

CasePlayer2 チュートリアル

Ver. 1.0.4
2015/06/10

CasePlayer2 チュートリアル	3
はじめに	3
CasePlayer2 概要	3
稼働環境	3
1. CasePlayer2 起動	4
2. 新規プロジェクト作成	4
その他の設定について(プロジェクトの新規作成ダイアログ)	6
プロジェクトの種類	6
オプションパラメータファイルの指定	6
言語設定	6
C ソースに関する設定	6
C++ソースに関する設定	6
アセンブラソースに関する設定	6
階層化を有効にする	6
プロジェクトの設定データファイル	6
3. ソースコードの登録	7
その他の設定について (ソースファイルの登録ダイアログ)	8
サブフォルダも含む	8
ユーザインクルードパスに追加する	8
ファイル構成に合わせてフォルダ付きで登録する	8
4. 仕様書生成(標準モード)	8
仕様書閲覧	10
チャートリンク機能	11
アセンブラソースのフローチャート	13
5. 詳細解析モード	15
詳細解析設定	15
仕様書生成	16
仕様書閲覧	17
変数参照情報の閲覧	19
6. MISRA-C ルールチェック	20
MISRA-C ルールチェックを実行	20
表示オプション	22
ルールの選択	22
ソースコード・メトリックス閲覧	23
7. 検索機能	25
8. 印刷機能	27
9. 仕様書の Word 変換	28
10. 仕様書の HTML 変換	30
11. 詳細解析モードの詳細設定について	32
プリプロセッサの設定	32
#define (デファイン名 [=数値])	32
#undef (デファイン名)	32
プレインクルードファイル	32
システムインクルードパス/ ユーザーインクルードパス	33
C オプションパラメータ	33
User Keyword ユーザーキーワード	33
置き換えキーワード	34
C オプションパラメータのその他の設定について	34
12. 最後に	34

CasePlayer2 チュートリアル

はじめに

この度は、仕様書作成・解析ツール、CasePlayer2 にご興味を頂き誠にありがとうございます。本チュートリアルは、初めて CasePlayer2 をご使用頂くお客様にサンプルプログラムを使用しながら、実際にツールをご操作頂き、基本的な使用方法と機能を理解して頂く事を目的として作成させて頂いております。

チュートリアル開始の前に CasePlayer2 をインストールして頂きますようお願い致します。CasePlayer2 はデフォルトでは”C:\¥Program Files¥gaio¥Caseplayer2”にインストールされます。

CasePlayer2 概要

CasePlayer2 は、ANSI 準拠 C 言語、C++、組み込み向け C 言語(非 ANSI)を解析して、フローチャートなどの仕様書を作成するツールです。「仕様書ブラウザ」を搭載し、ソースコードと各仕様書間の連携を自由に行うことができます。過去のソースコードのロジック解析や、新たに開発したプログラムソースの仕様書作成作業を強力に支援致します。ソースコードの静的解析機能として、「外部変数」の参照/代入の一覧作成機能や、C 言語コーディング規約「MISRA-C」のルールチェック機能も搭載しています。

The image displays the CasePlayer2 software interface with several key components highlighted by arrows and labels:

- 仕様書生成** (Specification Generation): An arrow points from the source code editor to the flowchart and various diagrams.
- 静的解析** (Static Analysis): An arrow points from the source code editor to the external variable reference list.
- ソースコードエディタ・仕様書ブラウザ** (Source Code Editor & Specification Browser): The main interface showing code and its corresponding specifications.
- フローチャート** (Flowchart): A visual representation of the program's logic.
- 各種構造図** (Various Structure Diagrams): A collection of different types of diagrams used in specifications.
- 外部変数参照リスト** (External Variable Reference List): A table listing variables used or assigned in the code.
- MISRA-Cルール解析** (MISRA-C Rule Analysis): A table showing the results of rule checks against the MISRA-C coding standard.

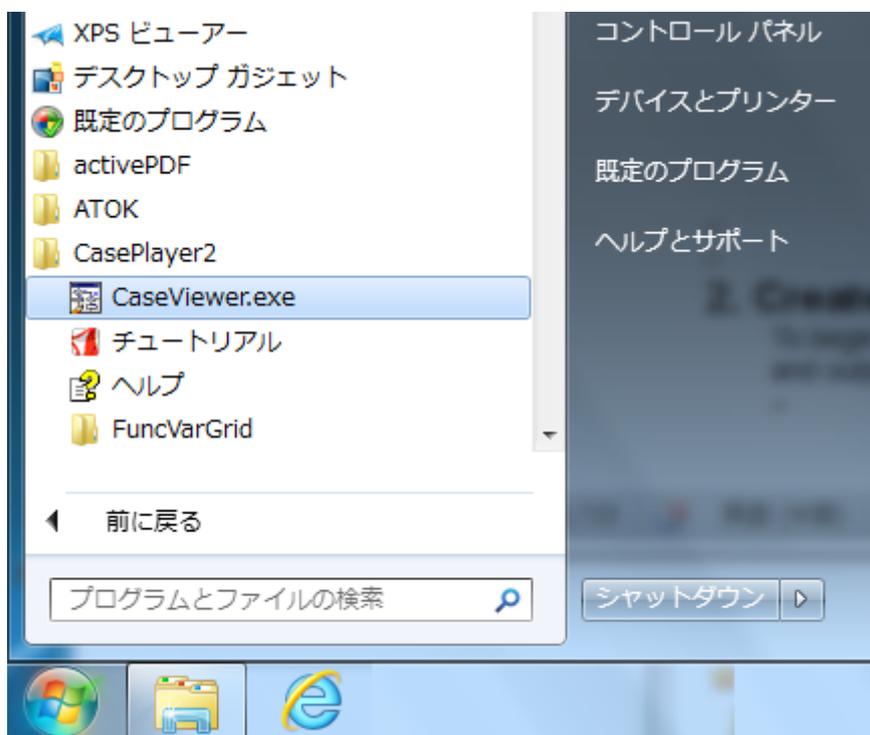
ID	ルール
<input checked="" type="checkbox"/>	1 すべてのコードは ISO 9899 ぞ
<input checked="" type="checkbox"/>	2 C言語以外で書かれたコードは
<input checked="" type="checkbox"/>	3 C言語から呼び出されるアセン
<input checked="" type="checkbox"/>	4 適切な実行時チェックのための
<input checked="" type="checkbox"/>	5 ISO9899で定義されている文字

稼働環境

- * IBM PC/AT 互換機
- * USB ポート (ライセンスキーの装着に使用します)
- * Windows XP, Windows Vista (32ビット), Windows 7 (32/64ビット), Windows 8.1 (32ビット、64ビット)
- * インストールに必要な HDD 容量:: 80MB
- * HTML フォーマット文書の閲覧: Microsoft Internet Explorer 6 以上
- * MS-Word 形式文書出力: Microsoft Word 2000 SP3 以降
Microsoft Word 2002 SP3 以降
Microsoft Word 2003 SP3 以降
Microsoft Word 2007 SP2 以降
Microsoft Word 2010 SP1 以降

1. CasePlayer2 起動

Windows のスタートメニュー 「全てのプログラム」 →「CasePlayer2」→「CaseViewer.exe」をクリックして下さい。



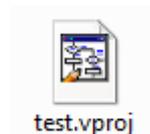
2. 新規プロジェクト作成

ソースコード解析、仕様書生成実行のために CasePlayer2 の新規プロジェクトを作成しましょう。

1. CasePlayer2 メニュー 「ファイル」→「新規作成」→「プロジェクト」をクリックして下さい。
2. プロジェクト名 : test
3. 場所: (任意)
4. プロジェクトの種類:: 仕様書生成のみ
5. 言語設定:: ANSI-C
6. オプションパラメータファイルの指定:[アセンブラオプションパラメータファイル]にチェックを付けてください。
[参照]ボタンで下記ファイルを指定してください。
7. アセンブラオプションパラメータファイル:C:\Program Files\gaio\CasePlayer2\sample\Z80\Z80.rev
8. 上記以外の設定はデフォルト設定のままにしてください。
9. [OK]ボタンをクリックしてください。



上記、3.場所: (任意)で選択して頂いたパスにプロジェクトフォルダが作成され、そのフォルダ配下にはプロジェクトファイル(test.vproj)と関連ファイルが配置されます。test.vproj をダブルクリックする事で CasePlayer2 を起動する事が可能です。



その他の設定について(プロジェクトの新規作成ダイアログ)

プロジェクトの種類

仕様書生成のみ: 本モードではソースコードから各種仕様書を生成することができます。
(チュートリアルではこちらを使用します)

概要説明をソースに戻す: 本モードでは上記の[仕様書生成のみ]に加え、仕様書に書き込んだコメントをソースファイルのコメントとしてソースに戻すことができます。[ソースファイルをコピーする]が ON になっている場合、コメントの反映はプロジェクト内にコピーされたソースに反映されます。

オプションパラメータファイルの指定

C オプションパラメータファイル:

C オプションパラメータファイルは C 言語のコンパイラ依存の表記などを定義します。
ANSI-C 規格に準拠している場合には設定する必要はありません。

アセンブラオプションパラメータファイル:

アセンブラオプションパラメータファイルは主にアセンブル言語の命令を定義します。
ガイオでは 40 以上のマイクロプロセッサのパラメータファイルをサンプルとしてご提供しております。
本チュートリアルでは Z80 マイコンのアセンブラ解析に使用致します。

言語設定

C ソースに関する設定

C の言語仕様を設定してください。本チュートリアルでは[ANSI-C]を使用します。

C++ソースに関する設定

C++解析を行う場合は、「C++解析を行う」のチェック指定を有効にしてください。本チュートリアルでは使用致しません。

アセンブラソースに関する設定

アセンブリマクロ記述を解析するための種類を選択して下さい。本チュートリアルでは[通常のアセンブラソース(ガイオ構造化マクロ)]を使用します。

階層化を有効にする

チェックを ON にした場合、プロジェクト内にフォルダを作成し、その中にソースファイルを登録することができます。本チュートリアルでは使用致しません。

プロジェクトの設定データファイル

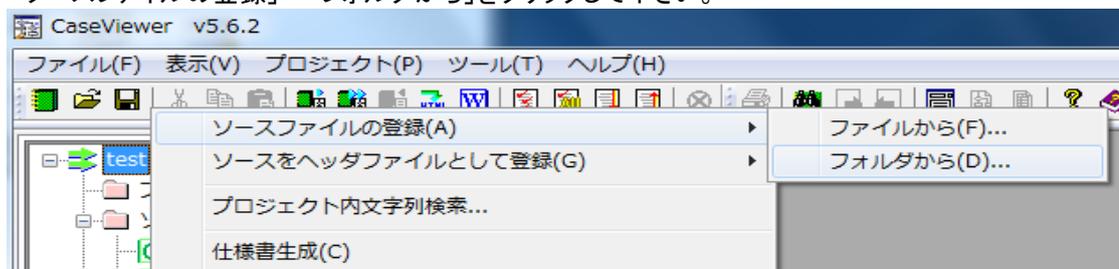
既存のプロジェクトの設定を新規に作成するプロジェクトに設定できます。設定を使用する場合は、あらかじめプロジェクトから設定データをエクスポートしておく必要があります。

本チュートリアルでは使用致しません。

3. ソースコードの登録

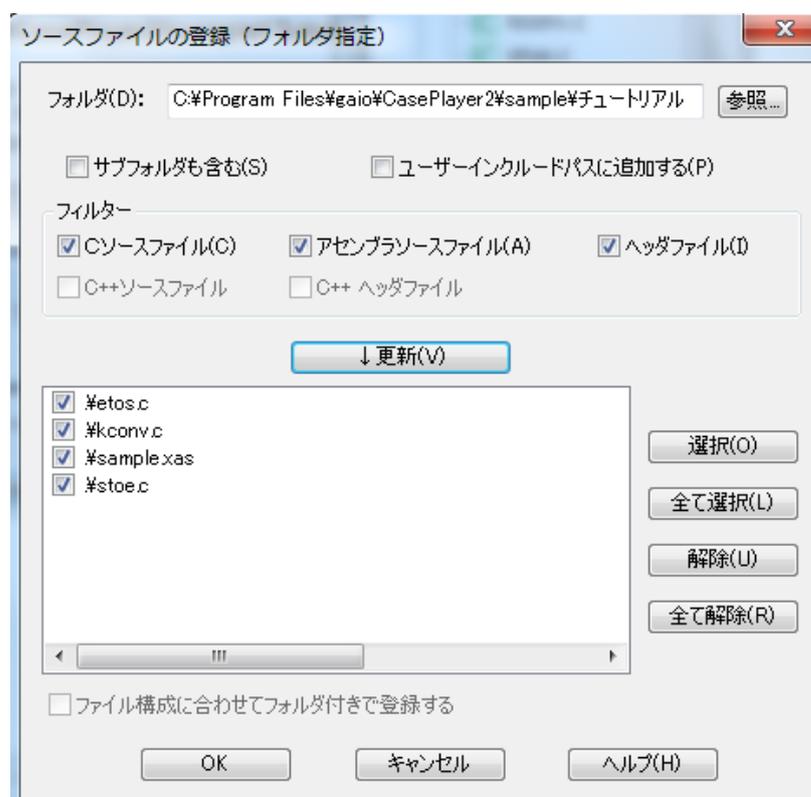
プロジェクト作成後、次に解析対象のソースコードをプロジェクト登録します。

1. プロジェクトビューのプロジェクト名 (test) を右クリックして、「ソースファイルの登録」→「フォルダから」をクリックして下さい。



2. チュートリアル用のソースが置かれている下記フォルダを選択して下さい。
C:\Program Files\gaio\CasePlayer2\sample\チュートリアル

