

DT-TREx ver.2.0.0

DT-TRExは、動的テストツールDT10 / DT-Winのプラグインとして動作するツールで、2つのテストレポートを比較表示します。

インストール方法

任意のフォルダにてsetup.exeを実行してください。インストールが完了すると、DT10、DT-WinのそれぞれのメニューにDT-TRExプラグインが起動できるメニューが追加されます。

尚、プラグインを起動するためには、以下のバージョンが必要です。プラグインのインストールより前にインストールすることをおすすめします。

- DT10 ver.12.1.0以降
- DT-Win ver.3.1.0以降
- DT10 Automotive Edition ver.12.0.0以降

注意

プラグインをインストールするとき、あらかじめDT10 / DT-Winアプリケーションは終了させてください。DT10 / DT-Winを実行中にプラグインをインストールすると、ツールメニューに登録されません。

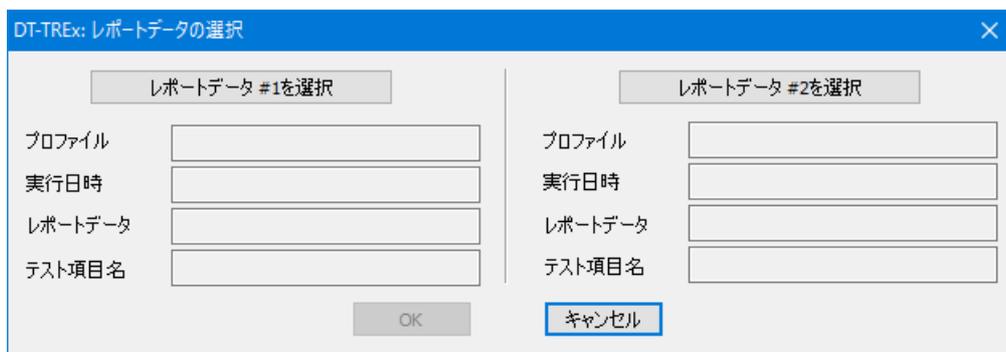
プラグインをインストールしてもDT10 / DT-Winの「ツール」メニューに「DT-TREx」が表示されない場合は、後述の「プラグインの手動登録方法」に従い、コマンドを登録することで、お使いいただけます。

