

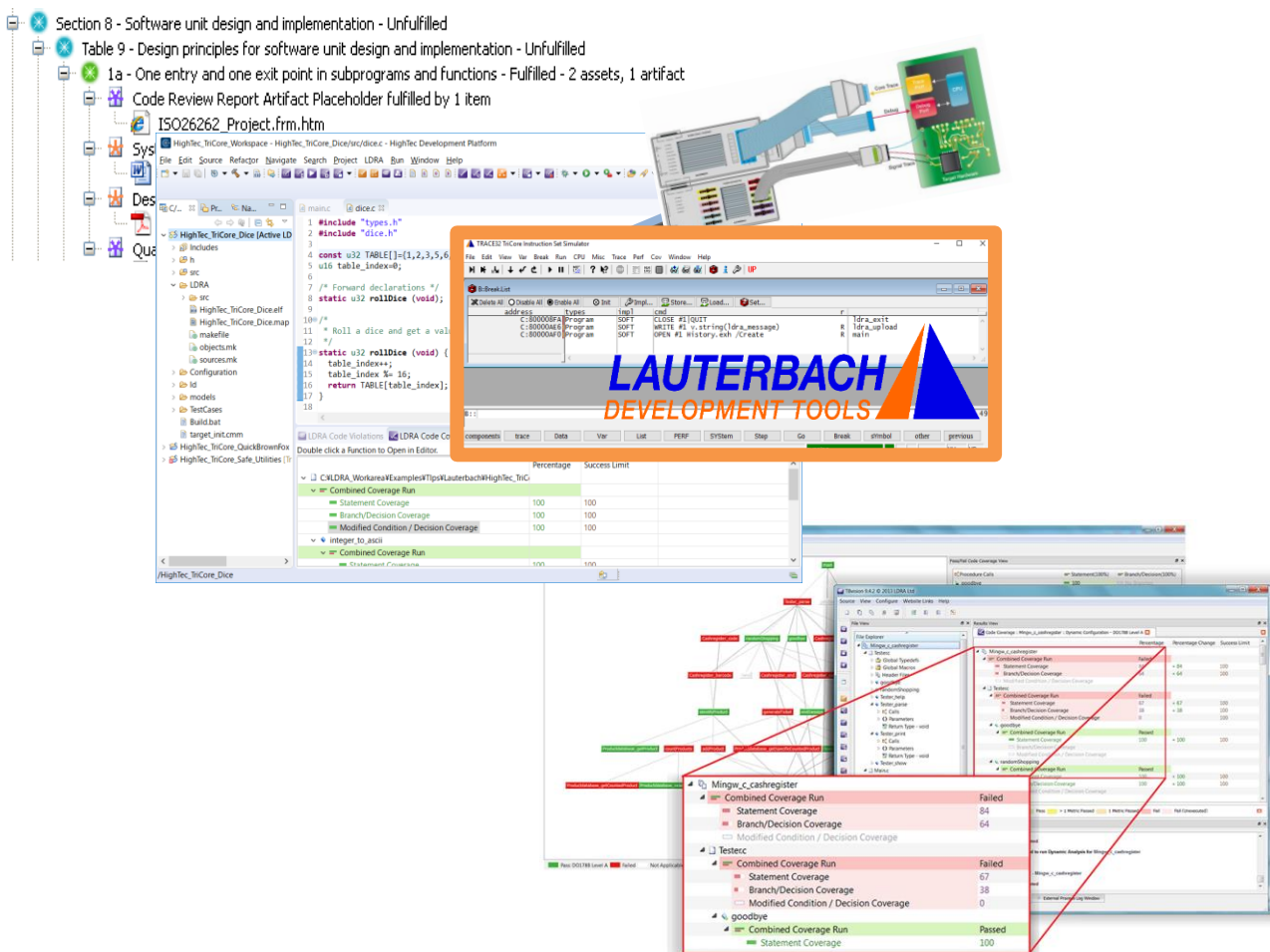
LDRA tool suite とターゲット開発環境の連携

国際スタンダード認証に最も実績のある LDRA 社テストツールは、ソフトウェア開発ライフサイクルの全般にわたって検証作業の自動化を支援します。その中で、単体テストの実行やカバレッジ解析にはコンパイラと実行環境が必要です。例えば、プログラムは統合開発環境 (IDE) でビルドされ、実ターゲットあるいはシミュレータ上でテストが実行されます。

LDRA tool suite は、あらゆるターゲット環境 (ターゲット CPU・ボード、コンパイラ・IDE、デバッガ/シミュレータ) を TLP (Target License Package) でサポートします。

以下、この TLP について、HighTec 社 Development Platform (IDE)・コンパイラとローターバハ社 TRACE32 シミュレータを用いて紹介します。プロジェクトには「HighTec 社コンパイラとローターバハ社デバッガチュートリアル」¹ の成果物を利用します。

TLP インストールでの注意点は、補足をご参考ください。



The screenshot displays the LDRA tool suite interface integrated with the HighTec Development Platform. The left sidebar shows a project tree for 'ISO26262_Project.frm.htm'. The main window is divided into several panes:

- Code Editor:** Displays C code for a dice simulation, including headers, constants, and a function `rollDice`.
- TRACE32 Simulator:** A window for running and debugging the code on a target or simulator.
- Coverage Reports:** A table showing the results of coverage analysis for different components.

Component	Statement Coverage	Branch/Decision Coverage	Modified Condition / Decision Coverage
Combined Coverage Run	100	100	100
integer_to_ascii	100	100	100
Combined Coverage Run	100	100	100

Below the coverage table, a detailed report for 'Mingw_cashregister' and 'Testerc' is shown, indicating 'Failed' status for some runs and providing specific coverage percentages.

¹ <https://www.fuji-setsu.co.jp/products/HighTec/#TriCore>

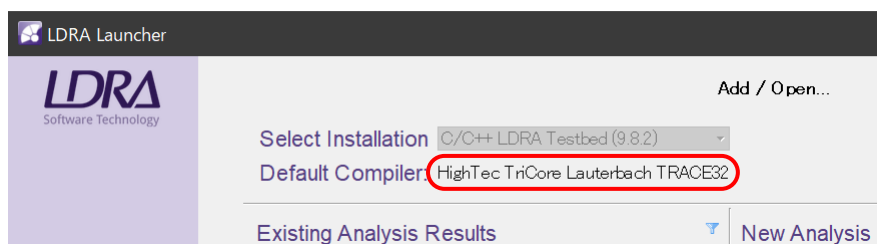
目次

1. LDRA tool suite の設定	3
2. HighTec Development Platform の起動とプロジェクトのインポート	3
3. プロジェクトの設定とビルド	6
• LDRA コンフィグレーション	6
4. 解析の準備 (HighTec IDE) – LDRA パースペクティブの選択	9
5. 静的解析の準備 (LDRA)	11
• 解析ファイルの選択	11
• 静的解析オプションの設定	12
♦ インクルードファイルのチェック	13
♦ 静的解析オプション：マクロ	15
6. 静的解析の実行と結果	16
7. 動的解析の実行	17
• カバレッジ解析の結果	18
8. 単体テストの自動生成と実行	19
補足：TLP のインストールと設定について	22
• インストールでの注意点	22
• LDRA Eclipse プラグインの選択	24

1. LDRA tool suite の設定

まず解析に利用するコンパイラ等のターゲット環境を選択します。

LDRA tool suite を起動し、LDRA Launcher で Default Compiler の設定が、使用するコンパイラ (ここでは HighTec TriCore) であることを確認します。異なる場合は Default Compiler 欄をクリックして変更してください。(TLP がインストールされていることが必要です。補足を参考)

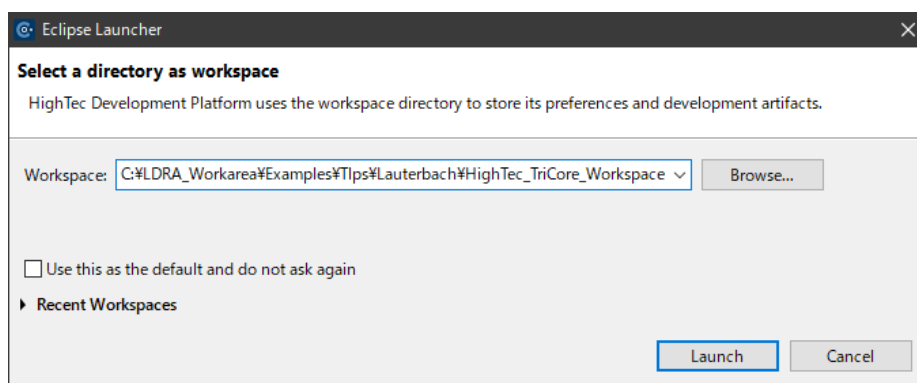


2. HighTec Development Platform の起動とプロジェクトのインポート

HighTec Development Platform を起動します。

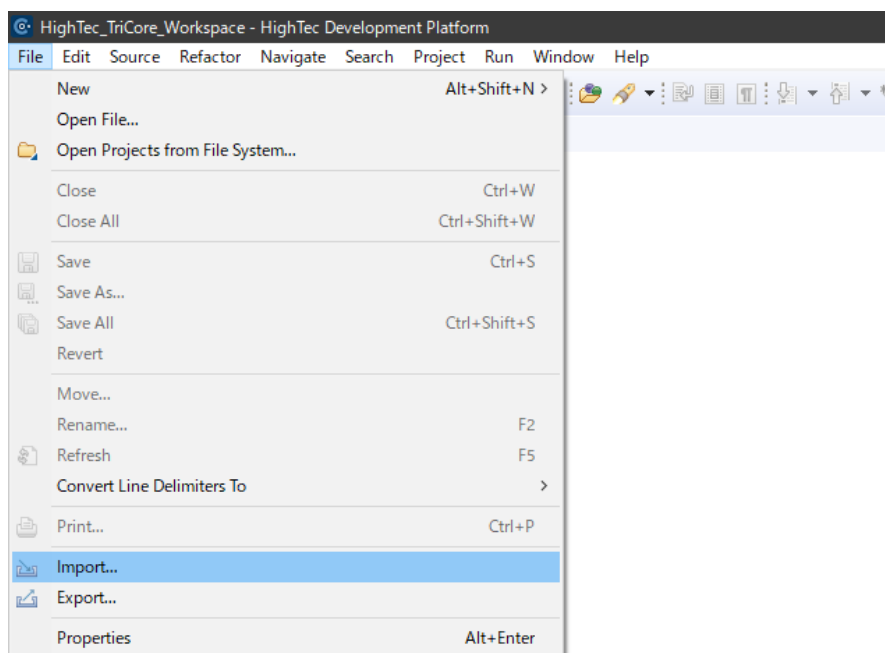
TLP のインストールによって、IDE に LDRA ツールがプラグインされると、TLP 用のワークスペースで起動されます。

(ここでは、C:\LDRA_Workarea\Examples\Tlps\Lauterbach\HighTec_TriCore_Workspace です)