拡張子 (.xas) のファイルはアセンブラソースになります。本チュートリアルでは z80 用のアセンブルソースを使用します。アセンブラソースとして認識する拡張子の設定は CasePlayer2 メニュー 「プロジェクト」→「プロパティ」で変更する事が可能です。



3. 「OK」ボタンをクリックして下さい。

その他の設定について (ソースファイルの登録ダイアログ)

サブフォルダも含む

このチェックボタンを ON にした場合、サブフォルダ内のソースファイルもファイルリストに表示する事ができます。

ユーザインクルードパスに追加する

このチェックボタンを ON にした場合、ヘッダファイルを登録する際、そのファイルのあるフォルダパスをプリプロセッサ設定のユーザインクルードパスに追加する事ができます。

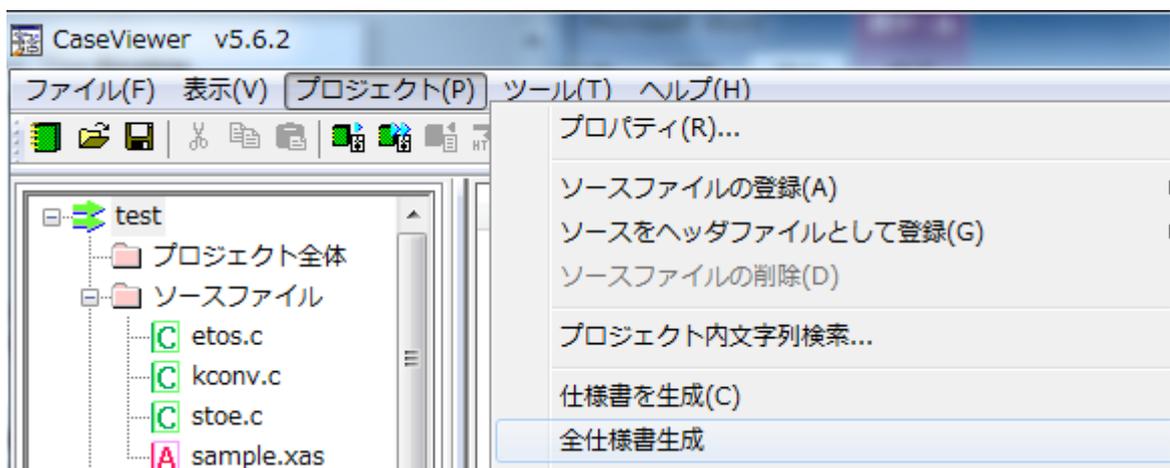
ファイル構成に合わせてフォルダ付きで登録する

このチェックボタンを ON にしてサブフォルダ内のファイルを登録した場合、プロジェクトにサブフォルダと同名のフォルダが自動的に作成され、そのフォルダ内にファイルを登録する事ができます。ただし、このチェックボタンは、プロジェクトの新規作成時に「階層化を有効にする」をチェックしている場合のみ有効です。

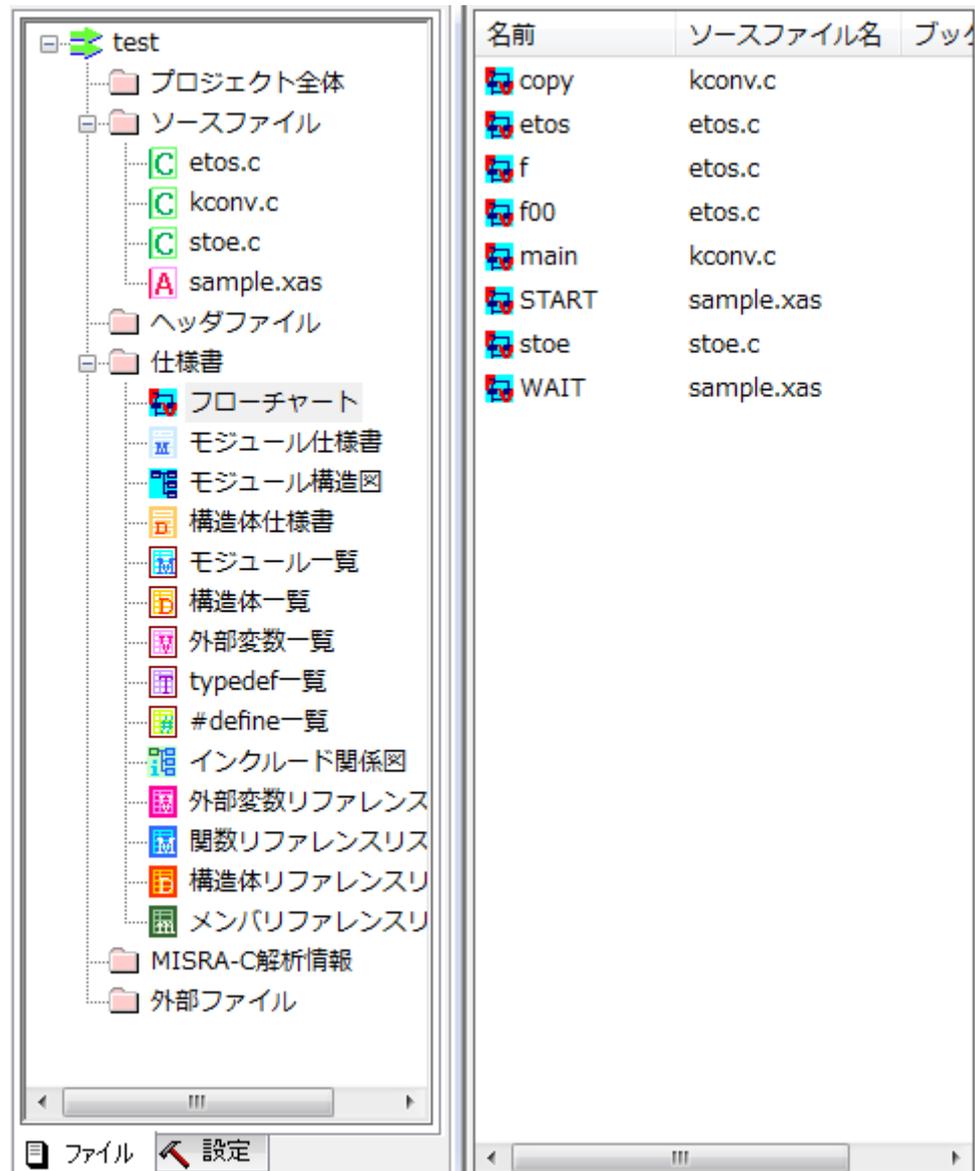
4. 仕様書生成(標準モード)

それでは標準モードで仕様書を生成してみましょう。本モードではフローチャート、モジュール仕様書などを生成する事ができます。

1. アプリケーションメニュー「プロジェクト」->「全仕様書生成」をクリックしてください。ソースファイルを解析し各種仕様書の生成を開始します。



2. 生成された各種仕様書はプロジェクトビュー内のツリーの[仕様書]を展開して頂くと表示する事ができます。



標準モードでは下記の仕様書が生成可能です。

フローチャート: 関数の構造をフローで表示する仕様書です。

モジュール仕様書: 関数の引数、戻り値などを表示する仕様書です。

モジュール構造図: 関数の呼出し関係を表示する仕様書です。

構造体仕様書: 構造体のメンバなどを表示する仕様書です。

モジュール一覧: 解析した関数の一覧を表示する仕様書です。

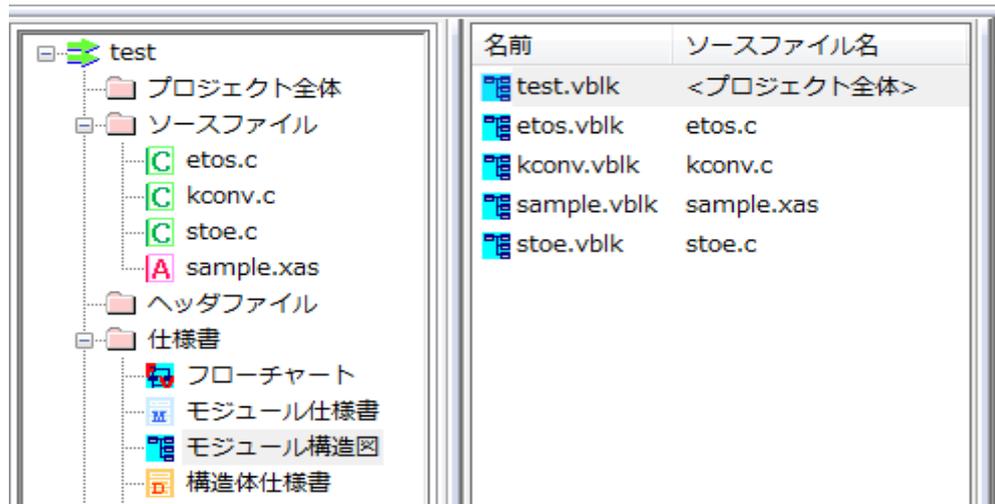
構造体一覧: 解析した構造体の一覧を表示する仕様書です。

上記以外のドキュメント生成は詳細解析モードで生成する必要があります。(本チュートリアル後半で説明します)。

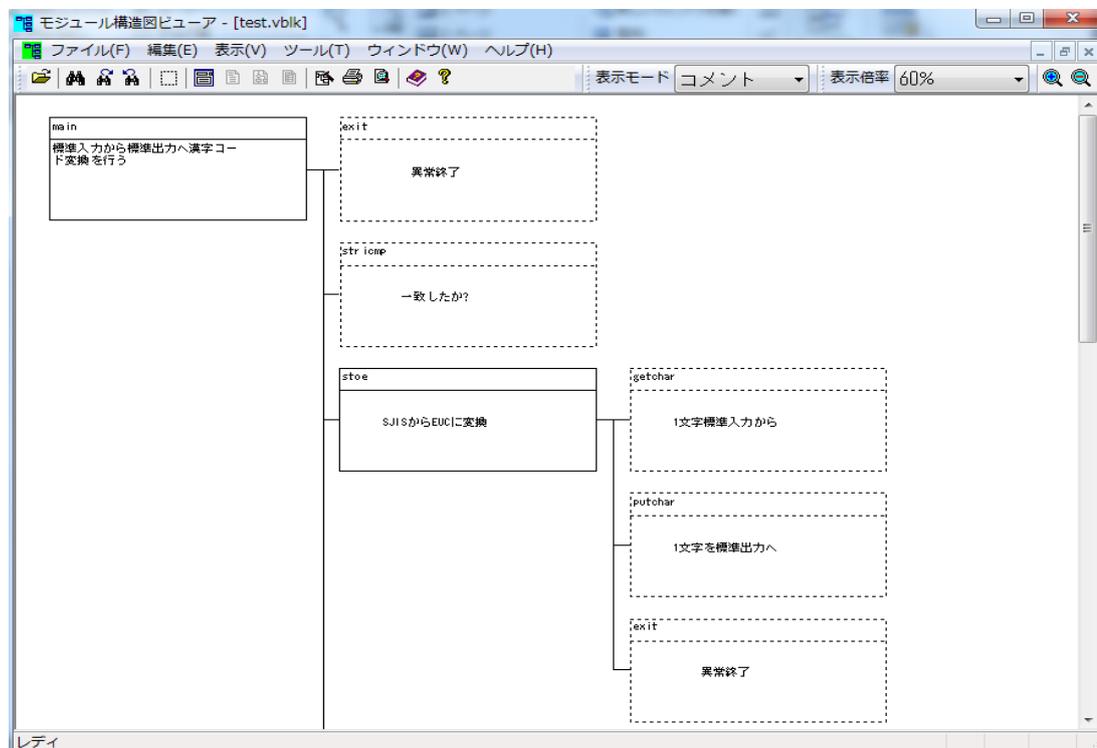
仕様書閲覧

モジュール構造図を見てみましょう

1. プロジェクトビューの[仕様書]→[モジュール構造図]を選択して下さい。
2. 仕様書リストビュー内にモジュール仕様書の一覧が表示されますので、Test.vblk をダブルクリックして下さい。



