-click on the Period time graph, and you can see a pop-up menu.

- **Change unit of time**: You can display the graph with changed time units of the time axis of the graph. You can select the next unit.
  - “Nano Seconds [ns]” “Micro Seconds [us]” “Milli Seconds [ms]” “Seconds [s]”
  - Changing the units of time value is synchronized with other reports.

- **Display the histogram**: Display the histogram of the function which is focused with the cursor.

- **Copy**: Copy the Period time Graph to clipboard
15.7. Period time histogram

You can check the period time's distribution of Period time Report to using the Histogram.

Select the function on Period time Report, and right-click it. Then select [Display the Period time Histogram], and you can see Period time Histogram.

Following Period time Histogram is displayed.

The vertical axis is the frequency, the horizontal axis is the data range.

The max value and min value is displayed at the bottom of the graph.
15.8. Basic movement of Period time histogram

- You can change Y axis, max / min value of data range, and resolution.

Click the button on Period time histogram, and you can see following dialog.

![Histogram Setting dialog](image)

In default, [Auto] is checked.

If you want to change the Maximum/Minimum value, check OFF the [Auto], and enter the value. Then click [OK] button and you can see graph with entered value.

Resolution of the data range can select [10 split] or [20 split].

- Double-click the any vertical axis on Period time histogram, and you can see Period Time Details List.
Right-click on Period time Histogram, and you can see pop-up menu.

- **Change unit of time**: You can display graph with change the time unit of the time axis of the graph.
  
  You can select next unit.

  “Nano Seconds[ns]” “Micro Seconds[us]” “Milli Seconds[ms]” “Seconds [s]”

  Changing the units of time value is synchronized with other reports.

- **Copy**: Copy the Period time Histogram to clipboard.
16. Loop Count Report

16.1. Setting of the Loop Count (Step Property)

“Loop Count” is the function which counts the executed count the Test Point has been run to reach the specify Test Point. If user set the count of loop by Step Property of after loop, user can check whether the number of the loop (for example, loop count of "for") is correct.

Analyze the Test Report with set the loop count by Step property, DT10 can check whether the looped count is within the set value. You can see the basic result of loop count check of DT10 (OK or NG) at Property view. And you can see detailed result of that at Loop Count Report.

<<Image>>

To reach the "free" step, the "for" step should be executed three times.

Check the result of the judgment in the Property view.

Test Report

<table>
<thead>
<tr>
<th>No.</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>138526</td>
<td>for</td>
</tr>
<tr>
<td>138527</td>
<td>for</td>
</tr>
<tr>
<td>138526</td>
<td>for</td>
</tr>
<tr>
<td>138527</td>
<td>free</td>
</tr>
</tbody>
</table>
You can set “Loop Count” by Step Property.

"Loop Count" is the function which count the executed count the Test Point has been run to reach the specify Test Point. After exit the loop, at the timing of execute the specify Test Point, DT10 execute judgement whether the loop is executed in success.

So, The Test Point which is executed after loop is becomes the standards of judgment.

User should set "Loop Count" at this Test Point's step property.

In above example, "Loop Count" is set in the Step Property of Test Point after exit the loop.

You can open the Step Property with select the [Open Property Editor...] in the right-click menu or select [Plan]→[Property Editor...] in the Menu.

Click the [add] button at [Loop Count] the item of Step Property, and you can see [Selection of the Test Point] dialog. You can specify the Test Point to check the loop count in the dialog.

Specified Test Point's line number is displayed to “Loop Count”, and enters the designed value of the loop count, then click [OK] button.
16.2. Loop Count Report

Loop Count Report is display the list of analysis result of the step which loops count is set in Step Property.

※To use Loop Count Report, user need to set the loop count in the Step Property.

When [Report Analysis] → [Loop Count Report] of the menu is selected, the following windows are displayed.

In the Loop Count Report, when the item name is clicked, it is displayed in the order of the item.

There are three types of Loop count Report.

The item from [Source] to [Loop count] is the item which is set loop count in the Step Property and it will be base of decision.

The item from [Source (Ref)] to [Module (Ref)] is the item which is related to the Test Point (Ref) specified in loop count.

The item from [Number of Max passes] to [Number of Min passes] is the result of the count of execution of Setting step to reach the Standard step.

■ Source
  - It is a source file name that the Step in a specified origin of the Loop Count belongs.

■ Function
  - It is a function name that the Step in a specified origin of the Loop Count belongs.

■ Step
  - It is a Step in a specified origin of the Loop Count.

■ Module
  - It is a module name that the Step in a specified origin of the Loop Count belongs.

■ Loop count
  - It is a Loop Count of the Step in a specified origin of the Loop Count.

■ Source (Ref)
  - It is a source file name that the Step in the specification destination of the Loop Count belongs.
■ Fuction (Ref)
  • It is a function name that the Step in the specification destination of the Loop Count belongs.

■ Ref
  • It is a Step in the specification destination of the Loop Count.

■ Module (Ref)
  • It is a module name that the Step in the specification destination of the Loop Count belongs.

■ Number of Max passes
  • It is the maximum value of the count in which the specified Test Point is executed by the attainment from the specified Test Point to this Test Point.

■ Number of Min passes
  • It is minimum value of the count in which the specified Test Point is executed by the attainment from the specified Test Point to this Test Point.

■ Average pass count
  • It is a mean value of the count in which the specified Test Point is executed by the attainment from the specified Test Point to this Test Point.

■ Total number of passing
  • It is a number of total passing.
16.3. Loop Count Detail List

When a mouse left button is double-clicked by an arbitrary line of the Loop Count Report, the Loop Count Details List of the Step is displayed.

The Loop Count Details List is all loop count of arbitrary Test Point displayed in the Loop Count Report list.

■ Report No.
  • It is a line number from the head of the Report Data (The filter is not applied).

■ Execution Count
  • It is a frequency that the specified step passed.

<table>
<thead>
<tr>
<th>Report No.</th>
<th>Loop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>3</td>
</tr>
<tr>
<td>157</td>
<td>3</td>
</tr>
<tr>
<td>222</td>
<td>3</td>
</tr>
<tr>
<td>287</td>
<td>3</td>
</tr>
<tr>
<td>352</td>
<td>3</td>
</tr>
<tr>
<td>417</td>
<td>3</td>
</tr>
<tr>
<td>482</td>
<td>3</td>
</tr>
<tr>
<td>547</td>
<td>3</td>
</tr>
<tr>
<td>612</td>
<td>3</td>
</tr>
<tr>
<td>677</td>
<td>3</td>
</tr>
<tr>
<td>742</td>
<td>3</td>
</tr>
</tbody>
</table>
16.4. Fundamental motion of Loop Count Detail List

In a Loop Count Details List, when the item name of "Report No" or "Count" is clicked, it is displayed in the order of the item. （※ It might take time for sorting when there is a lot of volume of data of the list. Please note it.）

A pertinent line and the source in the Report Data are displayed when double-clicking it by an arbitrary line of a detailed list. At this time, the report data list displays the column of the line number to which the selected Test Point corresponds in green.

The pop-up menu is displayed when right-clicking on the Loop Count Details List.

- **Copy**: The line that the Loop Count Details List has selected is copied onto the clipboard. To select two or more lines, two or more lines can be selected by clicking while pushing the "Shift" key at the selection end position after the selection beginning position is clicked.

- **Copy with header**: The line that has been selected is copied with the header.

- **MIN**: The minimum count line listing in detail is selected.

- **MAX**: The maximum count line listing in detail is selected.
**Function Trace Report**

- Display report data which selected line with Function Trace.
- It is "No," area of Test Report displayed with green when you double-click on Detail List.
- You can use this if only one data selected.
17. Function Trace Report

17.1. Specification of analysis range

When [Report analysis] -> [Function Trace Report] of the menu is selected, the following windows are displayed.

"Range of Trace" and "Resolution" can be specified by the Function Trace View display range specification dialog.

When the analysis range is specified by the dialog, and [OK] is clicked, the Function Trace Report is displayed.

In the range specification, there are following rules and limitations.

- Line number of Trace Report View specified “Range of Trace” is always an absolute number irrespective of the environment.

- The data number in the analysis range cannot be specified exceeding 2,097,152.
  To take time to analyze to exceed 1,048,576, the message is displayed.

- The analysis range cannot be specified exceeding 24 days (=2073600 seconds).
  When a long range is specified for 2073600 seconds, it traces from the starting position to 2073600 seconds by the resolution of 1ms.

- The display resolution of the Function Trace View changes automatically depending on analytical time when analyzing.

<table>
<thead>
<tr>
<th>Analytical time</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>10ns or more ~ 20sec or less</td>
<td>10ns (Minimum unit to be able to acquire DT10)</td>
</tr>
<tr>
<td>Time to exceed 20sec</td>
<td>1us</td>
</tr>
<tr>
<td>~ 30min(=1800sec) or less</td>
<td></td>
</tr>
<tr>
<td>Time to exceed 30min</td>
<td>1ms</td>
</tr>
<tr>
<td>~ 24days(=2073600sec) or less</td>
<td></td>
</tr>
</tbody>
</table>
When the unit of resolution is large, it is likely to be drawn to “□” that shows the Test Point in with a narrow Report Data acquisition interval coming in succession.

(Ex) For the Test Point of execution time 12,340ns and 12,450ns,

It is judged same 12.0us when analyzed by the unit of 1us, and is drawn to □ coming in succession.

When the TP acquisition interval is narrow, the Test Point within the range increasing and operating becomes heavy. When the Report Data with a narrow TP acquisition interval is displayed in the Function Trace, it is recommended to narrow down the limits and to trace.

The next usual Step is automatically set to the beginning Point, except when the beginning Point is a usual Step of the Variable Test Point etc.

**Attention**

When the maximum number is analyzed, the free space of about 100MB is needed. The performance falls, and note specs beforehand, please when an empty memory becomes insufficient because of the analysis. Please execute measures such as narrowing the analysis range that increase an empty memory capacity of PC. Moreover, please note that it similarly takes time to not only the analysis but also the memory liberating when it ends beforehand.
17.2. Function Trace Report

Function Trace Report is displayed the execution route of the program and the execution time.

The execution route of the program and the execution time are displayed graphically.

The dispatch part and the interrupt generation part of the task can be easily confirmed by setting the task name etc. in the Property Editor.

Moreover, the measurement of the time of a specified section can be easily done.

The nest and Y axis of the processing of the function show time in X axis.

However, because the Function Trace is the one that it advances to a nest (right side) deep by one, and movement of advancing to a nest (left side) shallow by one was shown in figure every time FuncOut is done every time FuncIn is done, the exact analysis should be not able to be done with no correct burial of the Test Point of FuncIn/FuncOut, and the trace Report when the Test Point is not buried under the dispatched task, please.
17.3. Fundamental motion of Function Trace Report

- When the mouse moves while pushing right-clicking, it is possible to scroll in all directions at the speed corresponding to the distance from the center position of the drawing window though the movement with the sucrose bar is also possible in the Function Trace Report.

- When the mouse pointer is moved on the Function Trace, drawing of red line trace is done. This line displays 'Difference time' from 'Elapsed time' and the line that has been selected at any time.

- The clicked part is set as "Line that has been selected" when left-clicking on the Function Trace View. It becomes a line in starting points of the operation of the zoom and the movement, etc., saying that "Line that has been selected". In the line that has been selected, it is displayed that the position of the beginning value of 'Elapsed time' and the line the reference and assumption want and is displayed belonging 'Task name'.
  (※The task name displayed Function Trace Report is reflected "Processed task name" item in Module Property. If you want to know task name configuration, Please refer to 【9.4. Module Property】.)

- When the mouse pointer is put on the rectangle of each Step on the Function Trace View, information on the Step is displayed by a green text box. Step information is displayed when there is "Selected line" on the Step by a red text box.

- If click the □ represented each Step on the Function Trace View, or move the Step by button on the toolbar, Source code window is displayed conjunction.

- If click the □ represented each Step on the Function Trace View, it is possible to see both the Test Report in conjunction with the Source Code Window.
  
  Ctrl + F7 : Go to previous Step with conjunction the Source Code Window and Test Report.
  Ctrl + F8 : Go to next Step with conjunction the Source Code Window and Test Report.

- A blue line on the Function Trace View shows the boundary of the task. The Step below a blue line belongs to the task displayed on a right edge.

- If the task changes, the nest becomes 0 for the task from which the following task is traced for the first time. The task has already been traced, and it becomes a nest number that the task saved when changing from the task into another task, and returning there.

- The content of the toolbar of the Function Trace is as follows.

  📐 : Zoom mode switch. It time axis zooms based on the click position.
: Zoom in of time axis

: Zoom out of time axis

: The first Step is retrieved and selected.

: The last Step is retrieved and selected.

: The previous Step is retrieved and selected from present location.

"F7" Even if the key is pushed, it is similar.

: The next Step is retrieved and selected from present location.

"F8" Even if the key is pushed, it is similar.

: The previous task is retrieved and selected from present location.

: The following task is retrieved and selected from present location.

: Zoom in of nest of function

: Zoom out of nest of function

: It jumps to the position of the line that has been selected.

: Information on the line that has been selected is clear.

: Information on all Steps is always displayed.

When it has neither enough CPU power nor empty physical memory so that the load of processing of drawing on the screen may increase if it switches to 'Always drawing mode' when a Step a large amount of by the low magnification is displayed on the screen, it is necessary to note it.

: The display resolution of the view is changed to 10ns.

: The display resolution of the view is changed to 1us.

: The display resolution of the view is changed to 1ms.
<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
</table>
| The unit change in the display resolution on the trace view can change only the unit more than the unit of the display resolution when analyzing.  
Ex) When analyzed by the unit of 1us, the unit of 1us $\rightarrow$ 1ms can be changed.  
   It is not revocable for the unit of 10ns. |
17.4. Display Trace Report from the Test Report

In DT10, the Test Report can display the Function Trace Report. There are two kinds of methods of the Test Report's displaying the Function Trace Report.

One is a method of displaying the Function Trace Report specifying the range on the Test Report View. The procedure selects two or more items from the list, and displays the pop-up menu by right-clicking. The trace Report can display it within the range specified that it is clicked to [Display the Function Trace Report (Range specification)] from the pop-up menu.

Another one is a method of selecting one item that doesn't belong from the list to FuncOut on the Test Report view, detecting FuncOut of the item with DT10 automatically, and displaying the Function Trace Report. One item of the Step to which it doesn't belong from the list to FuncOut is selected on the Test Report view, it right-clicks, and the pop-up menu is displayed. The Function Trace Report from the item specified that it is clicked to [Display Function Trace Report (Automatic picking)] from the pop-up menu to FuncOut of the function can be displayed.
17.5. The transition of the task is display by using Event Trigger

The transition of the task can be displayed by using the Event Trigger by the Function Trace.
Please see [20. Event Trace Function] about Event Trigger Point.

To use this task analysis mode, the check box of “The Event Trigger is used for the task judgment when the Function Trace is analyzed” is turned ON by an environmental set dialog ([Tool] → [Setting] of the menu).
(※When the check box of “When the Test Report is opened, the Event Trigger is analyzed” is only turning ON, the item of “The Event Trigger is used for the task judgment when the Function Trace is analyzed” becomes effective.)

And, the Event Trigger is analyzed at this time when the Test Report is opened.
It is possible to display it by using the Event Trigger for the task analysis when the Function Trace Report is opened when the Test Report of the Event Data analysis is opened.
17.6. Function Trace display that considers filter setting

Only the Test Report Data to set the filter and to display can be displayed in the Function Trace.

Please see [18. Set the display filter to the Test Report] about filter setting.

The execution route only of a pertinent task omitting another task of interrupting can be confirmed by setting the filter by the task name and displaying the Function Trace Report.

To do the Function Trace display that considers the filter setting, the check box of "When the Function Trace is analyzed, the filter setting is considered" is turned on by setting dialog ([Tool] → [setting] of the menu).

And, when the Function Trace Report is displayed when the filter is set and the Test Report is displayed, the Function Trace Report that considers the filter setting is displayed.

<table>
<thead>
<tr>
<th>Attention</th>
<th>As for the Report to which the filter is set, because all the filter settings are considered, the analytical result of the Function Trace is not warrantable. It is necessary to set the filter that is appropriate for the Function Trace by the user's judgment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention</td>
<td>The Function Trace Report is detected and when the range is detected with the Test Report automatically, the Test Point of FuncOut of correspondence can be detected regardless of display/non-display. However, please note no display in the Function Trace Report when FuncOut is non-display.</td>
</tr>
</tbody>
</table>

※ Image of a setting dialog.
18. Set the display filter to the Test Report

Test Report is set filter from "Detail setting of Test Report display" that is displayed by selecting [Report analysis] -> [Set the display filter of Test Report] of the pop-up menu. Even button of toolbar is similar.

Or, filter condition is added by clicking [Add to the Report display filter condition] of pop-up menu that is displayed by right-clicking when source file, function, Step, and module that add to filter condition select from Folder View or Module View.

Moreover, the filter condition of setting it from "Filter" list in the past can be selected from the list. The settings can be registered up to five.

It is possible to put by preserving the condition in five patterns or less if this function is used, and it becomes easy to set the filter.

Moreover, when "no filter" is selected, it is also possible to return to the state without the filter easily.

The check box of "Valid" of the type to be assumed to be a filter setting condition is turned on to set the filter and the "Setting value" column is double-clicked. Because "Test Report display setting" dialog of each item is displayed, the setting of the filter condition is specified by the dialog. Moreover, it is possible to display the condition of setting it by selecting "Show / Hide" by "Processing" item or to non-display or to select it.

① To enable the filter type
② Double-click "Set value" item.
③ Double-click "Hide / Show" switch.
When the filter setting is changed, the change is reflected in the Test Report.

[OK] button : The change is reflected, and the dialog is shut.

[Cancel] button : The setting is canceled, and the dialog is shut.

[Apply] button : The change is reflected with the dialog opened.
Source

- The filter is set subject to the source file name.
- The check is put in the check box, "Test Report display setting: the source item" dialog is displayed to double-click "Set value" column, and select the item that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.
- The addition of the source is possible from the folder view. Please select "Add to the Report display filter Setting" from the menu displayed to select a pertinent source on the folder view, and to right-click.
- The deletion of the source is possible from the folder view. Please select "Deletion from the Report display filter setting" from the menu displayed to select a pertinent source on the folder view, and to right-click.

Function

- The filter is set subject to the function name.
- The check is put in the check box, "Test Report display setting: the function item" dialog is displayed to double-click "Set value" column, and select the item that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.
- When "[ ](check box item name)" is clicked, it is possible to sort it by the item to which the check has adhered.
- After sort by function name, a character is inputted and move to the position of the function of the character.
- The addition of the function is possible from the folder view. Please select "Add to the Report display filter Setting" from the menu displayed to select a pertinent source on the folder view, and to right-click.
- The deletion of the function is possible from the folder view. Please select "Deletion from the Report display filter setting" from the menu displayed to select a pertinent source on the folder view, and to right-click.
**Step Type**

- The filter is set subject to the Step type.
- The check is put in the check box, “Test Report display setting: Step Type item” dialog is displayed to double-click “Set value” column, and select the item that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching “Show/Hide” of “Processing” item.

**Elapsed time**

- The filter is set subject to the elapsed time.
- The check is put in the check box, “Test Report display setting: elapsed time” dialog is displayed to double-click “Set value” column, and select the item that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching “Show/Hide” of “Processing” item.
Module

- The filter is set subject to the module name.
- The check is put in the check box, “Test Report display setting: Module item” dialog is displayed to double-click “Set value” column, and select the item that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching “Show/Hide” of “Processing” item.
- When the check on “The check is applied to the child item” is turned on, and the parents module that wants to set the filter is selected, the child module is selected together.
- The addition of the module is possible from the folder view. Please select “Add to the Report display filter Setting” from the menu displayed to select a pertinent source on the folder view, and to right-click.
- The deletion of the module is possible from the folder view. Please select “Deletion from the Report display filter setting” from the menu displayed to select a pertinent source on the folder view, and to right-click.

![Module screenshot]

Task

- The filter is set subject to the task name.
- The check is put in the check box, “Test Report display setting: Task item” dialog is displayed to double-click “Set value” column, and select the item that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching “Show/Hide” of “Processing” item.

![Task screenshot]
Mark

- The filter is set subject to the Mark specification character string. Please see [10.6. Bookmark Function] about how to set mark.
- The check is put in the check box, "Test Report display setting: Mark item" dialog is displayed to double-click "Set value" column, and select the item that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.