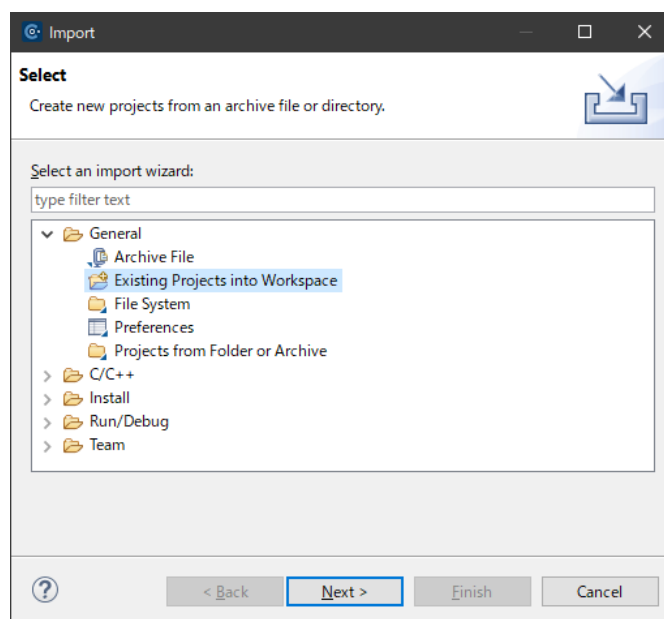


このワークスペースに、「HighTec 社コンパイラとローターバツハ社デバッガチュートリアル」で作成したプロジェクト(demo)をインポートします。

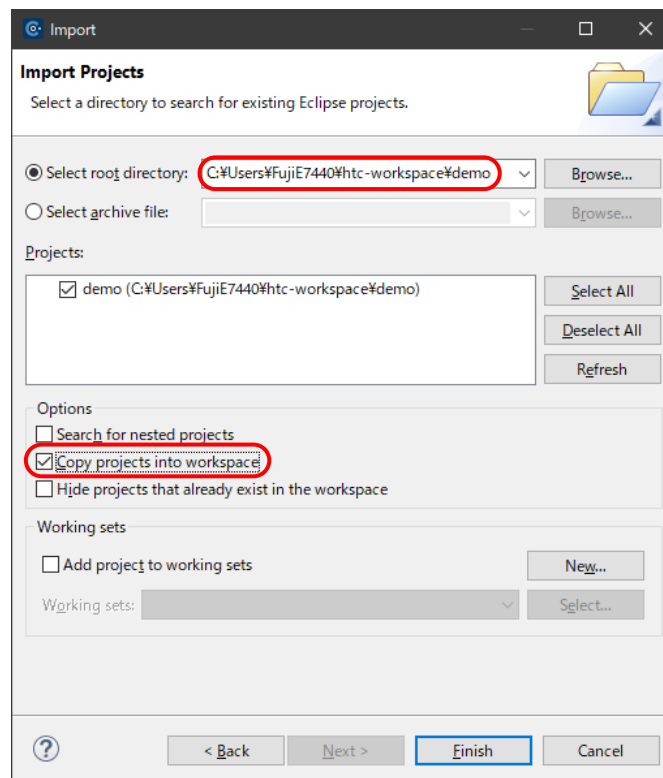
Step 1. File メニューから Import...を選択



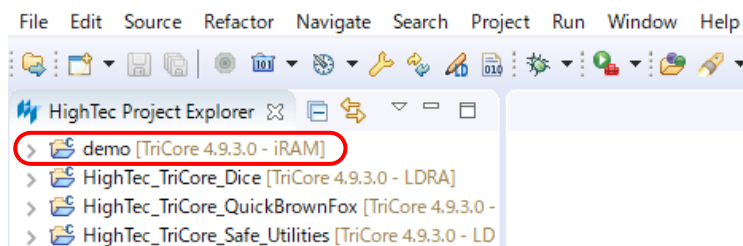
Step 2. Select ダイアログで Existing Projects into Workspace を選択し、Next > をクリック



Step 3. Import Projects ダイアログでインポート元のルートディレクトリを指定し、このワークスペースにコピーするよう Copy projects into workspace をチェックして、Finish します



demo がインポートされ、Project Explorer に表示されます。



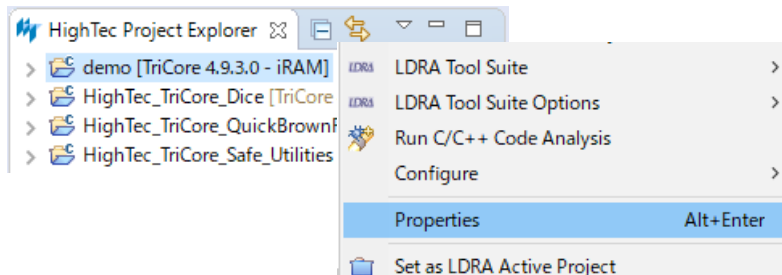
以降、IDE に統合された LDRA ツールで各種テストを行います。

3. プロジェクトの設定とビルド

• LDRA コンフィグレーション

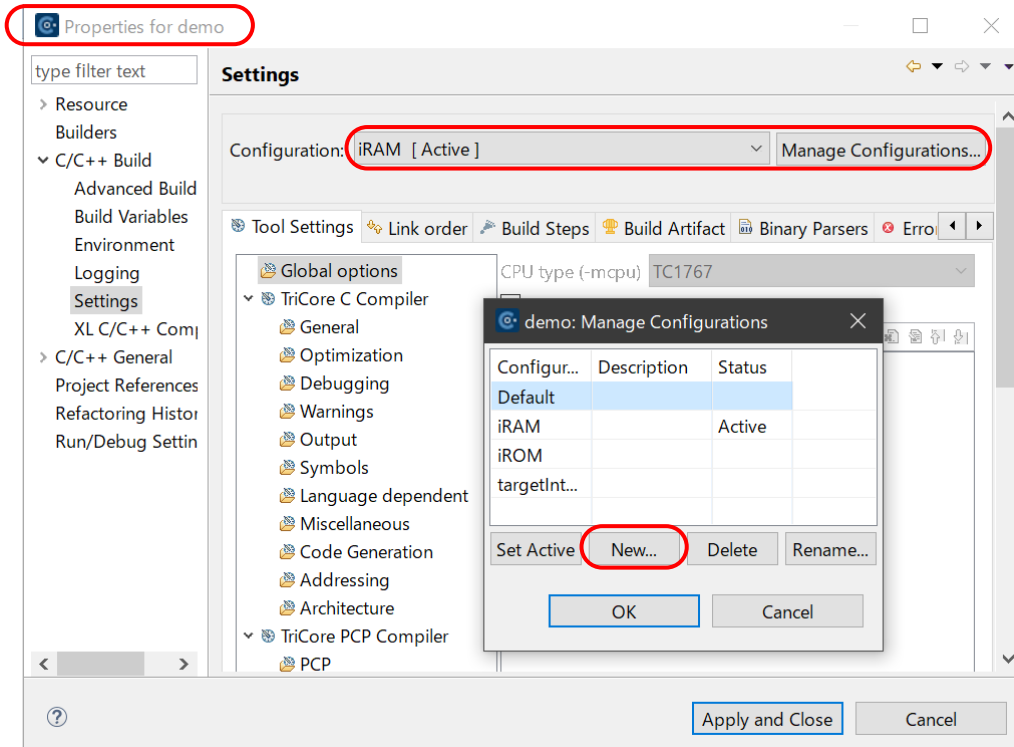
アクティブなコンフィグレーションを LDRA に設定します。LDRA で動的解析を行うために LDRA コンフィグレーションが必要になります。

demo[TriCore 4.9.3.0 - iRAM]を右クリックし、メニューから Properties を選択します。



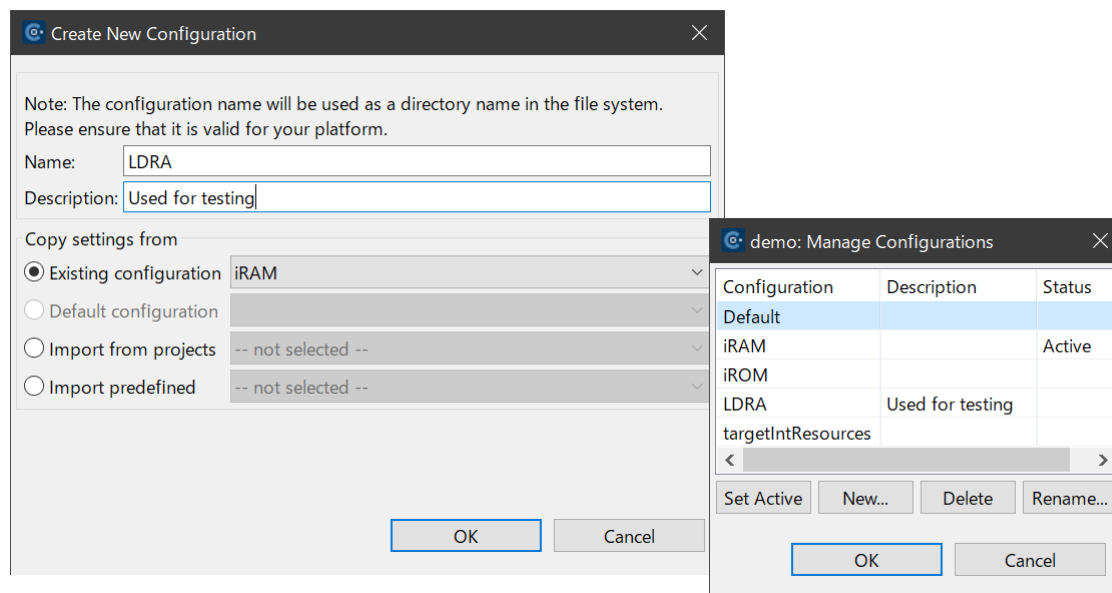
Properties for demo の C/C++ Build > Settings ダイアログで、下図のように Manage Configurations...をクリックします。

次に開く demo: Manage Configurations ダイアログで New... をクリックします。



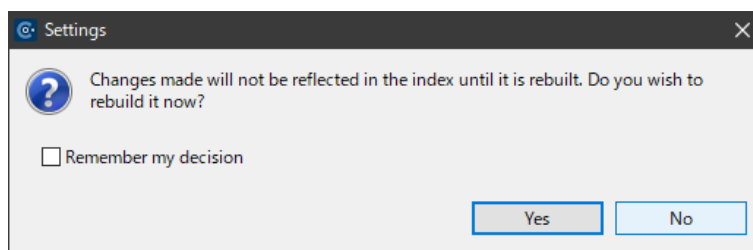
次に開く Create New Configuration ダイアログで Name に「LDRA」を追加し、OK します。
(Description 欄の「Used for testing」の入力は任意です)

demo: Manage Configurations ダイアログに戻るので、OK して Settings ダイアログに戻ります。

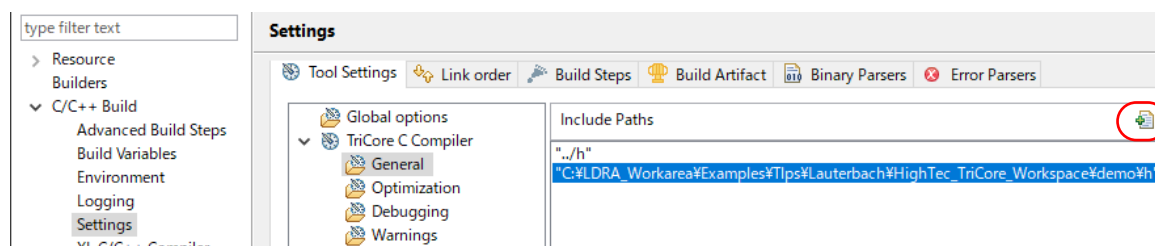


戻った Settings ダイアログで Apply and Close します²。

もし以下のダイアログが表示されたら、No をクリックします。

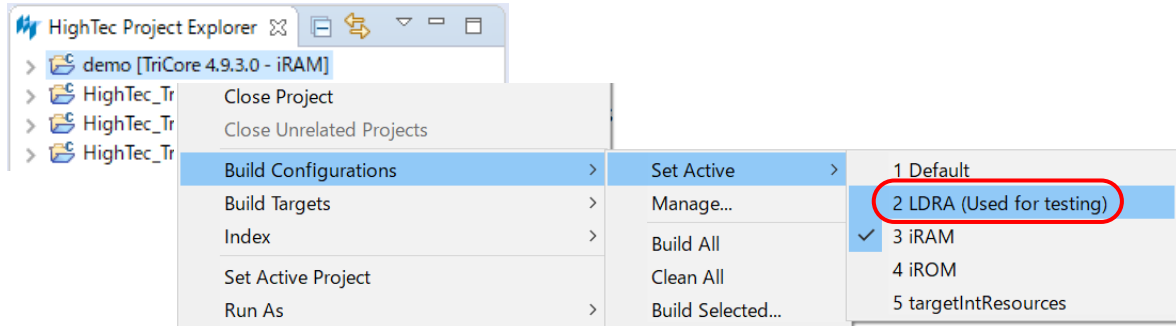


² この時点で、ビルドに必要な情報はインポートによって引き継がれていますが、単体テスト時のコンパイルにおいて、ヘッダファイルの場所によってはビルドエラーになることがあるかもしれません。その場合には、Settings の Include Paths で自身への絶対指定のインクルードパスを含むよう追加します。