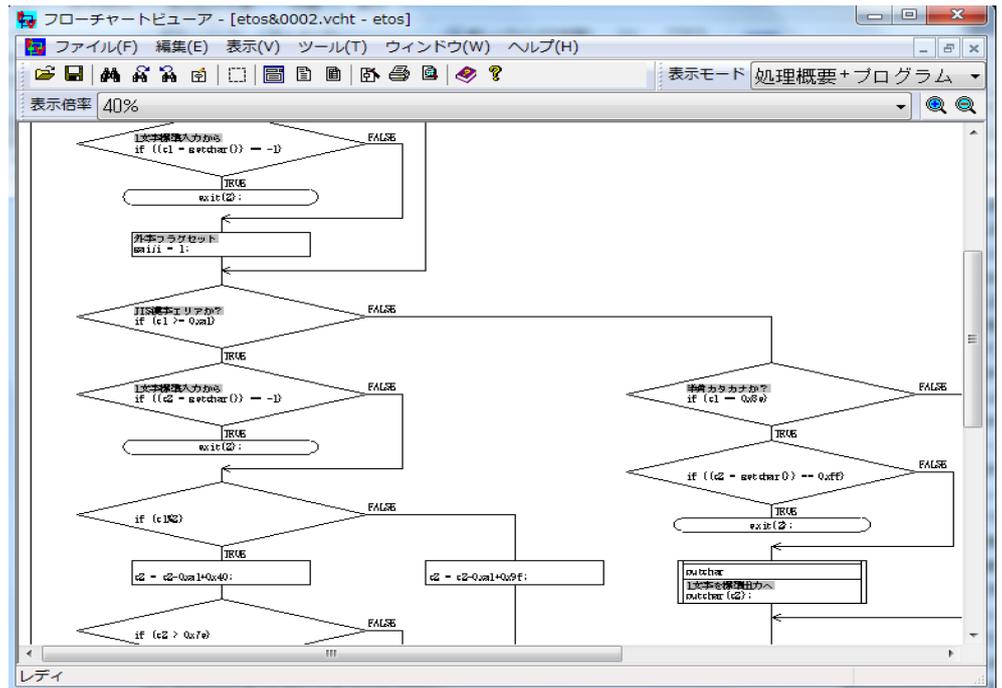
表示倍率、表示モードの変更などが可能です。



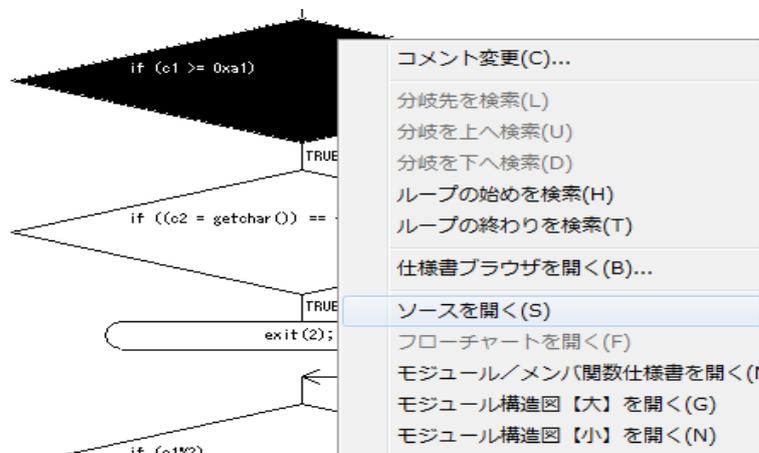
チャートリンク機能

簡単にソースコード⇄仕様書ドキュメント間をジャンプできます。
フローチャートの右クリックメニューからソースコードの該当箇所へジャンプしてみましょう。

1. プロジェクトビューの[仕様書]→[フローチャート]を選択して下さい。
2. 仕様書リストビュー内にフローチャートの一覧が表示されますので、`etos` をダブルクリックして下さい



3. 関数 `etos` のフローチャートが表示されます。表示倍率は、プルダウンメニューから変更する事が可能です。
4. チャートボックスを選択して、右クリックメニュー→「ソースを開く」をクリックして下さい。



選択したチャートボックスに対応したソースコードの行が表示されます。

```

8  /* EUCからSJISに変換 */
9  void etos()
10 {
11     int c1,c2;
12     int gaiji;
13
14     while ((c1 = getchar()) != -1) { /* 1文字標準入力から */
15         gaiji = 0; /* 外字フラグのクリア */
16         if (c1 == 0x8f) { /* 外字エリアか? */
17             if ((c1 = getchar()) == -1) /* 1文字標準入力から */
18                 exit(2); /* 異常終了 */
19             gaiji = 1; /* 外字フラグセット */
20         }
21         if (c1 >= 0xa1) { /* JIS漢字エリアか? */
22             if ((c2 = getchar()) == -1) /* 1文字標準入力から */
23                 exit(2); /* 異常終了 */
24             if (c1%2) {
25                 c2 = c2-0xa1+0x40;
26                 if (c2 > 0x7e)
27                     c2++;
28             }
29         }
30     }

```

仕様書とソースコードとが互いにリンクされているので、容易に右クリックメニューからプログラム文書とソースコードを開くことができます。

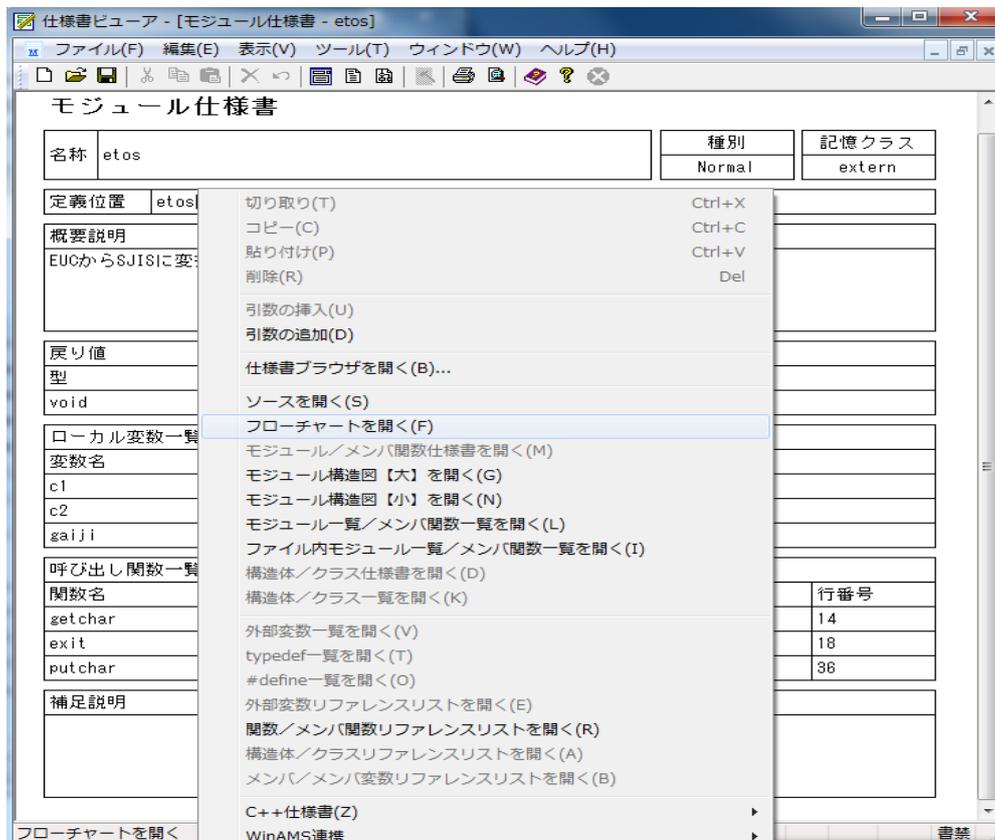


図 1 モジュール仕様書からフローチャートへ移動

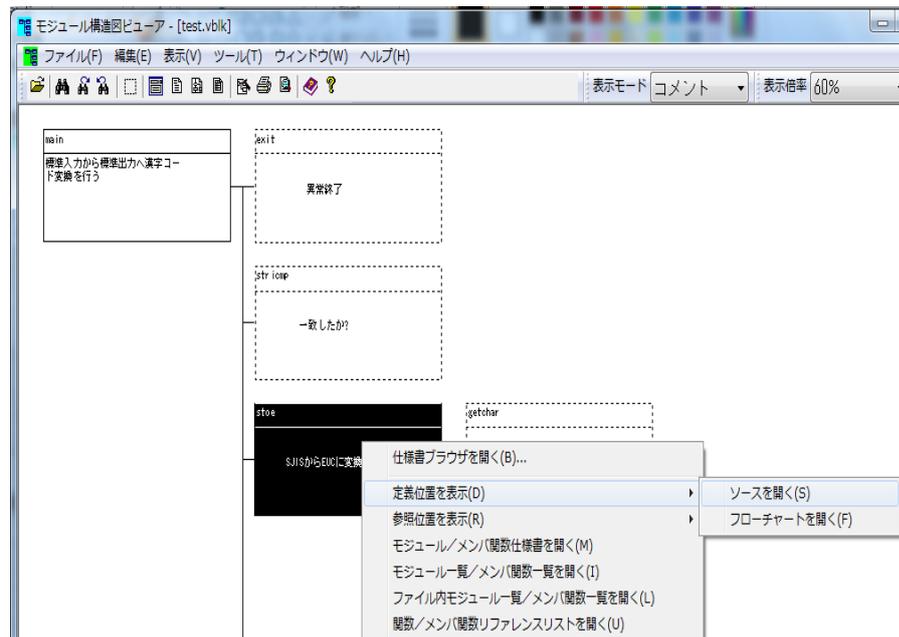


図 2 モジュール構造図からソースコードへ移動

アセンブラソースのフローチャート

CasePlayer2 はアセンブラソースからも仕様書を生成する事が可能です。本チュートリアルでは Z80 デバイスのアセンブラソースを含んでいます。

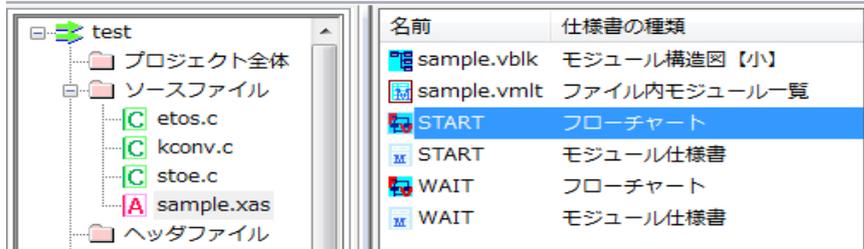
```

1  ;
2  ;   ザイログ Z80
3  ;   4相ステッピングモータの加減速運転プログラム
4  ;
5  ;
6  N EQU 200      ;Number of steps before reaching slew
7  S EQU 399     ;2*N-1
8  PTO EQU 0CDH  ; ポート 0
9  ;
10 PROG SECT CODE,ORG=9000H
11 ;
12 START: LD B,33H      ;データ出力 (0011)
13        LD C,0FDH    ;
14        LD A,B
15        OUT (PTO),A
16        EXX          ;励磁データの待避
17        LD IY,Y      ;コマンドテーブルのアドレス設定
18 ;
19 LOOP1: LD E,(IY+0)   ;ステップ数の入力
20        LD D,(IY+1)
21        LD A,D
22        OR E
23        JP Z,STOP    ;コマンド継続中か?
24 ;
25        LD A,D
26        ADD A,0
27 ;
28        JP M,MINUS   ;回転方向はCWか?
29 PLUS:  NEG          ;CW処理
30        NEG
31        LD A,0
32        JR .X CLEAR
33 ;
34 MINUS: CPL          ;CCW処理
35        LD D,A

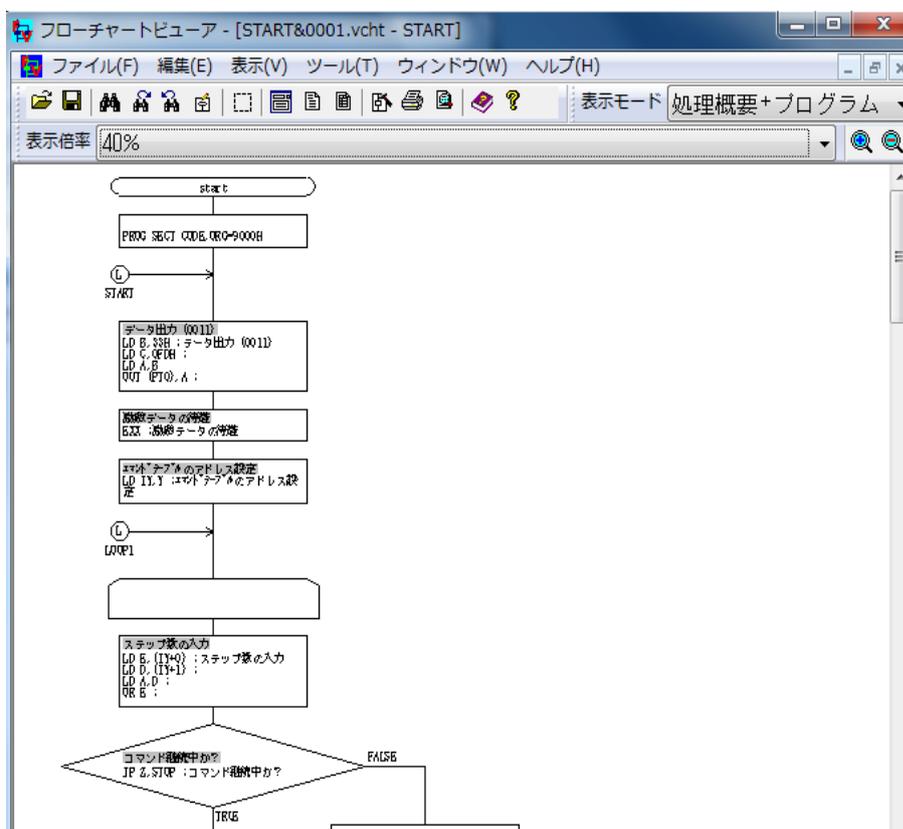
```

