

# Helix QAC 利用手順書

2020.1 ~ 2022.3

“はかる”技術で未来を創る



## contents

---

<b>1</b>	<b>はじめに</b>	<b>1</b>
1.1	インストール	1
1.2	マニュアルについて	4
<b>2</b>	<b>サンプルプロジェクトの解析</b>	<b>5</b>
2.1	サンプルプロジェクトを開く	5
2.2	解析	6
2.3	解析結果の確認	7
<b>3</b>	<b>新規プロジェクトの作成</b>	<b>9</b>
3.1	はじめに	9
3.2	新規プロジェクトの作成	10
3.3	ファイルを登録する	12
3.4	プロジェクトプロパティの設定(基本)	13
3.5	解析	16
<b>4</b>	<b>プロジェクトプロパティの設定(詳細)</b>	<b>18</b>
4.1	データフロー解析(DFA)の設定	18
4.2	クロスモジュール解析(CMA)、マルチスレッド解析(MTA)の設定	19
4.3	コンプライアンスモジュールの設定	21
4.4	ルールの設定	22
<b>5</b>	<b>TOYO DEVELOPER PORTAL</b>	<b>23</b>
<b>6</b>	<b>お問い合わせ先</b>	<b>24</b>

# 1 はじめに

本書は Helix QAC の最も基本的な使用方法をまとめたものです。

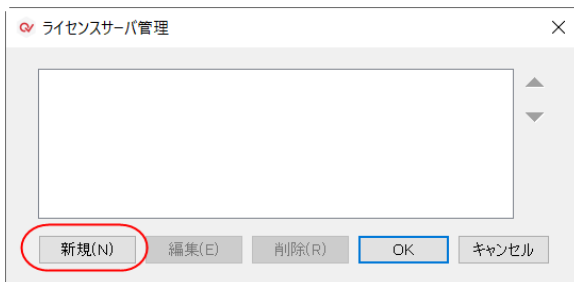
## ■ 1.1 インストール

Helix QAC に付属のインストールノートに従って Helix QAC をインストールします。  
MISRA C などのコンプライアンスモジュールをお使いの場合も同様にインストールします。

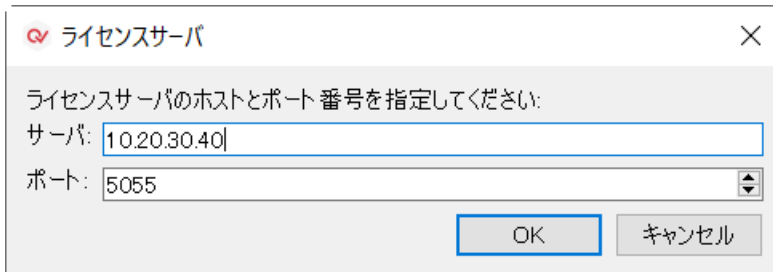
### ■ 1.1.1 ライセンスサーバの設定

先にライセンスサーバを起動しておく必要があります。ライセンスサーバの起動は  
TOYO DEVELOPER PORTAL から Helix QAC の FAQ に入り  
[ライセンス関連] > [ライセンス設定解説書] をご覧ください。  
※ TOYO DEVELOPER PORTAL については 5 章を参照ください

- 1 Helix QAC の初回起動時にライセンスサーバ設定画面が表示されます。  
[新規]を選びサーバを設定します。

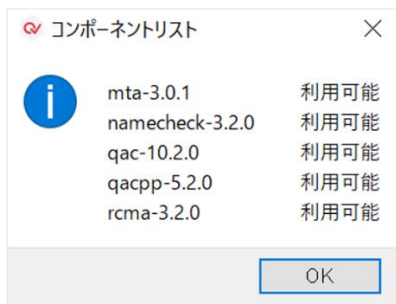


- 2 [サーバ]欄にホスト名もしくはIPアドレス、[ポート]欄に5055を入力して、「OK」を選びます。



※ローカルのコンピュータ内にライセンスサーバを構築した場合は、「127.0.0.1」を入力します。  
※ポート番号の「5055」はデフォルトです。ライセンスサーバの設定で変更した場合はその値を指定します。

### 3 正常に設定が完了した場合、使用可能なライセンスの情報(左側)が表示されます。



- 「利用可能(Available)」の状態は、製品の購入、ライセンスの発行状態により、個々に変わります。
- ライセンスが取得できなかった場合は「Not Available、ライセンスなし」と表示されます。

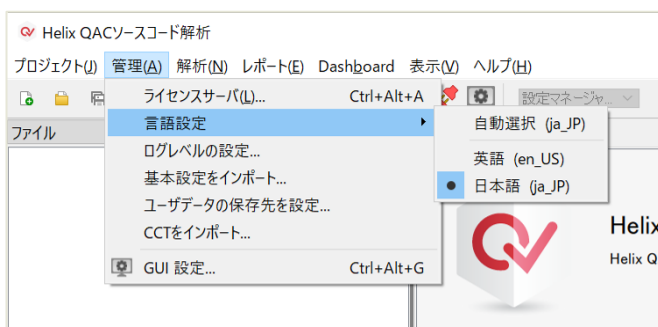
コンポーネント	説明
qac	C言語用アナライザです。QACのライセンスで動作します。
qacpp	C++言語用アナライザです。QAC++ のライセンスで動作します。
rcma	クロスモジュール解析(CMA)用コンポーネントです。QACまたはQAC++のライセンスで動作します。
mta	マルチスレッド解析(MTA)コンポーネントです。QACまたはQAC++のライセンスで動作します。
namecheck	命名規則チェック用コンポーネントです。QACまたはQAC++のライセンスで動作します。
m2cm	MISRA C:2004 コンプライアンスモジュールです。m2cmのライセンスで動作します。
m3cm	MISRA C:2012 コンプライアンスモジュールです。m3cmのライセンスで動作します。
certccm	CERT C コンプライアンスモジュールです。certccmのライセンスで動作します。
cweccm	CWE(C言語)コンプライアンスモジュールです。cweccmのライセンスで動作します。
mcpp	MISRA C++ コンプライアンスモジュールです。mcppのライセンスで動作します。
ascm	AUTOSAR コンプライアンスモジュールです。ascmのライセンスで動作します。
certcppcm	CERT C++ コンプライアンスモジュールです。certcppcmのライセンスで動作します。
cwecppcm	CWE(C++言語)コンプライアンスモジュールです。cwecppcmのライセンスで動作します。

### 4 [OK]を選び、画面を閉じて設定を終了します。

## 1.1.2 ユーザーインターフェース言語の切り替え

ユーザーインターフェース言語の切り替えることができます。

- 1 Helix QACを起動します。
- 2 メニューバーから [管理] > [言語設定] を選んで切り替えます。



- 3 設定完了後、Helix QAC を再起動して言語が切り替わったことを確認します。

- 日本語表示モードと英語表示モードの切り替えは、いつでも行うことができます。解析結果が存在する場合、再解析することなく結果の警告メッセージの言語を切り替えることができます。

## ■ 1.2 マニュアルについて

マニュアルは Helix QAC のインストールディレクトリ配下に生成されます。  
ここでは Helix QAC のインストールディレクトリを <Helix> と記します。

### ▶ Helix QAC コンポーネントマニュアル

GUI やコマンドからの使用方法についての説明です。

英語	<Helix>%common%doc-en_US%doc%manual
日本語	<Helix>%common%doc-ja_JP%doc%manual

### ▶ Helix QAC リリースノート

Helix QAC の旧バージョンからの変更点についての説明です。

英語	<Helix>%common%doc-en_US%doc%release_notes
日本語	<Helix>%common%doc-ja_JP%doc%release_notes