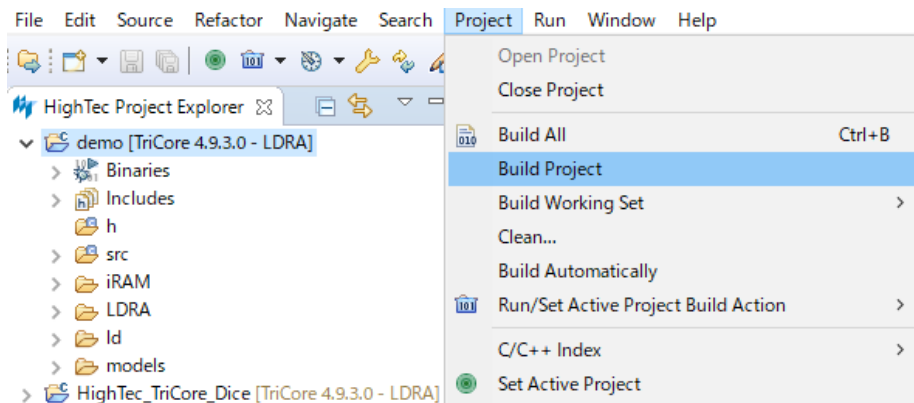


インポートしたプロジェクトが、問題無くビルドできることを確認します。

demo[TriCore 4.9.3.0 - iRAM]を右クリックし、Build Configuration > Set Active > 2 LDRA を選択して、チェックが入りアクティブとなるようにします。

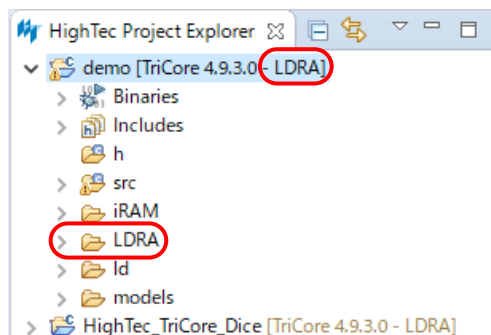


次にビルドします。



ビルド結果の Project Explorer ビューは以下になります。

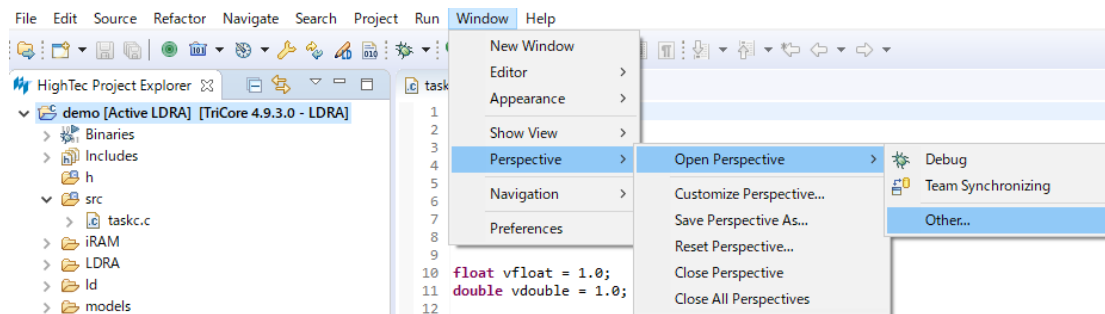
demo[TriCore 4.9.3.0 - iRAM]の iRAM が LDRA となり、ツリーに LDRA フォルダがあることを確認してください。



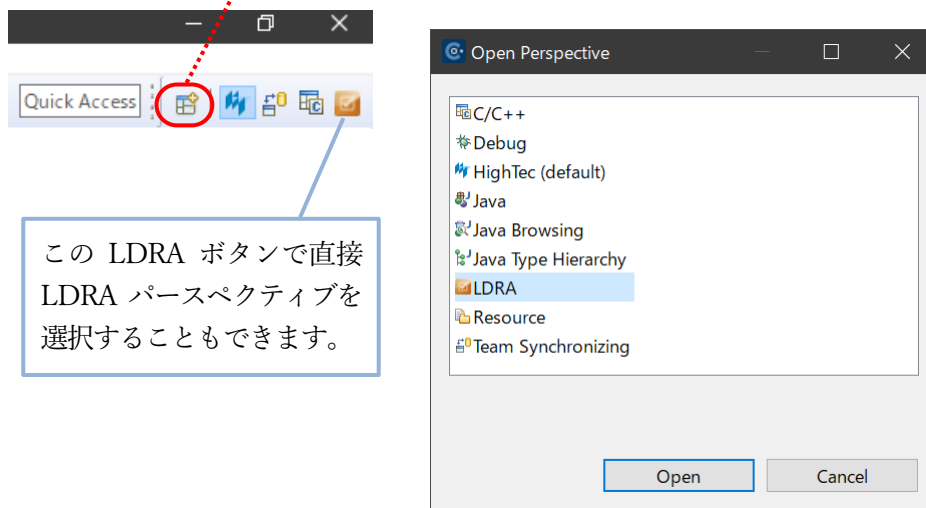
4. 解析の準備 (HighTec IDE) – LDRA パースペクティブの選択

HighTec IDE のユーザーインターフェース画面から直接 LDRA tool suite を利用できるようにします。

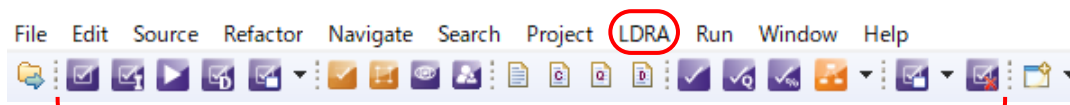
Window メニュー から Perspective > Open Perspective > Other を選択するか、もしくは



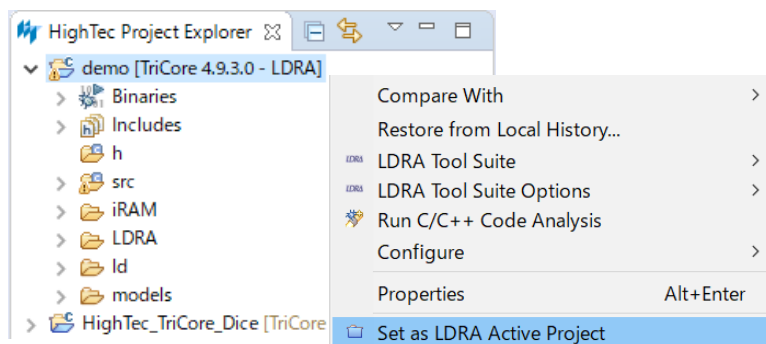
Quick Access から Open Perspective を選択し、Open Perspective ダイアログで LDRA を選択します。



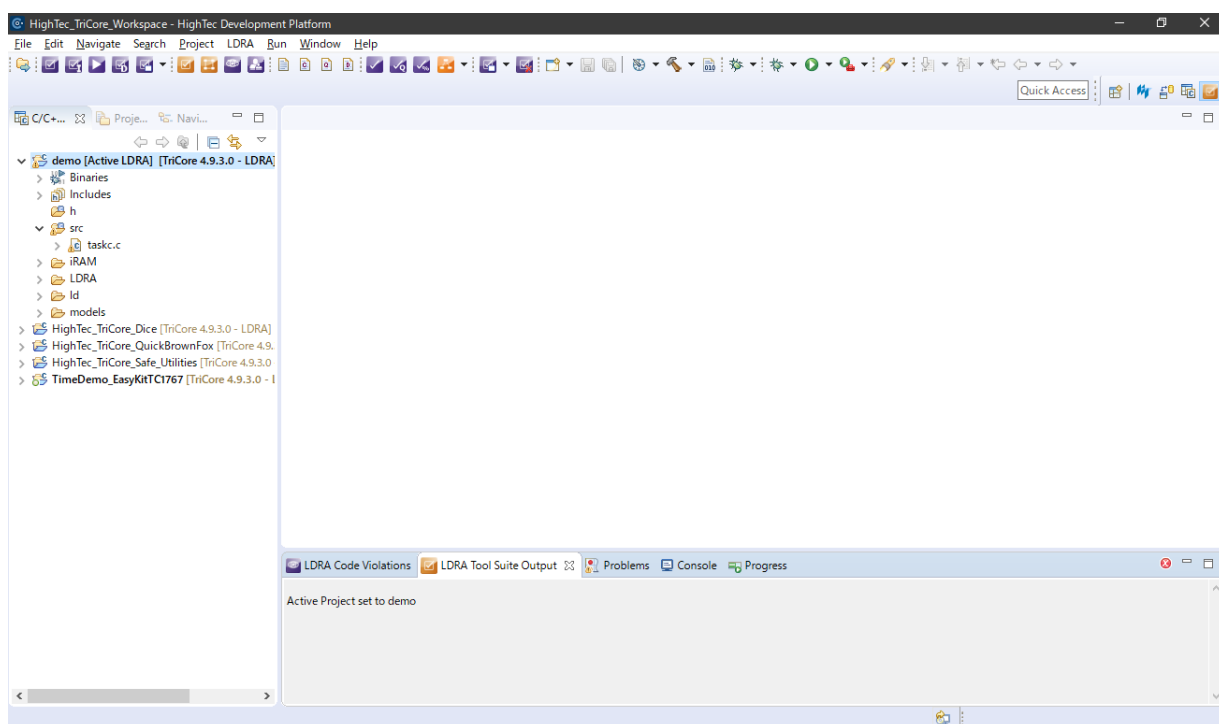
LDRA パースペクティブを選択すると、メニューに LDRA が追加され、ツールバーに LDRA のメニューアイコン(紫色やオレンジ色のボタン)も追加されます。



demo[TriCore 4.9.3.0 - iRAM]を右クリックし、プロジェクトを LDRA Active Project と設定します。



HighTec Development Platform の画面は以下のようになっています。

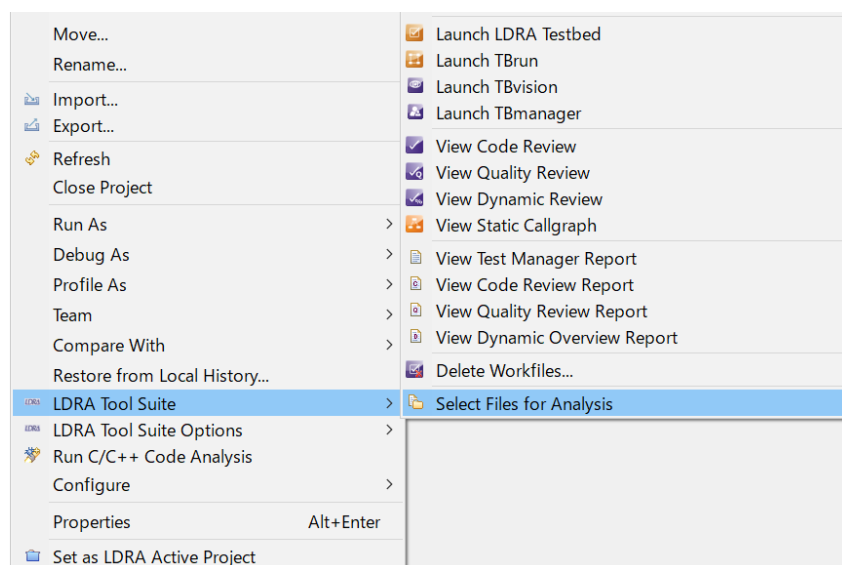


次に、LDRA tool suite でコードレビューや単体テストを行います。

5. 静的解析の準備 (LDRA)