Mark Type

- The filter is set subject to the Mark Type specification character string. Please see [10.6. Bookmark Function] about how to set mark type.
- The check is put in the check box, "Test Report display setting: Mark Type item" dialog is displayed to double-click "Set value" column, and select the item that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.
Step

- The filter is set subject to a specific Test Point.
- The check is put in the check box, “Test Report display setting: Step item” dialog is displayed to double-click “Set value” column, and select the item that sets the filter from the dialog, please.
- Selection of Test Point” dialog is displayed when the check is put in the check box, and the [Add] button is clicked, and Select the Test Point that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching “Show/Hide” of “Processing” item.
- The addition of the Step is possible from the folder view. Please select “Add to the Report display filter Setting” from the menu displayed to select a pertinent source on the folder view, and to right-click.
- The deletion of the Step is possible from the folder view. Please select “Deletion from the Report display filter setting” from the menu displayed to select a pertinent source on the folder view, and to right-click.

Input data (logic and AD)

- The filter is set subject to the input data acquired from an Analog Box.
- The check is put in the check box, “Test Report display setting: input data (logic and AD)” dialog is displayed to double-click “Set value” column, and select the item that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching “Show/Hide” of “Processing” item.
Interface

- The filter is set subject to a specific “Interface”. (“Interface” is configuration item of function called by specific Module/Function. Please refer to [9.5. Function Property] to know how to configure it.)
- The check is put in the check box, “Test Report display setting: Interface item” dialog is displayed to double-click “Set value” column, and select the item that sets the filter from the dialog, please.
- Click the “Add” button and you can see “Interface setting” dialog. If you select the called Function/Module in that dialog, you can see the Report data which Function called.
- Only the item of the condition is displayed by switching “Show/Hide” of “Processing” item.

Common Interface

- The filter is set subject to a specific “Common Interface”. (“Common Interface” is configuration item of function called by specific Module/Function. Please refer to [9.5. Function Property] to know how to configure it.)
- The check is put in the check box, “Test Report display setting: Common Interface item” dialog is displayed to double-click “Set value” column, and select the item that sets the filter from the dialog, please.
- Click the “Add” button and you can see “Common Interface setting” dialog. If you select the called Function/Module/Task in that dialog, you can see the Report data which Function called.
- Only the item of the condition is displayed by switching “Show/Hide” of “Processing” item.
Module start

- The filter is set subject to a specific "Module start". ("Module start" is configuration item for Step which before transitioned. Please refer to [9.6. Step Property] to know how to configure it.)
- The check is put in the check box, "Test Report display setting: Module start item" dialog is displayed to double-click "Set value" column, and select the item that sets the filter from the dialog, please.
- Click the "Add" button and you can see "Module start" dialog. If you select the transitioned Function/Module at that dialog, you can see the Step Report data which before transitioned.
- Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.

![Module start dialog](image)

Module end

- The filter is set subject to a specific "Module end". ("Module end" is configuration item for Step which after transitioned. Please refer to [9.6. Step Property] to know how to configure it.)
- The check is put in the check box, "Test Report display setting: Module end item" dialog is displayed to double-click "Set value" column, and select the item that sets the filter from the dialog, please.
- Click the "Add" button and you can see "Module end setting" dialog. If you select the transitioned Function/Module in that dialog, you can see the Step Report data which after transitioned.
- Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.

![Module end dialog](image)
### Task start

- The filter is set subject to a specific "Task start". ("Task start" is configuration item for Step which before transitioned. Please refer to [9.6. Step Property] to know how to configure it.)
- The check is put in the check box, "Test Report display setting: Task start item" dialog is displayed to double-click "Set value" column, and select the item that sets the filter from the dialog, please.
- Click the "Add" button and you can see "Task start setting" dialog. If you select the transitioned Function/Module in that dialog, you can see the Step Report data which before transitioned.
- Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.

![Task start dialog]

### Task end

- The filter is set subject to a specific "Task end". ("Task end" is configuration item for Step which after transitioned. Please refer to [9.6. Step Property] to know how to configure it.)
- The check is put in the check box, "Test Report display setting: Task end item" dialog is displayed to double-click "Set value" column, and select the item that sets the filter from the dialog, please.
- Click the "Add" button and you can see "Task end setting" dialog. If you select the transitioned Function/Module in that dialog, you can see the Step Report data which after transitioned.
- Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.

![Task end dialog]
■ Execution attribute
  - The filter is set subject to a specific Execution attribute.
  - The check is put in the check box. "Test Report display setting: Execution attribute item" dialog is displayed to double-click "Set value" column, and select the item that sets the filter from the dialog, please.
  - Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.

■ CPU load status
  - The filter is set subject to a specific CPU load status.
  - The check is put in the check box. "Test Report display setting: CPU load situation" dialog is displayed to double-click "Set value" column, and select the range of CPU load situation in which the filter is set from the dialog, please.
  - Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.

■ Display only value of variable in Report Data
  - Please put the check in this item when you do not want to make the Report Data of each byte of the variable value output data displayed.

■ Hide the Event Trigger Data
  - Please put the check in this item when you do not want to make the Report Data of each byte of the Event Trigger displayed.
Core item

- Set a filter condition the core number.
- The check is put in the check box, "Test Report display setting: Core item" dialog is displayed to double-click "Set value" column, and select the item that sets the filter from the dialog, please.
- Only the item of the condition is displayed by switching "Show/Hide" of "Processing" item.

Memo The Test Report is acquired with the filter set.

When the Test Report is acquired with the filter set, the filtered Test Report is displayed.

This doesn't filter the acquired data but filters and displays the data acquired at any time.

If the line number of the Report Data list is set to "Sequential number for the all data" in [Tool] -> [Setting] of the menu, the filtered thing becomes comprehensible.

Attention

Please note that the load to the personal computer grows when the acquired data is filtered and displayed by you.
19. Look at CPU load detection information in the Performance Monitor

19.1. CPU Load detection

19.1.1. Mechanism of CPU Load Detection

It is a function that can be used in the software structure of the multi task.

The priority inserts CPU Load Measurement Point of the Test Point in the task of the lowest level, and a simple CPU load is detected from the number of total Test Points that can be detected between the frequency detected with DynamicTracer near 100msec and the 100msec.

DynamicTracer sends the data of CPU load measurement to the DT10 side in each 100msec.

It is necessary to make the task of the lowest level by which CPU Load Measurement Point of the Test Point is infinitely looped as a preparation for CPU load detection.

19.1.2. Specify CPU Load Detection

When the Performance Monitor is used when the Test Report is acquired, it is necessary to apply the check on the item of [Plan]→[CPU Load Detection] from the menu.

Attention

Even if the Test Point was inserted, it doesn't operate normally if the above-mentioned check is not turned on to display the Performance Monitor when the Test Report is acquired.
19.1.3. Insert CPU Load Measurement Point

"CPU load measurement point" is inserted the lowest priority task, such as the idle task.

CPU Load Measurement Point is infinitely looped in the task.

In that case, please delete the Test Point, and **loop only CPU Load Measurement Point** when other Test Points have already been inserted in the loop processing.

The method of inserting CPU Load Measurement Point right-clicks in the part where "CPU Load Measurement Point" is inserted, and displays the pop-up menu in the source code window. "The insertion of CPU Load Measurement Point" is clicked from the pop-up menu.

```
void IdleTask(void)
{
    while(1)
    {
    
    }
}
```

When "Insertion of CPU Load Measurement Point" is clicked, the Test Point of the following forms are inserted.

<<C/CPP files>>

```
__DtTestPointIdle (__DtFunc_XXXX, __DtStep_yyyy, zzzz);
```

<table>
<thead>
<tr>
<th>__DtFunc_XXXX</th>
<th>Function name that inserted position manually Automatically determined.</th>
</tr>
</thead>
<tbody>
<tr>
<td>__DtStep_yyyy</td>
<td>Step No that inserted Point manually Automatically determined.</td>
</tr>
</tbody>
</table>

<<Java files>>

```
DtTestPointDriver.DtTestPointIdle(0xXXX0, 8);
```

<table>
<thead>
<tr>
<th>0xXXX0</th>
<th>Address for the CPU load detection It is a value that becomes basic of the base address. (Ex)When 0x80000000 is a base address, 0x80000000 is allotted for CPU load detection.</th>
</tr>
</thead>
<tbody>
<tr>
<td>yy</td>
<td>Serial number of Points in the source manual insertion. Automatically determined.</td>
</tr>
</tbody>
</table>

Please do the compilation and the work of the build etc., put the target into the state that can be operated, and acquire the Test Report with DT10 when you insert "CPU Load Measurement Point".

The Performance Monitor starts when the acquisition of the Test Report starts, and CPU load information is displayed in real time.
Please note that the load to the personal computer grows while the Performance Monitor is starting.

19.1.4. Test Report

Turn on CPU load detection and then get Test Report. You get the CPU load detection Report data like following data per 100msec. Performance monitor is displayed based on this Report data.

<table>
<thead>
<tr>
<th>No.</th>
<th>Source</th>
<th>Step</th>
<th>Elapsed Time (ms)</th>
<th>Difference (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>550871</td>
<td>CommMain_SwiTask7x.c</td>
<td>FuncOut</td>
<td>21,899</td>
<td>0</td>
</tr>
<tr>
<td>550872</td>
<td>CommMain_SwiTask7x.c</td>
<td>FuncOut</td>
<td>21,899</td>
<td>0</td>
</tr>
<tr>
<td>550873</td>
<td>***** CPU Performance *****</td>
<td>91 %</td>
<td>16789</td>
<td>0</td>
</tr>
<tr>
<td>550874</td>
<td>CommMain_SwiTask7x.c</td>
<td>FuncIn</td>
<td>21,901</td>
<td>1</td>
</tr>
<tr>
<td>550875</td>
<td>CommMain_SwiTask7x.c</td>
<td>FuncIn</td>
<td>21,901</td>
<td>0</td>
</tr>
<tr>
<td>550876</td>
<td>CommMain_SwiTask7x.c</td>
<td>if</td>
<td>21,901</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>***** CPU Performance *****</td>
</tr>
<tr>
<td></td>
<td>Indicate CPU load measurement data.</td>
</tr>
<tr>
<td>Step</td>
<td>CPU load</td>
</tr>
<tr>
<td>Elapsed time</td>
<td>Number of output CPU load measurement point.</td>
</tr>
<tr>
<td>Difference</td>
<td>The minimum time interval of all Test Points</td>
</tr>
</tbody>
</table>
19.2. Performance Monitor

CPU load information can be in real time displayed graphical by making CPU Load Detection function valid. Moreover, it jumps to a pertinent line in the test when double-clicking it with ■ in the Performance Monitor has been selected.

: Zoom in of time axis

: Zoom out of time axis

: Ratio zoom in of axis

: Ratio zoom out of axis

: Automatic scrolling. Focus is applied to final data at data acquisition
19.3. Re-analysis of CPU load measurement data

CPU load measurement data can be analyzed again by specifying the range of the Report Data. When “Report Analysis → Performance Monitor” of the menu is selected, the following dialogs are displayed. The Performance Monitor is displayed within the range specified that the analysis range is specified, and “OK” is clicked. The range specification can be selected from “All data “, “Target is specified data number ranges “, and “Target is data corresponding to the specified time range”. (* The range of beginning of “Target is specified data number ranges “ and “ Target is data corresponding to the specified time range “is fixed)

![Analytical range specification dialog]

19.4. The overhead is measured

The overhead to insert the Test Point when the Test Point is inserted, and the target is operated is generated. There is a method by which the overhead for each point of the one test can be easily measured. Two consecutive Test Points are inserted and confirming the difference with the Test Report.

It vanishes almost at the Execution time of the Test Point and the Test Point because the Test Points of two it continuous are buried. Therefore, the difference time between those and the output Execution time of the Test Point almost become equal signs.
19.5. The image of a graphical monitor is copied onto the clipboard

When it races with the Performance Monitor, the Function Trace Report, and the event, and the Variable Monitor is used, it introduces the method of preserving the image.

The capture function doesn't correspond to the application of DT10. However, if the print screen installed as a standard function of Windows is used, the image can be preserved. The window to be preserved is made active by floating (state that has not docked with other windows), and the screen in the window is copied onto the clipboard with [Alt]+[Print Screen] key.

①「▼」of the title bar is clicked, and “floating” is selected. Or, it is possible also by Drug and Drop.

②When floating keeps active, [Alt]+[Print Screen] key is pushed, and it copies it onto the clipboard.

③It put on the painting software etc.
20. Event Trace Function

20.1. Outline

The Event Trace function is a function to analyze the situation of the event transition. In Event Trace, Event data output by burying special Test Point (Event Trigger Point) in the function (instruction) where the user generates the state transition.

- Usage example
  - The Test Point is set to the function that dispatches the task, and the task change is confirmed.
  - The Test Point is buried under the function that manages the sequence number and the flag, and the state transition is confirmed.

20.2. Insert Event Trigger Point

The user inserts the following Test Points manually

<<C/CPP files>>

```c
__DtTestPointTrigger(__DtFunc_XXXX, __DtStep_yyyy, zzzz);
```

<table>
<thead>
<tr>
<th>__DtFunc_XXXX</th>
<th>Function name of position in which manual embed was done.</th>
</tr>
</thead>
<tbody>
<tr>
<td>__DtFunc_XXXX</td>
<td>It is decided by the automatic operation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>__DtStep_yyyy</th>
<th>Step number of point in which manual embeds was done.</th>
</tr>
</thead>
<tbody>
<tr>
<td>__DtStep_yyyy</td>
<td>It is decided by the automatic operation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>zzzz</th>
<th>Variable identifier and value soon. From 0 to 0xFFF can be specified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>zzzz</td>
<td>The user specifies it.</td>
</tr>
</tbody>
</table>

<<Java files>>

```java
DtTestPointDriver.DtTestPointEventTrigger(0xXXXX, yy, zzzz);
```

<table>
<thead>
<tr>
<th>0xXXXX</th>
<th>Base address of source file that does manual burial</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xXXXX</td>
<td>It is decided by the automatic operation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>yy</th>
<th>Serial number in source of point that does manual burial.</th>
</tr>
</thead>
<tbody>
<tr>
<td>yy</td>
<td>It is decided by the automatic operation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>zzzz</th>
<th>Variable identifier and value soon. From 0 to 0xFFF can be specified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>zzzz</td>
<td>The user specifies it.</td>
</tr>
</tbody>
</table>
It buries it as well as a usual manual burial. It right-clicks at the position to be inserted on the source code window, and "Insertion of the Event Trigger Point" is selected from the pop-up menu.

```c
switch( Task_Number )
{
    case 1:
        _DtTestPoint(_DtFunc_UserMain, _DtStep_1)
        Insert Test Point
        Insert Variable Value Point
        Insert CPU Load Measurement Point
        Insert Event Trigger Point
        Insert Cancel Step
        break;
}
```

The dialog that sets the value (above-mentioned zzzz) specified the selection for the Event Trigger is displayed.

The value and the variable that becomes the third argument of the Test Point are specified in the dialog box in the figure below.

As for the specification method, the method to specify by 【character string】 of the method, the variable identifier, and the [define] name, etc. in which task ID etc. are specified by 【direct value】 can be selected.

【direct value】 When 【define】 is specified, the input range becomes from 0 to 4095.
When 【variable identifier】 is input, it is possible to input it up to 128 bytes. However, please input the character string that is appropriate for the compiler used. (Please avoid the description that becomes a compile error.)

When necessary information is described and 【OK】 is clicked, the Event Trigger Test Point is buried as follows.)

```c
if( MBX_pend( &MBX_CommToMain, &msgCom, 5 ) != FALSE )
    _DtTestPoint(_DtFunc_Process_TskTask, _DtStep_2)
    _DtTestPointEventTrigger(_DtFunc_Process_TskTask, _DtStep_3, fromTaskId)
    ProcessMsgGet_ComToMainSwiTAsk(&MBX_ComData);
```
20.3. Making of driver function

You need to generate another driver function (or binding another driver function to the target environment) to output the Event Trigger Test Point.

There are sample of driver function in the Install folder in [sample] folder. Copy the process of this sample code to Target's environment or register Source file to Target's Project.

20.4. Test Report Data

The Test Report that DynamicTracer acquired when the target that sets the Event Trigger is operated is displayed as follows.

<table>
<thead>
<tr>
<th>No.</th>
<th>Source</th>
<th>Function</th>
<th>Step</th>
<th>Memory</th>
<th>Value of a variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>15539</td>
<td>user.c</td>
<td>KeyMain</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15540</td>
<td>user.c</td>
<td>***** Event Trigger *****</td>
<td>10</td>
<td>level error</td>
<td></td>
</tr>
<tr>
<td>15541</td>
<td>user.c</td>
<td>***** Event Trigger *****</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15542</td>
<td>system.c</td>
<td>int_intt4</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15543</td>
<td>system.c</td>
<td>int_intt4</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source name in which Event Trigger is embedded.
The above-mentioned shows the thing that is the Event Trigger.
Value of zzzz that user specified at manual burial
Name that defines event (following description)

The Event Trigger, the difference between normal Test Point (FuncIn, if Step, etc.), is near CPU load detection Test Point. For example, it becomes off the subject of the filter, the retrieval, and the analysis, etc. Moreover, please note no jump to a pertinent Step like the normal Test Point even if you double-click the Event Trigger item in the test.
20.5. Event Definition

You can set "Name"(define of event) to Event Trigger ID. Event definition's setting is optionally.

When [Report analysis] →[Event Trace] →[Event definition] is selected from the menu, the following dialogs make the event definition possible.

The Project file maintains the event definition name set here as event definition data.
As for the definition data, 4096 defined things are possible.

Please click 【Add】 button in the dialog when you want to add the definition newly

The value of ID in that case can select even from0 to 4095. The character string can be specified for each ID. The character string can be input up to 32 bytes.
ID is already impossible as for overlap with the registered one in case of the 【Add】
When the 【Change】 button is clicked, this dialog is displayed. However, ID is not revocable in this case. It comes to change only the definition name.

![Definition name settings dialog](image)

Please click 【Delete】 and 【All delete】 on the dialog when you delete the event definition. Please click on a button at the 【Delete】 with the item to be deleted beforehand selected from the list. As for 【All delete】 the message of the confirmation is displayed as shown in the figure below. Please click OK when really deleting it.

![All Delete confirmation](image)

The data defined here is displayed by the Test Report Event Trace View (following description).
20.6. Event Analysis

When [Report analysis]→[Event Trace]→[Event Analysis] is selected from the menu, it becomes an Event Trace Analysis screen. Event information in the Test Report data being used now is analyzed when the Event Trace analysis is done, and it is possible to confirm details by the trace view.

The analysis range in the Report Data is specified first by the following dialogs:

![Analytical range specification dialog](image)

- **[All data]** has been selected and there is no problem usually. (It takes time for the analysis.)
- Please set the range by **[Target is specified data number ranges]** and **[Target is data corresponding to the specified time range]** when you want to squeeze it only to given categories wanting shorten analytical time.

The number in which **[Target is specified data number ranges]** is specified reaches the value of ‘No.’ of the Test Report View. It doesn't depend in the state of the environmental setting and the filter, and specifies by the number (absolute number at the following) in the sequential number for all data of the Test Report.

When **[Target is data corresponding to the specified time range]** is specified, an absolute number that corresponds at time within the range is detected by the automatic operation, and the range is analyzed. Moreover, it is judged the analysis to final data when exceeding it to the final time of the Report.
Please note that it takes time to analyze the event. Moreover, to use for the analysis, a large amount of memory is needed. Operation becomes heavy when a physical memory is insufficient, and notes the operation specs enough, please. Moreover, please note that it similarly takes time when the secured memory is liberated.

※Please execute measures such as narrowing the range of an analytical object to increase an empty memory capacity of PC when the performance falls greatly.
20.7. Event Trace View

When the analysis ends, the event transition chart (above graph) and the analysis result (under) are displayed in the Event Trace View.

- **Tool bar**
  - The zoom in of time (length) axis (expansion) is done. The maximum magnification is 1280ns/Pixel.
  - The zoom out of time (length) axis (reduction) is done. A minimum magnification is about 1.3S/Pixel.
  - The zoom in of ID (side) axis (expansion) is done. The maximum magnifications are 32 Pixel in width.
  - The zoom out of ID (side) axis (reduction) is done. The maximum magnification is 1ID/32Pixel → 512ID/32Pixel.
  - However, the number of total ID becomes when thinning out is generated and the thing becomes a condition the twice or more the requirement.
  - Page switch is done. It moves on the previous page. When the Report Data of ten minutes or more is analyzed, the switch can be used.
  - Page switch is done. It moves to the subsequent page. Width on page 1 is ten minutes, and the interval at page beginning position is five minutes.
  - It jumps one to the previous point about the point that has been selected. Even the F7 key is similar.
It jumps one of the points that have been selected to the following point. Even the F8 key is similar.

It jumps to the previous ID change point of one of the point that has been selected. Even the Shift+F7 key is similar.

It jumps to the next ID change point of one of the point that has been selected. Even the Shift+F8 key is similar.