### ▶ コンポーネントマニュアル

QAC, QAC++ など各コンポーネントの機能についての説明です。

英語	<Helix>%components%<コンポーネント名>%doc-en_US%doc%component_manual
日本語	<Helix>%components%<コンポーネント名>%doc-ja_JP%doc%component_manual

### ▶ コンポーネントリリースノート

QAC, QAC++ など各コンポーネントの旧バージョンからの変更点についての説明です。

英語	<Helix>%components%<コンポーネント名>%doc-en_US%doc%release_notes
日本語	<Helix>%components%<コンポーネント名>%doc-ja_JP%doc%release_notes

### ▶ プラグインマニュアル (Visual Studio)

Visual Studioと連携して使用するための拡張機能についての説明です。

インストレーションノート、コンポーネントマニュアル、リリースノートがあります。

英語	<Helix>%ide_plugins%visual_studio%doc-en_US%doc
日本語	<Helix>%ide_plugins%visual_studio%doc-ja_JP%doc

### ▶ プラグインマニュアル (Eclipse)

Eclipseと連携して使用するための拡張機能についての説明です。

インストレーションノート、コンポーネントマニュアル、リリースノートがあります。

英語	<Helix>%ide_plugins%qa_eclipse%doc-en_US%doc
日本語	<Helix>%ide_plugins%qa_eclipse%doc-ja_JP%doc

# 2 サンプルプロジェクトの解析

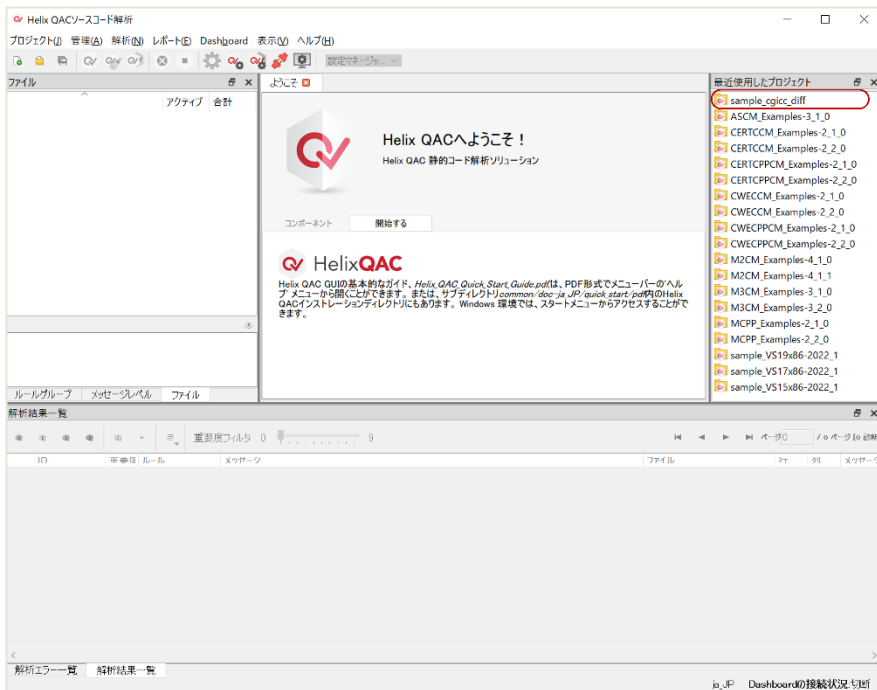
Helix QAC に付属のサンプルプロジェクトを用いて、解析方法と結果の確認方法を説明します。