Sample.xas のフローチャート表示方法:

1. プロジェクトビュー(ファイルタブ)の[ソースファイル]→[Sample.xas]を選択して下さい。
2. 仕様書リストビュー内に表示される START(フローチャート)をダブルクリックして下さい。



3. アセンブリルーチン START のフローチャートが表示されます。
(注釈)::アセンブラ命令は MPU に応じて異なるため(P5 参照)、プロジェクト生成時に適切なアセンブリパラメータファイル(*.rev)を設定する必要があります。



設定して頂く(*.rev)ファイルは C:\Program Files\gaio\CasePlayer2\sample 配下になんらかのサンプルがございます。

作成された仕様書が解析した MPU のフローとして正しく表示されない場合、(*.rev) ファイルをユーザー様で変更して頂く必要がございます。

5. 詳細解析モード

詳細解析モードでは、外部変数一覧、構造体リファレンスリストなどの仕様書を追加で作成することが可能です。これらの仕様書は、外部変数の予期せぬ書き込みや参照、構造体の構造上の問題などを早期に発見するのに役立ちます。

詳細解析モードでは、全てのコンパイルソースとヘッダファイルを解析のためにおプロジェクトに追加する必要があります。本チュートリアルはシンプルな構成ですので、詳細解析のために追加するファイル、設定はございません。実際の組み込みプロジェクトでは詳細解析のために、CasePlayer2 の設定項目「プリプロセッサの設定」、「C オプションパラメータの設定」をして頂く必要があります。これらの設定につきましては、チュートリアルの項目11以降にてご紹介させて頂いておりますので参照ください。

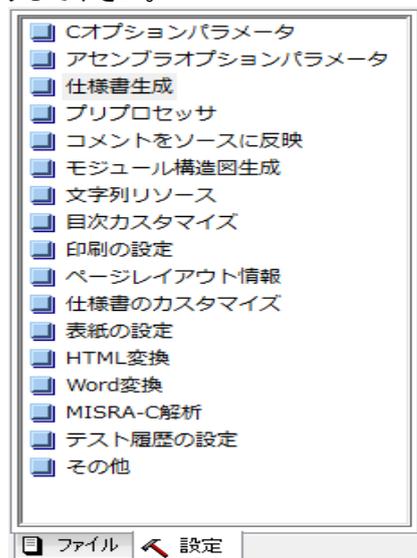
詳細解析モードでは標準モードの仕様書に加えて、下記の仕様書が生成可能です。

- 外部変数一覧
- typedef 一覧
- #define 一覧
- インクルード関係図
- 外部変数リファレンスリスト
- 関数リファレンスリスト
- 構造体リファレンスリスト
- メンバリファレンスリスト

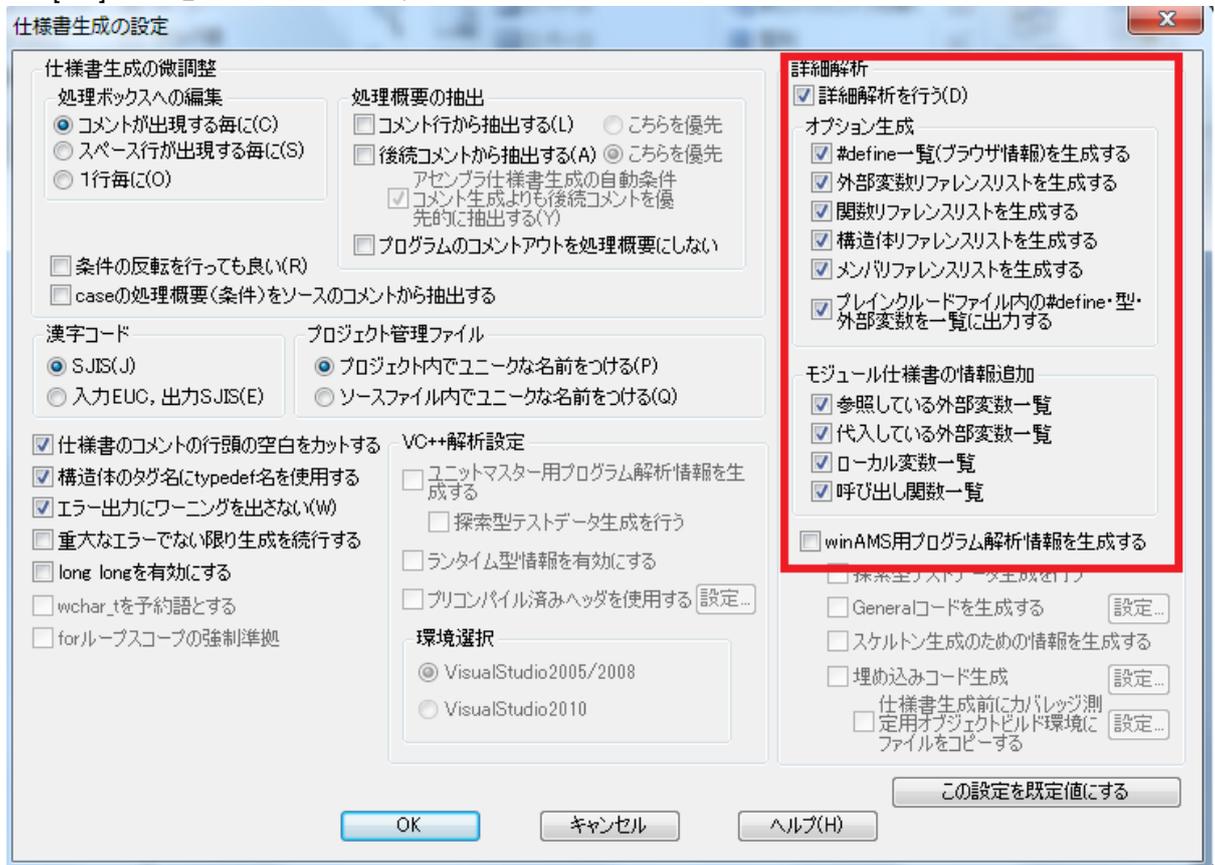
詳細解析設定

詳細解析モードの設定:

1. [プロジェクトビュー]の[設定]タブを表示して下さい。
2. [仕様書生成]をダブルクリックして下さい。

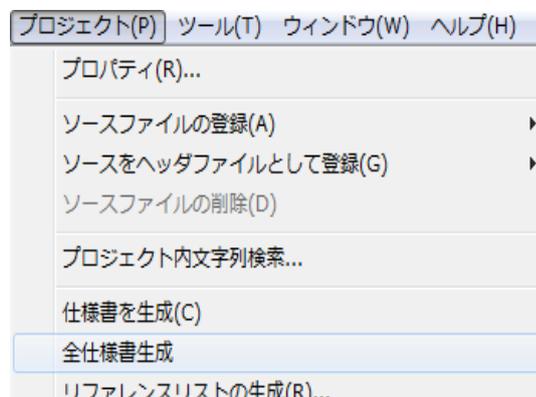


3. [詳細解析]の配下[winAMS 用プログラム解析情報を生成する]以外のすべてを ON にしてください。
4. [OK]ボタンをクリックして下さい。



仕様書生成

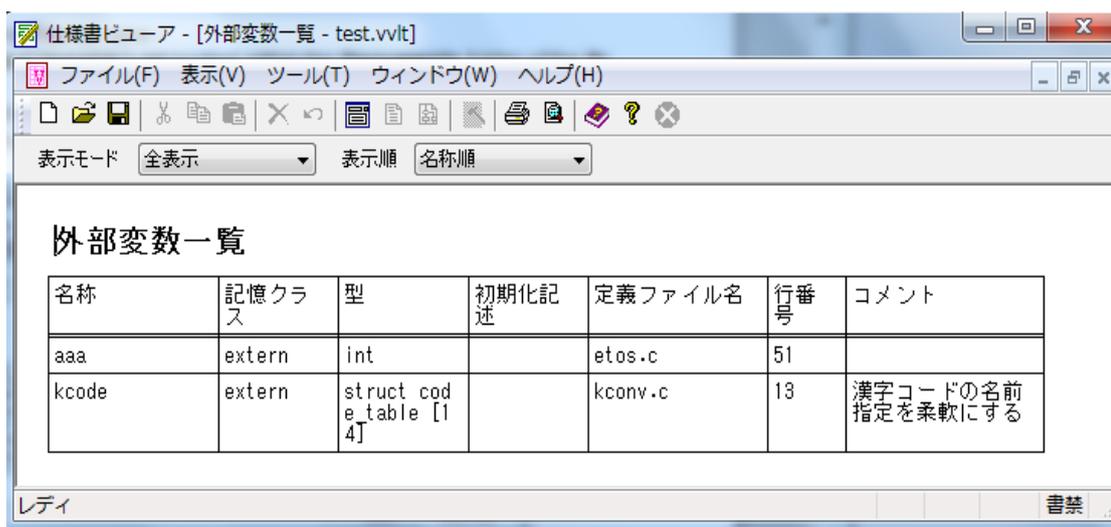
CasePlayer2 メニュー「プロジェクト」→「全仕様書生成」をクリックしてください。
ソースファイルを解析し各種仕様書を生成を開始します。



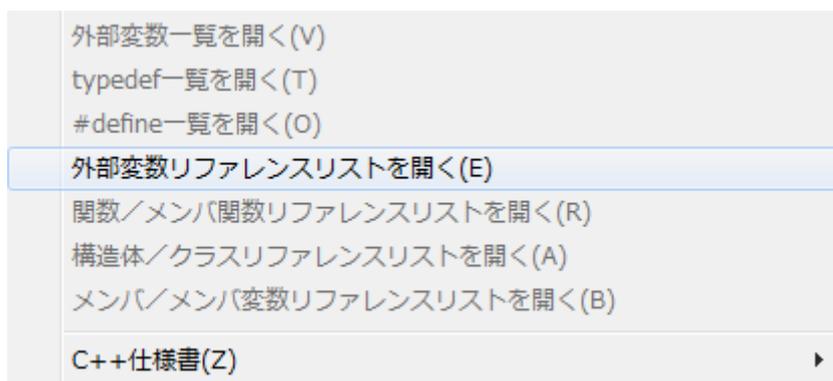
仕様書閲覧

外部変数一覧と外部変数リファレンスリストを見てみましょう。
作成したプログラムの仕様書を表示する操作方法は、前述の標準モードと同様になります。

1. プロジェクトビュー(ファイルタブ)を表示して下さい。
2. [仕様書]→[外部変数一覧]を選択して下さい。
3. 仕様書リストビュー内に表示される test.vvlt をダブルクリックして下さい。



4. **Kcode** を選択して右クリックして下さい。
5. 「外部変数リファレンスリストを開く」をクリックして下さい。



外部変数リファレンスリストは kcode が参照されている位置をソースファイル名、行番号、関数名を含むリストで表示します。リストの上で右クリックするとメニューが表示され、Kcode が参照されている各種仕様書にジャンプできます。

The screenshot shows a window titled "仕様書ビューア - [外部変数リファレンスリスト - kcode]". The main content area displays the "外部変数リファレンスリスト" (External Variable Reference List) for the variable 'kcode'. The list includes details such as the name, memory class, type, definition location, and comment. Below this, a table lists the reference locations, including the source file name, line number, and function name.