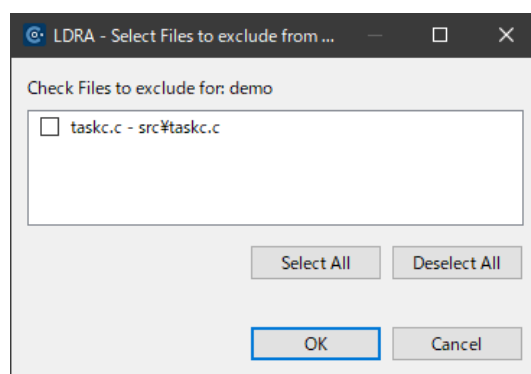
• 解析ファイルの選択

プロジェクト demo を右クリックし、メニューから LDRA Tool Suite > Select Files for Analysis を開きます。



ダイアログで除外するファイルを指定します。

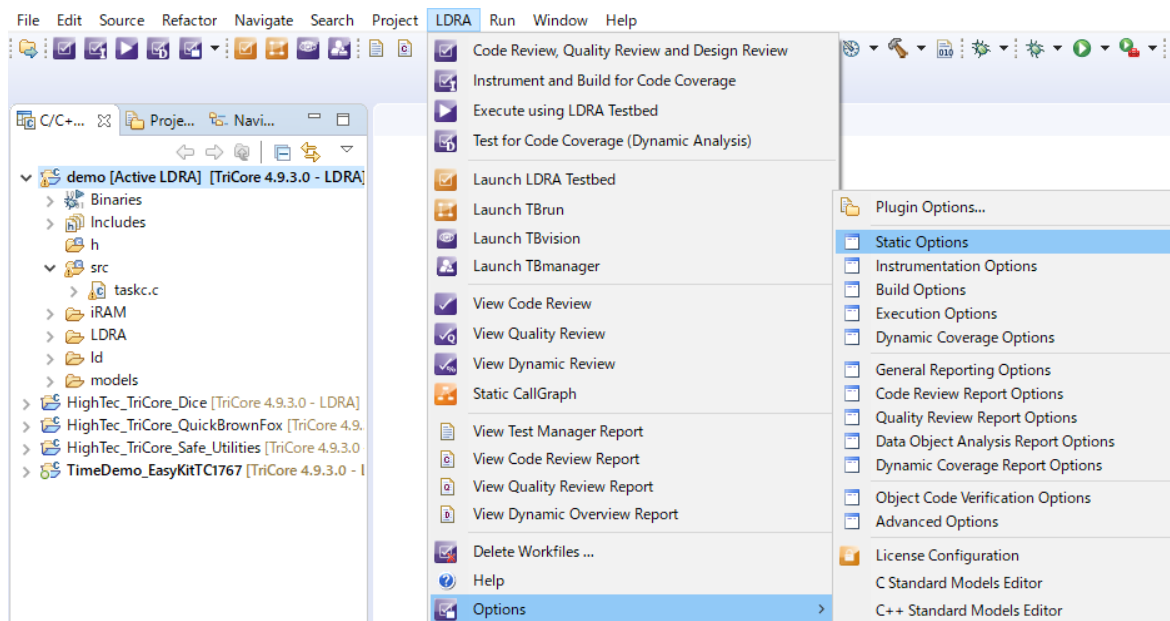
本例では `taskc.c` のみであり、解析しないファイルはありませんので、何も除外せず OK してメニューを閉じます



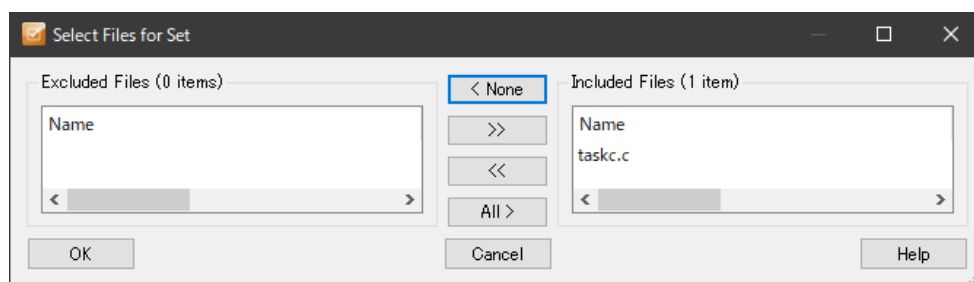
• 静的解析オプションの設定

解析を実行する前に、先にプロジェクトに設定したインクルードパスやマクロの設定が反映されていることを確認します。

LDRA メニューから Options > Static Options を開きます

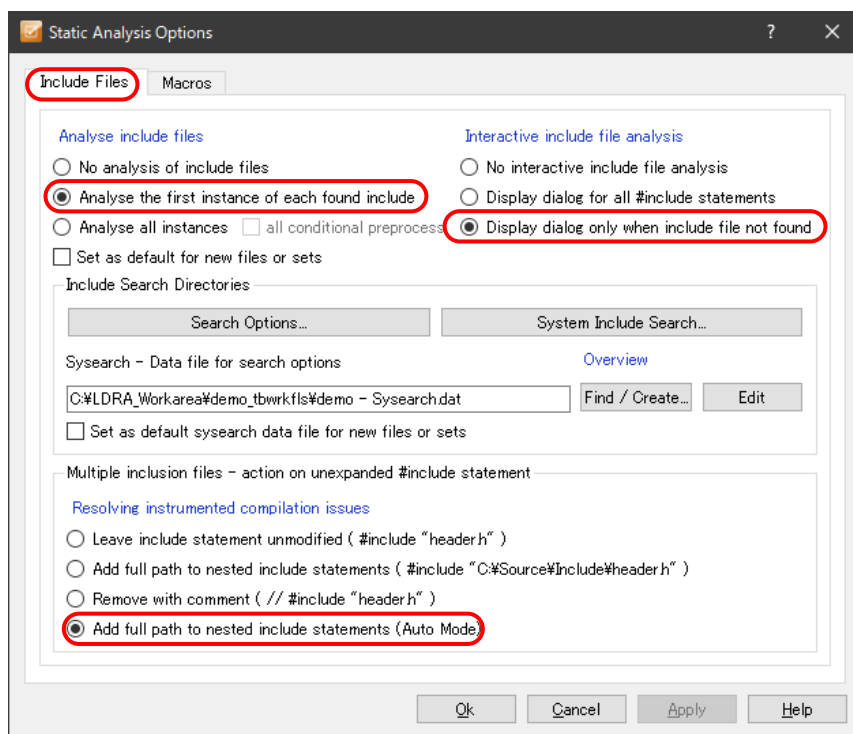


下図のように、解析対象のファイルすべてが含まれている(taskc.c が Included Files 側にある)ことを確認して OK をクリックし、Set を生成します。

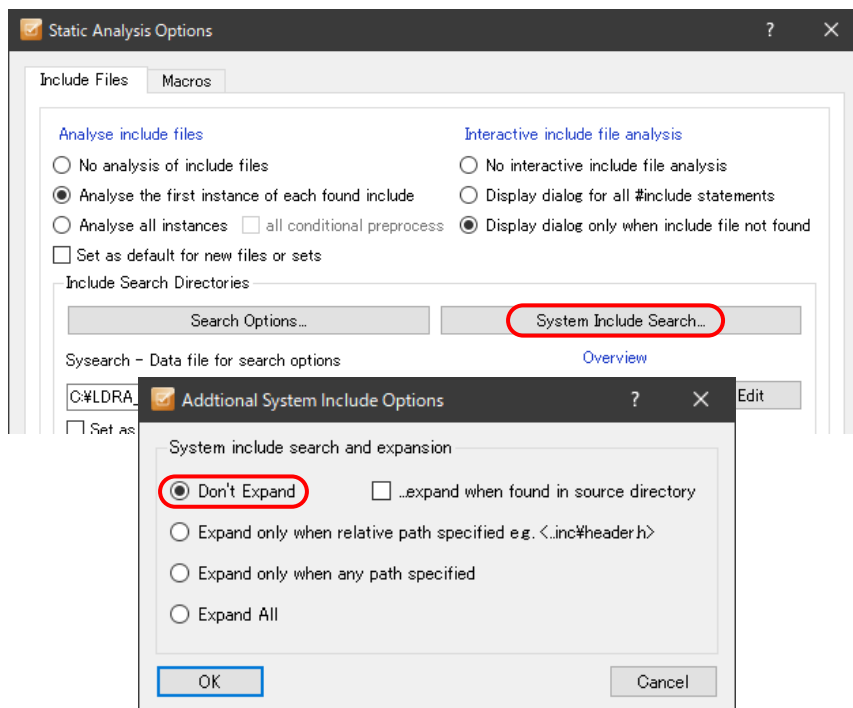


◆ インクルードファイルのチェック

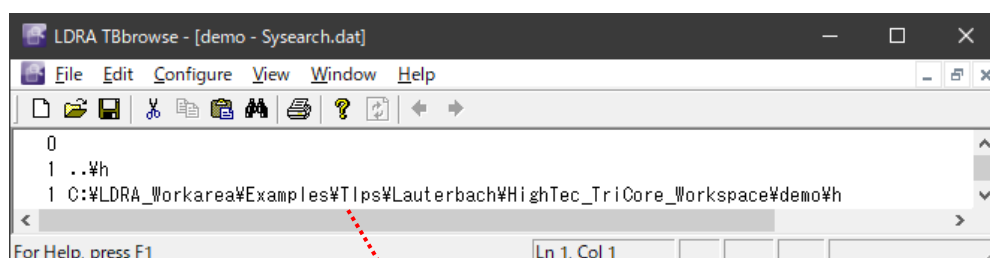
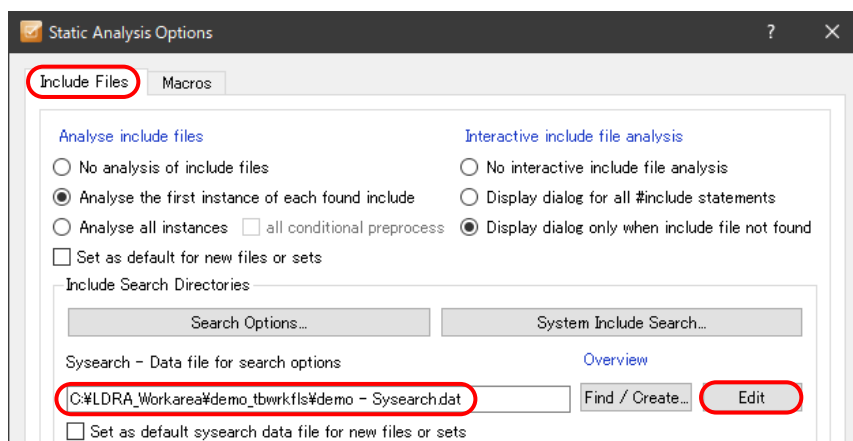
次のようにオプションが選択されていることをチェックします。



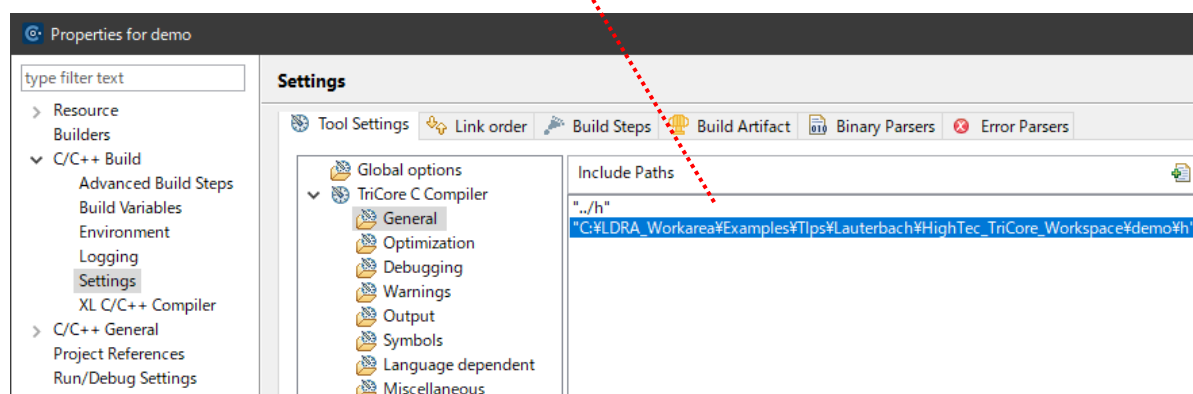
さらに System Include Search... をクリックし、Additional System Include Options で次のように Don't Expand オプションが選択されていることを確認して、OK します。