- サンプルプロジェクトは、問題が発生したときの問題の切り分けとしてご利用いただく場合がございます。

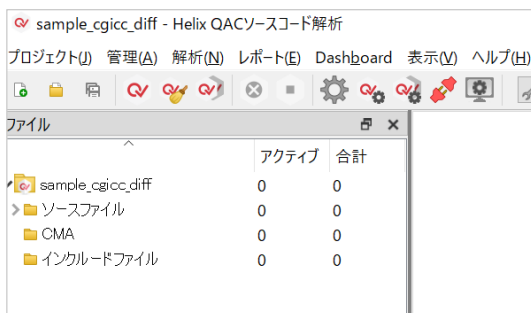
## 2.1 サンプルプロジェクトを開く

1 Helix QAC を起動します。

2 [最近使用したプロジェクト]の中から[sample\_cgicc\_diff]プロジェクトを選択します。



3 ファイルビューにフォルダが表示されることを確認します。

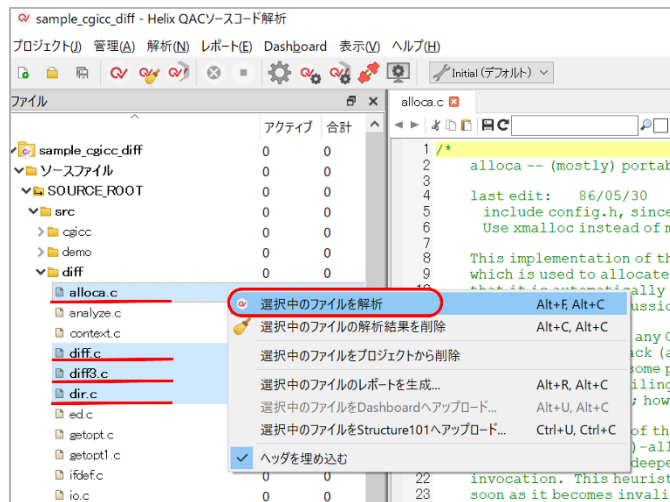


## 2.2 解析

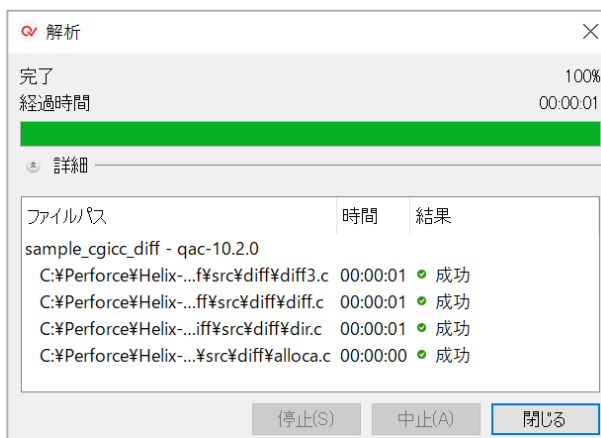
### 1 [ソースファイル]配下のフォルダを展開します。

QACをお使いの場合は diff、QAC++をお使いの場合は cgicc フォルダを選択ください。

### 2 展開後に現れるソースファイル上でファイルを選んで右クリック(1つ、または複数選択することが可能です)、メニューから[選択中のファイルを解析]を選びます。



### 3 解析完了後に、[成功]と表示されることを確認して、[閉じる]を選びます。

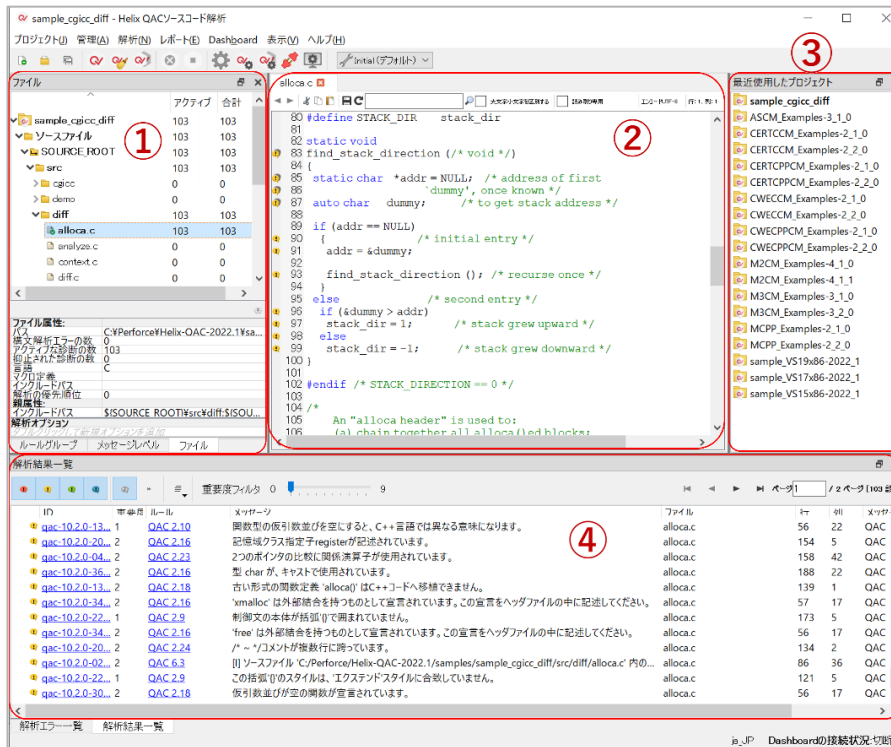


### 4 解析後、ソースコードビューと解析結果の一覧に結果が表示されます。



## 2.3 解析結果の確認

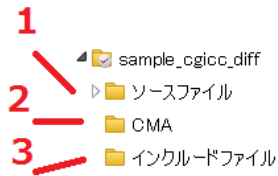
### 2.3.1 メイン画面



項目	説明
① ファイルビュー メッセージレベルビュー ルールグループビュー	<p>[ファイル] : 警告メッセージをファイル単位で確認します。</p> <p>[メッセージレベル] : 警告メッセージをメッセージ番号単位で確認します。</p> <p>[ルールグループ] : 警告メッセージをグループ単位で確認します。</p> <p>※ファイルビューで警告メッセージが出力されている箇所を選択していなければ、メッセージレベル、ルールグループには何も表示されません。</p>
② ソースコードビュー	<p>ソースコードを表示します。警告メッセージが出力されている箇所は左側のバルーンで示されます。バルーンをクリックすると警告メッセージがポップアップされます。</p>
③ 最近使用したプロジェクト	<p>最近使用したプロジェクトの一覧が表示されます。</p>
④ 解析結果一覧 解析エラー一覧	<p>[解析結果一覧]:</p> <p>①で選択したノードに応じて、解析結果の一覧を表示します。</p> <p>各行をダブルクリックすると、②に該当するソースコード行が表示されます。</p> <p>[解析エラー一覧]:</p> <p>開いているプロジェクト内で発生した解析エラーの一覧を表示します。</p>

## ■ 2.3.2 ファイルビュー内のノード

ファイルビュー内のノードは以下の構成になっています。



項目	説明
1. ソースファイル	解析対象として登録されたソースファイルです。
2. CMA	クロスモジュール解析(CMA)とマルチスレッド解析(MTA)による警告メッセージのうち、問題の箇所を1箇所に特定できない警告メッセージが表示されます。CMA, MTA は 4.2 で説明します。
3. インクルードファイル	ソースファイルからインクルードされているファイルです。

# 3

## 新規プロジェクトの作成

プロジェクトの作成、解析対象ソースファイルの追加、解析、解析結果の確認までの最も基本的な方法を説明します。



QAC/QAC++は、コンパイルはエラーなく完了するコードを解析することを前提とします。

### 3.1 はじめに

Helix QAC (QAC/QAC++) は、3つの設定ファイルを使用します。

- 解析設定ファイル(ACF)
- ルール設定ファイル(RCF)
- コンパイラ設定ファイル(CCT)

#### 3.1.1 解析設定ファイル(ACF)

ACF には、使用する解析コンポーネントや解析オプションを設定します。標準状態では default.acf だけが選択可能です。default.acf には、解析コンポーネント qac と qacpp だけが設定された状態です。default.acf を選択し、後の 3.4 節、および 4 章で調整します。

#### 3.1.2 ルール設定ファイル(RCF)

RCF には、使用する警告メッセージ、および警告メッセージのグループ分けを設定します。標準状態では、以下の3種類のファイルが準備されています。初めて QAC/QAC++ を使用する場合は、クリティカルファイルをお使いになることをお勧めします。後の 4.4 節で調整することができます。

・クリティカルファイル(default-critbug-ja\_JP.rcf)

メーカーが選定した特に重大な不具合に関する警告メッセージだけを使用する設定としたファイルです。

・セキュリティファイル(default-security-ja\_JP.rcf)

メーカーが選定したセキュリティに関する警告メッセージだけを使用する設定としたファイルです。

・デフォルトファイル(default-ja\_JP.rcf)

すべての警告メッセージを使用する設定としたファイルです。

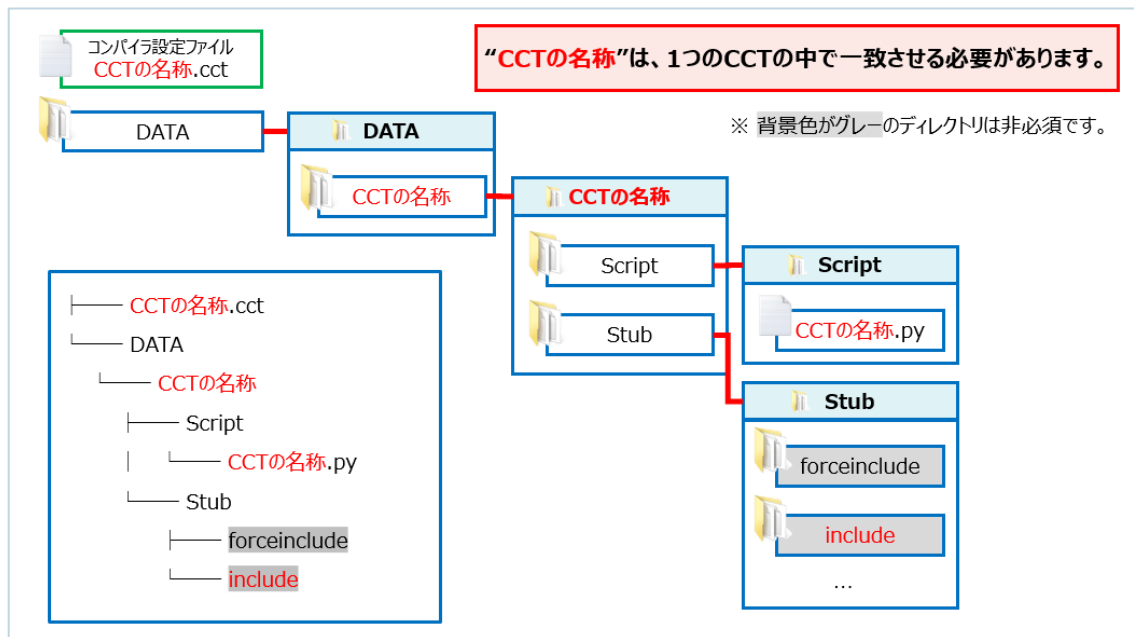
#### 3.1.3 コンパイラ設定ファイル(CCT)

CCT には、お使いのコンパイラの特性によって決まる設定を行います。お使いのコンパイラによっては、CCTの雛形ファイルが準備されている場合や、雛形ファイルを生成するツールが用意されている場合がございます。使用すべきCCTがご不明でしたら、**お使いのコンパイラとバージョンを弊社サポート宛て(ss\_support@toyco.co.jp)までご連絡ください。**