参照位置一覧	参照ファイル名	行番号	関数名
	kconv.c	41	main
	kconv.c	42	main
	kconv.c	43	main
	kconv.c	47	main
	kconv.c	48	main
	kconv.c	49	main

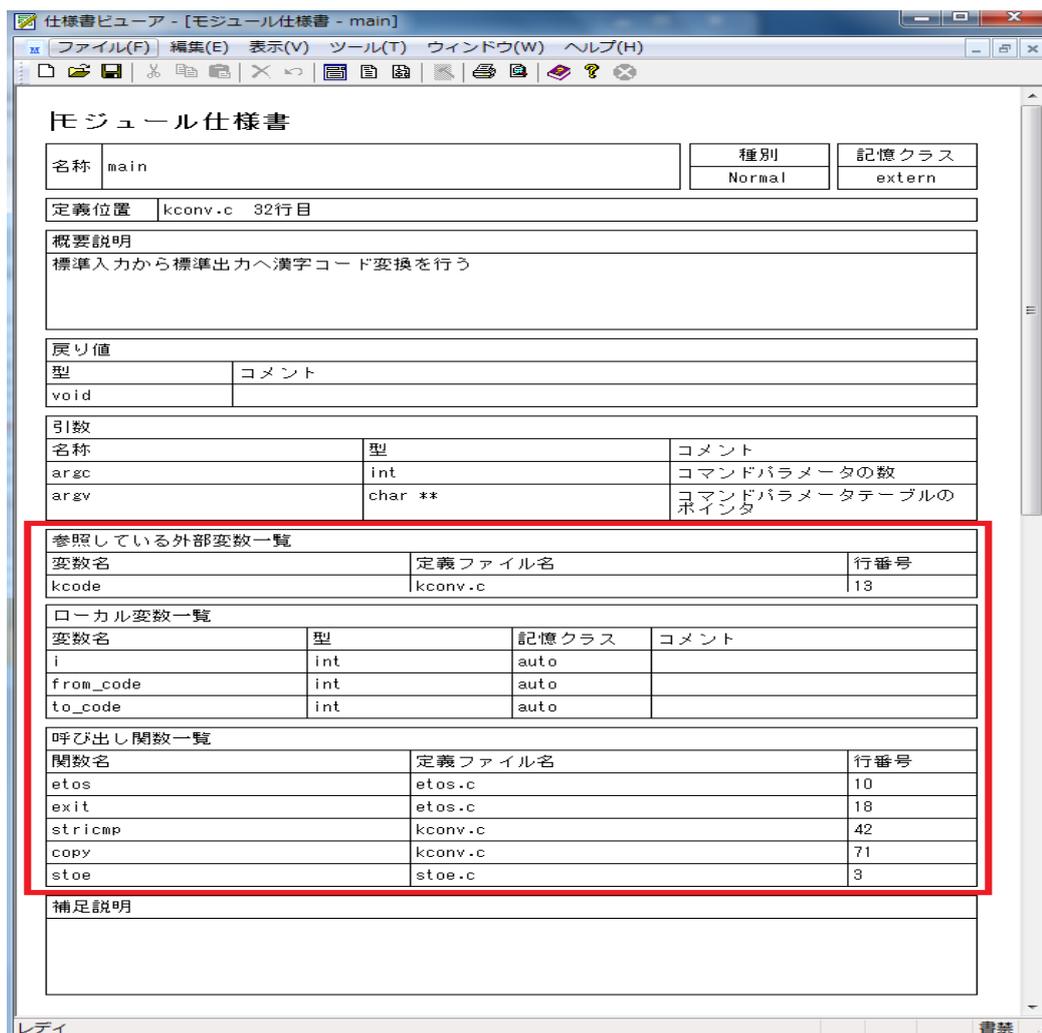
On the right side, a context menu is open, listing various actions such as "仕様書ブラウザを開く(B)...", "ソースを開く(S)", "モジュール/メンバ関数仕様書を開く(M)", and "C++仕様書(Z)".

変数参照情報の閲覧

詳細解析モードでの仕様書生成によりモジュール仕様書に追加された変数の参照情報を見てみましょう。

1. プロジェクトビュー(ファイルタブ)の[仕様書]→[モジュール仕様書]を選択して下さい。
2. 仕様書リストビュー内に表示される main をダブルクリックして下さい。

詳細解析モードで仕様書を生成した場合、次の項目が追加されます。
参照している外部変数一覧, ローカル変数一覧, 呼び出し関数一覧。



これにて、本チュートリアルでの仕様書生成についての説明は終了いたします。
ここまでの手順にて、標準モードと詳細解析モード、各種仕様書間の右クリックメニューでのリンク機能についてご説明させて頂きました。

6. MISRA-C ルールチェック

MISRA-C は、組込みソフトウェアのコードの安全性、移植性と信頼性を容易にするために、MISRA (自動車産業ソフトウェア信頼性協会)によって開発された C プログラミング言語のためのソフトウェア開発標準です。

ここでは、CasePlayer2 の MISRA-C ルールチェック機能についてご説明させていただきます。
CasePlayer2 にてサポートしている MISRA-C ルールセットは MISRA-C1998 及び MISRA-C2004 になります。

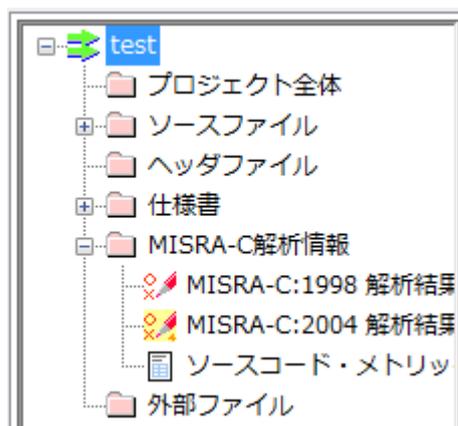
MISRA-C ルールチェックを実行

MISRA-C ルールチェックを行うためには、本チュートリアルの前篇でご説明させていただきました詳細解析モードの設定が必要になります。これ以降のご説明は、前篇で使用致しました test.vproj の詳細解析モードの設定が行われている事を前提にご説明させていただきます。

1. アプリケーションメニュー「プロジェクト」->「全 MISRA-C 解析」をクリックしてください。



2. プロジェクトビュー (ファイルタブ) の[MISRA-C 解析情報]を開いて下さい。



3. [MISRA-C:2004 解析結果]をダブルクリックして下さい。

MISRA-C:2004 解析結果 表示メッセージ数: 161 / 総メッセージ数: 161

ルール番号	説明	メッセージ数
ルール番号[1.2]	未定義又は未規定の動作に依存してはならない。	メッセージ数:1
ファイル名	行番号	補足説明
kconv.c	77	ファイルが改行文字で終了していない
ルール番号[3.3]	選択したコンパイラの整数除算の実装について確認し、文書化し、十分に配慮すべきである。	メッセージ数:2
ファイル名	行番号	補足説明
etos.c	24	
	31	
ルール番号[5.7]	識別子は、再使用すべきではない。	メッセージ数:5
ファイル名	行番号	補足説明
etos.c	11	動的変数が他の関数で使用されている (c1)
		動的変数が他の関数で使用されている (c2)
kconv.c	72	動的変数が他の関数で使用されている (c1)
stoe.c	4	動的変数が他の関数で使用されている (c1)
		動的変数が他の関数で使用されている (c2)
ルール番号[6.3]	基本型の代わりに、サイズ及び符号属性 (signedness) を示す typedef を用いなければならない。	メッセージ数:14
ファイル名	行番号	補足説明
etos.c	4	

MISRA-C のルールに違反しているコードがある場合、違反しているルール番号と共にソース名、行番号、補足説明がリストで表示されます。リストを選択してダブルクリックして頂くと該当するソース行にジャンプできます。

ファイル名	行番号	補足説明
kconv.c	77	ファイルが改行文字で終了していない



```

68
69 /* 無変換で標準入力から標準出力へコピー */
70 void copy()
71 {
72     int c1;
73
74     while ((c1 = getchar()) != -1) { /* 1文字標準入力から */
75         putchar(c1);             /* 1文字を標準出力へ */
76     }
77

```

表示オプション

表示順 - ルール番号順: 解析結果をルール番号順で表示します。

表示順 - ソース位置順: 解析結果をファイル毎に行番号順で表示します。

表示モード- 全体表示: 全ルールのメッセージを表示します。

表示モード - 簡易表示: ルール情報のみ表示し、ファイル名や行番号などソース位置に関する情報は表示されません。

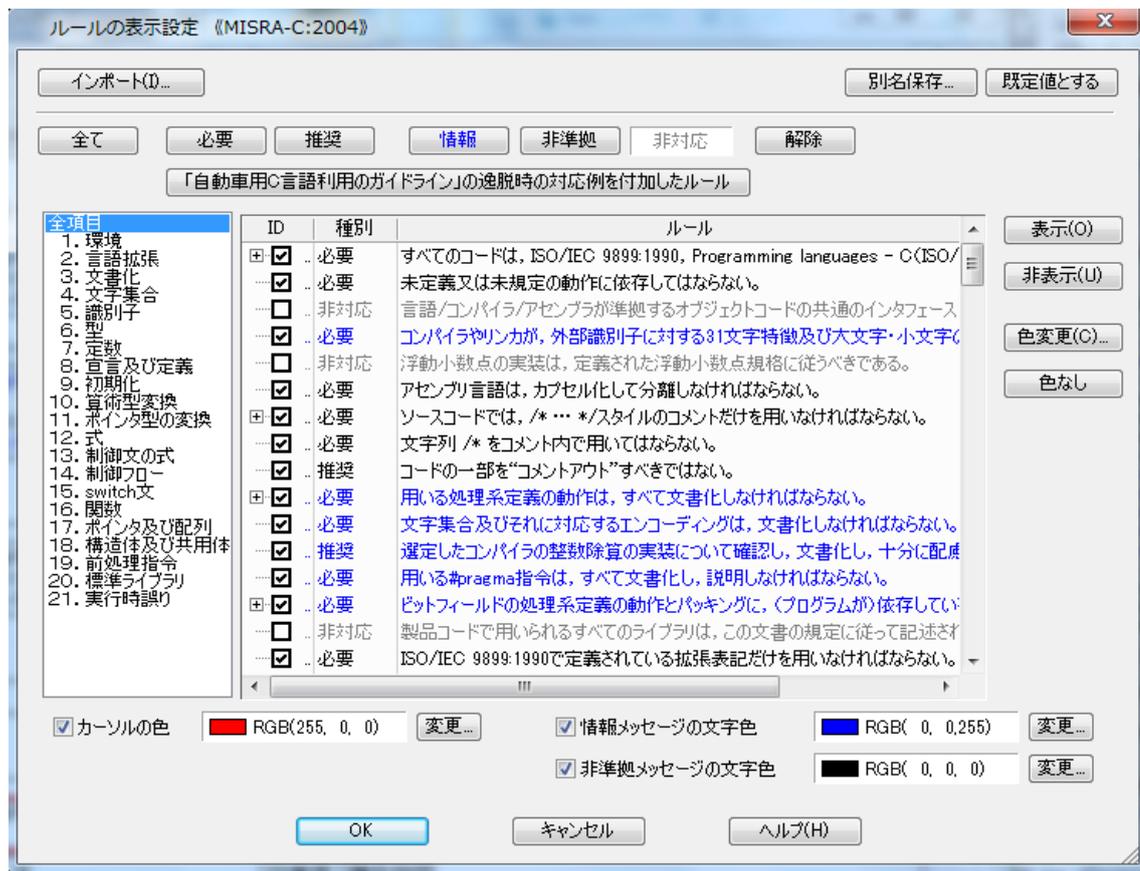
表示モード - 部分表示: 部分表示モードでは、本設定の右側にあるルールメニューで 選択して頂いたメニューについて表示します。

※上記表示モードの説明は表示順でルール番号順を選択頂いたのみのご説明になります。

ルールの選択

下記ダイアログにて結果表示させるルールを選択する事が可能です。

- CasePlayer2 メニュー「プロジェクト」→「全 MISRA-C 解析」.→「MISRA-C:2004 解析結果」をダブルクリックして下さい。
- 仕様書ビューア メニュー「ルール」→「表示ルール設定」をクリックして下さい。



ルールは項目別に分類されています。このリストの項目の選択によって、ルールリストの表示内容が更新されます。チェックを ON にして頂いたルールが解析結果に表示されます。

ソースコード・メトリックス閲覧

プロジェクトビュー(ファイルタブ)の[MISRA-C 解析情報]→[ソースコード・メトリックス]をダブルクリックして頂くと表示されます。

関数名	定義ファイル名	行番号	経路複雑度	マイヤーズ・インターバル	ネストの深さ	コード・サイズ	コメントを含む行数	コメント比率	静的パスカウント
etos	etos.c	10	13	1	4	40	18	45	85
main	kconv.c	34	11	0	2	34	18	52	90
stoe	stoe.c	3	9	1	5	36	12	33	27
copy	kconv.c	71	2	0	1	7	2	28	2
f	etos.c	54	1	0	0	3	0	0	1
f00	etos.c	5	1	0	0	2	0	0	1