The Function Trace Report is begun. After the icon is clicked, the beginning point and the end point are selected.

The re-execution analysis of the event is done. It is analyzed again within the range specified that it clicks on a button, and Start Point and End Point are specified on the Event Trace, and the Event Trace is displayed.

An analytical mode is switched. The method of analyzing the event is made 'Trigger end' specification.

Event point (■) becomes non-display.

### Event transition chart

The upper part of the Event Trace View is an event transition chart. The change of the data value of the output Event Trigger is pursued, and the state transition is displayed graphical. A horizontal axis is expressed at time in the test the spindle by the output data value (ID at the following).

It becomes easy to confirm the Event Transition because the name related to ID comes to be output by defining the event.
Point by which □ detected trigger output. The ruled line shows the transition of the state.
□ becomes green ■ by matching the mouse to, and information on the point is output.
Red ■ is a point that the user selected by clicking.
When □ is double-clicked, a pertinent point can be displayed to the Test Report (jump).
In the state with red ■, it becomes possible to display the difference time in green ■. The difference time is displayed as 'Diff:xxxx'.

If focus hits the Event Transition Chart, the jump of the point becomes possible by combining F7, F8, and the Shift key.
Please refer to the item of "Toolbar" for details.
Moreover, when mouse's right drug is done on the Event Transition Chart, the scroll toward the direction from the center is possible according to the distance from the center position.

The performance of PC decreases according to becoming if the number of information (* of the event point and ruled line etc. of the transition) displayed in the event transition chart increases. Please note specs enough when the drawing area is expanded or the zoom out is piled up.

It enters the range specification mode clicking  icon once when the Function Trace function is used. The Function Trace analysis is executed by the automatic operation when the beginning point and the end point are clicked in the state, and the Function Trace View is displayed. The end point makes the retrieved point an end point when the point is retrieved by  etc. and executes the Function Trace.

The arrangement of ID (side) axis usually queues up in order of ID acquiring it in the state so that event information may make the transition chart in order of the item of the Test Report. However, it might be convenient in the state to analyze all data that ID queues up in order with young number.
For that case, when  icon is clicked, ID (side) axis is sorted in order with young number.
Please click the icon again when you release sorting.

Event information has delimited the expressible information area for ten minutes as page 1 though it can use all of the one Test Reports. The point of page switch is every five minutes. The treatment of the page doesn't exist for the data installed at ten minutes.

Ex) Page 3 comes to switch at the first page from 0 to 10 minute, the 2nd page from 5 to 13 minute, and the 3rd page from 10 to 13 minute in case of the Test Report of 13 minutes.
### Analysis result

<table>
<thead>
<tr>
<th>ID / Name</th>
<th>Share-Time(μS)</th>
<th>Share-Ratio(%)</th>
<th>Min-Time(μS)</th>
<th>Max-Time(μS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 EEROM-W...</td>
<td>565,854</td>
<td>6.312</td>
<td>9,071</td>
<td>9,174</td>
</tr>
<tr>
<td>2 EEROM-R...</td>
<td>443,743</td>
<td>4.950</td>
<td>7,155</td>
<td>7,158</td>
</tr>
<tr>
<td>3 ProcessTask</td>
<td>450,962</td>
<td>5.030</td>
<td>7,157</td>
<td>7,159</td>
</tr>
<tr>
<td>4</td>
<td>7,504,365</td>
<td>83.708</td>
<td>120,958</td>
<td>121,352</td>
</tr>
</tbody>
</table>

- **ID** Data value that trigger output
- **Name** Definition name corresponding to ID
- **Share-Time** Between total time in the ID section
- **Share-Ratio** Ratio in the ID section in all event section
- **Min-Time** The section time in ID that was the shortest
- **Max-Time** The section time in ID that was the longest

The lower side of the Event Trace View is a list that analyzes information in the analysis range. Between total time and the schedule combination, etc. with each event can be confirmed. (* The display of the list becomes ---- for the item of 0 at the total time.)*

Especially, it becomes useful with the emergency though the task share is investigated when the task switch is made an event.

The pop-up menu is displayed when right-clicking on the list.

The unit can be changed to time in the list. (It doesn't influence another list in the Event Transition chart and the Test Report, etc.)

Moreover, it is possible to copy the line in selected list onto the clipboard. Please press the copy or 'Ctrl+C' with the line to be copied selected with the mouse. The item name can be copied together by selecting [the copy with the header].

It is possible to sort the list in the ascending order and the descending order of the row by the click of the column header of the list.

"Definition name" can be changed by double-clicking the item of Name of an analytical list, and starting the dialog. The event definition cannot be deleted by this dialog. Please delete the event definition by "Definition of the event" dialog (Display by [Report analysis] →[Event Trace] →[Definition of Event] of the menu).
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Share-Time(μS)</th>
<th>Share-Ratio%</th>
<th>Min-Time(μS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TaskA</td>
<td>5.194</td>
<td>8.599</td>
<td>6.194</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>8.452</td>
<td>8.915</td>
<td>6.452</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>5.421</td>
<td>8.432</td>
<td>5.421</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Setting of definition name**

- **ID**: 0
- **Definition name**: TaskA

[Image of dialog box for setting definition name]
20.8. Switch of analytical mode

An analytical mode can be switched by clicking buttons on the toolbar.
The trigger end mode that can analyze the Event Trigger as "Terminal of the event" can be specified when clicking on a button.

■ Analysis usually

- End mode analysis

It is an Event Trigger that the point of embedded. The segment to becomes the elapsed time of the Event Trigger.
Time and information on each event of the list changes comparatively at the same time, too.
20.9. Import/Export function of Event definition

20.9.1. Export function

It is said, "Export of the Event definition" to write the Event definition preserved in the DT Project to text file (*.txt).

The exporting Event definition can read to other Projects.

To export of Event definition, [Report Analysis] > [Event Trace] >[Event definition] > [Export...] of the menu is selected.

When "Export" is selected, "Export event definitions" dialog is displayed.

In the dialog, Event definitions are listed.

"Preserve it giving a name" dialog opens when the check on the exporting definition is turned on, and "OK" is clicked.

The preservation place and the name of the file are specified, and it Exports. （※There is no limitation in the file name. Please specify the file name in any of the user.)
The exported text file delimits the identification number and the definition name by the comma, formed ≪ID,Name≫. It is enumerated as one definition a line in order of ID.

<table>
<thead>
<tr>
<th>ID</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Level 0</td>
</tr>
<tr>
<td>2</td>
<td>Level 1</td>
</tr>
<tr>
<td>3</td>
<td>Level 2</td>
</tr>
<tr>
<td>4</td>
<td>Level 3</td>
</tr>
<tr>
<td>5</td>
<td>[EOF]</td>
</tr>
</tbody>
</table>
20.9.2. Import function

It is said, "Import of the Event definition" to read text file (*.txt) described in the form of specification, and to take the Event definition into the Project.

(※Import is possible if entering according to the description rule even if it is not a file output by the export function of DT10. Please see【20.9.3. Description rule of Event definition file】about details of description rule.)

To Import of Event definition, [Report Analysis]→[Event Trace]→[Event definition]→[Import...] of the menu is selected.

When "Import" is selected, "The file is opened" dialog is displayed. In the dialog, text file (*.txt) where the Event definition is described is selected. There is no limitation in the file name.

When the file is selected, "Import of the Event definition" dialog is displayed. In the dialog, Event definitions are listed.

Even if the description in the file is in any order, the list is sorted in order of ID and displayed.

Moreover, a pertinent line is displayed in red when there is something that an existing definition of the DT project overlaps with ID.

When the check on the definition that does import is turned on, and "OK" is clicked, import is done only as for the selected event definition.
20.9.3. Description rule of Event definition file

The import of the Event definition of DT10 can do the Import by using the text file edited in an external editor if entering according to the description rule even if it is not a file output by the Export function of DT10.

There is a description rule of the following files.

- The basis is «ID,Name».
  The identification number and the event definition name are delimited by the normal-width comma, and it enumerates it as one definition a line.

<table>
<thead>
<tr>
<th>Status</th>
<th>Example</th>
<th>DT10 Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is space between the ID number and the comma.</td>
<td>ID,...,Name</td>
<td>It is not recognized.</td>
</tr>
<tr>
<td>There is space between the comma and the definition name.</td>
<td>ID,...,Name</td>
<td>The definition name is recognized &quot;...Name&quot;.</td>
</tr>
</tbody>
</table>

- Please do not put space between ID and the name.

- Please change lines every one definition.

<table>
<thead>
<tr>
<th>Status</th>
<th>Example</th>
<th>DT10 Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two or more definitions were described in one line.</td>
<td>ID1,Name1,ID2,Name2,</td>
<td>The definition name is recognized &quot;Name1ID2Name2.&quot;</td>
</tr>
<tr>
<td>Two or more definitions were described in one line.</td>
<td>ID1,Name1,...,ID2,Name2,</td>
<td>The definition name is recognized &quot;Name1...,ID2Name2.&quot;</td>
</tr>
</tbody>
</table>

- Even if ID and each definition name are tied up by (normal-width double quotation "), it is recognized. However, it is not recognized that the entire line is bundled.

<table>
<thead>
<tr>
<th>Status</th>
<th>Example</th>
<th>DT10 Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID is tied up by &quot;</td>
<td>&quot;ID&quot;,Name</td>
<td>It is recognized.</td>
</tr>
<tr>
<td>Definition name is tied up by &quot;</td>
<td>ID,&quot;Name&quot;</td>
<td>It is recognized.</td>
</tr>
<tr>
<td>The entire line is bundled by &quot;</td>
<td>&quot;ID, Name&quot;</td>
<td>It is not recognized.</td>
</tr>
</tbody>
</table>
Please bundle the definition name by (normal-width double quotation ") when you want to use the definition name (normal-width comma).

<table>
<thead>
<tr>
<th>Status</th>
<th>Example</th>
<th>DT10 Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is a comma in the definition name, and it bundles it by &quot;</td>
<td>ID,&quot;Na,me&quot;</td>
<td>The definition name is recognized &quot;Na,me&quot;.</td>
</tr>
<tr>
<td>There is a comma in the definition name, and it doesn't bundle it by &quot;</td>
<td>ID,Na,me</td>
<td>The definition name is recognized &quot;Name&quot;.</td>
</tr>
</tbody>
</table>

Please bundle by "Definition name to use (normal-width double quotation)" in the definition name, and make "in the definition name two reams.

<table>
<thead>
<tr>
<th>Status</th>
<th>Example</th>
<th>DT10 Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is &quot; in definition name, and &quot; is made two reams.</td>
<td>ID,&quot;Na&quot;me&quot;</td>
<td>The definition name is recognized &quot;Na&quot;me &quot;.</td>
</tr>
<tr>
<td>There is “ in definition name, and “ is not made two reams.</td>
<td>ID, &quot;Na&quot;me&quot;</td>
<td>The definition name is recognized &quot;Name &quot;.</td>
</tr>
</tbody>
</table>
21. The specified value of a variable is displayed in the Test Report

21.1. Outline

When the program passes this Test Point when the Variable Value Point is inserted, the value of the specified variable can be displayed in the Test Report in DT10. The dump display of the variable in a specific part can be done by using this function.

It is necessary to insert function (*1) for the variable dump and the Test Point in the source code to use this function, and to register Property information on the variable. (*2)

The variable value output point is inserted by the following forms.

**<<C/CPP files>>**

```
__DtTestPointValue (__DtFunc_XXXX, __DtStep_yyyy, (void *) zzzz, aa);
```

<table>
<thead>
<tr>
<th>__DtFunc_XXXX</th>
<th>Function name of position in which manual embed was done.</th>
</tr>
</thead>
<tbody>
<tr>
<td>__DtStep_yyyy</td>
<td>Step number of point in which manual embeds was done.</td>
</tr>
<tr>
<td>zzzz</td>
<td>Variable identifier</td>
</tr>
<tr>
<td>aa</td>
<td>Number of output bytes of variable</td>
</tr>
</tbody>
</table>

**<<Java files>>**

```
DtTestPointDriver.DtTestPointValue ( 0xXXXX, yy, zzzz, aa);
```

<table>
<thead>
<tr>
<th>0xXXXX</th>
<th>Base address of source file that does manual burial</th>
</tr>
</thead>
<tbody>
<tr>
<td>yy</td>
<td>Serial number in source of point that does manual burial.</td>
</tr>
<tr>
<td>zzzz</td>
<td>Variable identifier</td>
</tr>
<tr>
<td>aa</td>
<td>Number of output array lengths of variable</td>
</tr>
</tbody>
</table>

※1 The function for the variable dump has the sample source in the installation folder in [sample] → [driver_samples], and the processing that exists in this sample source is copied onto the target environment or register the source file in the Project of the target, please.

※2 To reflect Test Point information in the Test Report after the Test Point is inserted, it compiles and the outside of the build etc. should work, and it download it to the target environment.
<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>The dump data of the variable is divided when another Test Point is inserted by the interrupt etc. when the value of a variable is output with the Variable Value Point while outputting and it is not data after the Report is analyzed warrantable. In the function for the variable value output, interrupt and the task switch are recommended to be prohibited.</td>
</tr>
</tbody>
</table>

*The variable interrupted is not evaluated when there are interrupt etc. of another task on the way of the variable dump. The Report that interrupts is evaluated.*

<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please register the variable substance (object) in the registered one when you register the variable identifier in the Property Editor of the function to insert the Variable Value Point. Please do not register 'Address'. Please work applying indirection reference (*) etc. for the pointer. (&quot;The purpose is Cast it as taking the address of the object automatically when the Variable Value Point is inserted. &quot;)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please note that only the primitive type (basic model) should prepare the driver for the variable output about the driver function for the variable value output when you use Java.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>The specification of the driver for the output variable value is changed from Ver7.00. When you collect the variable value with DT10 Ver7.00 application, before driver cannot use. DT10 Ver7.00 can only display the output data of variable value which collected with DT10 (before Ver6.11).</td>
</tr>
</tbody>
</table>
21.2. Variable Property value setting

**Belonging of Property**

The Variable Value Point divides into the following two kinds of properties.

- **Argument (Argument Type):** It is a Test Point that specializes in the argument by belonging to the number of properties.
  
  It is generated only with the Test Point automatic operation insertion function, and it manages and it operates it as a Test Property of the function.

- **Variable (Variable Type):** It is a Test Point for the memory dump that can be inserted in an arbitrary position in the source (manual insertion from the source code view).
  
  It manages and it operates it as a Step Property in the function as well as other Steps.

**Kind of Variable Test Point**

The variable Test Point has four kinds of the following according to the content to be output.

- **Size no specified memory dump**
  
  : When the number of output bytes is executed by sizeof(), it fixes it.
  
  Only the file of C/CPP can be selected.

- **Direct reference type memory dump**
  
  : The value of a variable is output specifying the number of output bytes directly.

- **Indirect reference type memory dump**
  
  : The variable of the object is treated as a pointer, a variable size and an indirect reference level are specified, and the content referring ahead is output.
  
  Only the file of C/CPP can be selected.

- **Array type memory dump**
  
  : The variable of the object is treated as an array, the number of array elements is specified, and the content referring ahead is output. Only the file of Java can be selected.
Setting of Variable Property

The Property value in which both Argument types (Function Property) and Variable types (Step Property) can be set is as follows.

Variable Name: It is variable name.

Please an arbitrary character string is input and input the code (content where the compile error is not generated) along the compiler used to the inserted character string though it is possible to input.

If you write the variable name enclose in "", you can register characters that are not used as a variable name, such as operator.

The automatic insertion function (burial library) acquires and sets the name from the source code for the Argument type. When the variable output point manual operation from the source code view is inserted, an arbitrary character string is specified for the Variable type. It is not possible to change in the variable identifier in the Property Editor.

I/O: The variable attributes are used to determine whether an input or output value.

If the input value attribute "type", the output value if the attribute "Output" is specified.

On the Test Report View, the selected attribute is displayed in the column of "Memory" of the DUMP_START line.

```c
void func_a(int arg_b)
{
    int a = 0;
    a = arg_b;
    _DtTestPointValue(_DtFunc_func_a, _DtStep_0, (void *)&a, 4)
    _DtTestPointValue(_DtFunc_func_a, _DtStep_1, (void *)&arg_b, 4)
}
```

Variable value “a” is “Output attribute”.
Variable value “arg_b” is “Input attribute”.

Reference: The reference method of the value of a variable is specified.

- When the source file is C/CPP
  Specify direct reference (Val) or indirect reference (Ptr).
  Indirection reference
- When the source file is Java
  Specify direct reference (Val) or array.

Level: The level of the indirection reference is specified by the item that becomes valid only when "Ptr" is selected by "Reference".

- For int **pData (The variable identifier is pData)
  Level 1 → (void *)pData ・・・Want to see the address of a double pointer.
  Level 2 → (void *)*pData ・・・Want to see the address of pointer
  Level 3 → (void *)__pData ・・・want to see the substance of the pointer.
Number of output bytes / output array length:

Specify the number of array elements, or dump. When the source file is C/CPP, it is possible to select it from "Direct value", "Character string" and "sizeof".

- **Direct value**
  The dump size of the variable is specified by the direct value.
  If unspecified, sizeof () will be.
  It becomes impossible to evaluate Range/unexpected value when the numerical value that exceeds eight bytes is input.

- **Character string**
  The dump size of the variable is specified by character string.
  A changeable value can be specified.

- **sizeof**
  The dump size of the variable is specified by sizeof.
  When "Reference" is only "Val (direct reference type)", it is effective.

  When "Reference" is "Val (direct reference type)" when the source file is Java, it is not necessary to change the number of bytes.
  When "Reference" is "Array", the array length of the output is specified.
  "Direct value" is turned on, and the number of array elements is specified.
  Because the array is dump data, the range judgment of the test property (Range and Illegal) is not done.

**Signed**: The value of a variable specifies whether there is a sign or is no sign. It influences the display and the evaluation of the value of a variable of the Report etc.

**Range**: The range assumed to be an error judgment standard when analyzing it is specified. When unspecified it, the range is not judged.

The range of the variable is specified by using '_ (under bar)'.

- **<Ex1>** Range : 1_10

  Moreover, two or more ranges can be specified by using ", (comma)" for the delimitation within the range

- **<Ex2>** Range : 1_10,20,30_40
Typ:  The expected value of this variable (theory value) is specified.

Illegal value: The unexpected value assumed to be an error judgment standard when analyzing it is specified. When unspecifying, the unexpected value is not judged.

Attention

In the Java source file, char type variable name is "unsigned", otherwise "signed" set, please. If not set correctly, since it affects the display of variable values and evaluation, please note.
21.3. Automatic insertion of Variable Value Point

It can be selected whether to insert the Test Point of the argument attribute (arg attribute) when the automatic insertion function of the Test Point is executed.

It is in new making wizard's "Specification of the automatic generation file" dialog on the third page, and can do the automatic insertion of the Variable Value Point concerning the argument by turning on the check box of "The Variable Value Point to the argument of each function is inserted automatically by the automatic insertion of the Test Point". Default is in the state of off.

Moreover, the switch is good at the check box with the third page of the Project setting dialog as well as the new Project making wizard.
In this case, it is necessary to select byte order specification the value of a variable is evaluated. (Process it by the byte order specified when it is evaluated to always change.)

Next, about the argument embed when inserts in the registered source file automatically. The check box of "Auto inserting Variable Value TP to Argument of each Function" in the displayed source selection dialog in the automatic insertion of the Test Point can be switched. Please make it to turning on when you embed the argument automatically.

However, when the check is turned on, [Remove all existing Test Points before insertion] is automatically turned on, too.

- Possible pattern

1. New embed (After it deletes it, bury it) + Argument none. Possible
2. New embed (After it deletes it, bury it) + Argument. Possible
3. Additional embed + Argument none Possible
4. Additional embed + Argument Impossible

Additional embed cannot do having argument. In that case, please execute the pattern of ② or embed the argument added by a manual burial after ③.

Attention

In the Java source files, when Test Point of the Argument attribute is inserted automatically, the Test Property is set to “Val type” at the “Reference” and “1” at the “Size”.

"Reference" is fixed to “Val “. Please change the Size according to the argument. If not set correctly, since it affects the display of variable values and evaluation, please note.
21.4. Manual insertion of Variable Value Point

The insertion of the Variable Value Point from the source code window selects "Insert Variable Value Point" from the pop-up menu displayed by right button click of the mouse by the line to be inserted thoroughly to the past.

When the above-mentioned command is selected, the following dialogs are displayed.

In a set dialog of the Variable Value Point, each item of the variable identifier, the reference selection, the size, the indirection reference level ("Ptr" is only selected by the reference), and the Test Property is input(Please see [Setting of Variable Property][21.2. Variable Property value setting] about the content of each item)

In the combo box of the drop down form, the variable identifier input column can select the variable identifier (And, it is argument name input automatically) input in the past from the list. When the variable identifier input in the past is selected, the input of each item when inserting it by the variable identifier is restored.