また、旧バージョン（QAC 8.1.2 / QAC++ 3.1. 以前）用のコンパイラパーソナリティをお持ちの場合、TOYO DEVELOPER PORTAL から Helix QAC の FAQ に入り [設定] > [コンパイラ用設定ファイル(CCT)] > [コンパイラパーソナリティからCCTを作成する] に記載の方法で CCT を作成することができます。

※ TOYO DEVELOPER PORTAL については 5 章を参照ください

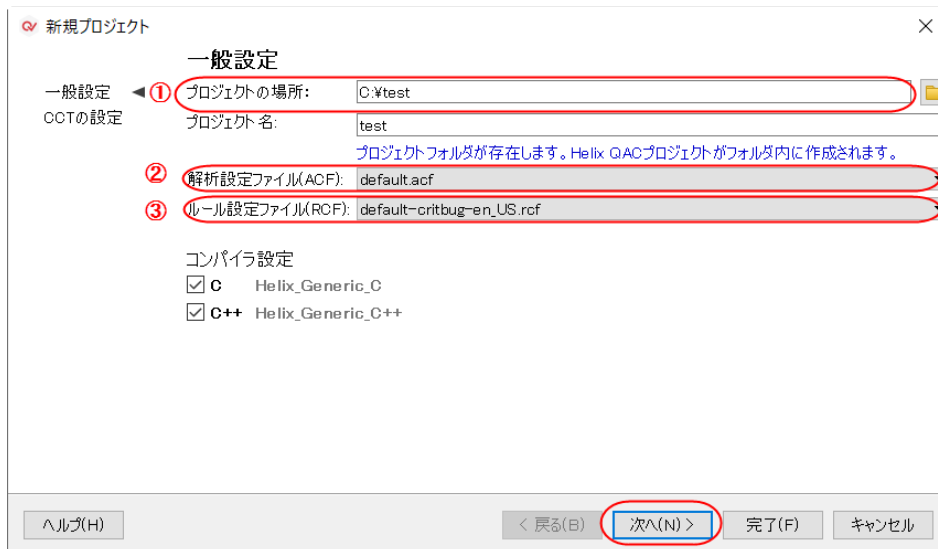
**CCT には下図に示す構造の DATA ディレクトリが付属している必要があります。**



## 3.2 新規プロジェクトの作成

新規にプロジェクトを作成します。作成した内容は、後で変更することが可能です。

- 1 Helix QAC を起動します。
- 2 メニューバーから [プロジェクト] > [新規プロジェクト] を選びます。
- 3 「一般設定」において、次のように設定し、「次へ」を押します。



項目	内容
① プロジェクトの場所	プロジェクトを作成するディレクトリを指定します。
② 解析設定ファイル(ACF)	3.1.1 を参照ください。
③ ルール設定ファイル(RCF)	3.1.2 を参照ください。

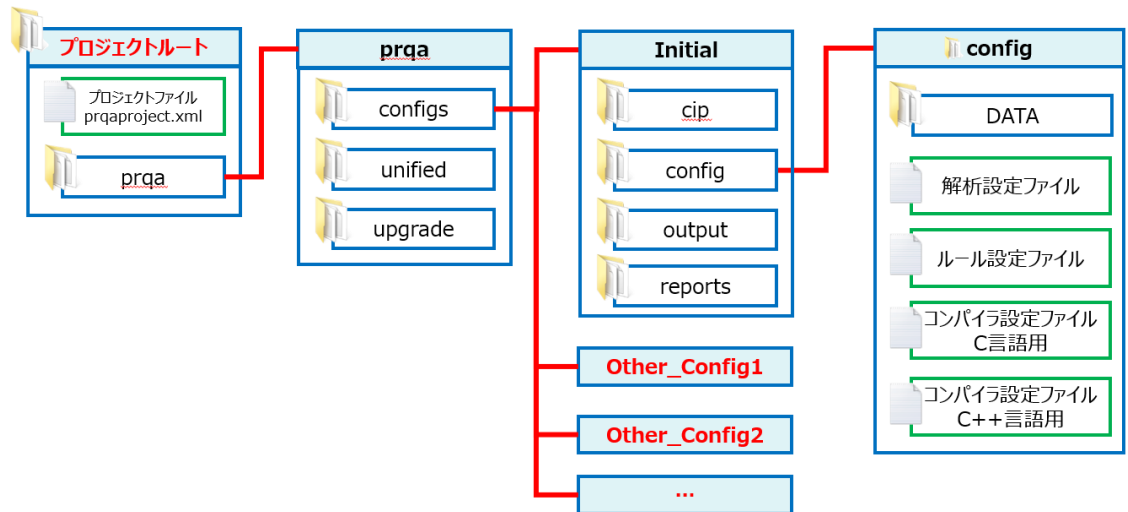
#### 4 お使いのコンパイラに対応する CCT を選択します。

CCTについては 3.1.3 を参照ください。

- ・ C言語だけのプロジェクトを解析する場合は、「C言語用のCCT」で CCT を選択し、「完了」を押します。
- ・ C++言語だけのプロジェクトを解析する場合は、「C言語用のCCT」では標準状態のまま「次へ」を押し、「C++言語用のCCT」で CCT を選択し、「完了」を押します。
- ・ C/C++両方を含むプロジェクトを解析する場合は、「C言語用のCCT」と「C++言語用のCCT」の両方で CCT を選択し、「完了」を押します。



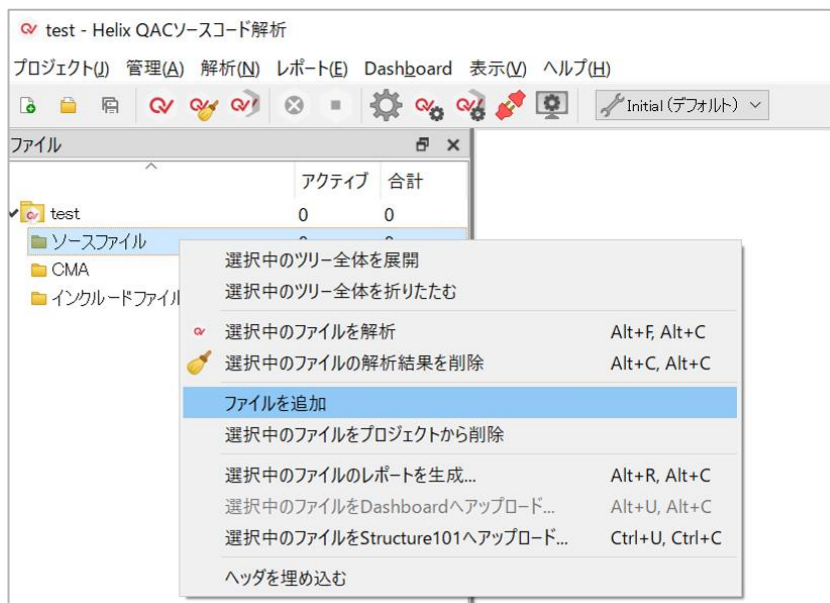
- 5** 下図の構造でプロジェクトが作成されます。  
 プロジェクト作成時に指定した設定ファイルは  
`<プロジェクトディレクトリ>%prqa%configs%<設定名>%config`  
 配下にコピーされて使用されます。  
 ※ <設定名>のデフォルトは Initial です