有効なサーチパスはプロジェクトから自動的に読み込まれ、Sysearch.dat ファイルに追加されます。

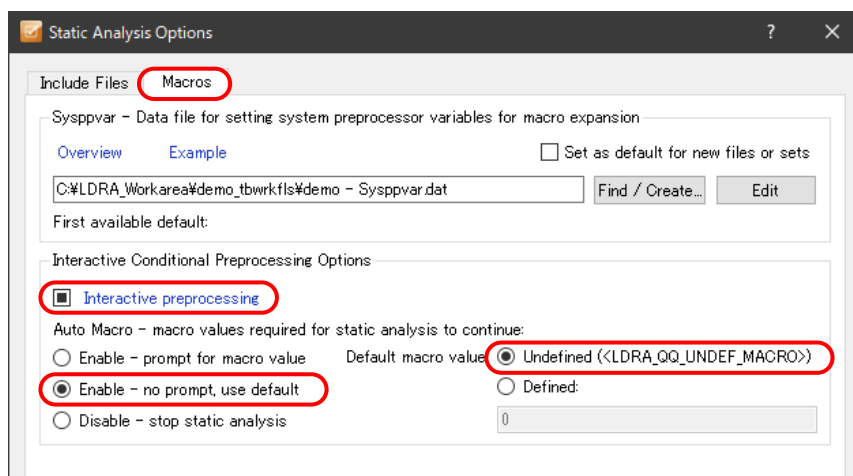


HighTec IDE で設定されている「インクルードパス」情報が反映されていることを確認してください。

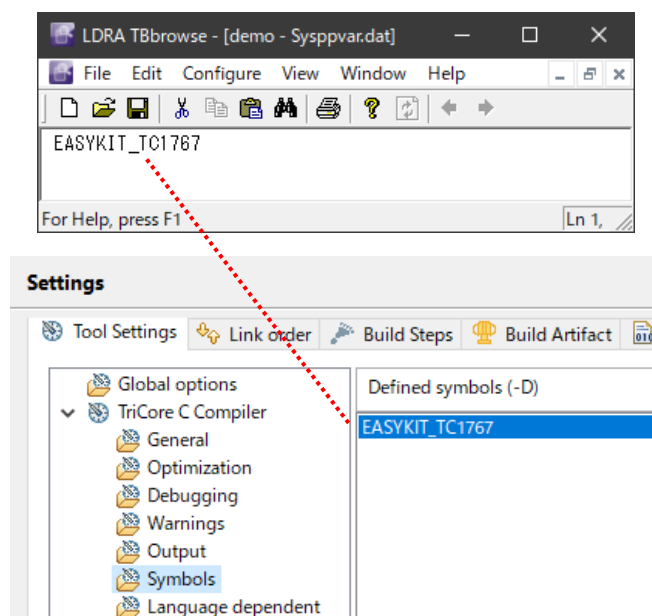
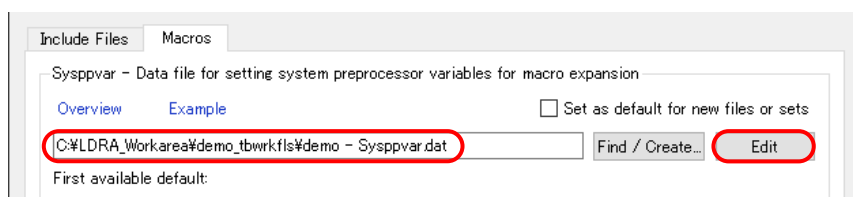


◆ 静的解析オプション：マクロ

次のようにオプションが選択されていることを確認します。



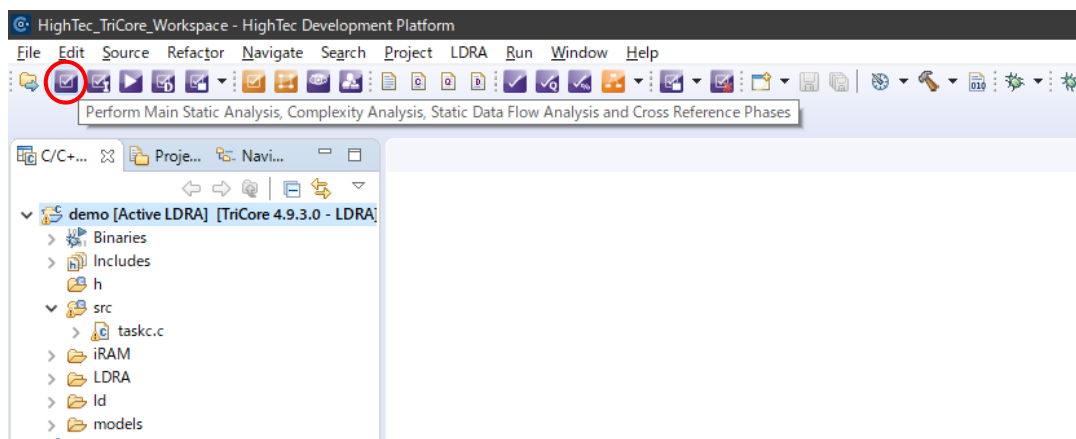
使用される define 定義はプロジェクトから自動的に読み込まれ、Sysppvar.dat ファイルに追加されます。




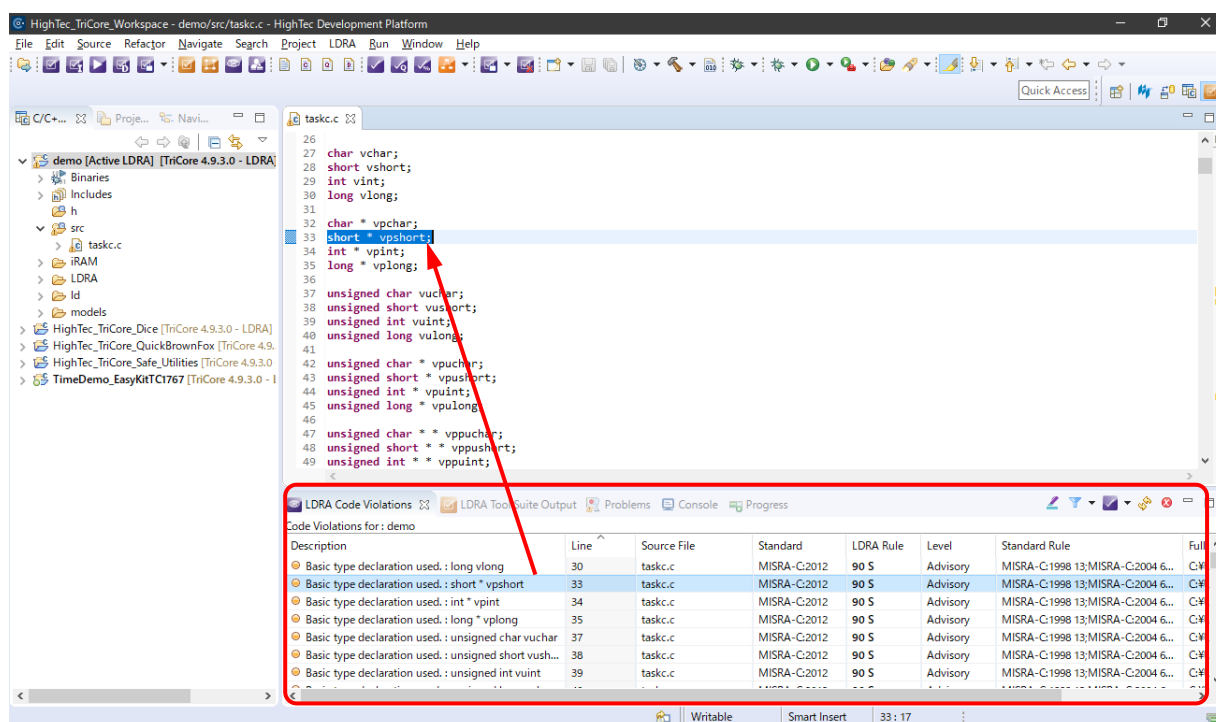
HighTec IDE で設定されている「define 定義」情報が反映されていることを確認してください。

Include Files タブと Macros タブの設定を確認できたら、Static Analysis Options ダイアログを OK して閉じます。

6. 静的解析の実行と結果





 ボタンで静的解析を実行します。解析結果は下記のようにになります。




LDRA Code Violations タブで違反が表示されており、エントリをダブルクリックすることで対応するソースコードがハイライトされます。

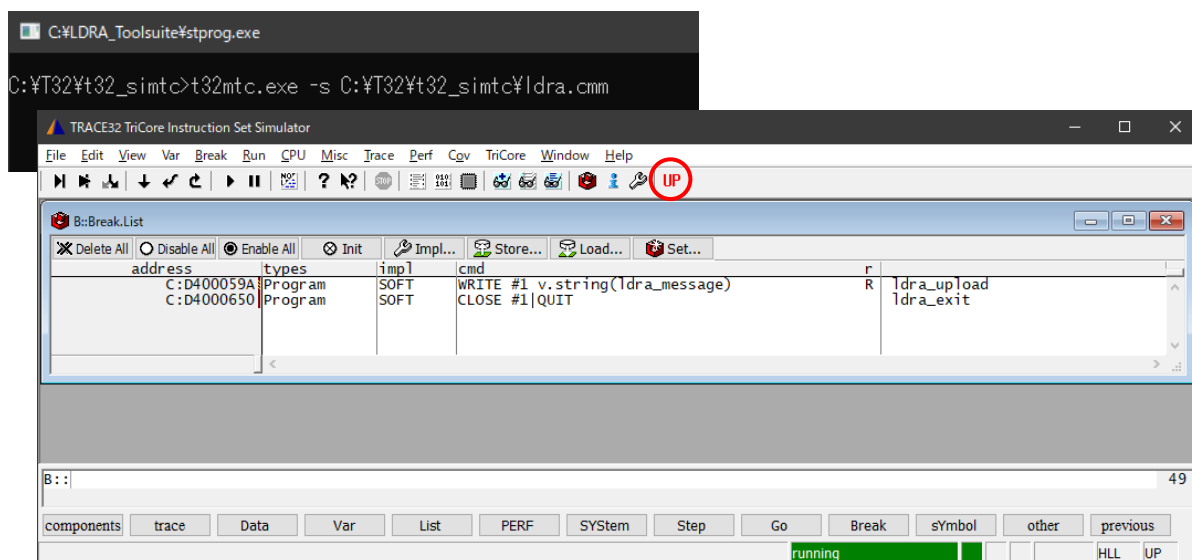
7. 動的解析の実行

ここでは、カバレッジ解析用にインスツルメントしたコードをビルドして、生成されたオブジェクトをターゲットあるいはシミュレータ環境にダウンロードして実行し、結果をアップロードして解析します。