Each item of the Test Property can be omitted (The omitted item is not evaluated).

The Variable Value Point is inserted in the source code by clicking the [OK] button.
21.5. Property Editor

After the Variable Value Point is inserted, the designed value can be changed in the Property Editor.

The case of variable (var) attributes changes by the function Property Editor for argument (Arg) attribute in the Step Property Editor.

■Argument (Arg) attribute

Variable Value Point of argument (Arg attribute) corresponds to the Test Property of the function that it belongs.
The Test Property of the argument Variable Value Point sets in the Function Property in the Property Editor.

Set values such as "I/O", "Reference", "Level", and "Signed" can be changed in the Property Editor. (Please see [Setting of Variable Property][21.2. Variable Property value setting] about the content of each item.).
※ The argument Variable Value Point is inserted and it is not possible to delete it from the Property Editor.

When the change is fixed (When you click time when another function was selected after it changes, the selection of another set page, and the OK button), "Reference", "Number of bytes(C/CPP)", "Output array length(Java)" and "Level" change the pertinent Test Point macro code in the source code.
At this point, to maintain the agreement of the source and the Project, a Project and a pertinent source are preserved in the superscription. Please note the right of access of the object source file when you change. (Please do not make the source file a read-only attribute etc.) When the superscription cannot be preserved, the change is canceled.
■ Variable (var) attribute

The Test Property of common Variable Value Point (Val attribute) inserted from source code view by pop-up menu sets it in the Step Property in the Property Editor.

Each item in the window the lower becomes valid when (The argument is included) is selected from the list of the left on the edit page of the Step Property, and the Test Property value concerning the variable can be input. The input of each item of the reference, the size, the level, the signed, Range, Typ, and the unexpected value is the same as the setting of the argument on the function page.

It is also possible to edit the Property value of the argument (Test Point of the Arg attribute) on this page. In that case, the change result is reflected in the Argument Property of the function

“All Steps are displayed” / “Only the variable value output Step is displayed” can be selected in the combo box arranged under the list of the window left end. When the latter is selected, the list only of the variable value output Step (The argument is contained) is displayed in the list.

21.6. Interchangeability with existing Project file

As for the specification of “Reference” added from Ver2.30, all things made by an existing version become “Val” types. Moreover, information registered in “Argument, variable, and return value” column that exists on the function page of the Property Editor in an existing version is added to the history (drop down of the variable identifier) in the Variable Value Point setting dialog displayed by manual insertion of the Variable Value Point.

Oppositely, please acknowledge not opening the Project file preserved by the version since ver2.30 in an existing version.
21.7. Test Report

When the test execution is done with DT10, and the Test Report is acquired, the result of the variable dump is displayed in the Test Report.

(* To reflect Test Point information in the Test Report after the Test Point is inserted, it compiles and the outside of the build etc. should work, and download it to the target environment.

---

**Memo** The display of the value of a variable is changed

The display of the value of a variable displayed in the Test Report can be set.

The method right-clicks on the Test Report view and display the pop-up menu. The display of the value of a variable can be selected when it is clicked to [Change the radix of variable] from the pop-up menu.
22. Variable Monitor Function

22.1. Outline

The appearance of the displacement of the time series of the variable that the user specified can be displayed in the graph by using the Variable Monitor function.

22.2. Setting of Variable Monitor

The Variable Monitor is set before the Variable Monitor is analyzed.

The variable to be displayed from the list of the Test Point of a valid variable attribute to the monitor is selected by a present Project. The selection maximum number becomes 50.

It sets by the following dialogs.

When of [Report Analysis] → [Variable Monitor] → [setting] of the menu is selected, the following dialogs are displayed.

![Variable Monitor Setting Dialog](image)

**Registration:** The item that turns on the check box is registered in the Variable Monitor. It is possible to register up to 50. The registration numbers is allocated in order of making to turning on, and adjust it in the check order, please when you want to change order.

**Variable Name:** It is a variable name.

**Source:** It is a source file name that the variable belongs.

**Function:** It is a function name that the variable belongs.

**Step:** It is a Step where the variable belongs.
### 22.3. Variable Monitor

In the Variable Monitor, the acquired Test Report is analyzed, and the appearance of the transition in the time series is displayed to the variable set beforehand in the graph as follows.

The window of the monitor divides into three blocks.

1. List that displays information on each variable in detail (under)
2. Single graph of choice variable of list (upper right)
3. Graph overlapping it with two or more variables chosen by list (The check box was turned on)(on the left).

※ The line in the graph is similar to the sample of the ruled line displayed when the check box of “No.” is turned on.
Display list

No.: Registration number of variable. It comes to be displayed that the check box is turned on in "Overlapping graph".

Variable Name: It is a variable name.

Source: It is a source file name that the variable belongs.

Function: It is a function name that the variable belongs.

Step: It is a Step where the variable belongs.

Present value: It is a value of a variable at the selection time (following description).

Output MIN: It is a variability region (Minimum value) in the acquisition Report of the variable.

Output MAX: It is a variability region (Maximum value) in the acquisition Report of the variable.

Output Count: It is an occurrence count in the acquisition Report of the variable.

Tool bar

- The zoom in of time (length) axis (expansion) is done.
- The zoom out of time (length) axis (reduction) is done.
- The zoom in of a numeric (side) axis (expansion) is done.
- The zoom out of a numeric (side) axis (reduction) is done.

Fundamental motion

- The scale of each variable of each variable in a single graph is maintained. An initial value is calculated from the variability region in the acquisition Report.

- The axis of the time of both graphs changes synchronously. The display of "Overlapping graph" synchronizes, too, when the magnification of the spindle and scrolling by "Single graph".

- When displayed "Selected line"(displayed green line when click on the Monitor), you can zoom in and out with respect to the line. If the line is not displayed, you can zoom in and out with respect to the top time on the Monitor.

- The value of each value of a variable at the time selected on the graph is displayed in "Present value" of the list. "Output MIN", "Output MAX", and "Output frequency" are fixed by the value in the acquisition Report (Do not depend at the selection time).
23. Multi Wave Scope

23.1. Outline

Multi Wave Scope (MWS) can display the data which collected by Analog Box (Logic input, analog input and so on) in graphical. It can use as simplified Oscilloscope. Multi Wave Scope can display Event trigger data and Variable Value Output data to the graph, not only Analog Box. Multi Wave Scope can display Event trigger data and Variable Value Output data to the graph, not only Analog Box. For this function, user can check the movement of the value that associated with the analog box in easier. Double-click the monitor of Multi Wave Scope, and you can jump to appropriate line of the Test Report. For this function, user can check conjunction of hardware and software.

※ Multi Wave Scope can use without Analog Box.

Only using of graph display of Event Trigger data and Variable Value Output data is available.

![Multi Wave Scope](image-url)
23.2. Collecting the display data

23.2.1. Collecting the data of the Analog Box

Analog Box added a measurement data (Logic input, Analog input) to the ever trace data, and collect those as Report data.

Analog box is connected to between Connect Box and DynamicTracer.

Connect the Analog Box's input terminal and target.

<table>
<thead>
<tr>
<th>Hardware Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AD Input</strong></td>
</tr>
<tr>
<td>Ch</td>
</tr>
<tr>
<td>Input voltage range</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Resolution</td>
</tr>
</tbody>
</table>

※Threshold value can be set for each Ch.
Analog Box setting for collect the data can set by DT10 application.

Select the [Analog Box data setting], and you can see [Analog Box data acquisition setting] dialog.

**Notify Setting**
- The auto notification mode is set.
- **Enable Interval Timer**: Voluntary is notified at the specified cycle.
- **Enable Trigger**: Voluntary is notified with the specified trigger.
- When an auto notification is generated, it is displayed in the column of 'Function' of the Test Report as "*****AnalogBoxData(Notify)*****".

**Logic Input Thresh Setting**
- The threshold of the logic input data is set.
- When the value more than a set value is acquired, it is judged the Hi level.

---

**Attention**

There is a possibility that IC is destroyed when the electric power that exceeds 24V in an Analog Box is acquired. Please do not input the electric power that exceeds.
To collect the Analog Box data, the collecting Option of Analog Box Data need to be enabled. Select [Test] → [Test Report collection condition setting] in Menu, and you can see [Test Report collection condition setting] dialog. Then check to the [Collect Data (Logic input and Analog input) from Analog Box] in dialog.

Analog Box Data is added to Report Data when collecting Test Report Data.

Multi Wave Scope Data is added to Report

In case of voluntary notification, "*****AnalogBoxData(Notify)*****" is displayed

Multi Wave Scope is displayed in auto when collecting the Test Report. Analog Box data can check at "A-Ch1 ~ 2" and "L-Ch1 ~ 4" of Multi Wave Scope.
23.2.2. Collect the Event Trigger data, Variable Value Output data

The Event Trigger data and Variable Value Output data can be displayed in Multi Wave Scope.

To display those data, the user needs to collect Event Trigger data and Variable Value Output data when collecting Test Report.

You can see the detail description about collecting Event Trigger data at [20. Event Trace Function].
You can see the detail description about collecting Variable Value Output data at [21. The specified value of a variable is displayed in the Test Report].

After collecting the Event Trigger data and Variable Value Output data, select [Report Analysis] -> [Multi Wave Scope] and open Multi Wave Scope.

You can see Event Trigger data at [Event] item. And you can see Variable Value Output data at [Var] item.

(※ In default, [Var] item is set to [No data] and it needs to set the value which the user wants to display to MWS.
You can see the detail description about value setting for display at [23.4.1. Display Variable value graph].)
23.3. Fundamental motion in Multi Wave Scope

- The value of each item is displayed by the spindle, and time is displayed with a horizontal axis in the Multi Wave Scope.

- It jumps to a pertinent line of the Report when double-clicking it on the monitor.

- The data acquired in the Analog Box displayed by the L-Ch1~4 item and the A-Ch1~2 item is displayed to acquire it in 8 bit resolution in the value as it is in default. It is displayed that the Analog Box scaling setting is done by "Environmental setting" dialog by the scaled value. Please see [26.1. Setting dialog] in detail.

- When set neither variable nor event information exist in the Test Report, the data is displayed as "NoData".
The item of the spindle displayed in the monitor is set by "Vertical control".

1. The item is selected.

2. The magnification and the position of the vertical direction are changed by the scrollbar in the vertical direction.

- The item displayed in the monitor is displayed in order in the list box of a vertical control.
- The order of each item displayed in the monitor with the button can be changed.
- The display and non-display can be switched by switching ON/OFF of the check box of the list.
- The item is selected, and the magnification (Mag) and the location (Loc) of the vertical direction are revocable in the scrollbar in the vertical direction.
- Mag and Loc can be changed, during Playback capability and getting the Test Report, too.
- "Event" The item displays the ID value of the Event Trace Point.
- "Var" The variable displayed by the item is selected by "The variable is selected" dialog displayed to click the Var item of the monitor screen.

The movement of the horizontal axis (time axis) displayed in the monitor uses the Loc control bar.

- In Loc, it is a Test Report that the Multi Wave Scope displays and the position of time during total time is specified, and it displays it in the monitor.
- The movement of every one scale is moved with the button and every one screen is moved with the button.
- The change in the magnification of the horizontal axis (time axis) displayed in the monitor uses the Mag control bar.
- Mag and Loc can be changed, during Playback capability and getting the Test Report, too.
- The width of the display of the window is changed when clicking on the button.
23.4. Variable value graph

23.4.1. Display Variable value graph

Multi Wave Scope can display transition of Value of a variable item "Var".

Multi Wave Scope can display transition of Value of a variable item "Var". To click "Var" item make display "Selection of variable" dialog.

Select the Variable you display on the dialog, click "OK". And you can see the graph on Multi Wave Scope.

Even if when Test Report is hiding, you can display Multi Wave Scope.

You can configure variable/bit field which display Multi Wave Scope item "Var" to get Test Report. You can this setting before getting Test Report.

However, Multi Wave Scope while hiding the Test Report has following limits.

- You can't control "Loc slider" "Loc navigation" and "Roll back", while hiding the Test Report.
- Each display's div value is displayed "1" while hiding the Test Report.
- Multi Wave Scope will be closed when you change viewing project. (Open the other profile report, etc...)
23.4.2. The bit specific inside variable
At the item “Var” in the Multi Wave Scope, you can specify bit of inside variable, and display it.
It makes you enable to check the each bit value.

Right click on the item “Var” in the Multi Wave Scope, and select “Specifies the bit” command.

After selected “Specifies the bit”, you can see following dialog.
This dialog can set bit number you want to show.

- **Bit number**
  - Specified the bit number you want to show.
  - The bit number has to specify at Decimal number (LSB to 0).
  - If the specified is blank, all bit value will be displayed.
    So if you want to see all bit value, this specified should be blank.

- **Name**
  - You can enter it in optional, if you have bit field define.
  - Maximum size is 32 characters.
  - Entered characters are displayed at under variable name of item “Var”, if you set Name.
  - If the specified is blank, specified bit number will be displayed like “Bit#bit number”.

![Dialog for specifying bit in variable](image)
23.5. Playback Function

The function to scroll analytical data from a specific position according to the passage of time automatically and to display is said, "Playback".

When "Playback" button in the Multi Wave Scope is clicked, the following dialogs are displayed.

![Playback Controller](image)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Pause]</td>
<td>It locates and the Playback is begun between while displaying it in the Multi Wave Scope now when clicking on a button. It stops when clicking on ![Play] a button again.</td>
</tr>
<tr>
<td>![Stop]</td>
<td>The time of the position is moved to the head of data.</td>
</tr>
<tr>
<td>![Play]</td>
<td>The time of the position is moved at the end of data.</td>
</tr>
</tbody>
</table>

**Speed**: The rollback speed can be selected. It is possible to select it from "Normal" "1/2" "1/10".

**Option**: When "Option" is clicked, the following dialogs are displayed. The conditions precedent of the Playback can be set by this dialog.

![Specify the option of Playback](image)
・The conditions precedent is set to an arbitrary channel of the logic input or the analog input, and it adds to the list box with the button of [Make condition].

・The OR condition or the AND condition can be selected by "Conditions".

・When the condition of existing in the list box is selected, and the deletion button is clicked, the condition of selecting it is deleted from the list box.

・When the button of the [deletion of all] is clicked, All conditions of existing in the list box are deleted.
24. Variable value writing function

24.1. Outline

In DT10, the value of the variable specified when the Test Report is acquired can be written, and the change in behavior be confirmed.

The Test Point for the variable value writing is inserted at an arbitrary position and the data specified for a specified variable at the position is stored.

The change in behavior by the parameter change can be evaluated without changing the target program in an arbitrary processing system.

It is necessary to insert the function and the Test Point for the variable value writing in the source code to use this function, and to register property information.

The Write Test Point is inserted by the following forms.

<<File of C/CPP>>

```c
__DtTest PointWrite (__DtFunc_XXXX, __DtStep_yyyy, (void *) zzzz, aa);
```

<table>
<thead>
<tr>
<th>__DtFunc_XXXX</th>
<th>Function name of position in which Test Point was inserted. It is decided by the automatic operation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>__DtStep_yyyy</td>
<td>Step number of inserted Test Point. It is decided by the automatic operation.</td>
</tr>
<tr>
<td>zzzz</td>
<td>Variable identifier</td>
</tr>
<tr>
<td>aa</td>
<td>Number of output bytes</td>
</tr>
</tbody>
</table>

Attention

The variable value writing function is a function that can be used when a source file and connected method of C/CPP are GPIO / SPI connections.

Please inquire of the support about details of the function for the variable value writing.

URL : [http://dt10.hldc.co.jp/](http://dt10.hldc.co.jp/)

E-mail : [user@hldc.co.jp](mailto:user@hldc.co.jp)
24.2. Insert Write Test Point

It is necessary to insert the Write Test Point to use the variable value writing function.

The Write Test Point clicks "Insert Write Test Point" of the menu displayed by right-clicking at an arbitrary position on the source code window.

When "Insert Write Test Point" is clicked, the following dialogs are displayed.

When the variable name and data that does the value of a variable writing are input and "OK" is clicked, the Write Test Point is inserted.
Variable name
- The variable name that does the value of a variable writing is specified.
- The history of the variable set to click "▼" in the past is displayed by the drop down list.

Description
- It is a column where the explanation of the Write Test Point is described. Please fill in arbitrarily.

Reference
- The reference method of a variable value is specified.
- Whether the variable value is direction reference (Val) or it is indirection reference (Ptr) that makes the value of a variable a pointer is specified.

Pointer Level
- The level of the indirection reference is specified by the item that becomes effective only when "Ptr" is selected by "Reference".
- The level that can be specified is 1~12.

Number of Bytes
- The dump size of the variable will be specified by the value.

Data
- The data of the written variable is specified. The specification method can be set from "Consecutive data specification" and "Individual designation".

Consecutive data specification
- The value of a variable writes whenever the Write Test Point is passed when "Start value" and "Increment value" are set, and the Write Test Point is inserted and it changes.
- "Start value" and "Increment value" are specified by the signed decimal.
- Writable data is ten times or less. It is not written that the Write Test Point is passed ten times after that.
Individual designation

- The value of a variable writes sequentially after it lists when the Write Test Point is passed and it changes when the given value is listed and the Write Test Point is inserted.
- Making the list selects whether the type of data is "Value" or "String" from the combo box, clicks on a button, and inputs the written data. The data that can be input is ten or less.
- The data under the selection is deleted when clicking on a button.
- The order of data can be changed with button.
- "Data" column of the list becomes possible to edit by double-clicking.
- When the type of data is "String", the character that can be input is only ASCII character. The em-size character cannot be input.

The number of maximum characters that can be input is 16 characters (byte). When a specified character string is longer than the size specification of the variable, the byte that exceeds a specified size when the variable value rewriting is executed is rounded down.
24.3. Change in Property setting

After the Write Test Point is inserted, the setting can be changed in the Property Editor.

The setting of the Write Test Point changes in the Step Property Editor.

In the Step Property of the Write Test Point, the value set when the Write Test Point is inserted can be changed.

A revocable item is "Reference", "Level", "Size", "Consecutive data specification" or "Individual designation." "Variable name" is not revocable.

Please see [24.2. Insert Write Test Point] about the explanation of each item.
24.4. Effective specification of Write Test Point

When it begins to acquire whether to do the value of a variable writing when the inserted Write Test Point is passed reporting the test, it is possible to specify.

"Execute program" dialog opens when [Test] → [Execute] of the menu is selected.

When the variable value writing is done, "Change" button of the dialog is clicked.

The following dialogs open when "Change" button is clicked.

When the variable value writing is done, the check on "Variable value is rewritten" is turned ON.

And, when writing the variable value specified that the written Test Point is selected from "Write Test Point list", and "OK" is clicked is testing executed, it does. Moreover, the data set to the Write Test Point selecting it by the list that "The data setting content is opened" button is clicked can be confirmed.

① The check is turned ON.

② The Write Test Point is selected.

③ "OK" The button is clicked.
The Test Point to make the variable value writing effective by "Write Test Point select" dialog is displayed in "Variable value rewriting" column of "Execute program" dialog.

Under such a condition, the Test Report from which the Write Test Point is displayed in "Variable value rewriting" item of "Getting Test Report" dialog when the Test is executed, and writing is executed is acquired.
24.5. Test Report

When the Write Test Point is set and the Test Report is acquired, it is displayed as follows.

As for the Report Data of the Write Test Point, the background is displayed in light blue.

<table>
<thead>
<tr>
<th>Item</th>
<th>Content of description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Source name in which Write Test Point is inserted.</td>
</tr>
<tr>
<td>Function</td>
<td>Function name in which Write Test Point is inserted.</td>
</tr>
<tr>
<td>Step</td>
<td>Variable identifier specified at manual insertion.</td>
</tr>
<tr>
<td>Value of variable</td>
<td>Written value.</td>
</tr>
</tbody>
</table>

"Value of variable" column of the Test Report is displayed in the writing data of the Write Test Point by the form set for "Consecutive data specification" or "Individual designation" in the Test Report as well as the variable value output. In the writing data of the Write Test Point, it is displayed by character string (ASCII) for "Individual designation".
25. Create Test Result

25.1. About Test Result

In DT10, the Test Result can be brought together as a Test Report based on the acquired Report Data. The outline of the test and the Test Result of going are input to the format of the Test Report prepared with DT10, and the Test Report is made.

(The Test Report is composed of the "Result List" dialog that displays "Test Summary" dialog and result of the test of filling in the outline of the test.)

It prints, and it is output outside, and it is rugged and it is possible to preserve and to open the made Test Report again on DT10.

The Report writer is a print function.

When the Report is made a file, the application such as Adobe Acrobat is separately needed.