### 3.3 ファイルを登録する

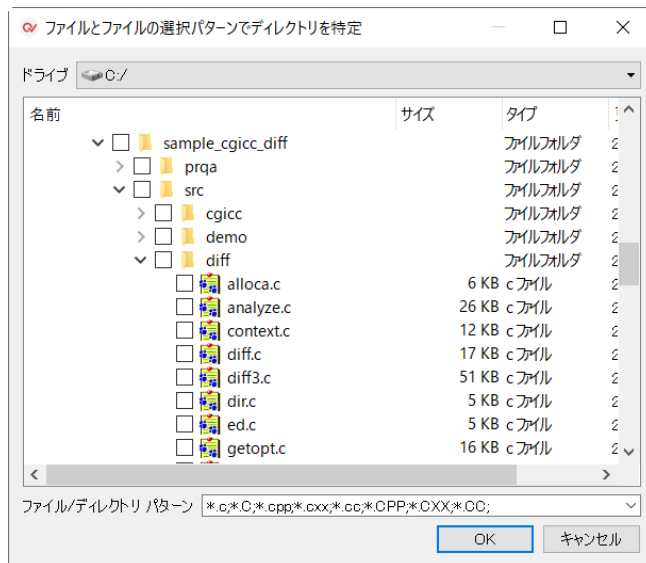
プロジェクトに解析対象のソースファイルを登録します。

- 1** [ソースファイル] を右クリック、ポップアップメニューから [ファイルを追加] を選びます。



## 2 登録したいファイルを選択し [OK] を押します。

ディレクトリを選択すると、その配下にある特定の拡張子をもつファイルが全て選択されます。



### 手動登録以外の方法について

ビルドを実行する、あるいはビルドログを読むことで、ビルド対象のファイルを解析対象として登録することも可能です。詳しくは、Helix QAC コンポーネントマニュアルの「7 Project Synchronization(プロジェクトの同期)」をご覧ください。

## 3.4 プロジェクトプロパティの設定(基本)

解析を行う上で重要な設定について説明します。

プロジェクトプロパティは [プロジェクト] > [プロジェクトプロパティ] を選択します。

### 3.4.1 文字コードの設定

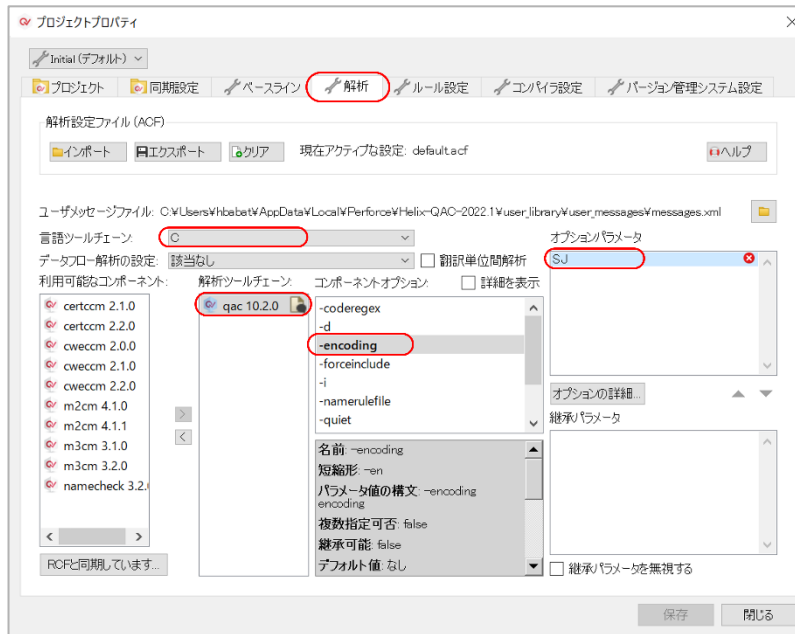
プロジェクトのソースコードの文字コードを設定します。



正しく設定できていない場合、GUI上で文字化けが発生するだけでなく、解析エラーが発生するなど正しく解析できない原因になります

- 1) プロジェクトプロパティの解析タブを開きます
- 2) “言語ツールチェーン”で C または C++、解析ツールチェーンで qac または qac++ を選択します
- 3) -encoding オプションを選択し、[オプションパラメータ]に以下の値を設定します

SJ	ソースコードの文字コードがShift-JISの場合
EUC-JP	ソースコードの文字コードがEUCの場合
UTF-8	ソースコードの文字コードがUTF-8(BOMつき/BOMなし)の場合



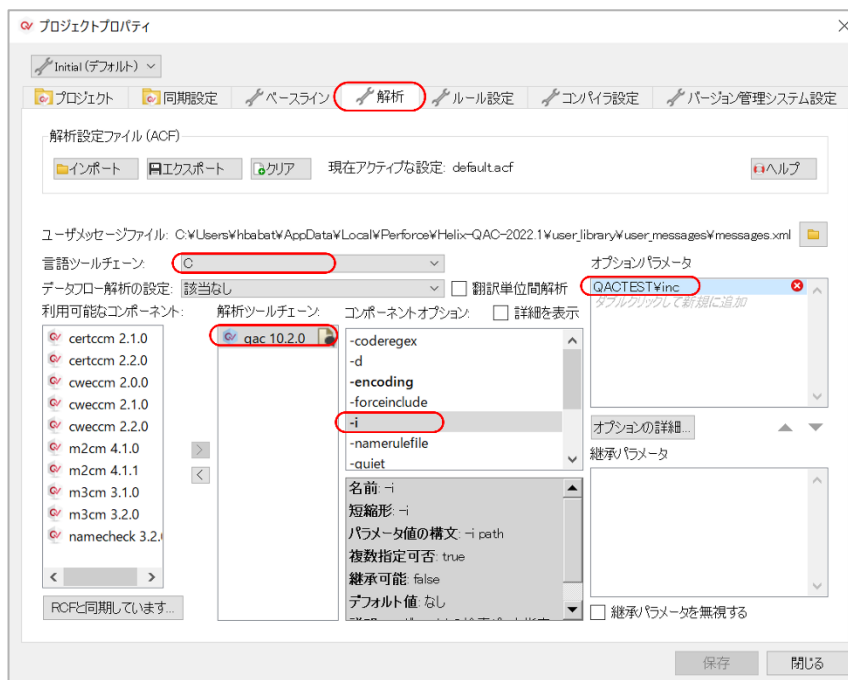
### 3.4.2 インクルードファイルの検索パスの設定

コンパイル時にコンパイラに与えているインクルードファイルの検索パスを設定します。



正しく設定ができていない場合、解析エラーが発生するなど正しく解析できない原因になります

- 1) プロジェクトプロパティの「解析タブ」を開きます
- 2) 言語ツールチェーンで C または C++、解析ツールチェーンで qac または qac++ を選択します
- 3) -i オプションを選択し、[オプションパラメータ]にインクルードファイルの検索ディレクトリを設定します  
 ※コンパイル時に複数の検索パスを与えている場合は、コンパイル時と同順で設定します





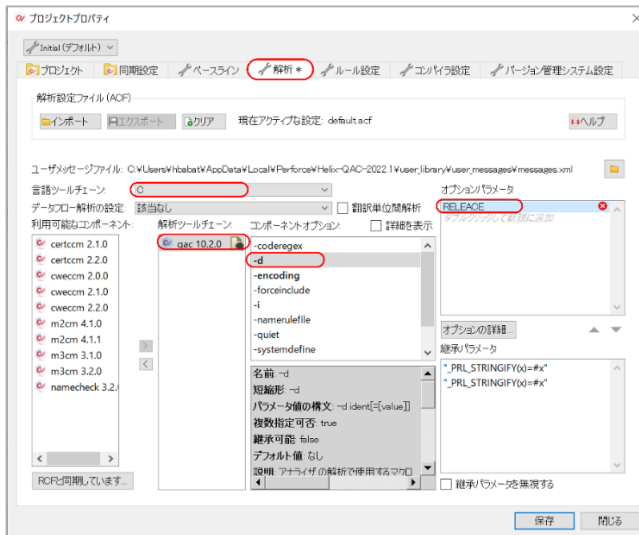
### 3.4.3 マクロの設定

コンパイル時にコンパイラに与えているマクロを設定します。



正しく設定ができていない場合、解析エラーが発生するなど正しく解析できない原因になります

- 1) プロジェクトプロパティの解析タブを開きます
- 2) 言語ツールチェーンで C または C++、解析ツールチェーンで qac または qac++ を選択します
- 3) -d オプションを選択し、[オプションパラメータ]にマクロを定義します