 でインスツルメントを行ってから、 で実行します。


ビルド後に TRACE32 シミュレータが、下図のように起動されます。

本例では main 関数が無限ループして実行が止まりませんので、 で一時停止させた後、**UP** ボタンを押します。



これでカバレッジ解析用の実行履歴がアップロードされ、シミュレータは終了して HighTec ツールに戻ります。

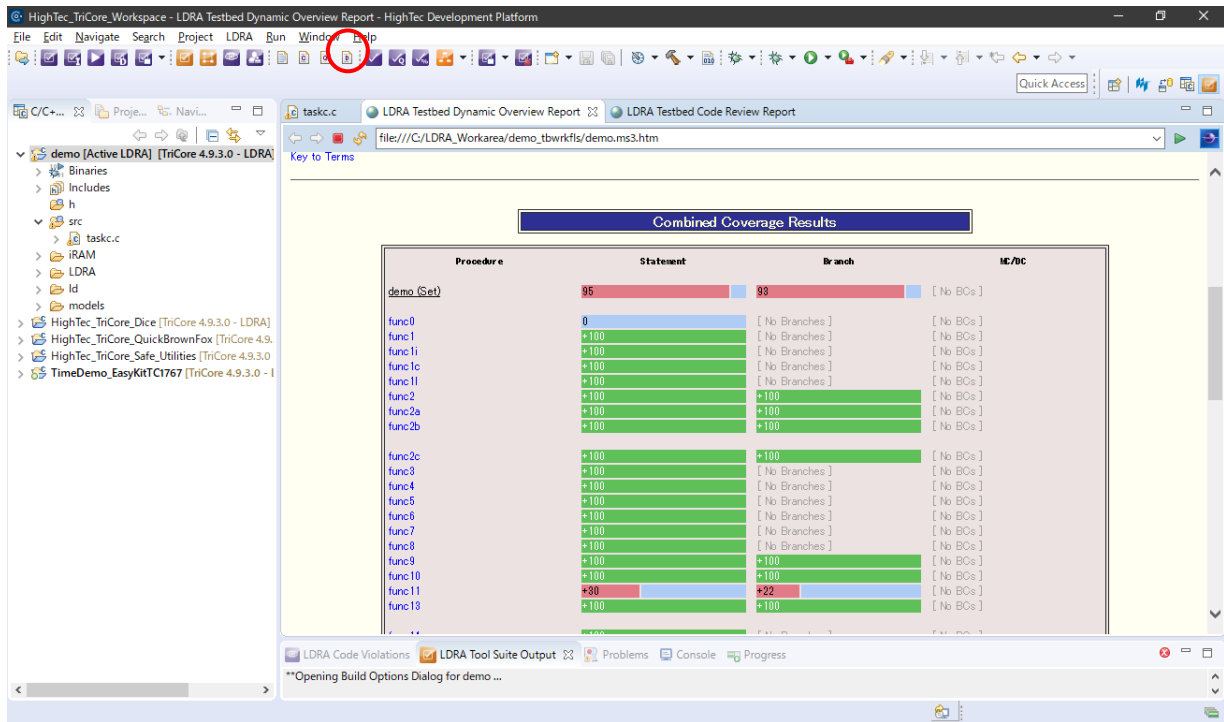
* 多くの組込みシステムでは main 関数は、一般には無限ループとなっています。その場合、上図のようにシミュレータが running のままとなります。この状態でプログラムの実行を停止させて実行履歴をアップロードする関数を呼び出す機構が必要になり、TLP では**UP** ボタンとしてその機能を用意しています。

次に、 でカバレッジ解析を実行します。

```
587 main(void)
588 {
589     int j;


670     while ( TRUE )
671     {
672         sieve();
673     }
674 }
```

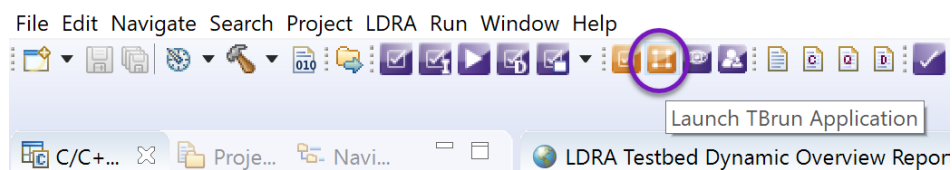
・カバレッジ解析の結果



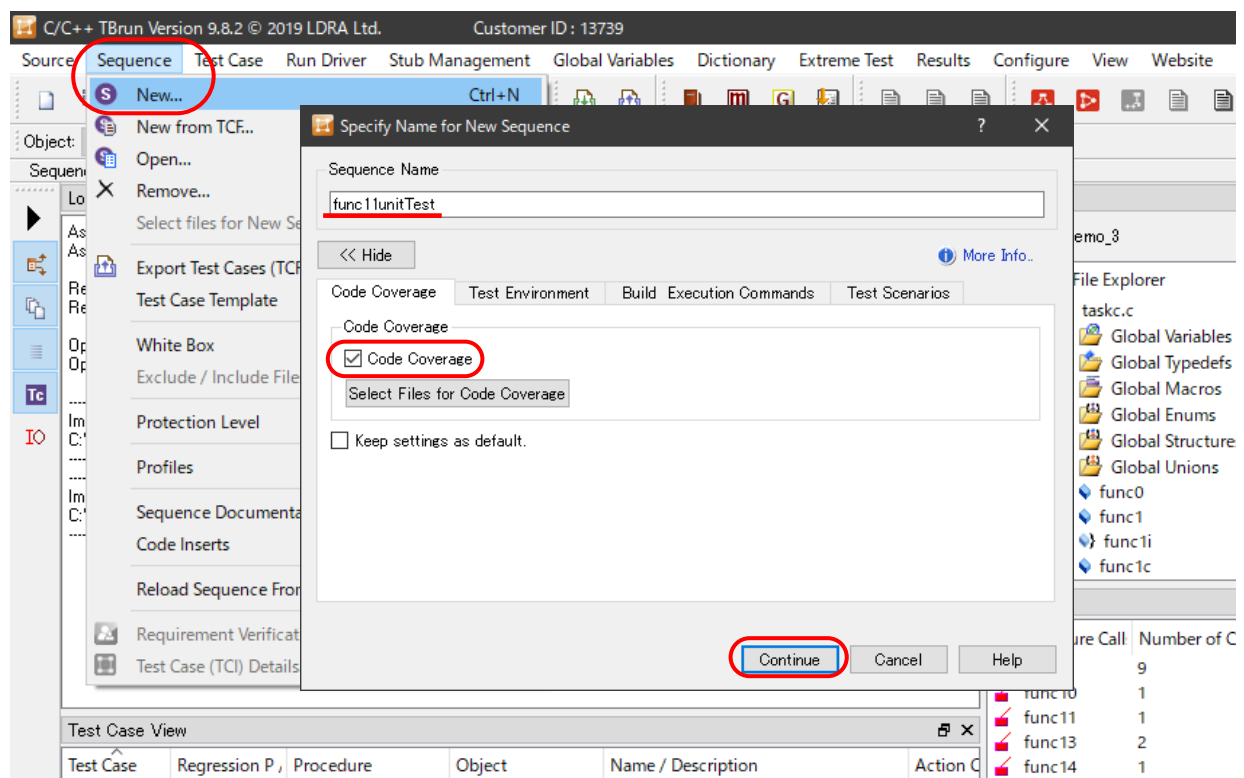
8. 単体テストの自動生成と実行

次に 100%カバレッジに満たなかった部分、ここでは **func11** (Statement 30%、Branch 22%) に対して単体テストの自動実行 (Extreme Test) を追加します。

TBrun は LDRA パースペクティブとしたツールバーから  ボタンをクリックして起動できます。また、LDRA メニューの Launch TBrun や、プロジェクトを右クリックしたメニューの LDRA Tool Suite > Launch TBrun から起動できます。



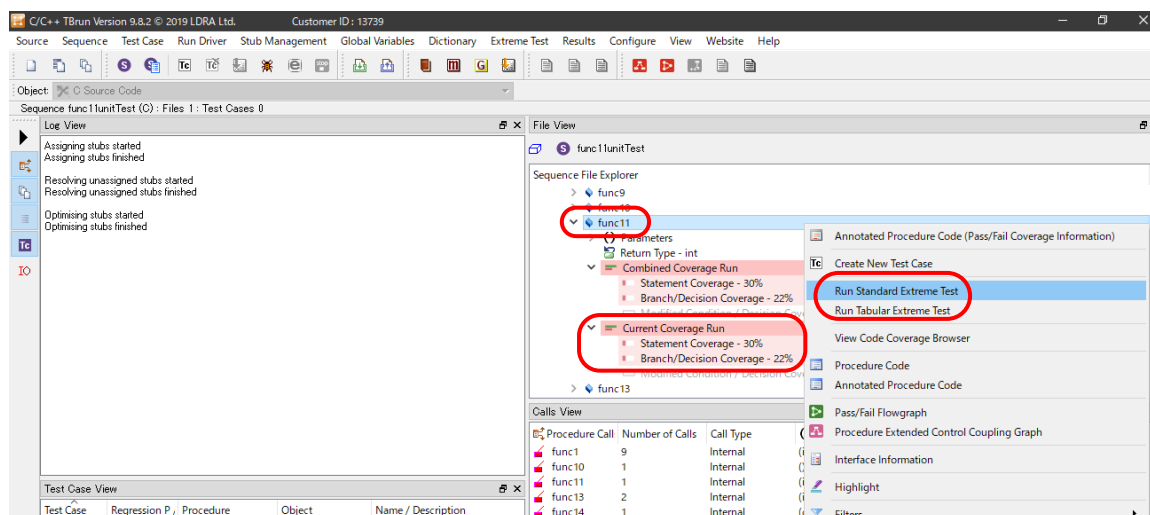
TBrun が起動したら、下図のように単体テストのシーケンスを新たに設定します。Sequence Name (func11unitTest) は任意に設定してください。



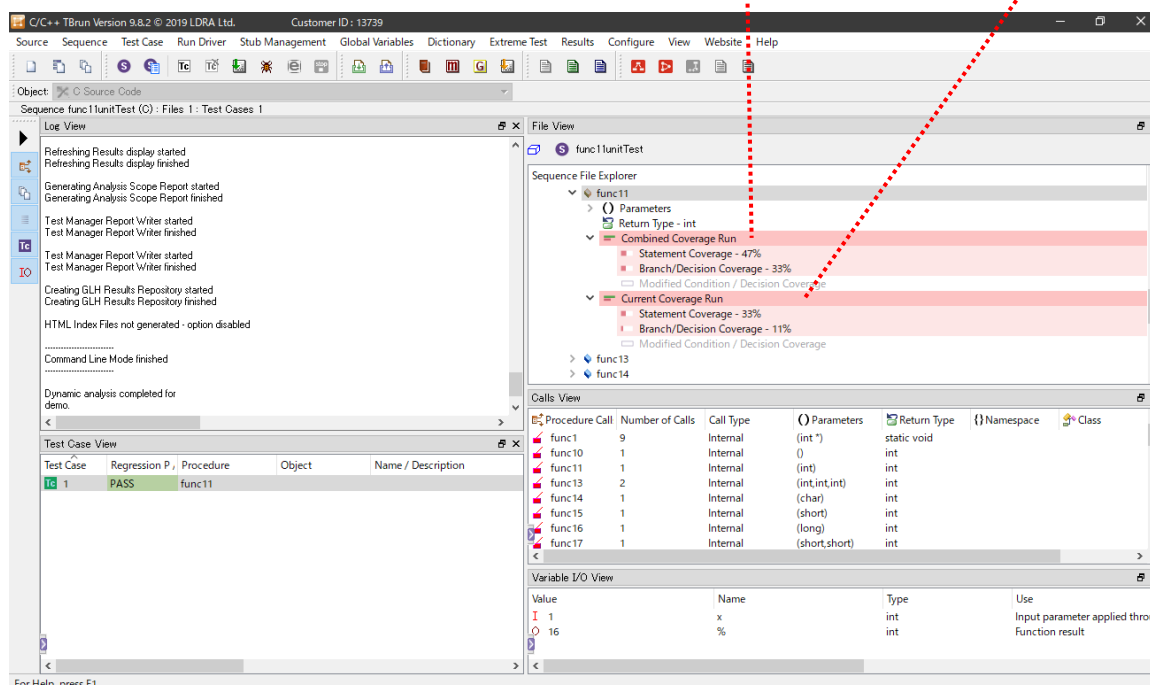
下図のように **func11** の現状のカバレッジ結果を見ることができます。