The Test Result is displayed in each module. The number of modules becomes the module in the state that nothing is registered and the total number value in all modules becomes 0 for 0, too. Therefore, the number etc. of Test Reports of total Test Points become 0, too and make the Test Report, please after making the module.
25.2. Create Test Result

25.2.1. Selection of Report Data

When [Test Result] → [Create...] of the menu is selected, "Selection of the Report Data that makes the Test Result" dialog is displayed.

In the dialog, the list of the Report Data that the profile is managing now is displayed (* Cleared Report Data etc. are not displayed). When only the Report Data that makes the Test Report applies the check by the dialog, and the "OK" button is clicked, the Test Result is made only by the Report Data to which the check attaches.
25.2.2. Input to the Test Result

The Test Result is displayed on the main screen of DT10.


Moreover, the format of the display of each size of the form when printing can be selected in "Display format". It is possible to select it from "A4 length" and "A4 side" "A3".
"Result List" In the dialog, each Test Result can be displayed.

The judgment item of each Test Result list is "NG" by default.

**Please switch "OK/NG" according to customer's Test Result by double-clicking the corresponding cell.** When the print and the print preview are executed when "Judgment" of "Judgment" of "Judgment" of "TI judgment", "WBT judgment" of the test synthesis result, and the range Test Result and the function execution time test results and the event execution time test results are not all OK, the alert message that informs them that there is NG item is displayed.

Moreover, the pop-up menu is displayed when right-clicking on the Test Result list.
All ranges of the selection are made OK: All the judgment items of the line that has been selected are made OK.

All ranges of the selection are made NG: All the judgment items of the line that has been selected are made NG.

Copy: The line that has been selected is copied onto the click board.

Copy with header: The item name is copied onto the clipboard together.

The following content is described in the Test Result.

- **Test synthesis result**
  - Overall information and the judgment result to test are displayed.
  - In the judgment items, select OK/NG of Test Result that you did, please.

- **C0 Coverage result**
  - The C0 coverage result of each module is displayed.
  - Coverage is calculated by using the data of the Test Point of "Unused", "Unexecute", and "Normality / Illegality" of "Attribute" set in the Step Property, based on the following calculating formula.
    
    $\frac{(Normal + Illegal)}{(Total - Unused - Unexecute - Non-pass)} \times 100[\%]
    $

- **C1 Coverage result**
  - The C1 coverage result of each module is displayed.
  - The item name can be copied together by selecting [the copy with the header].
    
    $\frac{(Execute \ Branch \ route)}{(Number \ of \ Branch \ route)} \times 100 \[\%\]

- **Range test result**
  - The result of each module of the range test of the value of a variable is displayed.
  - "Range rate" is the one that the value of a variable that inserted the Variable Value Point and obtained it output the range of which extent to the range set by "Variable Property setting" of "Function Property" was shown. It is calculated based on the following calculations.
    
    In "Range rate" of "Range test result", the mean value of all the range rates of each module is displayed.
    
    $\frac{Range \ of \ acquisition \ variable \ value (Max \ value - Min \ value)}{setting \ range \ value} \times 100[\%]
    $

  - In the judgment items, select OK/NG of Test Result that you did, please.
Function execution time test result

- The result of each module of the execution time test of the function is displayed.
- "Typ room degree" is the one that how it had room between the mean time for the value of "Typ" set by "Execution time" of the Function Property was shown. It is calculated based on the following calculating formula.
- In "Typ room degree" of "Function execution time test result", the mean value of "Typ room degree" of all functions set in each module is shown.
  
  "Typ setting value" of execution time / average execution time × 100[%]

- "Max room degree" is the one that how it had room between the mean time for the value of "Max" set by "Execution time" of the Function Property was shown. It is calculated based on the following calculating formula.
- In "Max room degree" of "Function execution time test result", the mean value of all functions set in each module is shown.
  
  "Max setting value" of execution time / average execution time × 100[%]

- In the judgment items, select OK/NG of Test Result that you did, please.

Event execution time test result

- The result of each module of the execution time test of the event is displayed.
- "Typ room degree" is the one that how it had room between the mean time for the value of "Typ" set by "Event" of the Function Property was shown. It is calculated based on the following calculating formula.
- In "Typ room degree" of "Event execution time test result", the mean value of "Typ room degree" of all functions set in each module is shown.
  
  "Typ setting value" of Event /Average execution time × 100[%]

- "Max room degree" is the one that how it had room between the mean time for the value of "Max" set by "Event" of the Function Property was shown. It is calculated based on the following calculating formula.
- In "Max room degree" of "Event execution time test result", the mean value of all functions set in each module is shown.
  
  "Max setting value" of Event / Average execution time × 100[%]

- As a result of the test that the customer did, please select OK/NG by the judgment item.
**Memo** About the function not registered in the module

The Test Result list is basically displayed in each module. However, when the function is not registered in the module, it displays it as a function outside the module. In that case, the analytical result of the function outside the module is displayed as a pseudo parent and child module by composing "-- -- of the function outside the module (top-level)→ file name (level 1)→ function name (level 2)".

<table>
<thead>
<tr>
<th>No.</th>
<th>Module name</th>
<th>Total TP number</th>
<th>Unused</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1_src</td>
<td>274</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>&gt; 1_1_Process_Task</td>
<td>274</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>&gt; &gt; 1_1_1_Process_Task</td>
<td>24</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>&gt; &gt; 1_1_2_ProcessCommon</td>
<td>250</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>--Function outside module--</td>
<td>303</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>&gt; ProcessSubPB.c</td>
<td>103</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>&gt; &gt; ProcessSubPbInit</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>&gt; &gt; UnitProcessEJECT</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
25.2.3. Confirmation of Test Results

The image printed by using the print preview before it prints can be confirmed.

When test [Test Result] → [Print preview] of the menu is clicked, the print preview is displayed.

In this case, the format when printing by "Display format" of the Test Summary dialog is revocable
25.2.4. Print the Test Result

The made Test Result can be printed on paper with the printer that ties to the personal computer.

"Print" dialog is displayed when [Test Result] → [Print] of the menu is clicked, and it prints from the dialog.

25.2.5. Save and Open the Test Result

The made Test Result can be preserved.

Because "Preserve it giving a name" dialog opens when [Test Result] → [Save] of the menu is selected, it preserves it from the dialog.

Moreover, because "The file is opened" dialog is displayed when [Test Result] → [Open] of the menu can open the preserved Test Result from this dialog in the preserved Test Result.
26. Setting

26.1. Setting dialog

When [Tool] -> [Setting] of the menu is selected, the following windows are displayed.

![Setting dialog](image)

The environmental set dialog can set the display setting and the analysis setting. Change setting and click [OK] button, then the change is reflected and the dialog is closed. When you click the [Apply] button, the change is reflected with the dialog opened.

However, the menu item of “analysis” is not reflected to the display immediately in the “Apply” button. It will be reflected in following timing.

<table>
<thead>
<tr>
<th>Item name</th>
<th>Timing of the Reflected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze the C1 Coverage at the timing of Report analysis.</td>
<td>The analysis timing of the valid report data.</td>
</tr>
<tr>
<td>Open the Test Report and also analyze the Event Trigger.</td>
<td>The Display timing of the Test Report.</td>
</tr>
<tr>
<td>Use the Event Trigger to task judgement for execution of the function trace analysis.</td>
<td>The analysis timing of the Function Trace Report.</td>
</tr>
<tr>
<td>Considering the filter settings at the timing to analyze of function trace.</td>
<td>The analysis timing of the Function Trace Report.</td>
</tr>
<tr>
<td>Do not display the function that has no valid Test Point to analysis Report.</td>
<td>The analysis timing of the report data.</td>
</tr>
</tbody>
</table>
### 26.1.1. Display – Document Window

**Document format**
- Select the display type of Document window.
- User can select the [Tabbed Document] or [Multi Document].

**View line number in source code window**
- When the check is put, the line number is displayed in the source code window.
- The line mark is a logical line mark (Changing line is considered to be a line).

**TAB stop of source code window**
- The tab position of the source code window is set.
- The range that can be set is 1～8.

**Change Font**
- When "Change font" button is clicked, the font change dialog is displayed.
- When it selects font by this dialog, and the "OK" button is clicked, the font is changed.
- "Style" and "Size" in the dialog are not revocable
- When being start next time, the font setting is succeeded because it is preserved in an environmental configuration file.
- The change in the font is shared in DTMerge / DTAdvisor.
- When you change the font, change the character set to source code window is closed. And when you view the source code window again displays the selected font.
  
  The change in the font is done without the source code window shutting when there is no change in the character set.

| Attention | When DT10 is used for the first time, it is necessary to do "Change font" by Setting dialog when Korean is used for the source code. The garble happens when there is Korean when the source code is displayed on the source code window when the font that doesn’t correspond to Korean is used. |
| Attention | Please do not copy a Korean character string from an external editor etc. in the state before the font is changed to Korean, and do not do the paste to the source code window. Save processing is not normally done. |
26.1.2. Display – Report Data

■ Line number of Report Data
- The method of displaying the line number of the list of the data of the Test Report can be specified.
  
  **Sequential number for the displayed data**
  - It becomes the sequential number for the displayed Report data.
  - In addition, all data's serial number is displayed to ( ).

  **Sequential number for the all data**
  - It becomes the sequential number for all data. If the Report display filter is set, it becomes the number of the tooth omission.

■ Step column display of Report Data
- The display method of the Step column of the list of the data of the Test Report can be specified.
  
  **Step number**
  - It is number given continuously in the function.

  **Step Type**
  - It is displayed in Folder View or Module View, and the character string that shows the kind of the Step number and the Step is displayed.

■ Customizing Test Report View
- The item of the Test Report View can be customized from the following dialogs
  
  - The display and non-display are switched with ON/OFF of the check on the list box.
  - The order of the item can be changed with "Up" and "Down" buttons.
  - In the button of "Width of the selected row (pixel)" , the width of the column is revocable.
  - It is possible to return to initialization with the button of "Return it to an initial value".
26.1.3. Analyze

■ Analyze C1 Coverage in Analysis of Test Report
  • When the Report is analyzed, the C1 coverage is analyzed when the check is put in this item.

■ Analyze Event Trigger when the Test Report is opened
  • When the Test Report is opened when the check is put in this item, the Event Trigger is analyzed, and it outputs to “Event ID” and “Event definition” of the Test Report View.
  • An analytical mode can be selected in the combo box.
    * Trigger Start : It analyzes in the ordinary mode.
    * Trigger End : It analyzes in Trigger End mode.

■ The Event Trigger is used for the task judgment when the Function Trace is analyzed
  • The transition of the task can be displayed by using the Event Trigger by the Function Trace when the check is put in this item.
  • It is possible to display by using the Event Trigger for the task analysis when the Function Trace Report is opened when the Test Report of the Event Data analysis is opened.
  • When “Analyze Event Trigger when the Test Report is opened” item is only turning ON, it becomes effective.

■ When the Function Trace is analyzed, the filter setting is considered
  • When the check is put in this item, the Function Trace can be displayed in consideration of the filter setting.

■ Hide the Function have no enabled Test Point from Analysis Report.
  • When you check this item, the function have no enable Test Point of “Folder View”/”Module View” will not displayed on Coverage Report/Execution Time Report/Period Time Report/Loop Count Report.
  • If there are disable Test Point (example, you have the disable Test Point which don't need check), this feature is effective to check only enable Test Point.
26.1.4. Analog Box

**Setting the scaling of Analog Box**

**Scaling to logic input**
- The scaling setting to the logic input is done.
- "There is not scaling" or "5V" can be selected.
- It becomes possible to set the threshold of the logic input by the scaling value by doing the scaling setting to the logic input in "Logic input Setting" of the Analog Box data acquisition setting dialog.

![Logic Input Setting](image)

**Scaling to AD input**
- The scaling setting to the AD input is done.
- It is possible to select it from each A-Ch1-A-ch2 and "There is not scaling" "200mV" "500mV" "2V" "5V" "12V" "24V".
- The part in doing the scaling setting to the AD input, with the influence is as follows.
  - The "A-Ch" item of the Test Report View is displayed by the scaling value. Moreover, the unit of this "A-Ch" can be changed by the pop-up menu.
  - The filter setting according to the AD input data can be done by the scaling value.
  - The value of the AD input can be confirmed by the scaling value in the Multi Wave Scope. Moreover, the unit of this "A-Ch" can be changed by the pop-up menu.
  - The conditions precedent of the AD input can be set by the scaling value in "Optional specification of the playback" dialog in the Multi Wave Scope.

![Test Report View](image)

Figure: Test Report(Unit change of A-Ch)
The numerical value for each minimum resolution is decided by setting the scaling value. Please note the setting by a high-ranking round-up when you set the numerical value between numerical values for each of the minimum resolution.

<Ex> When you set 2500mV when scaling 5V is set. Because it is a scaling of 5V setting, the numerical value for each minimum resolution is about 19mV. "2500" is treated as 'Fraction' that exists during 2490(127) ~ 2509(128) as a value that can be divided in resolution. In this case, it is set by a high-ranking round-up as 2509(128) mV.

26.1.5. Other

**Other options**

The Test Report that was the opening at the last end when the Project opens is displayed

- If you check this checkbox, when open the project, the Test Report which displayed when the last DT10's booting is displayed.

Display each core load to Performance Monitor.

- Select the number of core of displayed Performance Monitor.
  
The number you can select is 2~4. If you want to display only one core, you need set to off this checkbox.

**DynamicTracer connection test**

- When the connection of DynamicTracer is confirmed, it uses.
26.2. About connection test

26.2.1. Explanation of dialog

**Connection test**
- Connected test of DynamicTracer and the target equipment of the asynchronous bus connection starts.

**Width of address**
- The address bit of the NOR flash memory is specified. The maximum value is 24 bits.
  
  (Ex)

  When the range of byte address is 0x00000000~0x000FFFFF. Value=20

  When the range of byte address is 0x00000000~0x0000FFFF. Value=16
26.2.2. Connected test procedure

In the asynchronous bus connection and asynchronous bus (function call) connection, to confirm whether the connection can be corrected, connected test can be done.

In connected test, it writes aiming at a certain numerical value, and whether DT10 can correctly obtain the data within the fixed time is tested. It is judged that it is not correctly connected with DT10 if it is judged that the connection of DT10 is normal if data can be correctly obtained, and is not possible to obtain it.

There is the following sample sources in [Sample] -> [async_linecheck] in the installation folder. Connected test is done by using this sample source.

```c
/* Copyright (C) 2009, Heartland.Data Co., Ltd. All Rights Reserved. */
/* Author : JLOC */

/* ========= Parameter ========= */
/* Please change the content of this block according to the user environment. */
#define DT_UINT32  
#define DT_NOR_BASE_ADDR 0x80000000
#define DT_NOR_SIZE 0x02000000
#define DT_USE_FUNCTION 0
#define DT_INT_DISABLE() DI
#define DT_INT_RESTORE() EI

/* ========= Don't Change Parameter ========= */
/* Please do not usually change the content of this block. */
static const DT_UINT32 ch_test_res[13] = 
{ 0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
  0x00002B40, 0x5678, 
}; /* Test param */
```
Step1  The function for connected test is added to the target

First of all, to do connected test, the function for connected test is added to the source code.

It registers in [Sample]→[async_linecheck] in the installation folder and the function for connected test of "_DT_ConnectTest()" must be the written sample source, and must register the processing that exists in this sample source in the target environment and register the copy or the source file in the Project of the target.

```c
void _DT_ConnectTest(void)
{
    int i;

    DT_INT_DISABLE();  // Disable interrupt
    for (i = 0; i < 8; i++)
    {
        (volatile unsigned short *)((dt_test_ptr[i] & DT_MAX_ADDR_MASK) + DT_MBR_BASE_ADDR) = 0;
    }
    DT_INT_RESTORE();  // Enable Interrupt
}
```

Moreover, please change the item of /* Parameter*/ in the sample code according to the specification of the target in that case.

Step2  The function for connected test is called.

Please call this function in the place over which it passes without fail only once when starting.

Step3  The outside of the compilation build etc. works.

Please download data to the target environment by compiling the source code, and starting the emulator, and put the target into the state that can be operated.

※DT10 is not used in this work.

Step4  Connected test is executed.

[Tool]→[Setting] of the menu is selected with DT10, and "Environmental setting" dialog is displayed.

[Width of the address] of the dialog is set, and [Connected test] button is clicked.
Step 5  The target environment is actually done in the orchids.
Please start the target environment, and let me pass the function processing for connected test.

Step 6  The result of connected test is confirmed
The following dialogs are displayed when a normal connection can be confirmed, and select [Close], please.
When a normal connection cannot be confirmed within a given time period, it becomes "Time-out".
27. Display the Report Data

27.1. About the management of Report Data

In DT10, as a specification of the Report Data management, Test Report data and the source file for the test and the Test Point setting content were assumed to be a set and the history was managed.

The set of the Test Report and the Test Point setting corresponding to the source file and this of the same version is called “Profile”.

27.2. Structure of Profile management

The Profile of each source file version that is the set of the Test Report, the Test Point setting, and the source file is made as a folder in the folder where the DT10 Project file is stored.

It files in the Profile folder by the maintenance of the source folder and the Test Point setting content that backs up the source file at the Report Data folder and that time that stores collected the Test Reports, and it is stored to file it and by the maintenance of administrative information in the Profile.

When the Test Report collection is executed after it is changed for the source file composition to be tested by registering and correcting the source file, the Profile folder is made. When the composition of the source file is not changed, the Test Report data is added to an existing Profile folder.

When the Test Report of a past Profile with a different source composition is displayed, DT10 correctly displays the state of the function and the Step in the Test Report window referring to the source file and the Test Point setting content in a specified Profile.
27.3. Renewal of Profile

In "Specify Profile to store Test Report" of "Test Report collection condition setting" window that displayed by selecting [Test] -> [Set collecting condition of Test Report] of the menu, the Profile that preserves the Test Report data can be confirmed. Moreover, the change in the place where the Profile is kept is also possible.

When the Test Report collection is executed for the first time after the source file is updated, it is displayed in the column of "Profile that preserves the Test Report data" of figure above, "< new profile >". In that case, when clicking [Execute...] button, or selecting [Test] -> [Execute] of the menu, the Profile name is fixed. The Profile name at this time can be confirmed in "Execute program" window.

The Profile name is displayed as "Profile to store Test Report" in "Execute program" window.
A present source file and Test Point information are preserved in the Profile folder that is the Test Report preservation ahead by clicking [Execute] button when executing it for the first time by a present source composition.

When the source file is updated after the Test Reports are collected and the Report collection is executed again, a new Profile is made.

Delete Test Point

Test Report collection condition setting

It becomes New Profile.
27.4. Management of Report Data

Because collected the Test Reports of each Profile are managed, the Profile and the Test Report are selected in "Display of the Report Data" window displayed by [Display Test Report...] command of [Report analysis] menu.

Moreover, the deletion of the Profile and the reference and the Test Report can be deleted in "Display of the Report Data" window.

Delete Report Data

The deletion button of the Report Data clearly deletes the Test Report selected in the Report Data list.

The difference between clearness and the deletion is as follows.

Clear: The Test Report of the object is excluded from the list of the Test Report that DT10 manages. In this case, the substance of the Test Report data remains in the Profile folder.

Delete: The substance of the Test Report data is completely deleted.

When [Delete Report Data] button is clicked, the following windows are displayed.

The deletion of the Test Report is executed by clicking the clearness of the Test Report and [Delete] button in the click of [Clear] button. Nothing is executed and it returns to “Display of the Report Data” window when [Cancel] button is clicked.

![Profile selection combo box](image)

![Display list of Test Report contained in the selected profile.](image)

![Test item names can be edited by double-clicking any field test items.](image)

![Delete Report Data](image)

![Restructure Report Data List](image)

![Show](image)

![Cancel](image)
Restructure Report Data List

In the selection screen of the Report Data, it changed so that the list was made in the descending order at the date of the Report Data when [Restructure Report Data List] was executed.

The determining criterion of the Report Data at the date is chosen in order of

① ‘dvt’ file name (Repo_xxxxxx_yyyyyy.dvt)

② Preservation folder name when ‘dvt’ file doesn’t exist (Repo_xxxxxx_yyyyyy)

③ Report data (‘dat’ file) date and time of creation when preservation folder name has been changed by another name

Attention

Please do not change the name of the Report Data that DT10 generates.
It is not possible to be able to secure it at the execution date when changed.

Delete this Profile

The Profile that has been selected in the Profile selection combo box is clearly deleted.

The difference between clearness and the deletion is as follows.

**Clear:** The Profile of the object is excluded from the list of the Profile that DT10 manages. In this case, the substance of the Profile remains.

**Delete:** The substance of the Profile folder is completely deleted.

When [Delete this Profile] button is clicked, the following windows are displayed.