ソースコード・メトリックスでは下記の情報が出力されます：

関数名: 関数名が表示されます。

定義ファイル名: 関数の定義されているソースコード名が表示されます。

行番号: 関数の定義されているソースコード上での行番号が表示されます。

経路複雑度: 分岐の下図

マイヤーズ・インターバル: 分岐の数+"?:","&&","||"の数が表示できます。

ネストの深さ: 一番深いネスト。

コード・サイズ: 関数の行数 ({ から } までの行数)。

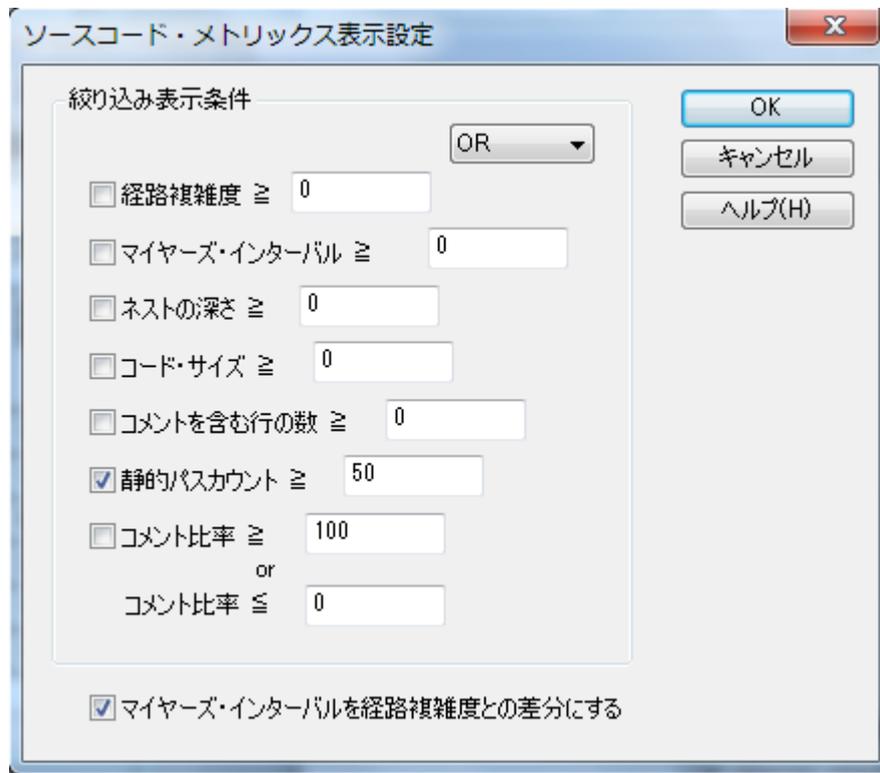
コメントを含む行数: 上記の行のうちコメントを含む行のカウンタ。

コメント比率: コメントを含む行のコード・サイズにおける割合。

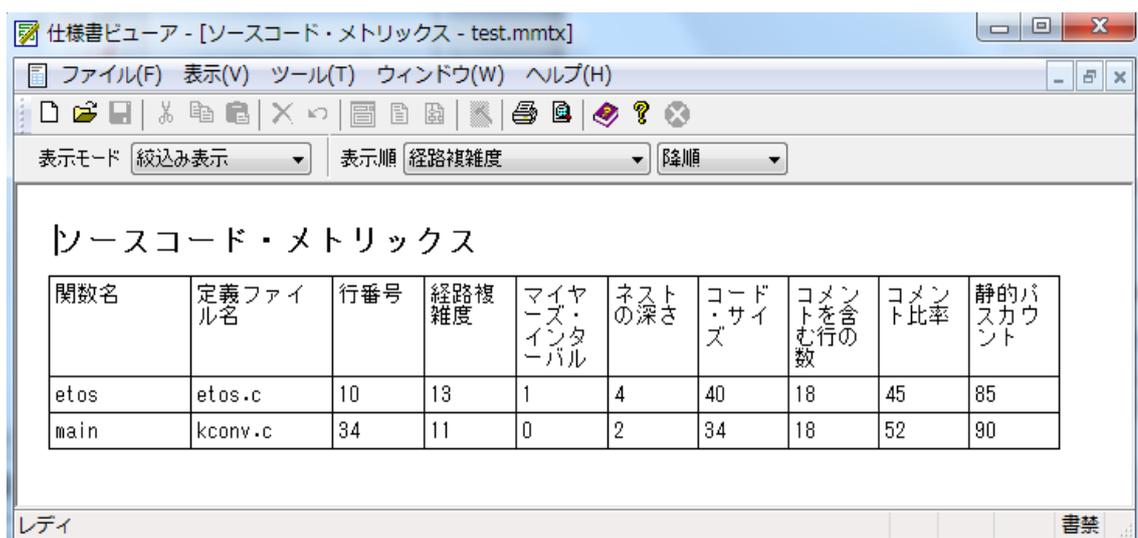
静的パスカウント: 静的パスをカウントしたもの(総経路数)。

ソースコード・メトリックスの解析結果表示を絞り込んで表示する事が可能です。
仕様書ビューア メニュー「ツール」→「表示設定」をクリックして下さい。

例として、静的パスカウントのチェックを ON、数値を 50 に設定して下さい。
静的パスカウントが 50 以上の関数がソースコード・メトリックスのリストに表示されます。



条件設定の絞り込みにより下記のように表示されます：

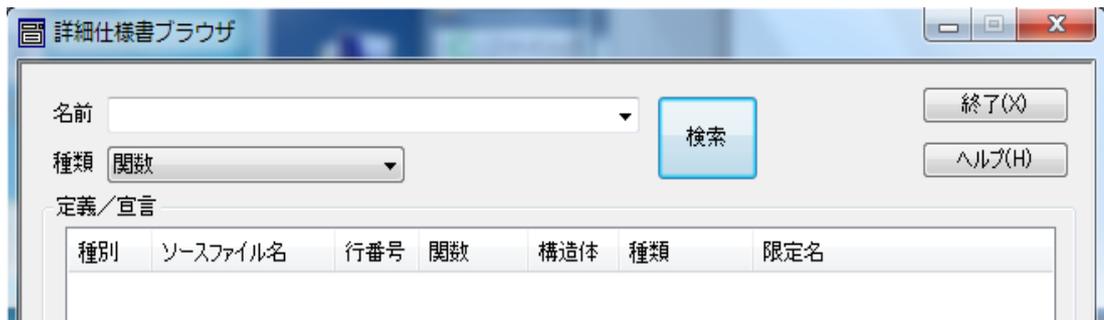


これで、MISRA-C 解析についての説明を終了します。

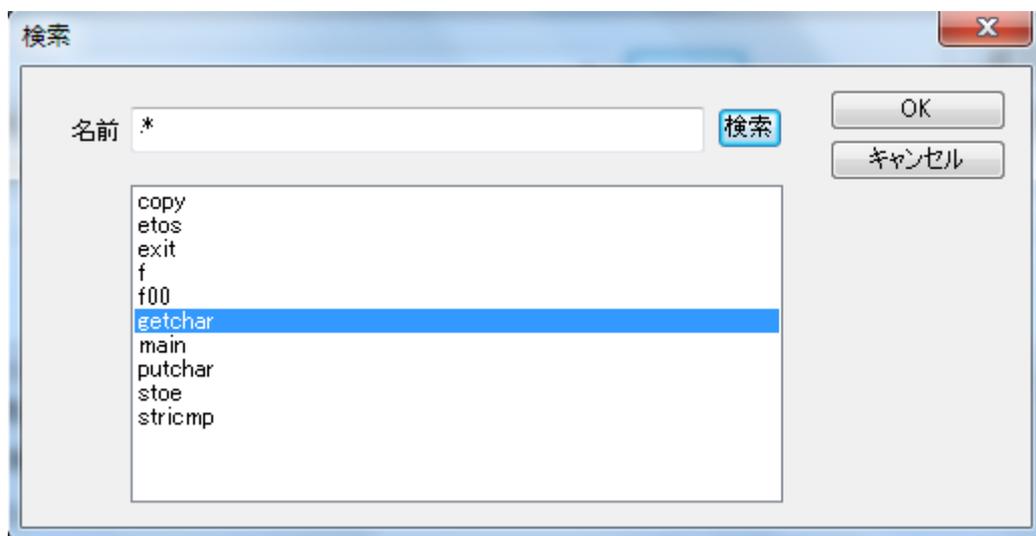
7. 検索機能

詳細仕様書ブラウザではプロジェクト内の、関数、変数、typedef、構造体などの様々な情報を検索する事が可能です。

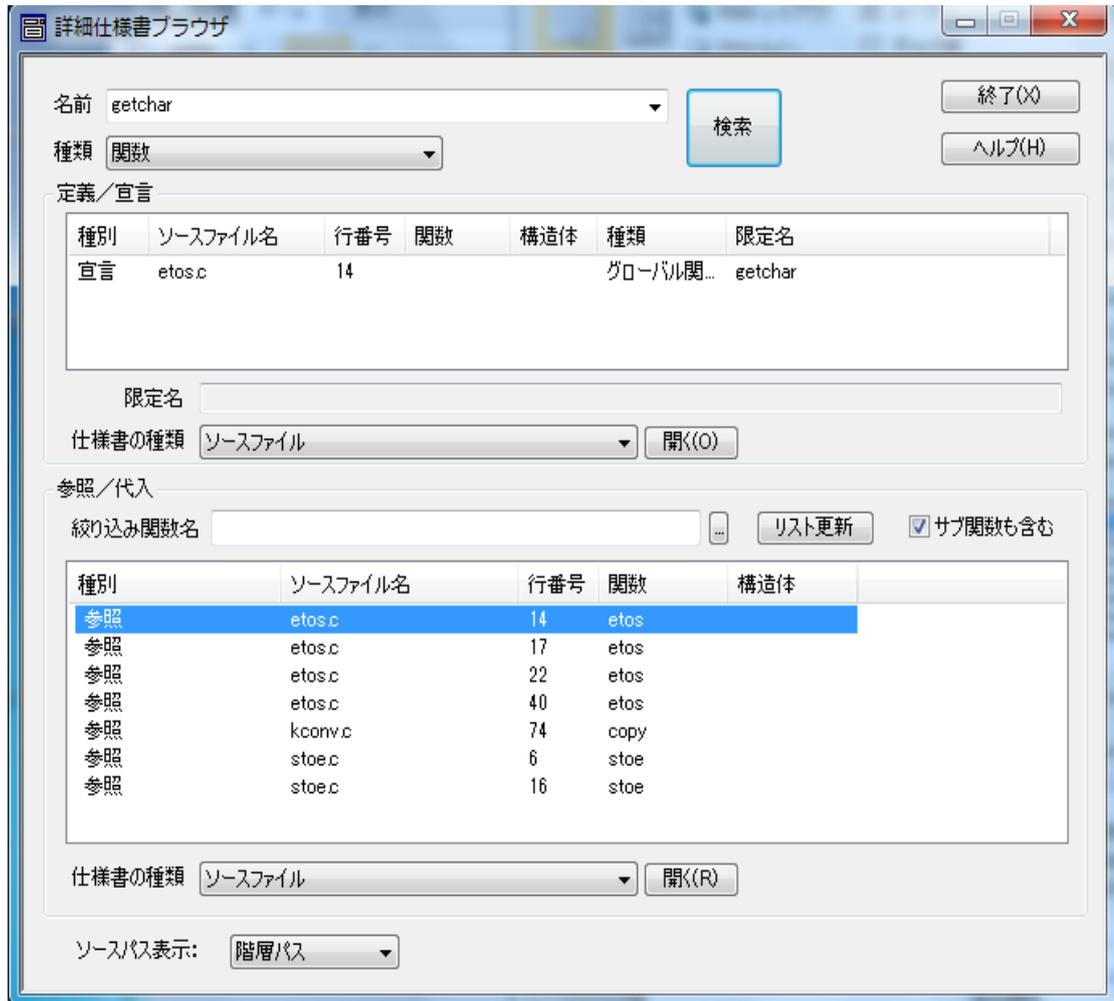
1. 本チュートリアルで使用している `test.vproj` を開いてください。
2. CasePlayer2 メニュー [ツール]→[仕様書ブラウザを開く]をクリックして下さい。
3. 種類を関数に設定して下さい。
4. 名前の項目は空白のままにしてください。
5. 検索ボタンをクリックして下さい。