### 3.4.4 データフロー解析(DFA)の設定

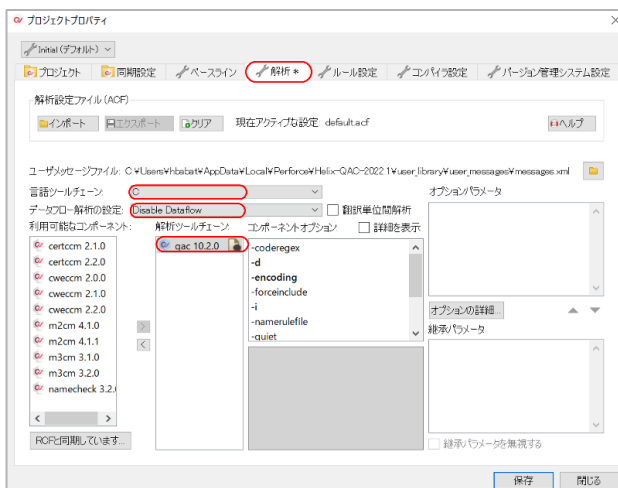
エラーなく解析が完了するまでは、DFAは無効に設定します。



解析エラーが発生する場合は、DFAを実施することはできません。

※エラーなく解析が完了となり、DFAを有効にする場合は 4.1 を参照ください

- 1) プロジェクトプロパティの解析タブを開きます
- 2) 言語ツールチェーンで C または C++、解析ツールチェーンで qac または qac++ を選択します
- 3) データフロー解析の設定で [Disable Dataflow] を選択します



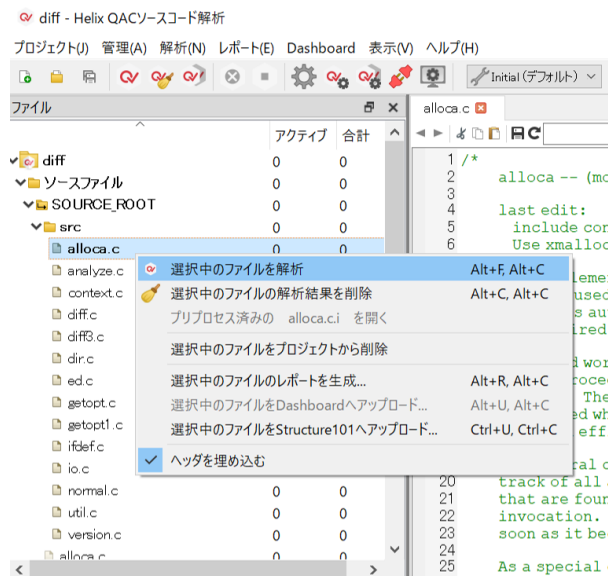
## 3.5 解析

解析を実施し、解析結果を確認します。

### 3.5.1 1つのファイルを解析する

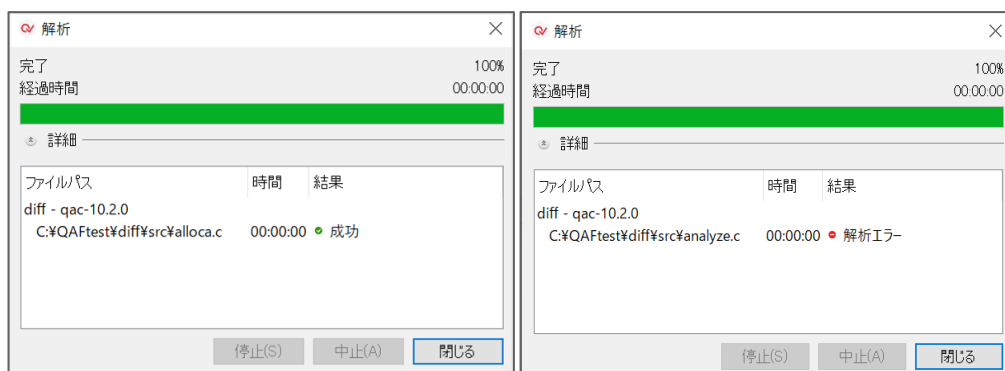
#### 1 プロジェクト内のソースファイルを1つ選択し、右クリックから【選択中のファイルを解析】を実施します。

※ 複数のファイルを選択して解析することや、プロジェクト全体を選択して解析することも可能ですが、いくつかのファイルについてエラーなく解析完了となるまでは1ファイルずつ解析することをお勧めします



#### 2 解析が実施され、結果が表示されます。

エラーなく解析が完了した場合は下図左のように「成功」、エラーが発生した場合は下図右のように「解析エラー」と表示されます。



#### 3 解析結果を確認します。

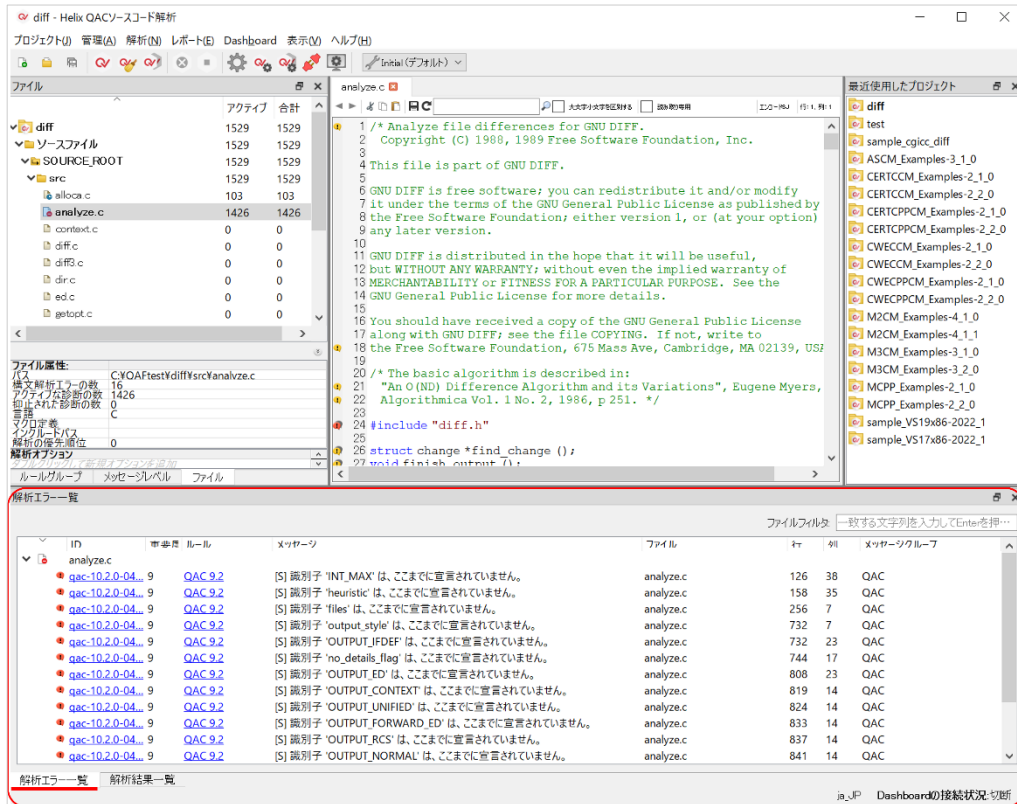
結果の確認方法は 2.3 を参照ください。

プロジェクト内の解析対象のソースファイルすべてにおいて、エラーなく解析完了となりましたら、データフロー解析(DFA)、クロスモジュール解析(CMA)、マルチスレッド解析(MTA)、コンプライアンスモジュールによる2次解析を実施することができます。詳しくは 4 章で説明します。