The deletion of the Profile is executed by clicking the clearness of the Profile and [Delete] button in the click of [Clear] button. Nothing is executed and it returns to “Display of the Report Data” window when [Cancel] button is clicked.
Reference to Profile

From the Profile selection combo box [Reference] The Profile not included in the Profile selection combo box can be added by selecting.

The profile that doesn't exist in the profile list is selected (The profile folder is selected).

The Profile selected in the folder selection dialog window is added to the last position of the Profile selection combo box in "Display of the Report Data” window (Ahead of <Reference>).

This function uses the Profile excluded from the Profile list clearly of the Profile to list to return it.

The error message was displayed, and it corresponded in such a case to return to the latest Profile environment because it did not correctly open it when the Profile folder where 'pprj' did not exist was selected.

Attention

If you read the DT project made before DT10 Ver4.11, Profile Folder have to be same folder saved DT project.

If you have used DT10 before Ver4.11, and if you have not saved Profile Folder at same folder saved DT project, you have to check your folder hierarchy.
### 27.5. Displays Test Report

When the Test Report of a past Profile is selected in "Displays of the Report Data" window and [Display] button is clicked, DT10 displays the Test Report referring to the source file and the Test Point setting included in the Profile of the object.

The background of Folder View/Module View is displayed in the gray, and it becomes a change prohibition while the Test Report of a past Profile is displayed. At this time, the setting of the Test Property and insertion of the Test Point/deletion cannot be executed basically.

However, the set value used for the range judgment in the test Property value can be changed. The judgment of the designed value can be confirmed by changing the Property value in a past Test Report, and analyzing the Report.

Concretely, the Property value of the following items can be changed.

<table>
<thead>
<tr>
<th>Function property</th>
<th>Step property</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Argument (※)</td>
<td>• Variable (※)</td>
</tr>
<tr>
<td>• Period</td>
<td>• Period</td>
</tr>
<tr>
<td>• Event</td>
<td>• Event</td>
</tr>
<tr>
<td>• Execution time</td>
<td>• Loop Count</td>
</tr>
</tbody>
</table>

※ "Reference", "Level", and "Number of bytes" items where rewriting the Test Point is generated by the change in the setting are not revocable.
To return from this state to the latest environment, [Close Report Data] command of [File] menu is executed.

Attention

Please change to a name similar also as for the PPRJ file name in the Profile folder when you change the Profile folder name. Please note becoming the error when the Test Report is displayed if the folder name of the Profile is not corresponding to the PPRJ file name.
27.6. Rollback of Project

The Rollback of the Project is a function to replace the content included in a past Profile with the latest test environment. If [Rollback] command of [File] menu is executed while displayed, the content being displayed now is replaced as the latest environment, and insertion, the deletion of the Test Point, and the setting of the Test Property become possible.
28. Project is shared by the Import Function

28.1. Sharing of Import Function with another Project

In the development of software, when two or more development teams share the same source code by a misappropriation of the source code of a different product (in-house code etc. with historical earnings), and a large-scale Project, the thing shared (succession) is possible with a present Project. The source file, the Test Point of other Projects, and Test Property information

In this application, this function is called 「Import function」

28.2. Outline of Import Function

28.2.1. Thing that Import Function enables

The following operations become possible by using the Import function.

- Misappropriation of source (And, Test Point and Test Property) used in a different development Project. (Succession of Property)

- Sharing of the same source file in team development. (Synchronization of Property)

※As for rprj, only information on a pertinent file is succeeded
28.2.2. About information that the Import is not done

- The base address is set again according to the administrative information of a new Project.
- As for the header file, the reproduction is done.
- When both files do not become complete according to the import timing, the event setting (When you Step over the file) is released.

<Ex> When you set the event from [Test Point a] in [Test Point b] in [File A] to [File B] (The file is Stepped over.)

- It is import only as for [A] ⇒ Event setting release of [a]→[b]
- It is import only as for [B] ⇒ Event setting release of [a]→[b]
- It is import as for [B] after only [A] does the import. ⇒ Event setting release of [a]→[b]
- It is this timing and import as for [A] and [B] ⇒ The event setting of [a]→[b] is succeeded to

<Example> When you set the event from [Test Point a] in [File A] to [Test Point b] in [File B] (The file is not stepped over.)

- Only [A] is succeeded to and the event setting of import ⇒ [aa]→[ab] is succeeded to.

28.2.3. About the error in Import

Import is discontinued when it puts on the import processing, and the following events occur, and it returns to the state before it processes it.

- When the error occurs by opening or reading an import former Project
- When the substance of the source file registered in an import former Project is not found
- When failing in the copy of the source file

Attention: The import function is a function to misappropriate and to share information in the Project file used before (Or, in the parallel). It is not a function to generate Test Point information from the Test Point statement character string in the source file.
28.3. Import for succession of information

When the Test Point is buried under the registered source file when the source file is registered by making and the Project setting of a new Project, the dialog to do import is displayed.

The source file that wants to do import is selected, and an import former Project can be registered to a pertinent file by selecting the import former Project file from a [Specified of Project].

Only the source file that registers an import former Project executes import. After all the Test Points that exist in the source file are deleted once, the source file (The Project name is an empty column) to which an import former Project is not registered is buried automatically (*).

- When “The Test Point automatic operation insertion is not executed” is selected, only the deletion of the Test Point is executed.

When the import origin is registered and the [OK] button is executed, the succession of the source file registration and Test Point information (The Test Property contains it) is executed to the file that sets the import origin.
※When no file is selected, cannot execute “Specify Project”

■ To the Project file when information on a pertinent source file is not found

The following error messages come out, and confirm whether there is in the file that selects or registers a correct Project file the mistake, please

※When two or more pertinent files exist, two or more file names are displayed.

■ When you want to clear the import former Project setting

Please select the source file that wants to clear the setting to change the import former Project setting and click a clear button.

The setting of an import former Project of the object file is deleted.

■ When canceling

When [Cancel] clicks on a button on “Import of Test Point information” dialog screen when the Project is newly made, the Project to which any source file is not registered is made to cancel the registration of the source file. Please execute “Import” and “Registration of the source file” if necessary.
28.4. Import for sharing of information

When it runs parallel to the Project (henceforth [Proj-A]) that is progressing now, an advanced Project (henceforth [Proj-B]) exists, and the same source file is shared between Projects, the Test Point and Property information inserted with [Proj-B] can be shared with [Proj-A]. (The opposite is also possible.) After the Project file is selected, the dialog of the import object source file selection is displayed when [File]→[Import] is executed from the menu.

The object source file is selected, whether the source file is copied (Where do you copy it in that case?) or the file passing that the import origin uses is used as it is selected, [OK] is clicked, and the import processing is executed.

When "Copy specified folder as a root" is checked when import is copied, it copies in consideration of the place from the root folder and the place from a specified folder.

The file of the same name cannot be selected because all selected files are copied onto the specified folder when not copying it onto the tree structure even by the one with another full path. Moreover, the root folder of the DT Project of the import origin is made a starting point, the tree structures is molded, and select the one in the route of the Project of the import origin about the selected file, please when copying it onto the tree structure.
A specified source file is copied onto the following folders

![Diagram showing source code sharing between Proj-A and Proj-B]

The object source file of [Proj-A] is copied onto an appropriate position in the area of the Project by using the import command to share the source code of "Shared library etc." changed in [Proj-A] (Test Point Property information) with [Proj-B], and information is succeeded.

A specified source file is used as it is.

Here is selected when the source file that shares information is in the same area (folder) in each Project.

![Diagram showing reference between Proj-A and Proj-B]

Information on [Proj-A] is importing done because the difference arises in information on "Shared library etc." seen with [Proj-B] when the source code of "Shared library etc." (Test Point Property information) was changed in [Proj-A] and the intelligence sharing is aimed at.
Writing confirmation after file is copied
When the same name file exists at the copy destination when the source file is copied by the import processing, the processing selection dialog is displayed.

![Select the import process](image)

- **Overwrite**
  The file is copied in the superscription.

- **Overwrite all file**
  The superscription copy is executed without displaying the dialog when two or more object files have been selected or the superscription confirmation is similarly generated by filing following.

- **Copy and rename**
  When copying it, the file name is changed and the copy is executed.
28.5. Import for adjustment of information

The following dialogs are displayed when the change takes place in the situation when the registered file is preserved last time and the adjustment of information (correspondence) is aimed at in the place where the Project file is opened.

■ Import Test Point information

The following dialog is displayed, and "Import to succeed information" is executed.

Please refer to [28.3. Import for succession of information] for the usage of the dialog.

■ OK

Processing to which the location information of the Test Point is updated is opened to all the corresponding files and the Project is opened with the correspondence of information on doing (The re-header file generation is executed when there is a difference in header file information) and the Project file and the source file was able to be taken.

Please select here when in another Project where the source file is shared, there is a change (The function was added on an external editor) into which the Test Point Property was not changed.

■ Cancel

Nothing is done, and the Project is opened as it is. Naturally, when such a file is opened because the source file with not correct location information of the Test Point etc. exist, the processing of "Update of the location information in the source" and "The header file is generated again" is generated.
29. Save the Report in CSV by Export Function

In DT10, various Reports can be preserved in the file by Comma Separated Value. The thing of this function is called Export.

[File] -> [Export] of the menu can be selected and the exporting Report be selected from among selection [Report Data], [Coverage Report], [Execution Time Report], [Period Time Report] and [Loop Count Report].


When the Exporting Report is selected, the following dialogs are displayed.

In this dialog, "Preserve it giving a name" dialog is displayed when setting [Export] clicks on a button, and the Test Point setting such as "Data for the output", "Order of the item", and "Output form" can be preserved by Comma Separated Value.
Output target data

- Whether all items of the Test Report are output or only the item displayed on the window now is output is specified.
- Even if the filter is set when "All data is output" is specified, all items of the Test Report displayed now are output.
- When "Only the displayed data is output" is specified, only the item being displayed in the Test Report now is output. When the filter setting is done, only the item displayed in the Report Data is output.
- Only the item that has been selected in the test when "Only the data that has been selected now is output" is specified is output.

Output form

- The text form of the CSV file is specified.
- The field district switching off specifies the delimiter between fields on the CSV record.
- The text district switching off specifies whether to bundle the field by (double quotation ").
- The character-code specifies the type of the output character.
- The line feed code specifies the line-feed character.
- The header specifies whether to export the item name.
**Order item**

- The field for the output is specified the order of the field outputting it by Comma Separated Value.
- The record of CSV is output in order displayed in the list.
- When the order is changed, the item in the list is selected, and [Up] [Down] clicks on a button.
- When an arbitrary item is not output to CSV, the item is selected, and the button [Not output] is clicked or the check is removed clicking the check box displayed in the left of the item name in the list.
- The order of the item is returned to default with the button [Set to init].
- The button [Set to custom] is returned to the value set by “Customize of the Test Report view” of an environmental, set dialog.

**Export**

- When [Export] clicks on a button, the file selection dialog of the output file specification is displayed.
- When the preservation button is clicked specifying the file name, the export processing is executed.
30. Use DTPlanner

30.1. Outline of DTPlanner

DTPlanner offers the Test Point and various test properties set now and the function that the list display can be done by the folder hierarchy and the module hierarchy expression and an incomplete setting of the Test Point and the range specification value etc. be confirmed easily is offered.

Moreover, it becomes possible to plan the best Test Point setting by being able to confirm the presence of the frequency and the range error that passed each Test Point from the analytical result of the Test Report, and repeating the collection of the Test Points and the confirmation by DTPlanner.

30.2. Main window

When [Tool] → [Open DTPlanner] of the menu of DT10 is selected, the following windows are displayed.

The upper row of the main screen displays the Test Point list by the folder hierarchy and the Test Point list by the module hierarchy is displayed in the lower by the table form.
30.2.1. Folder hierarchical form

- **Proj/Src/Func/Step**
  - The folder layered structure of the DT Project is displayed by the tree form as well as the Folder View of DT10.
  - The pop-up menu is displayed by the mouse's right-clicking on each item. All items below the selection item opens when [Open all items contained in this] by which the pop-up menu opens everything included in this item is selected. Moreover, when [Close all items contained in this] that shuts all items included in this item is selected, all items below the selection item are shut.

- **TP**
  - It is displayed to be valid and to invalid of the Test Point in the check box.
  - The check box is displayed by the source file, the function, and each item of the Step.
  - When the check box does mouse's left-click, the state is revocable.

- **Function**
  - The total of the function that is valid and included in the source file only in the item of the source file is displayed.

- **Step**
  - The number of total Steps that is valid and included in the source file and the function only in the item of the source file and the function is displayed.

- **SourceSteps**
  - The source code line number is displayed.
  - It is valid in the item of the source file and the item of the function in the folder hierarchy. It is valid in the item of the module and the item of the function in the module hierarchy.

- **Attribute**
  - A valid, and normality system and abnormality system set content is displayed only by the function and the item of the Step (It is not displayed in the variable dump Step). This cell can display the drop down list by double-clicking the left mouse button, and change the setting.
Execution attribute

- The content of the execution attribute set (Normal, Unused, Unexecute, and Non-pass) is validly displayed only by the function and the item of the Step. (It is not displayed in the variable dump Step).
  
  This cell can display the drop down list by double-clicking the left mouse button, and change the setting. A pertinent line is displayed in orange when judged passing over the Step of "Unused, Unexecute, and Non-pass" as a result of the Test Report analysis.

Description

- The content of "Description" column set to the Property in the Project, the source file, the function, and the Step (The variable dump Step is excluded) is displayed. This cell can edit it by double-clicking the left mouse button.

Description 2 (Comment)

- Display the contents of the column of "description 2 (comment)" which set in Step Property.
  
  Double-click your mouse's left button and you can edit this cell.

Variable

- It is valid only in the function and the item of the variable dump Step. The following content is reflected respectively.
  
  - The function item: The number of variable dump Test Points (Val attribute) set in the function is displayed.
  
  - For the variable dump Step: The variable identifier is displayed.

Reference

- Only the variable dump step item is effective, and whether the variable is direct reference (Val) or indirection reference (Ptr) is shown.

Level

- Only the item of the variable dump step is effective, and the reference level is shown when the variable is indirection reference (Ptr).

Bytes

- The number of bytes for the variable is validly displayed only by the item of the variable dump Step.

Signed

- Only the variable dump step item is effective, and whether the "Signed" is "On" or "Off" are shown.
■ Period

- When a Period set Property turns on, “*” is valid only displayed by the item of the function and the Step (The variable dump Step is excluded).
- The background color of the cell is displayed in yellow in turning on the Period setting but the item.

■ Event

- When it is valid, and set the Event Property only by the item of the function and the Step (The variable dump Step is excluded), a set number is displayed.
- When the Event Property is not set, the background color of the cell is displayed in yellow.

■ Time

- When it is valid, and set the Execution time Property only by the item of the function, a set number is displayed.
- When the Execution time Property doesn't exist, the background color of the cell is displayed in red.

■ Count

- A set number of execution frequency properties is valid only displayed by the item of the Step (The variable dump Step is excluded).
- When the execution frequency Property is not set, the background color of the cell is displayed in yellow.

■ Pass count

- The frequency that passed each Test Point when the Report Data is valid only analyzed by the function and the item of the Step is displayed.
- It agrees to the frequency that passed over the Step of the FuncIn attribute for the function.

■ Range error

- If there is a result of becoming outside the set value range of the Property set to each Test Point when the Report Data is valid only analyzed by the function and the item of the Step, it displays in background color red of the cell, and the type of the Property of which it made an error is shown respectively by combining the following characters.
  - P; It makes an error outside the range in a Period set Property
  - E; It makes an error outside the range in the event Property
  - T; It makes an error outside the range in the Execution time Property
  - C; It makes an error outside the range in the loop count Property
  - V; It makes an error outside the range in the variable dump Step.
30.2.2. Module hierarchical form

The display of a module hierarchical form of the function and the item of the Step is equal to a folder hierarchical form.
The content displayed only by a module hierarchical form is described as follows.

■ Type
  - The processing system that is valid, and set to the module only by the item of the module is displayed.
  - This cell can display the drop down list by double-clicking the left mouse button, and change a set content.

■ Task
  - The task that is valid, and set to the module by the item of the module is displayed.
  - This cell can edit it by double-clicking the left mouse button.

■ Priority
  - The priority that is valid, and set to the module by the item of the module is displayed.
  - This cell can edit it by double-clicking the left mouse button.

30.3. Menu

30.3.1. File

■ Export
  - The Test Point setting of Project being displayed now content is preserved in the file by Comma Separated Value.

■ Exit
  - The window of DTPlanner is closed.

30.3.2. Tool

■ Setting
  - Setting dialog is displayed.
  - In Setting dialog, the display of the list form can be customized.

■ View
  ▼ Status bar
    - The ON/OFF switch of the display of the status bar under the main window can be done.

  ▼ Application view
    - The design of the window is revocable.
30.4. Edit of Property value

The item of “TP” and “Attribute”, “Unused”, “Unexecuted”, “Description” and “Execute attribute” can change the setting of the Property on the main window of DTPlanner.

When the setting of the Property of other items is changed, the setting can be changed from the Property Editor because the Property Editor of DT10 can be opened when double-clicking it with the left mouse button on the column of the display of each item name or the Property setting (frequency at the variable, the number of bytes, the Period, the event, and time).

It double-clicks on the item.

The Property Editor of DT10 is opened.
DTPlanner cannot be operated while the Property Editor of DT10 is being opened. The window of DTPlanner is displayed as follows, and the key input and the input of the mouse become invalid the other day.

The operation is an invalid state

The following messages are displayed at the center of the window

During collecting Report Data and editing Property Editor, DTPlanner cannot be used

It automatically returns to the Planner screen when the Property Editor of DT10 is shut, and the invalid state is released.
30.5. Export

In DTPlanner, the Test Point setting of Project being displayed now content can be preserved in the file by Comma Separated Value. The thing of this function is called "Export".

When [File] → [Export] → [CSV format.] of the menu is clicked, the following dialogs are displayed. "Preserve it giving a name" dialog is displayed when the setting and the export buttons such as "Sort by data", "Output target data", "Order item", "Output format", and [Export] are clicked by this dialog and the Test Point setting can be preserved by Comma Separated Value.

![Image of Export to CSV format dialog]

Export to CSV format dialog:
- Sorted by data
- Output in the order of the folder hierarchy
- Output target data:
  - Output all data
  - Output only shown data
- Output all data selected currently
- Order item
- Output format:
  - Field Separator: Comma
  - Delimited Text: Unicode
  - Character: CRLF
  - Line feed: Non Header
- Output Property
  - Output Variable, Period, Event, Execution Time, and Property value for Function
  - All the data included in the selection item is output
- Export
- Cancel
■ Sorted by data

- Whether data is output in order displayed in the upper row in the window (folder hierarchy) or data is output in order displayed in the window the lower (module hierarchy) is specified.
- When the hierarchical order of the folder is specified, the line only of the folder is not output. The source file is output in the full path.

■ Output target data

- Whether all items in the Project are output or only the item that is the selective state on the window now is output is specified.
- When "Output only shown data" is specified, the selection of the item of the content of the display of the form specified by “Sorted by data" in the preceding clause (If it is a window upper row, and a modular form in case of the folder form, it is window the lower) is applied.
- When "Output all data" is checked, selected items output all the child items to other items if it is a parent item. (In the state to remove this check, only the selected item is output. The selection item is output in this case in the folder form specification and nothing is output only by the folder item. )

■ Order item

- The field for the output is specified the order of the field outputting it by Comma Separated Value.
- The record of CSV is output in order displayed in the list.
- When the order is changed, the item in the list is selected, and [Up] [Down] clicks on a button.
- When an arbitrary item is [Not output] to CSV, the item is selected, and the button not output is clicked or the check is removed clicking the check box displayed in the left of the item name in the list.
- It is possible to return to initialization with the button of [Set to init].
Output form

- The text form of the CSV file is specified.
- The field district switching off specifies the delimiter between fields on the CSV record.

![Field Separator and Delimited Text](image1)

- The text district switching off specifies whether to bundle the field by (double quotation ").

![Delimited Text and Character](image2)

- The character-code specifies the type of the output character.

![Character and Line feed](image3)

- The line feed code specifies the line-feed character.

![Line feed and Header](image4)

- The header specifies whether to export the item name.
■ Output Property

- When a set value of various properties set to the function and the Step is output, it checks it.
- The output of the Property value is always output by the record format of a Test Point, minimum value, the maximum value, Typ, and the unexpected value in fixation regardless of the item order setting of the export dialog. (There is a difference in the output of peculiar information depending on the Property type of the setting of the variable setting and the Period etc. Peculiar information is output from the minimum value field to previous.)
- When the Property output is specified, the record of all Property value included from the following line to the item to which the record of function information or Step information according to the item order setting is output is output.