プロジェクト内にある関数名が表示されます。



6. リストの中から `getchar` を選択して、OK ボタンをクリックして下さい。



getchar 関数が定義/宣言されている位置、また参照/代入されている位置の情報が表示されます。関連するソースファイルの行を表示するには、リストをダブルクリックして下さい。

```

12     int gaiji;
13
14     while ((c1 = getchar()) != -1) { /* 1文字標準入力から */
15         gaiji = 0; /* 外字フラグのクリア */
16         if (c1 == 0x8f) { /* 外字エリアか? */
17             if ((c1 = getchar()) == -1) /* 1文字標準入力から */
18                 exit(2); /* 異常終了 */
19             gaiji = 1; /* 外字フラグセット */
20         }
21         if (c1 >= 0xa1) { /* JIS漢字エリアか? */
22             if ((c2 = getchar()) == -1) /* 1文字標準入力から */
23                 exit(2); /* 異常終了 */
24             if (c1%2) {
25                 c2 = c2-0xa1+0x40;
26                 if (c2 > 0x7e)
27                     c2++;
28             }
29             else
30                 c2 = c2-0xa1+0x9f;

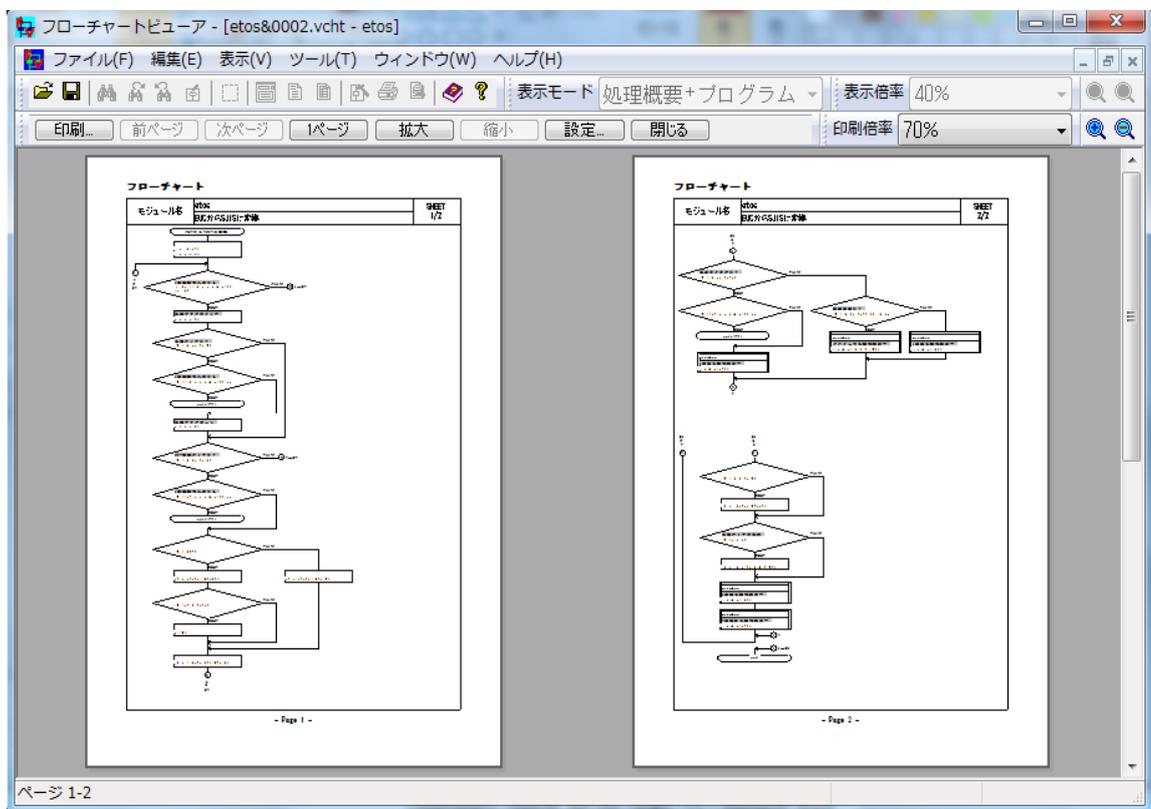
```

詳細仕様書ブラウザでは検索したい項目/設定をして頂く事によりプロジェクト内の情報を検索する事が可能です。

8. 印刷機能

CasePlayer2 では仕様書を実際に印刷する前に、印刷プレビュー機能を使用して実際の印刷イメージを確認し、カスタマイズする事ができます。印刷ドキュメントに無駄な空白部分がない事やページ数が無駄に増えていないか等の確認に役立てて頂けます。

1. プロジェクトビュー(ファイルタブ)を表示して下さい。
2. [仕様書]→[フローチャート]をクリックして下さい。
3. 仕様書リストビュー内に表示される etos をダブルクリックして下さい。
4. フローチャートビューア メニュー [ファイル] →[印刷プレビュー]をクリックして下さい。
5. メニューバーの下に現れる[2 ページ]ボタンをクリックして下さい。



6. [印刷倍率]の右側に位置する (+) or (-)をクリックする事によって、印刷倍率を設定する事ができます。

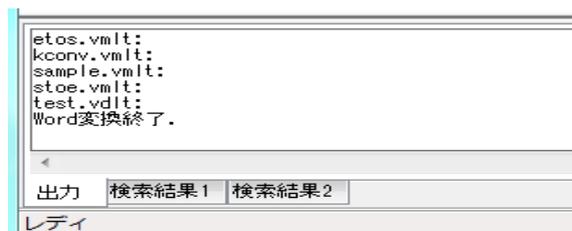
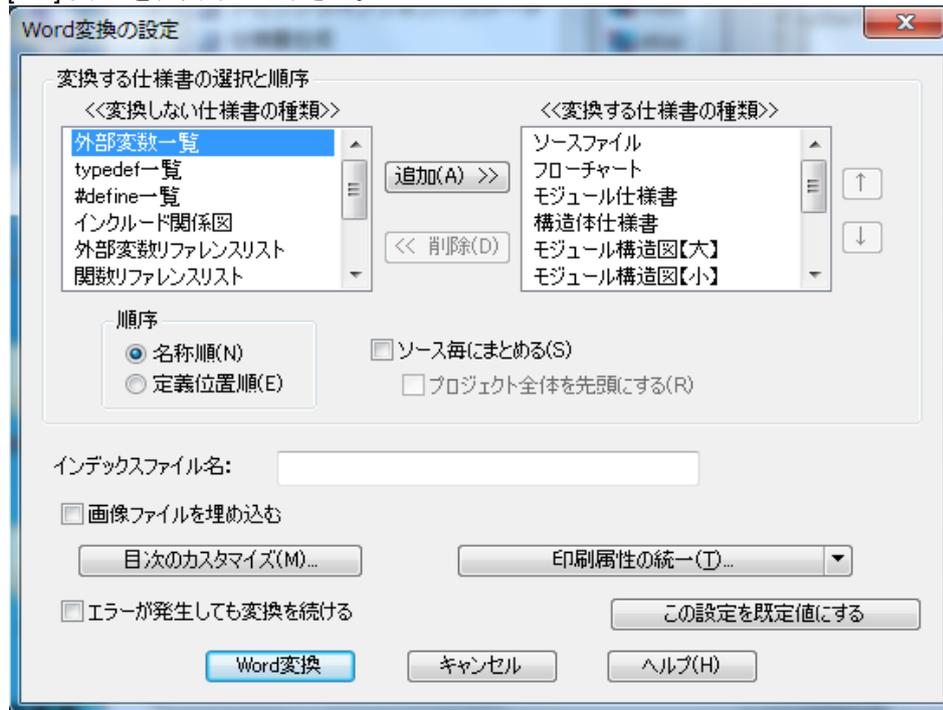


変更した印刷プレビューの設定は、仕様書を再生成した後も維持されます。

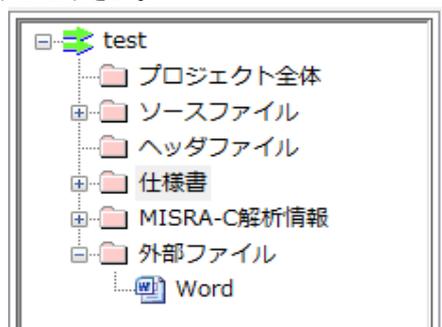
9. 仕様書の Word 変換

仕様書は、CasePlayer2 の機能を使用して Word 変換する事ができます。

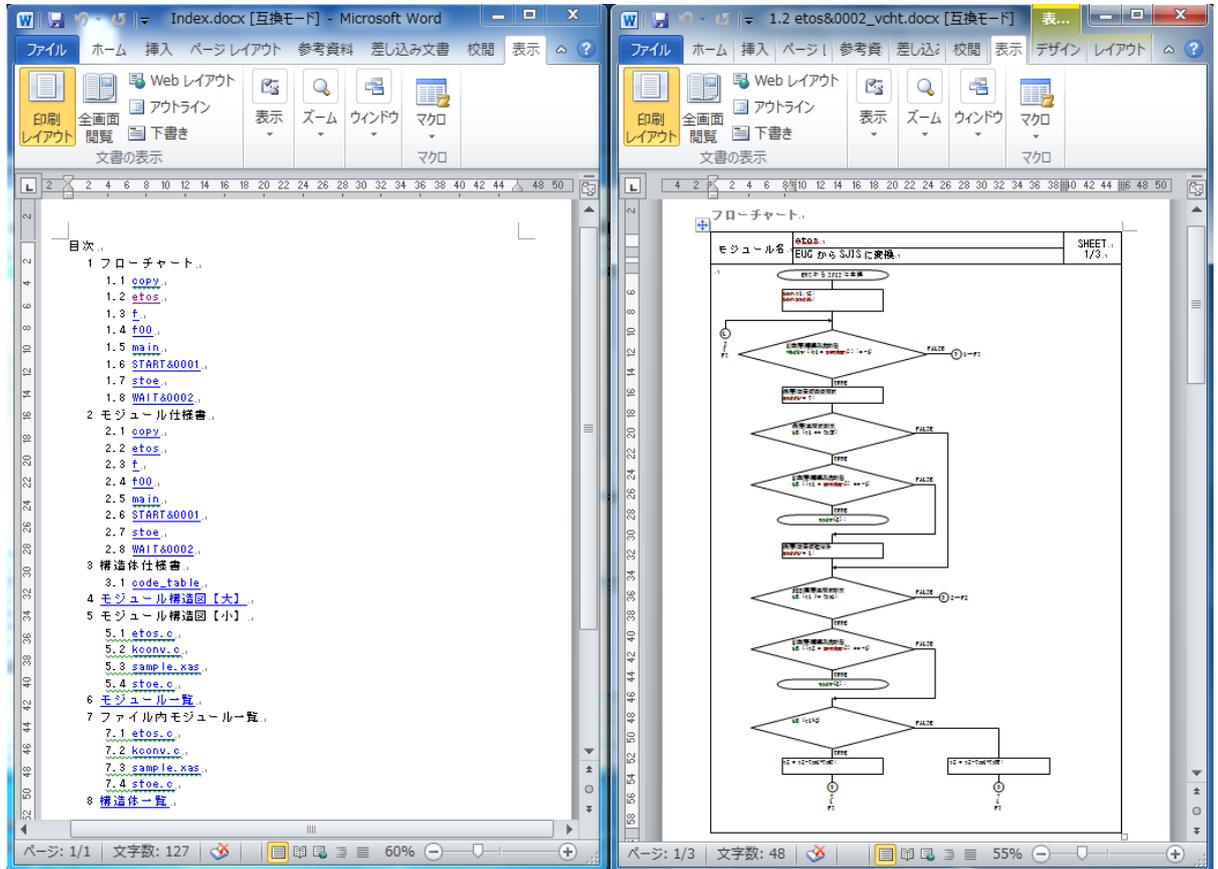
1. CasePlayer2 メニュー「プロジェクト」→「Word 変換」をクリックして下さい。
2. Word 変換する仕様書の種類を右側の[<<変換する仕様書の種類>>]に追加して下さい。
3. [OK]ボタンをクリックして下さい。



4. プロジェクトビュー(ファイルタブ)の[外部ファイル]を展開して下さい。
5. [Word]をダブルクリックして下さい。

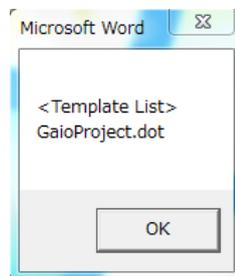


目次ページより各種仕様書の word 文書へジャンプする事ができます。
目次の各項目を[Ctrl+クリック] する事で該当文書を表示できます。



注釈:

本機能のご利用には、お使いの PC に Microsoft Word がインストールされている必要があります。
Word 変換操作を実行する前にご使用中の全ての Microsoft Word 文書を閉じて下さい。
変換時に<Template List>エラーが発生した場合、下記の操作にしたがって、Microsoft Word のセキュリティ設定を変更して下さい。