## 3.5.2 解析エラーが発生した場合

解析エラーが発生した場合は、画面左下の【解析エラー一覧】で確認します。

解析エラーは解析が正しく完了していないことを表します。最初に検出された箇所以外のメッセージは信頼性の低いものになりますので、必ず最初に検出されたエラーから順にエラーを解消する必要があります。



エラーの除去方法がご不明でしたら、弊社サポート宛て([ss\\_support@toyo.co.jp](mailto:ss_support@toyo.co.jp))までお問い合わせください。その際は、可能な範囲でエラーの状況をお知らせください。

項目	内容
1 エラーの内容	最初に発生しているエラーメッセージ番号と、その箇所のコードの記述をお知らせください。
2 各設定ファイル(.oct/.acf/.rcf)	プロジェクトディレクトリ下の「prqa¥configs¥<設定名※1>¥config」にコピーされて使用されています。3.2の5の図をご覧ください。
3 プリプロセス済みファイル	プリプロセス済みファイルを出力し、最初にエラーが発生している箇所がプリプロセス済みファイルではどのように展開されているかお知らせください。プリプロセス済みファイルについては、TOYO DEVELOPER PORTAL (※2) から Helix QAC の FAQ に入り [ツール操作] > [プリプロセス済みファイルを生成する] をご覧ください。
4 解析時の設定ファイル	TOYO DEVELOPER PORTAL (※2) から Helix QAC の FAQ に入り [設定] > [解析時に使用された設定ファイル] をご覧ください

※1 : <設定名> のデフォルトは Initial です

※2 : TOYO DEVELOPER PORTAL については 5 章を参照ください

# 4 プロジェクトプロパティの設定(詳細)

解析がエラーなく完了しましたら追加の解析を行うことができます。それらの詳細設定について説明します。プロジェクトプロパティは [プロジェクト] > [プロジェクトプロパティ] で開きます。

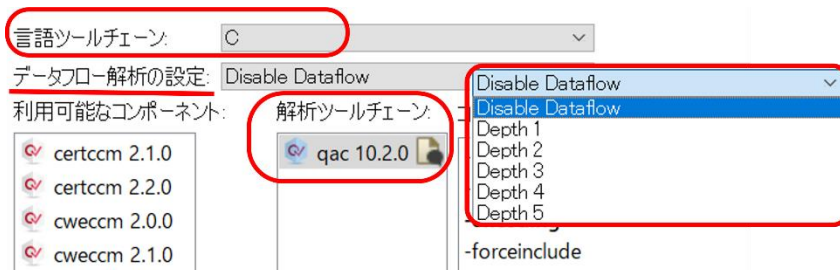
## 4.1 データフロー解析(DFA)の設定

DFA について詳しくは、TOYO DEVELOPER PORTAL から Helix QAC の FAQ に入り [解析] > [データフロー解析(DFA)] > [DFAとは] をご覧ください。  
※ TOYO DEVELOPER PORTAL については 5 章を参照ください

1 プロジェクトプロパティの解析タブで、言語ツールチェーン C または C++、解析ツールチェーン qac または qac++ を選択します。

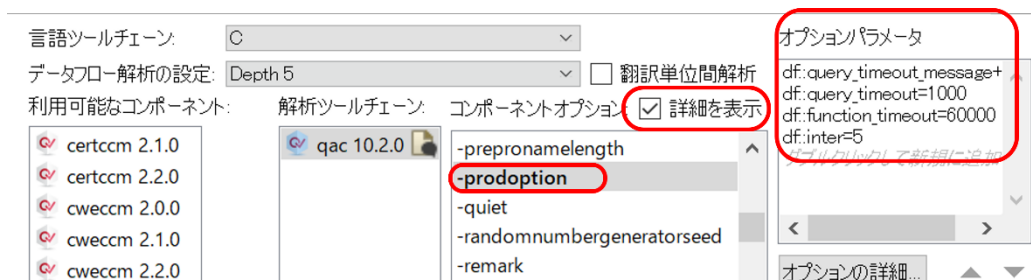
2 「データフロー解析の設定」で Depth 1 ~ 5 のいずれかを選択します。

※数値が大きくなるに従って、深く長時間かけて解析する設定になります。



3 設定値は prodoption で確認、調整することができます。

※prodoptionを表示するには「詳細を表示」を選択しておく必要があります。



4 設定後、再度解析を実施しますと DFA も含めて解析が実施されます。

DFAによる警告メッセージは、TOYO DEVELOPER PORTAL から Helix QAC の FAQ に入り [警告メッセージ] > [データフロー解析(DFA)のメッセージ] をご覧ください。  
※ TOYO DEVELOPER PORTAL については 5 章を参照ください

## 4.2 クロスモジュール解析(CMA)、マルチスレッド解析(MTA)の設定

CMA、MTA については TOYO DEVELOPER PORTAL から Helix QAC の FAQ に入り  
 [解析] > [クロスモジュール解析(CMA)] > [CMAとは]  
 [解析] > [マルチスレッド解析(MTA)] をご覧ください。  
 ※ MTAを実施するにはデータフロー解析(DFA)を実施する必要があります(4.1参照)  
 ※ TOYO DEVELOPER PORTAL については 5 章を参照ください

- 1 プロジェクトプロパティの解析タブで言語ツールチェーンに「C/C++に共通するコンポーネント」を選択します。

言語ツールチェーン: C/C++に共通するコンポーネント

データフロー解析の設定: ツールチェーンのコンポーネントを選択

利用可能なコンポーネント: 解析ツールチェーン: コンポーネントオプション:

<input checked="" type="checkbox"/> mta 3.0.1		
<input checked="" type="checkbox"/> rcma 3.2.0		

- 2 「利用可能なコンポーネント」で rcma または mta を選択し右矢印を押します。

言語ツールチェーン: C/C++に共通するコンポーネント

データフロー解析の設定: ツールチェーンのコンポーネントを選択

利用可能なコンポーネント: 解析ツールチェーン: コンポーネントオプション:

<input checked="" type="checkbox"/> mta 3.0.1		
<input checked="" type="checkbox"/> rcma 3.2.0		

>

- 3 「解析ツールチェーン」に選択したコンポーネントが追加されたことを確認します。

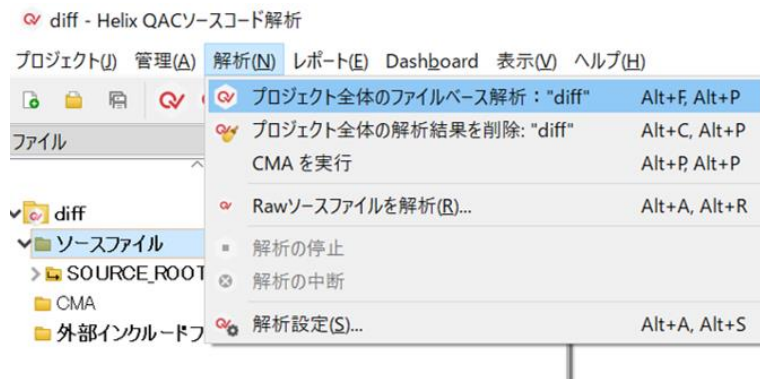
言語ツールチェーン: C/C++に共通するコンポーネント

データフロー解析の設定: ツールチェーンのコンポーネントを選択

利用可能なコンポーネント: 解析ツールチェーン: コンポーネントオプション:

<input checked="" type="checkbox"/> mta 3.0.1	<input checked="" type="checkbox"/> rcma 3.2.0	
---	--	--

#### 4 設定後 [解析] > [プロジェクト全体のファイルベース解析] を実施しますと CMA、MTA も含めて解析が実施されます。



#### 5 CMA、MTA による警告メッセージには問題の箇所を1箇所に特定できるものと、特定できないものが存在します。

特定できる警告メッセージはソースファイルノード下、特定できない警告メッセージはCMAノード下に表示されます。

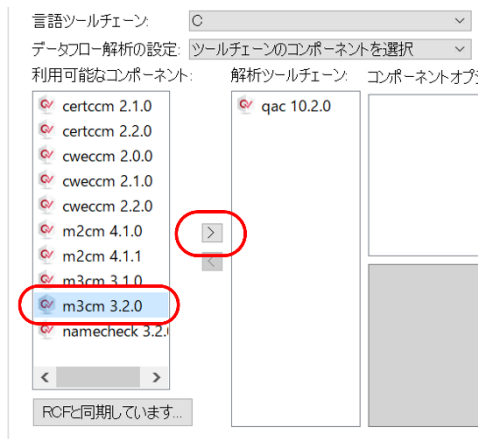
詳しくは TOYO DEVELOPER PORTAL から Helix QAC の FAQ に入り  
 [警告メッセージ] > [multi-homedメッセージ] をご覧ください。  
 ※ TOYO DEVELOPER PORTAL については 5 章を参照ください



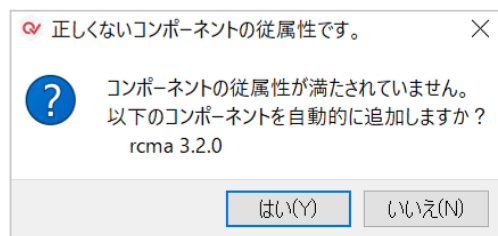
## 4.3 コンプライアンスモジュールの設定

MISRA C や CERT C などのコンプライアンスモジュールをお使いの場合、QAC/QAC++ が検出する警告メッセージに加えて、コンプライアンスモジュールが検出する警告メッセージも使用します。ここでは、解析にコンプライアンスモジュールを追加する方法を説明します。

- 1 プロジェクトプロパティの解析タブで言語ツールチェーンに C または C++ を選択します。
- 2 「利用可能なコンポーネント」でコンプライアンスモジュールを選択し右矢印を選択します。

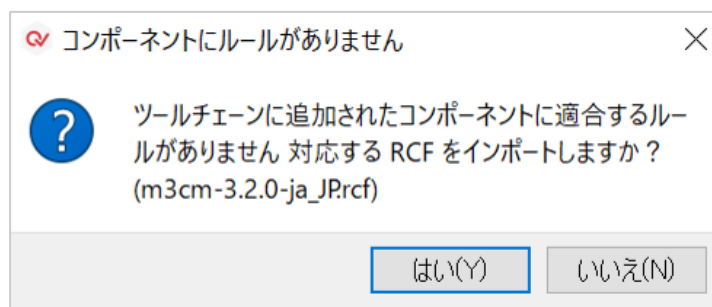


- 3 追加しようとしたコンポーネントを使用するために必要なコンポーネントがまだ追加されていない場合、下図のように追加することを促されます。「はい」を選択することをお勧めします。



- 4 追加しようとしたコンポーネントで検出される警告メッセージがルール設定で有効になっていない場合、下図のようにデフォルト設定のインポートを促されます。「はい」を選択することをお勧めします。

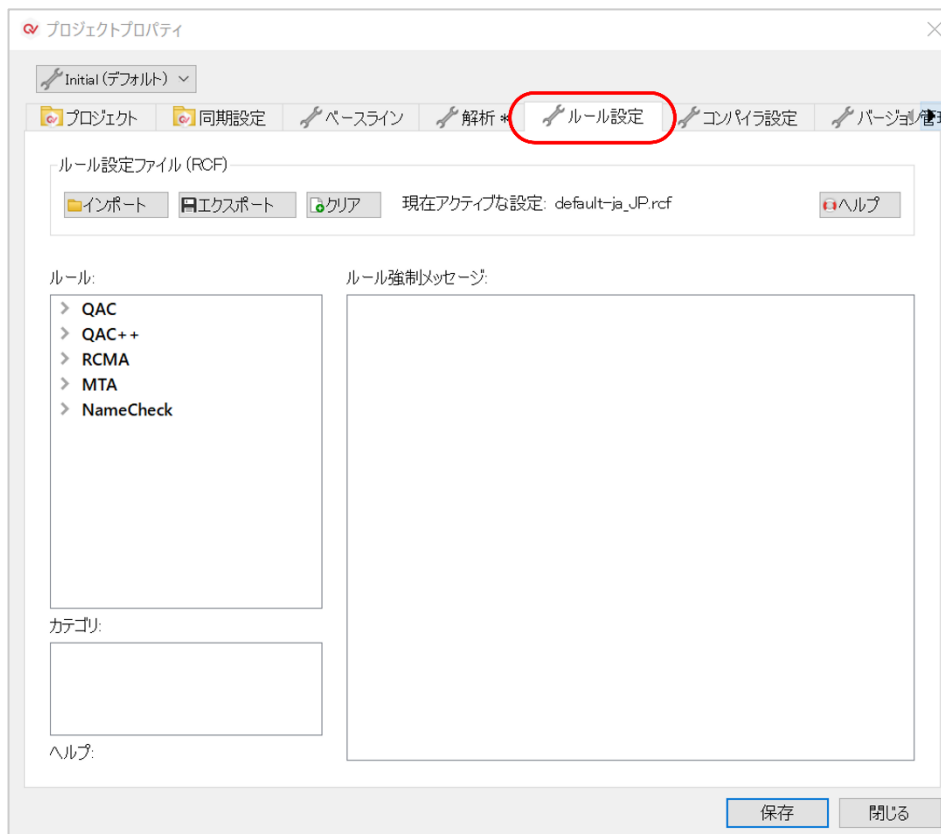
※ルール設定については4.4をご覧ください



## 4.4 ルールの設定

プロジェクトプロパティのルール設定タブで、使用するルール設定ファイル(RCF)の変更や、ルール設定の調整を行うことができます。

詳しくは TOYO DEVELOPER PORTAL から Helix QAC の FAQ に入り  
[設定] > [ルール設定ファイル(RCF)] 下の各項目をご覧ください。  
※TOYO DEVELOPER PORTAL については 5 章を参照ください





# 5 TOYO DEVELOPER PORTAL

本書の文中に TOYO DEVELOPER PORTAL に言及している箇所があります。  
TOYO DEVELOPER PORTAL には弊社取り扱い製品についてのナレッジを掲載しております。  
TOYO DEVELOPER PORTAL の情報は随時追加してまいります。ぜひご利用ください。

## TOYO DEVELOPER PORTAL

<https://kb.toyo.co.jp/wiki/display/KBTOP>

※Helix QACのページにアクセスするにはアカウントを登録いただく必要がございます

# 6 お問い合わせ先

本書の内容に関するご質問がございましたら、  
お気軽にメール/電話/FAXにてご連絡下さい。

---

<b>メールでのお問い合わせ先</b>	製品の使用方法などに関するお問い合わせ： <b>ss_support@toyota.co.jp</b>
	価格などに関するお問い合わせ： <b>ss_sales@toyota.co.jp</b>

---

<b>電話でのお問い合わせ先</b>	03-3245-1248 (直通)
--------------------	-------------------

---

<b>FAXでのお問い合わせ先</b>	03-3246-0645
---------------------	--------------

---