■ Inform

- When [Export] clicks on a button, the file selection dialog of the output file specification is displayed.
- When the preservation button is clicked specifying the file name, the export processing is executed.
30.6. Setting

The following dialogs open when [Tool] -> [Setting] of the menu is selected.

In this dialog, order of the column display and display/non-display specification of the list and the width specification of the column can be done.
30.6.1. Order of column display and display/non-display specification of list

The column is displayed from the left to the right in order displayed in the item name list. When the order of the display is changed, the item name is selected, and [Up] and [Down] button are clicked.

When non-display/display is switched, the item is selected, [View][Hide] button is clicked or the check box displayed in the left of the item is clicked.

When the column display is set, and [OK] is clicked, it is reflected in the main screen.

① Order of the display and display/non-display of the column is set.

② [OK] is clicked

③ The display setting of the column is changed.

![Settings window](image)
30.6.2. Width specification of column

When the column width is changed, the numerical value is input directly to "Width of the selected column Pixels" in Set dialog by the pixel value.

When the column width is set, and [OK] is clicked, it is reflected in the main screen.
30.7. DTPlanner-automatic operation close

The window of DTPlanner closes by the automatic operation when the following operating it with DT10.

- End of DT10
- Creating of new Project
- Open the Project
- Setting change of Project
- Automatic insertion
- Deletion of batch of Test Point
- Import
31. DTMerge Function

31.1. Outline of DTMerge Function

If the embed of the Test Point macro statement in the target source file for the Test Point that Project file (*.rprj) maintains is not completely corresponding, the collection and the analysis of the Report Data cannot be correctly executed in DT10.

For instance, when it cannot help deleting the Test Point by using an external editor, and the Test Point being deleted by the convenience of the management of the source code, the Project file cannot be continuously used under such a condition.

Then, the **DTMerge function** is used.

In DTMerge, the modified date of registration Source is checked when reading a Project file. And if there are differences, compare backup source and insert Test Point to the range there is no difference. In the range there are differences, Test Point information of DT Project is deleted, and insert Test Point Automatically.

In DT10, There is a backup function of source files are subjected to the Test Point setting. Therefore, the exact differences between the lasted sources (Corresponding source to Test Point setting) are acquired when starting DT10 if the source file is updated with.

The backup saves in the **[DT_XXXX (DT Project name)]** folder that created in the folder saved DT Project file.

The source is backed up according to the following timing.

- New
- Setting Project
- Insert Test Point automatically
- Delete Test Point by the batch
- Insert Test Point manually
- Delete Test Point manually
- Edit Source file
- Import
- Merge when the starting

When no backup file, the lasted sources are executed Test Point insertion automatically, and compare created Test Point and Test Point of DT Project, and copy of Property only as for the one corresponding (function and Step attribute), Test Property setting can be used continually.

In this case, The Test Point inserted and deleted manually need to insert and delete by manual merge function.
31.2. DTMerge usage example

<Example> For the operation method of not leaving Test Point (TP) on the configuration management tool.

The macro statement of the Test Point is deleted from the source code by "Test Point information on the project is not deleted" (the reference [7.3.2. Delete Test Point by the batch]) after operation is confirmed with DT10, and the check-in is done. Afterwards, DTMerge starts when the DT Project file is opened when making a change to the source, and the Test Point is inserted.

Server

Check Out the source.
Change the source.

Check In the file.

Check Out the source.
Change the source.

Check In the file.

Check operation using DT10

.C  rprj

• Insert TP to source.
• Create Profile.

.C  rprj

• After check, TP is deleted
without deleting Test Point information...

.C  rprj

• Open the file saved by another name.
• DTMerge is operated, and insert TP.

.C  rprj

• After check, TP is deleted
without deleting Test Point information...

Copyright © 2009-2012 Heartland.Data Co., Ltd.
31.3. Operational procedure

1. Project file (*.rprj) is opened by starting DT10.

2. The automatic merging function operates, the latest source is compared with the content of the backup source, and the Test Point is inserted.

When all existing Test Points can be allocated only by the Automatic Merge function (The change in the one and the source where last time and the Test Point composition do not change (※) is only an addition and the deletions, etc. of each function), the following messages can be displayed, and it shift to the manual merge function (It is a reorganization collection as for the automatic merge result). It returns to DT10 when not shifting to the manual merge function (the case of clicking OK button).

![Test Point correction completion]

Attention

When DTMerge function use, it may become an unintended result of Test Point insertion. Please click “The content is confirmed” button, and check the result of Test Point insertion.

3. If configuration changes in the function statement, the Merge function is operated manually because the agreement between Projects cannot be taken by auto Merge function.

The manual Merge function can select insertion and the deletion of no Project agreement with Test Point information in the source in the following windows.

![Manual Merge function window]
4. The object source of the merging work is selected in "Source file for processing" window. The list of the source with a different Test Point setting is displayed in this window. If the file name is double-clicked, it is reflected in the main screen, and it becomes possible for the source to do the merging setting.

- About the source that specifies the merging setting, *** is displayed in the file name, and it is shown that it is a merge only application object.
- The merging setting specified to the source that "Cancellation of the change" is selected by the menu displayed by the mouse's right-clicking on the merging setting and specification ending source is annulled everything, and removes *** display.
- When returning to DT10 about the one that the merging setting was not specified by the source displayed in the source list, the state of an old Test Point setting is maintained. (In this case, the source file cannot be opened with DT10)
5. The Test Point that merges it by the TP view is selected.

- In the TP view (project), Test Point information that exists in the specified Project is displayed by the tree view form (The route of the tree is a source file).
- It is shown that it is a Test Point to which it has not been fixed to merge it when the color of the Test Point is a black. It is shown that it is a Test Point to which it has already been fixed to merge it when the color of the Test Point is a gray.
- When the Test Point is double-clicked, the Property of the Test Point is displayed.
- The pop-up menu is displayed by right-clicking. (A similar operation is possible from "Merging setting" of the menu.)

**Auto merge**: An automatic merging is executed for the item that has been selected now.
(When the item that has been selected is only a source, and a function, it is valid.)

**Delete**: The item that has been selected now is deleted.

**Cancel delete**: The item that has been selected now is processed.

- The mark of the icon has the following meanings

<table>
<thead>
<tr>
<th>Icon</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>🗔️</td>
<td>Annulled Test Point</td>
</tr>
<tr>
<td>🛠️</td>
<td>Test Point of merging indetermination</td>
</tr>
<tr>
<td>🧵</td>
<td>Test Point of Property indetermination When the TP specified in property is deleted, this mark is shown.</td>
</tr>
</tbody>
</table>
6. The Test Point is inserted in the source code window.

```java
switch( UnitStatus->CurLoadState )
{
    case Unit_INITIAL_STATE : // 0x00
        _DLTestPoint(__DLfunc_ProcessSubPb, __DLStep_1);
    break;

    case Unit__INITIAL_STATE : // 0x01
        _DLTestPoint(__DLfunc_ProcessSubPb, __DLStep_2);
    break;

    case Unit__INITIAL_STATE : // 0x02
        _DLTestPoint(__DLfunc_ProcessSubPb, __DLStep_3);
    break;
}
```

- The content of the source file has been selected by the TP view is displayed.
- In the TP view (project), Test Point of unspecified marge is selecting, it right-clicks in the Test Point insertion line, and click "Insert the selected Test Point" command. Then macro statements are inserted in source core, and add Test Point at the line of new Project. (Test property value and Step attribute are inherited.)

7. The Test Point inserted in the TP view (source) by [6.] is displayed.

- In TP view (source), it is displayed Test Point info existing in source file by tree view form.
- Operation of window and display are the same as TP view (project).
- When Test Point is selecting in TP view (source), right-click and select “delete”, and the icon become ☒.
  In this state, merge function uses, and the TP is deleted from DTProject.
8. “Execute Merge and return to DT10” of the file menu is selected, and merging is executed.

The merge only application is not executed though a current operation does a relation of the Test Point between Projects and unnecessary Test Point deletion specification. When DTMerge ends, the merge only application is executed.

![DT10 User's Manual](image)

**Execute Merge and return to DT10**

: DTMerge is ended according to present Test Point allocation specification and the deletion specification after the merge only application that executes the merge only application ends and it returns to DT10. DT10 opens the new Project made by merging.

**Cancel the merge setting and return to DT10**

: DTMerge is ended without executing merging, and the control is returned to DT10. DT10 opens an old Project before merging is executed.

### 31.4. Output window

The processing result to various operations and operation is displayed in the output window.

![Output window](image)

The content of the display can be preserved in the text file as a log of the merge only application.

*The content of the output window is preserved in the file* of the file menu is selected.
32. DTAdvisor function

32.1. Outline of DTAdvisor function

DTAdvisor passes a normal route to program in visually showing the flow of the processing of the target as a result (Test Report) by the execution of the target program actually and is executed or is a function to judge whether to go as for an abnormal route.

Making a more appropriate test specification is facilitated from the insertion of the Test Point immediately after the introduction and the state of the setting of the Test Property the Test Report of the program in the target Project.

32.2. Usage example of DTAdvisor

DTAdvisor has the following functions.

- Function to detect short processing at execution time intervals extremely, and for Test Point with possibility to influence operation of program to investigate.
- Function to make dynamic module that suits real operation of function besides definition of module of DT10 Project, and to facilitate analysis of program based on real operation DT10.
- Function made problem and characteristic of execution easy to understand by displaying execution situation of function with different route that passed at the same time

A dynamic module that relates the module in which DT10 is defined statically to the real operation of the function by DTAdvisor and is defined and is made is said, "Dynamic Module". DTAdvisor is valid in the case enumerated as follows.

(Ex 1) When the error goes out as a result of the Test Analysis for the designed value set in the Test Property of DT10.
⇒ The Test Point where the error is detected is confirmed by "Error Test Point list" of DTAdvisor.

"Step Tracer" is opened, and it is specified that the cause of the error is normal by comparing the execution result in abnormal circumstances with operation.

(Ex 2) Confirmation of execution time intervals
⇒ After the Test Report is acquired, DTAdvisor is started. "Check on the Test Point at execution intervals" function of DTAdvisor is executed, the Test Point with the doubt that influences the behavior of the program is extracted, and the Test Point is invalidated. The Test Report is acquired with DT10 in the state, and whether the problem was solved is confirmed.

(Ex 3) Confirmation of execution route that uses Dynamic Module
A Dynamic Module is made from the processing executed by using "The execution situation of the function is displayed" function on the route by starting DTAdvisor. And, whether it is executed is confirmed as intended when the operation of the program on the target designs.
32.3. Main screen

When it is selected to [Tool] -> [Execute DTAdvisor] of the menu, the following screens are displayed.

DT10 cannot be operated while using DTAdvisor.

When the Test Report is not displayed, DTAdvisor cannot be used.

When DTAdvisor is used displaying a past Test Report, a part of function (function to change the Project) cannot be used.

Folder View

- It is a window where the source file in the Project is displayed by the tree view.
- The tree is displayed similarly to the folder view of DT10 in order of [Folder, File, Function, Step] function Step".
- Effective/invalidity of the Test Point is revocable in the check box.

(※If [Return to DT10] of the file menu is not executed when the setting of the check box is changed, the change is not reflected in the main body of DT10. When return to DT10 with [Return to DT10] command, DT Project is not saved. You have to save it.)
- The Step over which it doesn't pass is displayed in the gray.
- The pop-up menu is displayed when each item is selected, and it right-clicks.
  "Find this Test Report", "Add to the Report View filter condition", "Open Function execution situation", "Open Function History List", and "Open Step Tracer" can be executed.
- Two or more condition items can be selected while pushing the Shift key or the Ctrl key and, then, plurals of the item is selected. It is convenient to set the filter by the batch.
Module View

- It is a window where the function in the Project is displayed by the tree view.
- The tree is displayed similarly to the module view of DT10 in order of [Module, Function, Step]
- Effective/invalidity of the Test Point is revocable in the check box.
  (※If [Return to DT10] of the file menu is not executed when the setting of the check box is changed, the change is not reflected in the main body of DT10. When return to DT10 with [Return to DT10] command, DT Project is not saved. You have to save it.)
- The Step over which it doesn't pass is displayed in the gray.
- The pop-up menu is displayed when each item is selected, and it right-clicks.
  "Find this Test Report", "Add to the Report View filter condition", "Open Function execution situation", "Open Function History List", and "Open Step Tracer" can be executed.
- Two or more condition items can be selected while pushing the Shift key or the Ctrl key and, then, plurals of the item is selected. It is convenient to set the filter by the batch.

Dynamic Module View

- It is a window where a Dynamic Module is displayed by the tree view.
- The tree is displayed in order of [Dynamic Module, Function].
- Effective/invalidity of the Test Point is revocable in the check box.
  (※If [Return to DT10] of the file menu is not executed when the setting of the check box is changed, the change is not reflected in the main body of DT10. When return to DT10 with [Return to DT10] command, DT Project is not saved. You have to save it.)
- The pop-up menu is displayed when each item is selected, and it right-clicks.
- The function item can be added from the Folder View/Module View by drag & drop.
  The added function can do "Adds as a child function", and does "Adds as a parents function" or select the item that drops by "Setting of Dynamic Module" dialog.
Test Report List

- The Test Report data is displayed.
- The following line that has been selected now can be selected with [F10] key. Moreover, the previous line is selected with [Shift + F10] key. The Test Report View synchronizes with the Folder View, the Module View, and the Source Code Window.

Execution Time Report

- The Execution Time Report is displayed.

Execution interval warning TP Report

- The Report Data interval is analyzed, and the list of the part of the interval of less than 1ms is displayed.
- It synchronizes with the Test Report View and the Source Code Window when an arbitrary line is double-clicked and displays it.

Error Test Point list

- When there is a Test Point that is the error for the Test Property, it is displayed in this list when as a result of the Report analysis.
- When there is no Test Point that is the error, this list is not displayed when as a result of the Report analysis.
32.4. Menu

32.4.1. File

- **Return to DT10**
  - The window of DTAdvisor is shut, and it returns to DT10.
  - The content that is edited and set with DTAdvisor begins to appear doing this operation and is reflected in the Project on the DT10 side.
  - At this time, the opened Report Data is shut in DT10, and it enters the state when the Project is opened.
  - When return to DT10 with [Return to DT10] command, DT Project is not saved. You have to save it.

32.4.2. Edit

- **To the next execution interval warning TP**
  - It jumps to the appearing warning point next to present location the selection line in the Test Report when "Check on the Test Point at execution intervals" is executed.

- **To the previous execution interval warning TP**
  - It jumps to the appearing warning point previous to present location the selection line in the Test Report when "Check on the Test Point at execution intervals" is executed.

- **Disable TP with execution interval warning**
  - "Check at Test Point Execution Intervals" is executed, and the detected Test Point is invalidated.
    - **Selected warning TP**
      - The Test Point of the line that has been selected in "Warning TP Report of the execution interval" window is invalidated, and the header file is made again.
    - **All warning TP**
      - All the Test Points displayed in "Warning TP Report of the execution interval" window are invalidated, and header file is made again.

- **To the next execution Step**
  - The selection line of the Test Report is moved to the following line.

- **To the previous execution Step**
  - The selection line of the Test Report is moved to the previous line.

- **To the next execution Step all together**
  - Other Step Tracer windows and content are matched, and the selection line of the Report list is moved to the next line.
  - It is displayed only when the Step Tracer window is active.
■ To the previous execution Step all together
  • Other Step Tracer windows and content are matched, and the selection line of the Report list is moved to the previous line.
  • It is displayed only when the Step Tracer window is active.

■ Set the position to divide Step Tracer View
  • It is displayed only when the Step Tracer window is active.
  • The position where the Step Tracer is divided the display is changed. The following commands can be selected.
    Show the next divide position
    Show the previous divide position
    Show the divide position of maximum execution time
    Show the divide position of minimum execution time
    Specify the divide position to display

■ Change the position of execution situation display
  • It is displayed only when the execution situation display window is active.
  • The position where the execution situation display is divided is changed. The following commands can be selected.
    Show the next same function
    Show the previous same function
    Show the maximum time position of the same function
    Show the minimum time position of the same function
    Specify the execution position to display

■ Filter setting Test Report..
  • A detailed set dialog to display the Test Report is displayed.
32.4.3. Report Analysis

■ Check Execution Interval of TPs
  • The execution interval of the Test Point is checked, and the illegal location is detected.

■ Clear the check result Execution Interval of TPs
  • The content of an Execution interval warning TP Report displayed by executing "Check at Test Point execution intervals" function is cleared.

■ Open the function execution situation
  • The execution situation of the function displayed with the Execution Time Report now is displayed.
  • The route displayed first is the first route in all execution routes.

■ Open Step Tracer
  • It displays it by dividing the Report based on the function or the Step that has been selected now.
  • The pattern of division is as follows.
    - Divide by the Period
    - Divide by the Time
    - Divide by the Event

32.4.4. Tool

■ setting
  • An environmental set dialog is displayed.

■ Show
  ▼ Docking Window
  • The ON/OFF switch of the display of each window can be done.
  ▼ Status Bar
  • The ON/OFF switch of the display of the status bar under the main window can be done.

■ Line of MDI window
  • The MDI window displayed in DTAdvisor is made to line up.
  • It is possible to select it from "Cascade", "Tile Horizontally", "Tile Vertically", "Arrange Icons", and "Close all".
32.5. Step Tracer

The Test Report data is divided, and the function to display the execution route of each unit of division tracing is said, "Step Tracer".
The Step Tracer can confirm the Report data list by specifying the range of division, and dividing an arbitrary function.

32.5.1. Display and fundamental motion of Step Tracer

An arbitrary function is selected on the execution time, and the pop-up menu is opened by right-clicking, and "The Step Tracer is opened" is selected.
The range of division can be selected from "Divide by the Period", "Divide by the Time", and "Divide by the Event".

The division route displayed first by the Step Tracer displays the first route when all routes manse.
The division position can be changed divided the display from [Edit] -> [Set the position to divide Step Tracer View] of menu.
A revocable position is as follows.

**Set the position to divide Step Tracer View**

- Show the previous divide position (Shift+F7)
- Show the next divide position (Shift+F8)
- Show the divide position of maximum execution time (F5)
- Show the divide position of minimum execution time (F6)
- Specify the divide position to display

Moreover, it is possible to move to the next Step with the [F10] key and to move to the previous Step with [Shift + F10] key.

Each route is displayed because the Step Tracer is an independent window respectively and it is also possible to compare it.

Specific of the cause of the error becomes easy by comparing a normal execution route to the route where the error occurs with the Step Tracer.

The execution route of two or more Step Tracers can be synchronously confirmed by [Edit]→[To the next execution Step all together (Ctrl+F10)] or [To the previous execution Step all together (Ctrl+Shift+F10)] of the menu when two or more Step Tracers are open.
32.5.2. Dynamic Module Tree part

Moreover, when a Dynamic Module is defined in the Dynamic Module tree part, and the Step Tracer is opened, the tree and the number of passages of a Dynamic Module are displayed.

In the Step Tracer, the Playback function is provided.

The route displayed to select the execution speed by the toolbar, and to start with the button by the Step Tracer is matched at the elapsed time and it traces it automatically.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>![PlayButton]</td>
<td>The Playback is begun at the time located while displaying it with the Step Tracer now when clicking on a button. It stops when clicking on the button again.</td>
</tr>
<tr>
<td>![StopButton]</td>
<td>The located time is moved to the head of data.</td>
</tr>
<tr>
<td>![StepButton]</td>
<td>It locates and in is moved at the end of data.</td>
</tr>
<tr>
<td>![NormalButton]</td>
<td>The rollback speed can be selected. It is possible to select it from &quot;Normal&quot; &quot;1/2&quot; &quot;1/10&quot;.</td>
</tr>
</tbody>
</table>

Moreover, the frequency that passed is counted when the button is clicked and turns on when the Playback and the Step are executed, and "Pass Count" item of the Dynamic Module part is updated.

The count is cleared with the button.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>![PassCountButton]</td>
<td>The frequency that passed when this icon is turned ON is counted.</td>
</tr>
</tbody>
</table>
32.6. Check function of Test Point at execution intervals

The Test Point (Report data) that seems that the function of "Check at execution intervals of Test Point" influences defective operation of the program it is detected, and displays it in the list.

To do "Check on the Test Point at execution intervals", when [Report analysis] -> [Check Execution Interval of TPs] of the menu is selected, the check is executed. When a warning Test Point is detected as a result of checking it, the list of the warning Test Point is displayed in "Execution interval warning TP Report".

This warning Test Point is a part of less than time and a continuous frequency from which the execution time difference from the front Report Data is specified in the Report Data. This warning Test Point is specified by "Setting dialog".

A pertinent line and the source code window of the Test Report view synchronize when an arbitrary line is double-clicked in Execution interval warning TP Report and it is displayed.

Moreover, can it right-click on Execution interval warning TP Report, and the warning Test Point be invalidated by the pop-up menu.