起動方法

DT10 / DT-Winで、プロジェクトを開いている状態で、「ツール」から「DT-TREx」を選択し起動してください。

操作方法

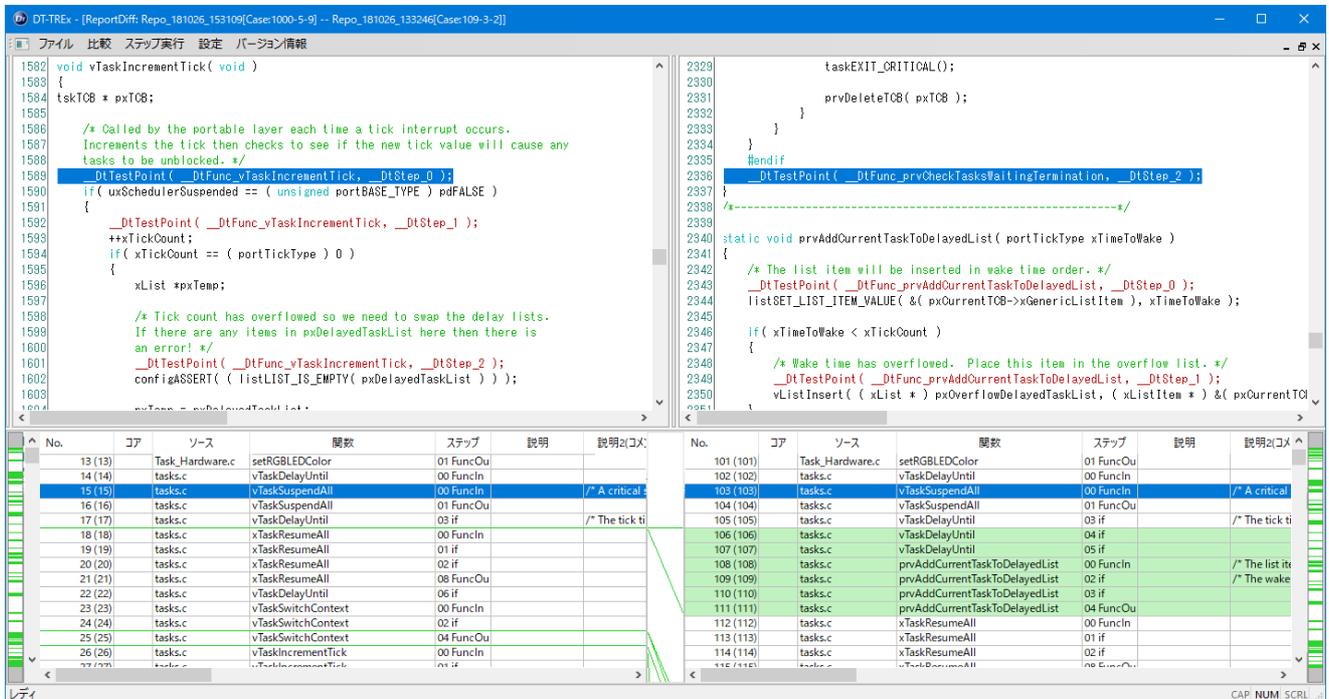
DT-TRExが起動すると、「レポートデータの選択」ダイアログが開きます。



「レポートデータ#1を選択」、「レポートデータ#2を選択」ボタンをクリックして比較したいレポートデータを選択してください。

「OK」ボタンをクリックすることで、各レポートデータの解析を実行します。

解析が終了すると、レポートデータ比較ウィンドウが表示されます。



レポートデータ比較ウィンドウには、選択したレポートデータ同士の差異、レポートに対応するソースが表示されます。

左右で差異があるレポートデータ行は緑色の背景とリンク線で強調表示されます。

レポートデータの比較は、「比較」メニューの「レポートデータの比較」からも実行できます。

レポートデータリストは、左右が同期して表示されます。

「ステップ実行」メニューの「次の実行ステップへ」(またはF10)、「前の実行ステップへ」(またはShift+F10)で、選択行が1行ずつ移動します。

また、「ステップ実行」メニューの「次の差分位置へ」(またはF8)、「前の差分位置へ」(またはF7)で、選択行が次または前の差分範囲の先頭行に移動します。

DT10/DT-Winのテストレポートフィルタと同等のフィルタ機能により、レポートデータ行の表示/非表示を切り替えることができます。フィルタ設定を有効にした場合、フィルタを適用した状態のレポートデータを比較します。