(Statement 30%、Branch 22%) となっています。

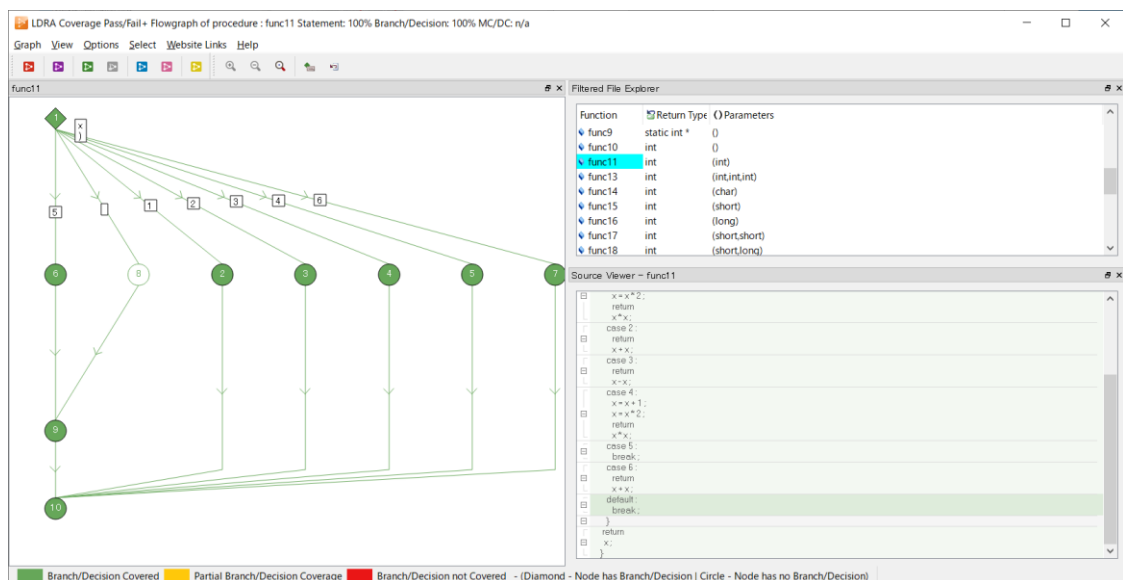
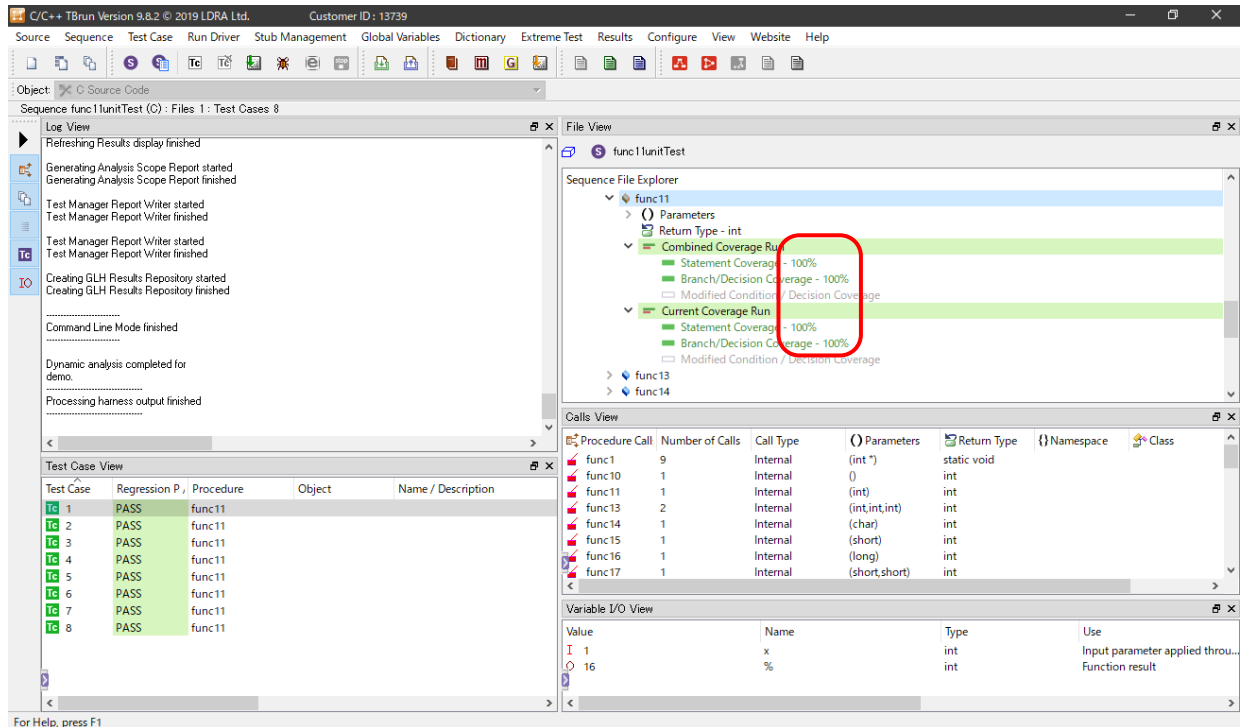
次に、**func11** を右クリックして開くメニューから **Run Standard Extreme Test** を選択して、1つのテストケースの自動生成と実行を行います。



実行結果として、**func11** の累積されたカバレッジ結果 (Combined) と、今回の結果 (Current) を、下図のように見ることができます。



次に、Run Tabular Extreme Test でカバレッジ 100%を狙った複数テストケースの自動生成と実行を行うと、今回のカバレッジ結果 (Current)、累積された結果 (Combined) の両方で 100%になっていることが下図のように確認できます。



以上、このチュートリアルでは、単体テスト自動生成機能の紹介までとします。その他、詳しい機能について、以下に動画やスライドを公開していますので、ご参考ください。

<https://www.fuji-setsu.co.jp/products/LDRA/#TriCore>

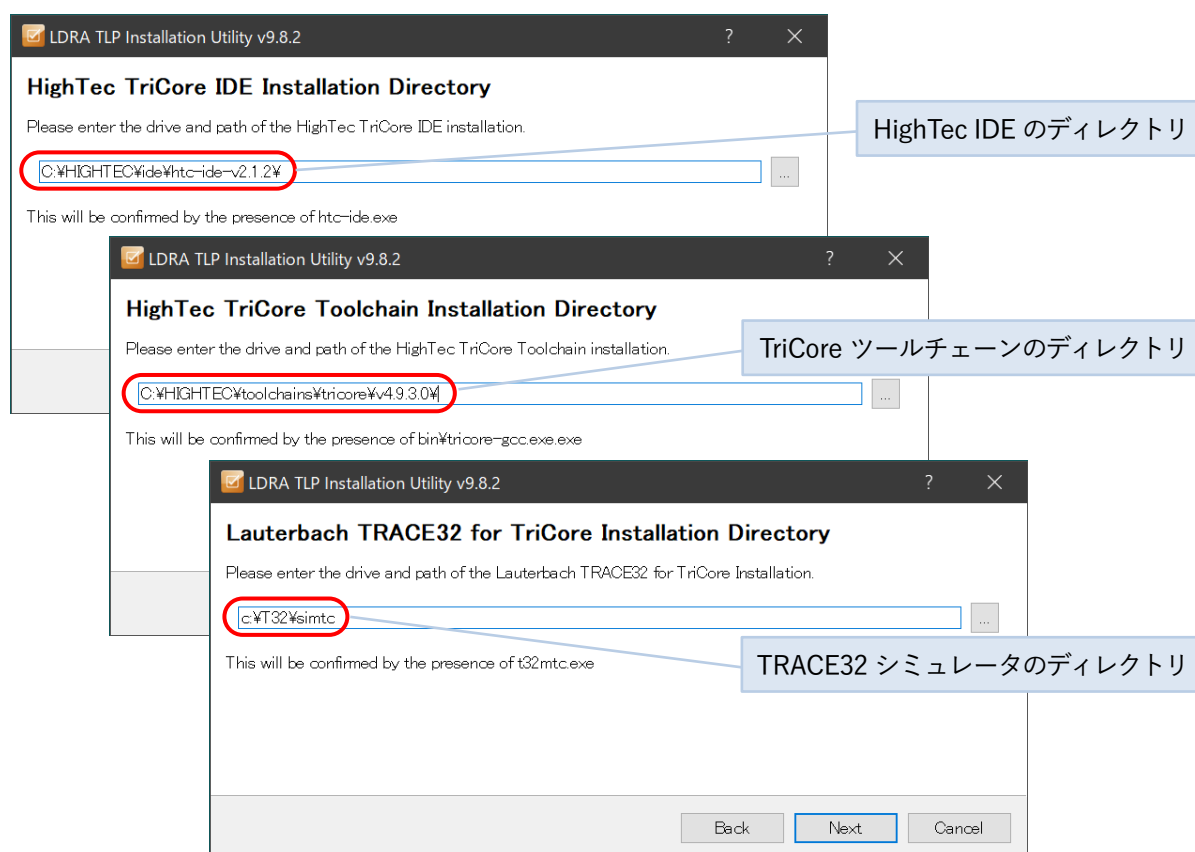
補足：TLP のインストールと設定について

TLP では、ターゲットごとにインストーラが作成されます。TLP のインストールにより **tlps** フォルダにサンプルプロジェクトやドキュメント、コマンドファイルが配置されます。

本チュートリアル例では、HighTec C/C++ Development Platform を利用して、HighTec IDE と TriCore コンパイラツールチェーン、および Lauterbach TRACE32 シミュレータを使用して TLP を適用しています。