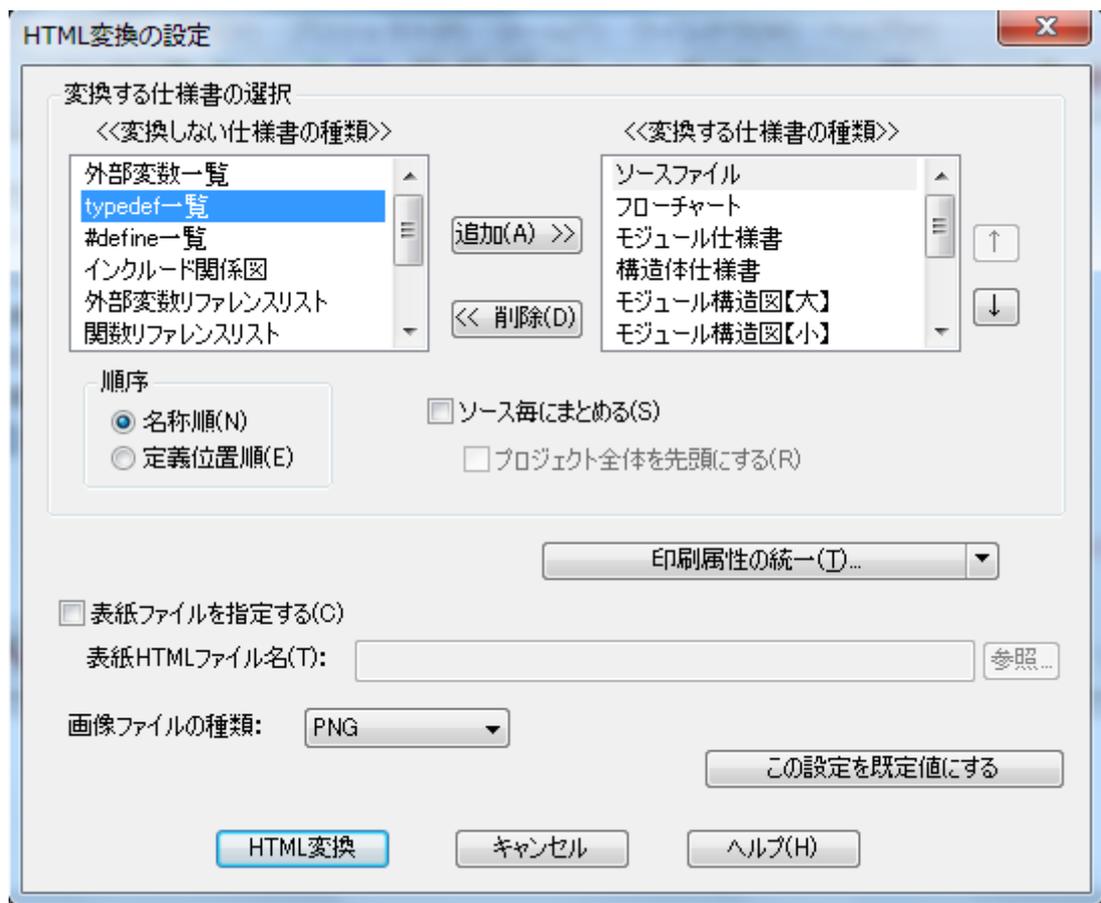


CasePlayer2 メニュー [ヘルプ]→[ヘルプ]をクリックして CasePlayer2 ヘルプを起動して下さい。
[検索]タブの検索エリアに” Word 変換時のエラー”を入力して[検索開始]ボタンをクリックして下さい。検索結果に”Word 変換時のエラー”が表示されますのでクリックして下さい。
お使いの Word バージョンに合わせて、セキュリティ設定をご変更下さい。

10. 仕様書の HTML 変換

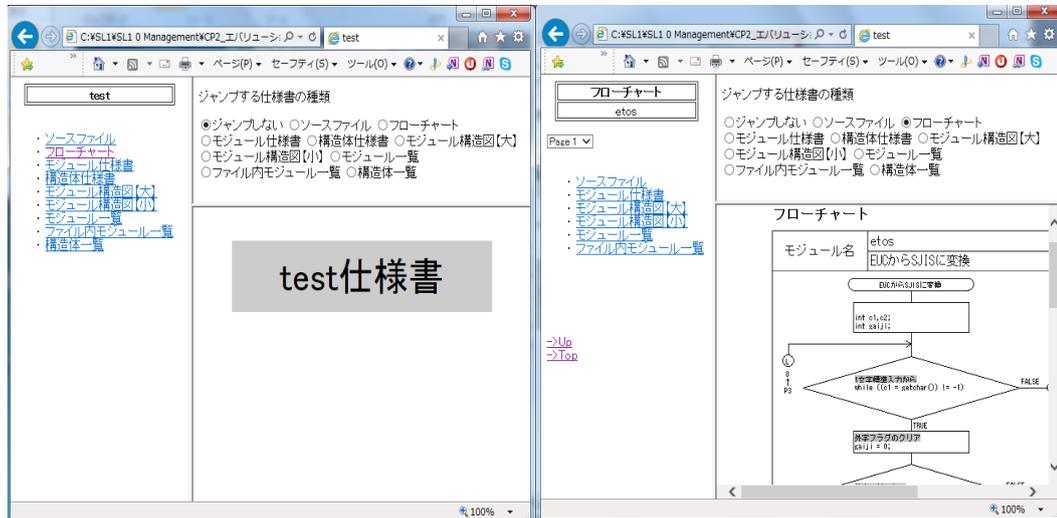
生成した仕様書は、CasePlayer2 の機能を使用して HTML 変換する事ができます。

1. CasePlayer2 メニュー「プロジェクト」→「HTML 変換」をクリックして下さい。
2. HTML 変換する仕様書の種類を右側の[<<変換する仕様書の種類>>]に追加して下さい。
3. [OK]ボタンをクリックして下さい。



4. プロジェクトビュー(ファイルタブ)の[外部ファイル]を展開して下さい。
5. [HTML]をダブルクリックして下さい。

HTM 形式の文書が作成され、各種仕様書がリンクされますのでクリックする事で該当する仕様書へジャンプする事ができます。



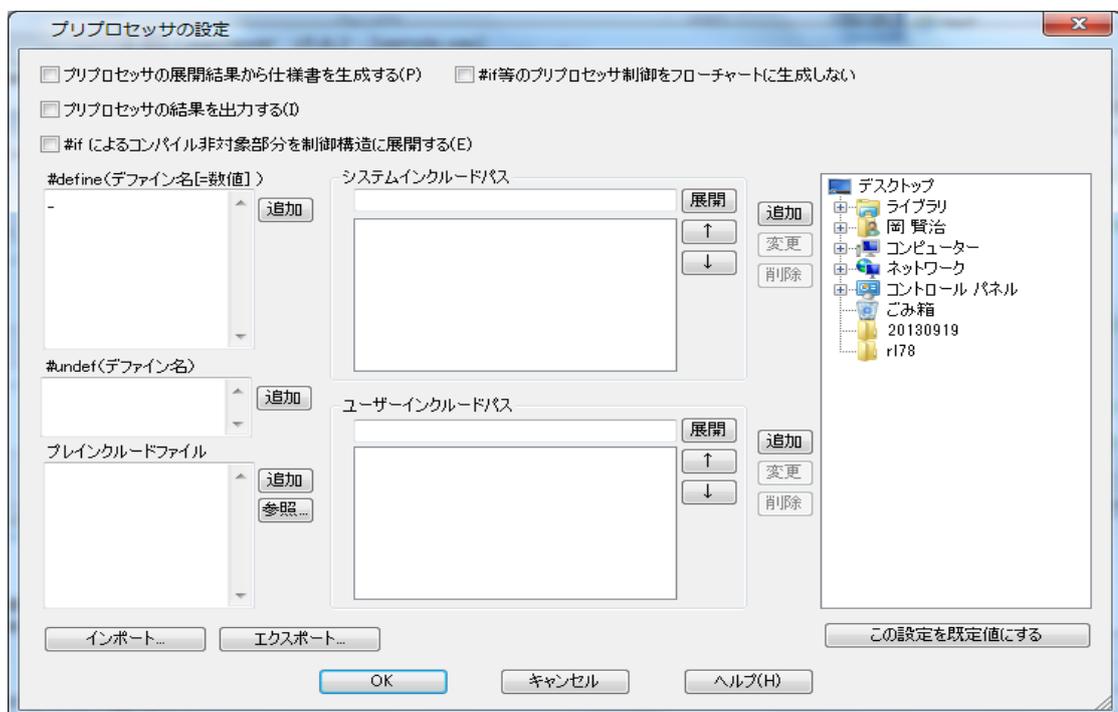
11. 詳細解析モードの詳細設定について

本項目では、詳細解析モードの追加説明と致しまして、詳細解析モードで仕様書生成するための設定方法をご紹介します。本チュートリアルにて使用するサンプルプログラムはシンプルな構成であるため詳細設定は必要ありません。しかし、実際の組込み開発プロジェクトでは、詳細設定が必要な場合があります。

プリプロセッサの設定

CasePlayer2 では解析時にプリプロセッサの設定情報を使用してソースコードの解析を行うため、開発時にクロスコンパイラで設定されているプリプロセッサの設定情報を CasePlayer2 側にも与えてください。

1. プロジェクトビュー(設定タブ) を表示して下さい。
2. [プリプロセッサ]をダブルクリックして下さい。



#define (デファイン名 [=数値])

プリプロセッサに渡すデファインマクロを指定してください。複数指定する場合には、改行で区切ってください。改行するには、Ctrl+Enter キーを押下してください。[追加]ボタンを押下すると、デファインマクロを追加できます。

#undef (デファイン名)

ソース解析前に、指定したデファイン名を無効にします。

プレインクルードファイル

全ての C ソースファイルに予めインクルードさせるファイルのファイル名を指定します。お使いのクロスコンパイラが暗黙のキーワードや定義を有する場合、これらの解析エラーを回避するために使用します。

システムインクルードパス/ ユーザーインクルードパス

プリプロセッサに渡すシステムインクルードパス(#include <…>のパス)、ユーザーインクルードパス(#include “…”のパス)を設定します。パスを追加するには、ストボックスの右側にある「追加」ボタンをクリックして、[フォルダの選択]ダイアログを表示し、指定するフォルダを選択した状態で[OK]ボタンをクリックしてください。

C オプションパラメータ

この設定は、開発に使用しているクロスコンパイラ独自の方言に対応するためのものです。CasePlayer2 の解析エンジンは、指定した言語仕様に対応していますが、クロスコンパイラ独自の言語仕様(方言)には対応できません。このため、ソースコードに記述された方言を、解析時にどのように扱うかを設定します。

1. プロジェクトビュー(設定タブ) を表示して下さい。
2. [C オプションパラメータ]をダブルクリックして下さい。



User Keyword ユーザーキーワード

非 ANSI-C またコンパイラ固有のキーワードはユーザーキーワードにて登録して下さい。以下に登録方法の例を挙げます。

例1 :int near p1;

near は「型修飾子」です。これを型修飾子として認識させるために、near のキーワードに「TYPEATT」指定します。

キーワード: near、種類: TYPEATT、後述記述: [空白]

例2: direct int array[100];

direct は「記憶クラス」です。これを記憶クラスとして認識させるために、direct のキーワードに「STRAGE」指定します。

キーワード: direct、種類: STRAGE、後述記述: [空白]

例3: _asm (“ “)

_asm は「インラインアセンブラ記述」です。これをインラインアセンブラ記述として認識させるために、_asm のキーワードに「ASM」指定します。また、このキーワードに続く記述が _asm の内容の記述であることを示すために、後置記述に「(EXPRESSION)」を指定します。

キーワード: _asm、種類: ASM、後述記述: (EXPRESSION)

例4: _except (.....)

_except は例外処理を示すコンパイラ独自の記述文法です。この記述全体を無視させるために、_except のキーワードに「IGNORE」指定します。また、このキーワードに続く記述が _except の内容の記述であり、同時に無視させるために、後置記述に「(EXPRESSION)」を指定します。

キーワード: _except、種類: IGNORE、後述記述: (EXPRESSION)

例5: INT32 val;

INT32 は変数の型を表します。型として認識させるために INT32 のキーワードに「TYPE」を指定します。

キーワード:INT32、種類:TYPE、後述記述:[空白]

置き換えキーワード

コード内のキーワードを置き換えるために使用します。

例1: typedef _WCHAR_T_TYPE_ _Wchart;

このエラーを回避するために、_WCHAR_T_TYPE_ を int 型に置換して解析させます。

新キーワード:_WCHAR_T_TYPE_, 既にあるキーワード int

例2: typedef _SIZE_T_TYPE_ _Sizet;

このエラーを回避するために、_SIZE_T_TYPE_ を int 型に置換して解析させます。

新キーワード:_SIZE_T_TYPE_, 既にあるキーワード int

C オプションパラメータのその他の設定について

大部分のエラーは、上記のユーザーキーワードと置き換えキーワードを使用して回避することができます。上記以外には下記の設定がございます。

非抽出コメント

モジュール仕様書・概要説明等の抽出定義

構造体仕様書・概要説明等の抽出定義

#pragma 定義

12. 最後に

これにて、CasePlayer2 のチュートリアルを終了致します。

本チュートリアルにて、CasePlayer2 の操作方法について少しでもご理解を深めて頂ければ幸いです。

CasePlayer2 チュートリアル

※会社名・商品名は各社の商標または登録商標です。

※本資料の無断転載、複写は禁止しております。

ガイオ・テクノロジー株式会社**■ユーザーサポートのご案内**

http://www.gaio.co.jp/support/support_about.html

■使用方法に関するお問い合わせ方法

ご質問は、ユーザーサポート窓口をご利用ください。

http://www.gaio.co.jp/support/support_entry.html

ユーザーサポート窓口へのお問い合わせには、ユーザーIDが必要です。

※保守契約がない場合は、いかなるサポートも提供致しません。