フィルタ設定は、DT-TRExでのみ適用され、プロファイルごとに設定・保存されます。

「設定」メニューの「フィルタ設定」からプロファイル別に表示されるメニューを選択すると、フィルタ設定のウィンドウが表示されます。

フィルタ設定を編集した後は、「比較」メニューの「レポートデータの比較」を選択して、改めて差分の解析を行う必要があります。

ソースファイル	関数	#1. Prof_181026_093316 / Repo_181026_153109 [Case:1000-5-9]	#2. Prof_181026_093316 / Repo_181026_133246 [Case:109-3-2]
DisplayRendererc	DRenderer_BeginScene	平均実行時間：10,249,740ns / 最大実行時間：10,249,750ns / 最小実行時間：10,249,740ns [実行回数：10]	平均実行時間：10,249,730ns / 最大実行時間：10,249,750ns / 最小実行時間：10,249,730ns [実行回数：10]
DisplayRendererc	DRenderer_EndScene	平均実行時間：60,150ns / 最大実行時間：60,160ns / 最小実行時間：60,150ns [実行回数：11]	平均実行時間：60,150ns / 最大実行時間：60,160ns / 最小実行時間：60,150ns [実行回数：11]
DisplayRendererc	DRenderer_DrawDeviceFormatIma	平均実行時間：25,150ns / 最大実行時間：31,340ns / 最小実行時間：17,000ns [実行回数：44]	平均実行時間：24,770ns / 最大実行時間：36,480ns / 最小実行時間：14,490ns [実行回数：44]
DisplayRendererc	DRenderer_Present	平均実行時間：10,833,670ns / 最大実行時間：10,836,910ns / 最小実行時間：10,827,890ns [実行回数：10]	平均実行時間：10,830,410ns / 最大実行時間：10,836,590ns / 最小実行時間：10,827,890ns [実行回数：10]
Task_Hardware.c	getDistance	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：32]	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：32]
Task_Hardware.c	getDMSVoltage	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：11]	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：11]
Task_Hardware.c	setLEDState	平均実行時間：14,400ns / 最大実行時間：14,400ns / 最小実行時間：14,400ns [実行回数：11]	平均実行時間：14,390ns / 最大実行時間：14,400ns / 最小実行時間：14,390ns [実行回数：11]
Task_Hardware.c	setBuzzerPitch	平均実行時間：28,890ns / 最大実行時間：28,900ns / 最小実行時間：28,890ns [実行回数：10]	平均実行時間：28,890ns / 最大実行時間：28,900ns / 最小実行時間：28,890ns [実行回数：10]
Task_Hardware.c	getColorVolume	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：219]	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：219]
Task_Hardware.c	setRGBLEDCol	平均実行時間：14,490ns / 最大実行時間：14,500ns / 最小実行時間：14,490ns [実行回数：219]	平均実行時間：14,490ns / 最大実行時間：14,500ns / 最小実行時間：14,490ns [実行回数：219]
Task_Hardware.c	procHardware_LED_AD	平均実行時間：10,971,110ns / 最大実行時間：10,974,350ns / 最小実行時間：10,965,320ns [実行回数：10]	平均実行時間：10,967,840ns / 最大実行時間：10,974,020ns / 最小実行時間：10,965,320ns [実行回数：10]
tasks.c	vTaskDelayUntil	平均実行時間：666,040ns / 最大実行時間：1,171,970ns / 最小実行時間：177,010ns [実行回数：239]	平均実行時間：663,340ns / 最大実行時間：1,171,980ns / 最小実行時間：177,010ns [実行回数：239]
tasks.c	vTaskSuspendAll	平均実行時間：15,980ns / 最大実行時間：104,000ns / 最小実行時間：14,330ns [実行回数：273]	平均実行時間：14,330ns / 最大実行時間：14,340ns / 最小実行時間：14,330ns [実行回数：273]

「比較」メニューの「パフォーマンス解析結果の比較」から「実行時間(In-Out)」または「周期時間」を選択すると、各レポートデータ解析結果を比較します。「解析オプション設定...」を選択すると、「レポート解析オプション設定」が表示され、解析時の設定を変更できます。