"Selected warning TP " or " All warning TP " can be selected.

- **Selected warning TP**
  - The Test Point that has been selected is invalidated, and the header file is made again.

- **All warning TP**
  - All the Test Points displayed in the Report are invalidated and the header file is made again.

On the Test Report view, when warning TP is detected, a pertinent line is displayed by the color.

It is possible to move from the pop-up menu to the warning point by right-clicking in the Test Report at this time. (The Folder View and the Source Code Window synchronize, too and it is displayed.)

"To the next execution interval warning TP " or " To the previous execution interval warning TP " can be selected.

- **To the next execution interval warning TP**
  - It moves to the next warning point.

- **To the previous execution interval warning TP**
  - It moves to the previous warning point.
32.7. Execute situation display function of Function

32.7.1. Execute situation display of function

All routes of the executed function (only by one hierarchy) are displayed in the function to which it pays attention, and the function being executed by the route of the function now is displayed in green in “Execution situation display” function.

When "Execution situation display" function of this function selects the function in the Execution Time Report, and selects “Show the function execution situation” of the pop-up menu displayed by right-clicking, the execution situation of the function is displayed.

At this time, the displayed route is the first route in all routes.

The display position is revocable from [Edit] → [Change the position of execution situation display] of the menu. The display position that can be selected can be selected from “Show the next same function ”, “Show the previous same function ”, “Show the maximum time position of the same function ”, “Show the minimum time position of the same function ”, and “Specify the execution position to display ”.

Moreover, because the function execution situation display of each route can be displayed in the window, the difference of the behavior of the function by different pathways can be confirmed.
### Origin of call of function

- The list of the function where the function that pays attention is called is displayed.
- The origin of call of the execution route assumed to be a display object is displayed in green now.

### Now played i and call stack of function

- Information on the function that pays attention (execution time, belonging module, and task, etc.) is displayed and the call stack in a present route is displayed.

### Function executed from function to which it pays attention

- The list of the function (Only by one hierarchy) executed in the function that pays attention is displayed.
- The origin of call of the execution route assumed to be a display object is displayed in green now.
- When button in the lower right of each function name is clicked, the starting position in the execution section is revocable.
  - It jumps to the starting position in the previous execution section of the execution sequence chart.
  - It jumps to the starting position in the next execution section of the execution sequence chart.

### Execution sequence chart

- A horizontal axis is made a time axis, and the execution sequence chart of the function executed from the function that pays attention is displayed.
- It changes into the display of the route within the range if the execution display part is double-clicked.
- A present time position is displayed in a blue line.
- It is displayed that the step execution and the Report Data of the Test Report are double-clicked with the check on “Synchronize with double-clicking and the step execution of the Report Data List.”(※Please see 【32.9. Setting】 for details of the item.) is turned ON in “Function execution situation display” tab of setting dialog. “Function execution situation” is displayed, and when Report Data is double-clicked, it is displayed synchronizing with the execution sequence chart of the function execution situation.
32.7.2. Setting of the Dynamic Module

"Execution situation display of the function" function simply displays the change of the function in the Report Data in the chart. Therefore, it is not possible to distinguish by the function execution situation display when there is function call (FuncIn) the execution route’s suddenly changing by the time slice in the Multithreading ahead.

For this case, the Step of the Test Report can be executed with the F10 key with "Execution situation display of the function" displayed.

When a certain function is obviously judged that it not is a thread switch, and a simple function call from the display of the source and the Report by the Step execution, a Dynamic Module is set in the property of the function of the object. A Dynamic Module is grouping that groups the relationship between a child and his parents to the lamplight and the settlement as a result of actually executing the target program (Test Report), and judges.

On the figure that is shown the call stack, and it is displayed popup menu.

In the menu, "Define the Dynamic Module so this function to be root" command is selected, and “Definition of a Dynamic Module” dialog is displayed.

In the dialog, the Dynamic Module name is inputted.

![Definition of Dynamic Module](image)

- **Dynamic Module name**
  - It is a dynamic module name.

- **The function not called in a present execution route is set.**
  - When the check is turned on, and a Dynamic Module is set, all the child functions in a present execution route without are registered in a Dynamic Module.

- **Refer to a present definition situation**
  - When "Refer to a present definition situation" button is clicked, the definition situation of a Dynamic Module is displayed under the definition dialog of a Dynamic Module.
  - The display is shut with "Hide the Dynamic Module View" button while displaying the definition situation of a Dynamic Module.

- **OK**
  - A Dynamic Module is registered when clicking on a button.
32.8. Function History List

The Function History List is tabulation of the number of passages of each execution route of the Dynamic Module definition function.

To open Function History List, it right-clicks first by Execution Time Report or Error TP List and Folder View/Module View and the pop-up menu is displayed. And [Open Function History List] of the menu is selected and the range of division is specified.

When the range of division is specified, the following Function History List is displayed.

How many degrees of the defining function of a Dynamic Module were passed at each division position (No. of the division position) is displayed in the Function History List.

The division position displayed with buttons on the toolbar is revocable.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>◀</td>
<td>The first division position is displayed.</td>
</tr>
<tr>
<td>▶</td>
<td>The last division position is displayed.</td>
</tr>
<tr>
<td>◄</td>
<td>The previous division position is displayed.</td>
</tr>
<tr>
<td>►</td>
<td>The next division position is displayed.</td>
</tr>
</tbody>
</table>
32.9. Setting

When [Tool] → [Setting] of the menu is selected, an environmental setting dialog is displayed.

The setting of the Source Code Window and the display item etc. of the Test Report can be selected.

![Setting dialog]

- **General**
  - **Window Style**
    - The display form of the source code window can be selected from "Document with the tab" or "Multi document".
  - **Scaling of AD input**
    - The analog box data can be displayed by the scaling value.

- **Test Report**
  - **Line number of Report Data list**
    - The display of the line number of the Report Data list can be selected from "Data for the display is reaming watched" or "All data is reaming watched".
  - **Display the Step field**
    - User can select display of the Report data list of the Step field from [Step number] or [Step type].
  - **Change the list alignment, show or hide the item**
    - The order of the row of the Test Report, the display, non-display, and the width of the item are revocable.
■ Execution interval warning TP

▼ Set judgment of Execution Interval

・The value of an execution interval of "Check on the Test Point at execution intervals" function abnormal judgment (time and continuous frequency) is specified.

▼ Change the list alignment, show or hide the item

・You can change the list alignment, show or hide the item.

■ Execution Time Report

▼ Change the list alignment, show or hide the item

・You can change the list alignment, show or hide the item.

■ Error Test Point list

▼ Change the list alignment, show or hide the item

・You can change the list alignment, show or hide the item.

■ Source code window

▼ Show the line number

・When the check box is turned on, the line number is displayed in the source code window

▼ Number of characters of TAB stops

・The tab position of the source code window is set.

▼ Synchronizes with double-clicking and the step execution of the Report Data List

・When the check box is turned ON, the source code window is opened synchronizing with double-clicking of the Test Report and step execution.
## Function execution situation display

▼Other window synchronizes when becoming active window.
- It synchronizes with other windows when "Function execution situation display" window becomes active when the check box is turned ON.

▼Synchronizes with double-clicking and the step execution of the Report Data List
- It synchronizes with the function execution situation display according to double-clicking of the Test Report and the step execution when the check box is turned ON.
- When the check box is turned on, the function execution situation display can specify the operation when the function changes synchronizing with the movement of the Test Report.

### The display function is not changed
- Even when the function changes, the display function of the function execution situation display is not changed.

### The display function is changed
- When the function changes, the display function of the function execution situation display displays a changeable function.

### Displays in another window
- When the function changes, the function execution situation display is displayed in another window.
33. Multi-core functions

33.1. Outline

DT10 can trace the Multi Core (Up to 4 cores) Target's operation.

After get Test Report as Multi Core Target, You can use following specific core operation.

- Identification core in a one Test Report.
- Display the Test Report of each core.
- Display the CPU Load Rate of each core. (Performance Monitor)
- Setting for Filter of Test Report on condition that the core.

33.2. The data format of Driver Functions

In case that Target CPU is Multi Core, and when output the data by DT10's Driver Function 【_TP_BusOut function】,

The Core number indicator is set and output at the upper two bits of address data's 20 bit.

If you use the GPIO/SPI connection, The Output is set MSB First Output in the order of dat, addr.

```
16 bit
  dat
20 bit
  addr
```

Upper 2 bit of addr data is Core number.

dat= Argument “dat” of _TP_BusOut function
addr= Argument “addr” of _TP_BusOut function
33.3. Get Test Report

To use DT10’s Multi Core functions, you need to check the checkbox which [Target CPU : Multi Core] in [Execute program] dialog.

You can get following Test Report.

You can see the core which got Report Data belongs at the item "Core" in Test Report View.

Marker’s colors identify the core.  **Core0: Blue**  **Core1: Green**  **Core2: Yellow**  **Core3: Red**
33.4. Display the Test Report each core

"Cores' Test Report" is the Report displayed each core's Test Report which has more than one core information. You can use it to trace specific core operation.


You can see Core0 Test Report ~ Core3 Test Report by Test Report's core information.

(*Core Test Report can display if Test Report include Core information.)

Each Core Test Report can display with Test Report at the same time.

If you enter [F10] key on the Test Report, Core Test Report is synchronized with the Report. It is effective to check the route.
33.5. Performance Monitor of each core

You can see each core's CPU Load Rate at Performance Monitor.

DT10 can get Test Report of up to 4 cores. And Performance Monitor can display some Core's Load Rate (up to 4 cores). Select [Tool] -> [Setting] to display [Setting dialog], if you want to see Performance Monitor of each core.

Check [Display each core load to Performance Monitor.] in Setting dialog, and select the number of Target core.

You can see following display at Performance Monitor.

※ This image is case of 4 core.
33.6. Setting Filter condition the core

Set a Test Report filter condition the core.

To setting the Filter at Test Report, select [Report Analysis] -> [Set the View filter of Test Report], and then you can see [A detailed set dialog to display the Test Report]. check the item “Core select” and double-click it and you can select core which you want to filtering in the [Test Report display setting: Core item] dialog.

[Core select] in Filter setting is applied at main Test Report only. It is not applied Core Test Report. It is applied Core Test Report in case the filter setting is not [Core select].
34. Source code analysis function

34.1. Outline

User can analyze and display the problems of source code by using source code analysis function of the "klocwork INSIGHT" from DT10.

Insert Test Point to the point where is warned by static analysis, and analyze it in dynamic analysis of DT10.

To use this function, you can detect problems more efficiently.

Attention

If you want to use static analysis in DT10, it need that the "klocwork INSIGHT" is introduced in development process of target project.

34.2. To Specify the Local Project Folder

Specify the Local Project Folder generated by "klocwork INSIGHT".

Local project folder is named ".kwlp" or "~.kwlp", in usually.

User can specify the local project folder in [Local project folder of "klocwork INSIGHT"] which is the item of [new making wizard] dialog (first page of new making wizard).
After generating DT10 project, users can also setting of local project folder of "klocwork INSIGHT" by the first page of the [Project setting dialog] (select the [File]→[Project setting...] in Menu)
34.3. Execution of Source code Analysis

To execute source code analysis, click [Tool]→[Analyze source code with "klocwork INSIGHT"] of Menu.

In this case, the source code which is targeted in the local project of "klocwork INSIGHT" is analyzed. (Source file which is not registered to DT project is also targeted to analysis.)

When select the source file user wants to analyze in folder view, select [Analyze source code with "klocwork INSIGHT"] in right-click menu, and the selected source file's analysis is executed. (Analysis target is the selected source file only.)

Following window is displayed during analyzing.

During display the past Profile, User cannot to analyze source code with the "klocwork INSIGHT".
At the end of analysis, the result of the analysis is displayed as [Source code analysis Report] window.

![Source code analysis Report](image)

Double-Click the displayed message, and user can jump to appropriate line of the source code.

When analyzing the source code, if there are source codes which is not registered to DT project or its path is changed, following message is displayed, and those source codes is displayed in gray background in [Source code analysis Report] window.

![Message](image)

### 34.4. Limitations

When moved the local project of "klocwork INSIGHT", user should re-execute analysis of the local project by "klocwork INSIGHT". If user does not re-execute the analyzing of the local folder, DT10 can not correctly display for the result of the local project analysis.
35. Shortcut key list

35.1. Shortcut key on source code window

■ Shortcut key concerning source code edit

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+C</td>
<td>Copy</td>
</tr>
<tr>
<td>Ctrl+X</td>
<td>Cut</td>
</tr>
<tr>
<td>Ctrl+V</td>
<td>Paste</td>
</tr>
<tr>
<td>Ctrl+A</td>
<td>All select</td>
</tr>
<tr>
<td>Ctrl+Enter</td>
<td>Changing line is added to the cursor position.</td>
</tr>
<tr>
<td>Ctrl+BS</td>
<td>When the cursor has come to the head of line, the previous changing line is deleted</td>
</tr>
<tr>
<td>Ctrl+Delete</td>
<td>When the cursor has come to the head of line, the next changing line is deleted</td>
</tr>
</tbody>
</table>

■ Shortcut key that can be used only in edit mode

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESC</td>
<td>The edit is canceled, and returns at the time of the edit.</td>
</tr>
<tr>
<td>Ctrl+Z</td>
<td>Return</td>
</tr>
<tr>
<td>Ctrl+Y</td>
<td>Repeat</td>
</tr>
</tbody>
</table>

■ Shortcut key concerning jump

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+G</td>
<td>Jumps to specified position</td>
</tr>
<tr>
<td>Ctrl+P</td>
<td>Jumps to the head of the previous function.</td>
</tr>
<tr>
<td>Ctrl+K</td>
<td>Jumps to the head of the current function.</td>
</tr>
<tr>
<td>Ctrl+N</td>
<td>Jumps to the head of the next function.</td>
</tr>
<tr>
<td>Ctrl+U</td>
<td>Jumps to the previous Test Point.</td>
</tr>
<tr>
<td>Ctrl+D</td>
<td>Jumps to the next Test Point.</td>
</tr>
</tbody>
</table>

■ Other shortcut key

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+F</td>
<td>The search dialog is opened.</td>
</tr>
<tr>
<td>F3</td>
<td>Retrieves to directions of easy flow.</td>
</tr>
<tr>
<td>Shift+F3</td>
<td>Retrieves in the opposite direction.</td>
</tr>
<tr>
<td>Ctrl+W</td>
<td>The source code window that has been selected now is shut</td>
</tr>
<tr>
<td>Ctrl+I</td>
<td>The pop-up menu is opened at the position with the cursor now. When two or more lines are selected, the pop-up menu is opened in the first line.</td>
</tr>
</tbody>
</table>
### 35.2. Shortcut key on Test Report View

#### Valid shortcut key for display of Test Report

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F10</td>
<td>The following line that has been selected now is selected.</td>
</tr>
<tr>
<td>Shift+F10</td>
<td>The line before it is selected now is selected.</td>
</tr>
</tbody>
</table>

#### The shortcut key enabled when title bar of Test Report View is selected color.

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3</td>
<td>It retrieves to direction.</td>
</tr>
<tr>
<td>Shift+F3</td>
<td>It retrieves it in the opposite direction.</td>
</tr>
<tr>
<td>Ctrl+S</td>
<td>&quot;Retrieval of the Test Report&quot; dialog can be opened.</td>
</tr>
<tr>
<td>Ctrl+F3</td>
<td>It is search in conjunction with Folder view and Source code window.</td>
</tr>
<tr>
<td>Shift+Ctrl+F3</td>
<td>It is backward search in conjunction with Folder view and Source code window.</td>
</tr>
</tbody>
</table>

### 35.3. Shortcut key on the Function Trace Report

#### Shortcut key

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F7</td>
<td>The previous Step of present location is retrieved and selects it.</td>
</tr>
<tr>
<td>F8</td>
<td>The next Step of present location is retrieved and it selects it.</td>
</tr>
<tr>
<td>Shift+F7</td>
<td>In the same Task, in the same Nest, go to previous Step.</td>
</tr>
<tr>
<td>Shift+F8</td>
<td>In the same Task, in the same Nest, go to next Step.</td>
</tr>
</tbody>
</table>

### 35.4. Shortcut key on Event Trace

#### Shortcut key

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F7</td>
<td>Jumps one to the previous point about the point that has been selected.</td>
</tr>
<tr>
<td>F8</td>
<td>Jumps one to the next point about the point that has been selected.</td>
</tr>
<tr>
<td>Shift+F7</td>
<td>Jumps to the previous ID change point of one of the point that has been selected.</td>
</tr>
<tr>
<td>Shift+F8</td>
<td>Jumps to the next ID change point of one of the point that has been selected.</td>
</tr>
</tbody>
</table>
35.5. Shortcut key on DTAdvisor

**Shortcut key concerning Test Report**

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F10</td>
<td>The next line that has been selected now is selected.</td>
</tr>
<tr>
<td>Shift+F10</td>
<td>The previous line that has been selected now is selected.</td>
</tr>
</tbody>
</table>

**Shortcut key concerning Step Tracer**

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F10</td>
<td>The next line from which the execution route of the Step Tracer has been selected now is selected. It synchronizes with the Test Report /source code window /Folder View.</td>
</tr>
<tr>
<td>Shift+F10</td>
<td>The previous line from which the execution route of the Step Tracer has been selected now is selected. It synchronizes with the Test Report /source code window /Folder View.</td>
</tr>
<tr>
<td>Ctrl+F10</td>
<td>The next line that is synchronized with when two or more Step Tracers are opened and has selected now is selected. It synchronizes with the Test Report /source code window /Folder View.</td>
</tr>
<tr>
<td>Ctrl+Shift+F10</td>
<td>The previous line that is synchronized with when two or more Step Tracers are opened and has selected now is selected. It synchronizes with the Test Report /source code window /Folder View.</td>
</tr>
<tr>
<td>Shift+F7</td>
<td>The previous division position is displayed.</td>
</tr>
<tr>
<td>Shift+F8</td>
<td>The next division position is displayed.</td>
</tr>
<tr>
<td>F5</td>
<td>The execution time displays the position where the maximum value is divided.</td>
</tr>
<tr>
<td>F6</td>
<td>The execution time displays the position where the minimum value is divided.</td>
</tr>
</tbody>
</table>

**Shortcut key concerning Execution interval warning TP Report**

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F7</td>
<td>The previous Warning Point in of the Test Report (Point judged that the execution interval is abnormal) is selected. It synchronizes with the source code window /Folder View.</td>
</tr>
<tr>
<td>F8</td>
<td>The next Warning Point in of the Test Report (Point judged that the execution interval is abnormal) is selected. It synchronizes with the source code window /Folder View.</td>
</tr>
</tbody>
</table>
Shortcut key concerning execution situation display window

<table>
<thead>
<tr>
<th>Key operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift + F7</td>
<td>The previous same name function execution position is displayed.</td>
</tr>
<tr>
<td>Shift + F8</td>
<td>The next same name function execution position is displayed.</td>
</tr>
<tr>
<td>F5</td>
<td>The position of the maximum is displayed.</td>
</tr>
<tr>
<td>F6</td>
<td>The position of the minimum is displayed.</td>
</tr>
</tbody>
</table>
36. Function addition item list (ver7.01)

36.1. Correction of already-known problem

- Connection when SD / IF, to fix the problem of missing data occur in the specific equipment.
- Fixed an issue that Fast Test Report is not activated when it is set as Start/Stop trigger.
- Fixed an issue that the colored range of Test Report set in Detail Time List is not match with actual range.
- Fixed an issue that Fast Test Point is not activated when it is set as Restart Option of "Ignore the collected data Mode".
- Fixed an issue that the Attribute/Execution Attribute setting in function property can not be changed in property editor while combined report is displayed.
- In functionality combined with WATCHPOINT, Fixed some internal mismatch.
- In functionality combined with WATCHPOINT, when user set the extremely small value to the timeout value, force-quit is caused.
- Fixed an issue that Combo Box is displayed as Attribute item in Step Property/Function Property item on Property Editor.
- Fixed an issue that an error occurs that the search results by mark or mark type is first data.
- Fixed an issue that the colored range of Test Report set in Error range list is not match with actual range.
- Fixed the problem that even after DTMerge execute, DTMerge would execute again.
- Fixed an issue that "Coverage of variable property setting" item is not enabled in Step Property.
## 37. Change tracking

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Content of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012.05.22</td>
<td>V7.00</td>
<td>・Reform of user's manual.</td>
</tr>
<tr>
<td>2012.06.22</td>
<td>V7.01</td>
<td>・Function addition item list (ver7.01) are added【36. Function addition item list (ver</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.01) (p371)】</td>
</tr>
<tr>
<td></td>
<td></td>
<td>・Correction of typographical errors.</td>
</tr>
</tbody>
</table>