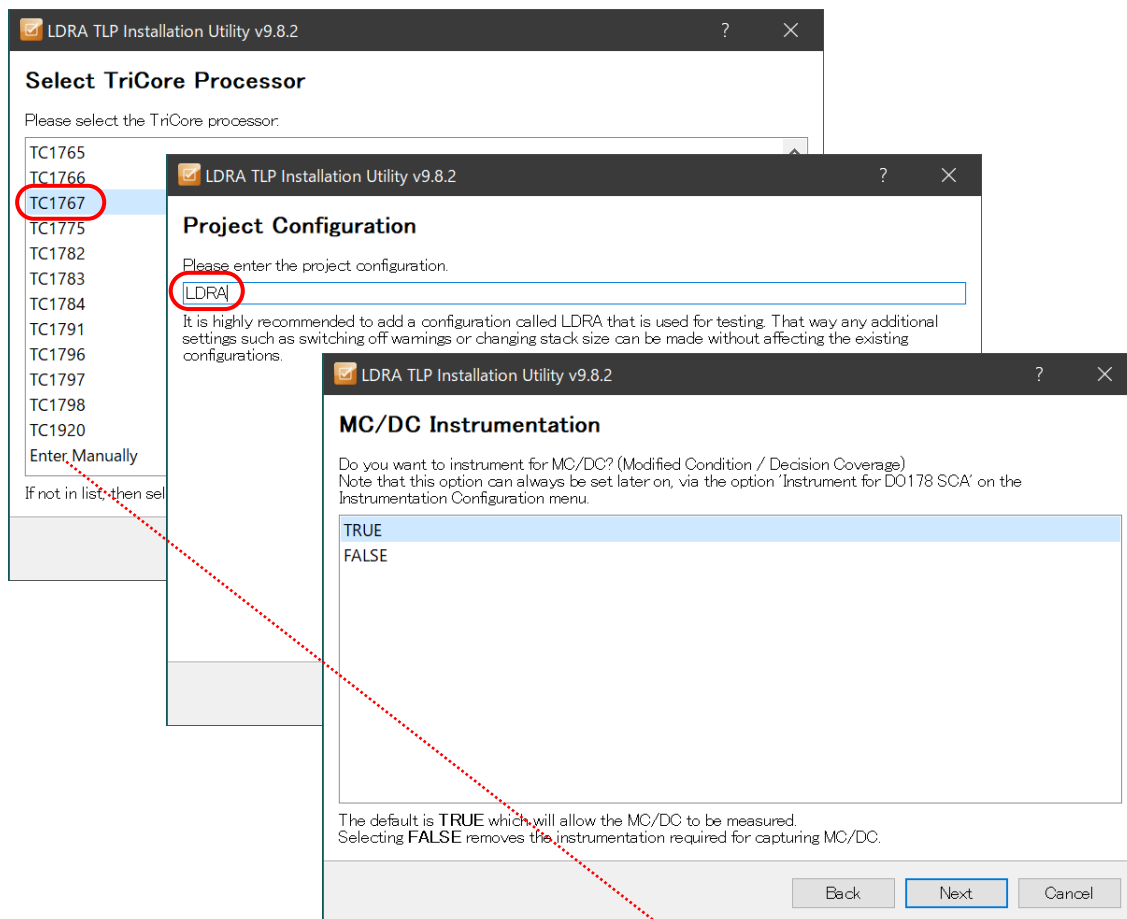
・インストールでの注意点

TLP のインストール中に、ダイアログでコンパイラツールチェーンとデバッガ/シミュレータのインストールディレクトリを確認します³。

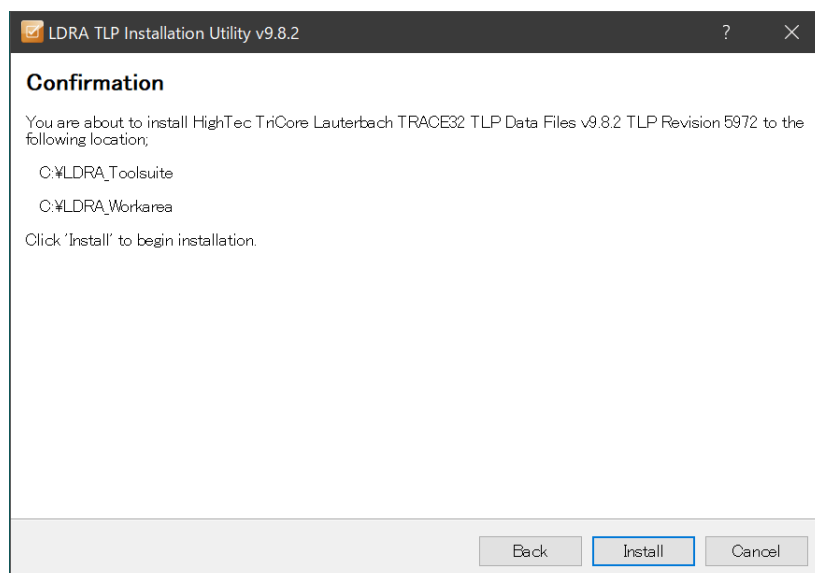


使用するプロセッサの指定（本例では TC1767）、プロジェクトコンフィグレーション（LDRA）の設定、解析用処理の設定するダイアログにしたがってインストール手順を進めてください。

³ TLP インストールの前に、ご使用になるコンパイラ・IDE とデバッガ/シミュレータがインストールされていることをご確認ください。



デバイスのリスト内に使用したいバリエーション名が無い場合は、「Enter Manually」から、そのバリエーション用の各オプション(コンパイラとデバッガ)を指定いただけます。

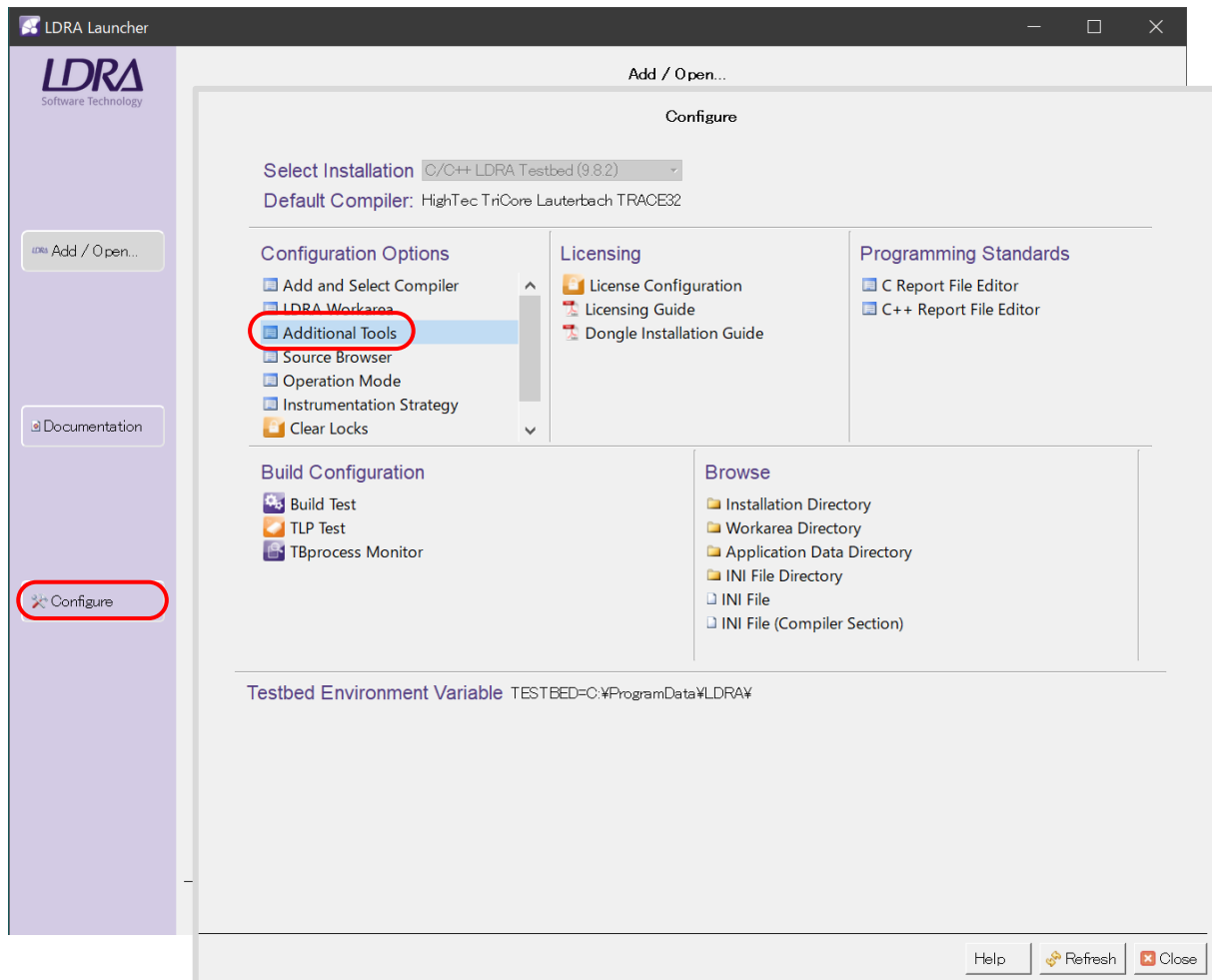


確認して、Install します。

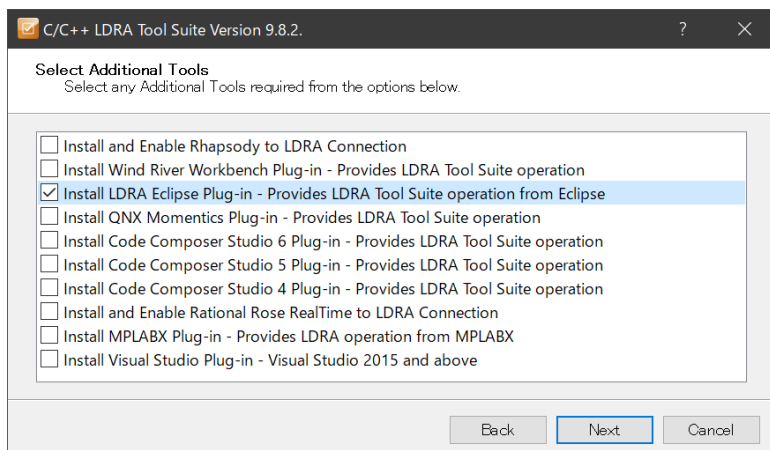
• LDRA Eclipseプラグインの選択

HighTec IDE (Eclipse) に LDRA tool suite をプラグインとして組み込んで、HighTec IDE の GUI から LDRA の機能を実行できるようにします。LDRA ツールから以下の手順で行います。

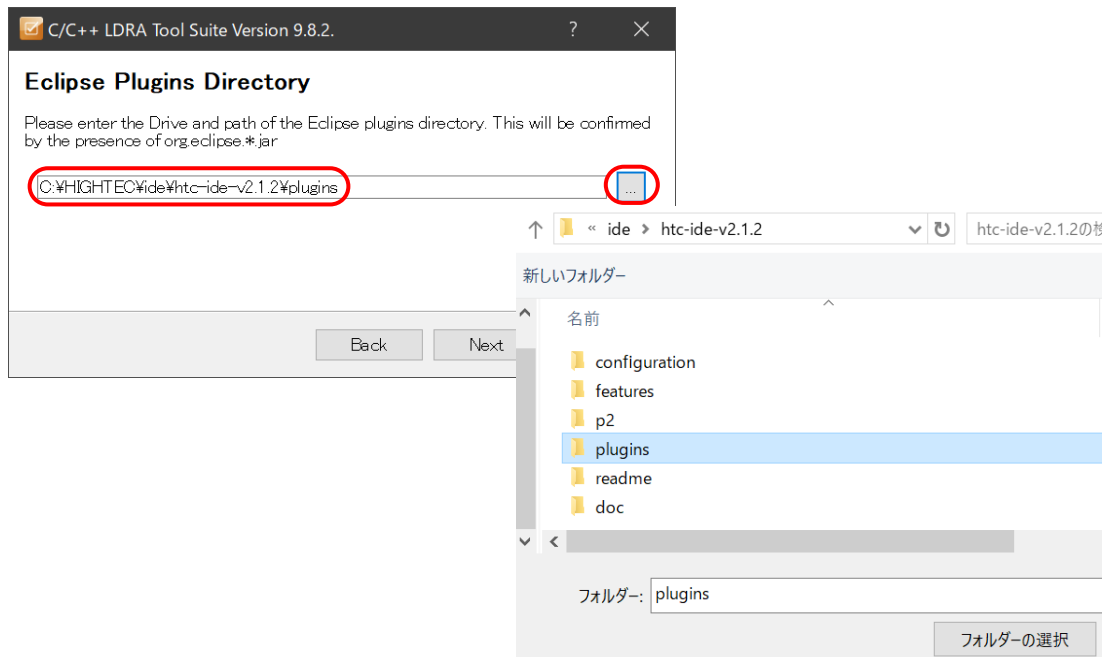
Step 1. LDRA ランチャーを起動して **Configure** をクリックし、Configure 画面から **Additional Tools** を選択します



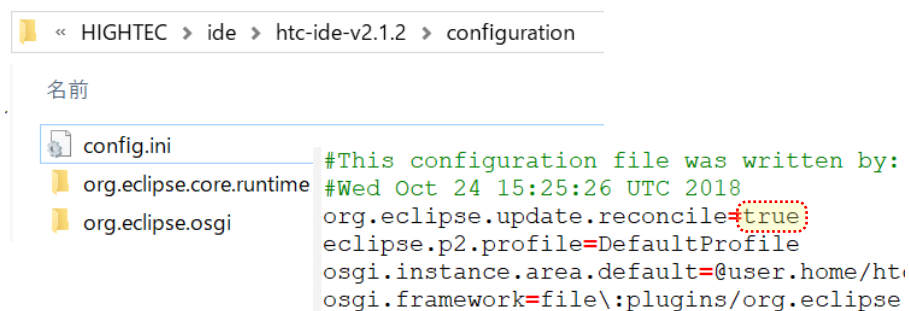
ダイアログで **Install LDRA Eclipse Plug-in ...** を選択し、Next します



HighTec Development Platform のプラグインディレクトリを選択します (インストールされている HighTec Development Platform に合わせてフォルダを選択し、設定します)



- Step 2. 次に、HighTec IDE で LDRA パースペクティブが有効になるようにします
設定した上記プラグインを使用できるようにするため、**config.ini** に次の変更を行います。
(org.eclipse.update.reconcile=true とします)



以上で、インストールと設定が終了します。



富士設備工業株式会社 電子機器事業部 www.fuji-setsu.co.jp