レポート解析オプション設定

プロセス解析モードを有効にする 全てのコアを統合して解析する

※これらの設定は、DT10/DT-Winの「環境設定」の解析の設定に合わせてください

OK キャンセル

「解析結果比較」ウィンドウに、各レポートを解析した結果が表示されます。

各平均値、最大値、最小値が左右の解析結果で一致している関数は、グレー色の文字で表示されます。

また、左右とも実行回数が0回の関数、および、どちらかにしか登録されていない関数は、リストに表示されません。

「設定」メニューから「比較結果表示設定...」を選択することで、リスト表示の色分けを設定することができます。

ソースファイル	関数	#1. Prof_181026_093316 / Repo_181026_153109 [Case:1000-5-9]	#2. Prof_181026_093316 / Repo_181026_133246 [Case:109-3-2]
DisplayRendererc	DRenderer_BeginScene	平均実行時間：10,249,740ns / 最大実行時間：10,249,750ns / 最小実行時間：10,249,740ns [実行回数：10]	平均実行時間：10,249,730ns / 最大実行時間：10,249,750ns / 最小実行時間：10,249,730ns [実行回数：10]
DisplayRendererc	DRenderer_EndScene	平均実行時間：60,150ns / 最大実行時間：60,160ns / 最小実行時間：60,150ns [実行回数：11]	平均実行時間：60,150ns / 最大実行時間：60,160ns / 最小実行時間：60,150ns [実行回数：11]
DisplayRendererc	DRenderer_DrawDeviceFormatIma	平均実行時間：25,150ns / 最大実行時間：31,340ns / 最小実行時間：17,000ns [実行回数：44]	平均実行時間：24,770ns / 最大実行時間：36,480ns / 最小実行時間：14,490ns [実行回数：44]
DisplayRendererc	DRenderer_Present	平均実行時間：10,833,670ns / 最大実行時間：10,836,910ns / 最小実行時間：10,827,890ns [実行回数：10]	平均実行時間：10,830,410ns / 最大実行時間：10,836,590ns / 最小実行時間：10,827,890ns [実行回数：10]
Task_Hardware.c	getDistance	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：32]	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：32]
Task_Hardware.c	getDMSVoltage	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：11]	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：11]
Task_Hardware.c	setLEDState	平均実行時間：14,400ns / 最大実行時間：14,400ns / 最小実行時間：14,400ns [実行回数：11]	平均実行時間：14,390ns / 最大実行時間：14,400ns / 最小実行時間：14,390ns [実行回数：11]
Task_Hardware.c	setBuzzerPitch	平均実行時間：28,890ns / 最大実行時間：28,900ns / 最小実行時間：28,890ns [実行回数：10]	平均実行時間：28,890ns / 最大実行時間：28,900ns / 最小実行時間：28,890ns [実行回数：10]
Task_Hardware.c	getColorVolume	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：219]	平均実行時間：0ns / 最大実行時間：0ns / 最小実行時間：0ns [実行回数：219]
Task_Hardware.c	setRGBLEDCol	平均実行時間：14,490ns / 最大実行時間：14,500ns / 最小実行時間：14,490ns [実行回数：219]	平均実行時間：14,490ns / 最大実行時間：14,500ns / 最小実行時間：14,490ns [実行回数：219]
Task_Hardware.c	procHardware_LED_AD	平均実行時間：10,971,110ns / 最大実行時間：10,974,350ns / 最小実行時間：10,965,320ns [実行回数：10]	平均実行時間：10,967,840ns / 最大実行時間：10,974,020ns / 最小実行時間：10,965,320ns [実行回数：10]
tasks.c	vTaskDelayUntil	平均実行時間：666,040ns / 最大実行時間：1,171,970ns / 最小実行時間：177,010ns [実行回数：239]	平均実行時間：663,340ns / 最大実行時間：1,171,980ns / 最小実行時間：177,010ns [実行回数：239]
tasks.c	vTaskSuspendAll	平均実行時間：15,980ns / 最大実行時間：104,000ns / 最小実行時間：14,330ns [実行回数：273]	平均実行時間：14,330ns / 最大実行時間：14,340ns / 最小実行時間：14,330ns [実行回数：273]

解析結果比較リストの任意の行をダブルクリックすると、その関数の全ての経路の実行時間のリストが表示されます。

更に、実行時間のリストの任意の行をダブルクリックすると、ソースファイルとレポートデータを表示するウィンドウが表示されます。このウィンドウを「ステップトレーサー」と呼びます。

ステップトレーサーのウィンドウは複数表示可能で、「ステップ実行」メニューから、「次の実行ステップへ」(F10)または「前の実行ステップへ」(Shift+F10)を選択することで、全てのステップトレーサーウィンドウを同期してトレースします。

これにより、異なるレポート間のプログラムの実行経路の違いを確認することが可能です。

(トレースの同期は、レポートデータを各経路で分割し、スタート位置からのオフセット時間で同期して表示されます。基準となるレポートは、現在アクティブになっているウィンドウです)

The screenshot displays the DT-TREx application interface. At the top, there's a menu bar with options like 'ファイル', '比較', 'ステップ実行', '設定', '表示', and 'バージョン情報'. Below the menu, there are several panes. On the left, a list shows execution times for 'vTaskDelayUntil' with columns for '開始時刻' and '実行時間(ns)'. The central pane shows a code editor with C code for 'vTaskDelayUntil' and a table below it with columns: 'No.', 'コア', 'ソース', '関数', 'ステップ', '説明', '説明2(コメント)'. The table lists function calls like 'vTaskDelayUntil', 'vTaskSuspendAll', and 'vTaskIncrementTick'. The bottom pane shows a comparison table of analysis results with columns for 'ソースファイル', '関数', and performance metrics for two different reports.

プラグインの手動登録方法

インストール時にDT10 / DT-Winを実行していたことでメニューに登録できなかった等で、プラグインを手動で登録する場合、以下の手順に従い、プラグインを登録してください。

1. 「ツール」から「プラグインの登録と解除...」を選択
2. 「プラグインの登録と解除」ダイアログで「追加」ボタンをクリック
3. 「プラグインの登録」ダイアログで、以下の内容を入力し、「登録」ボタンをクリック
 - メニューコマンド名: DT-TREx
 - 実行ファイル: DT-TREx.exeのフルパス名
 - コマンド引数: `$(ProjectFile) 10` (DT-Winの場合は、`$(ProjectFile) win`、Automotive Editionの場合は `$(ProjectFile) AE`)
4. 「プラグインの登録と解除」ダイアログで「OK」ボタンをクリック