

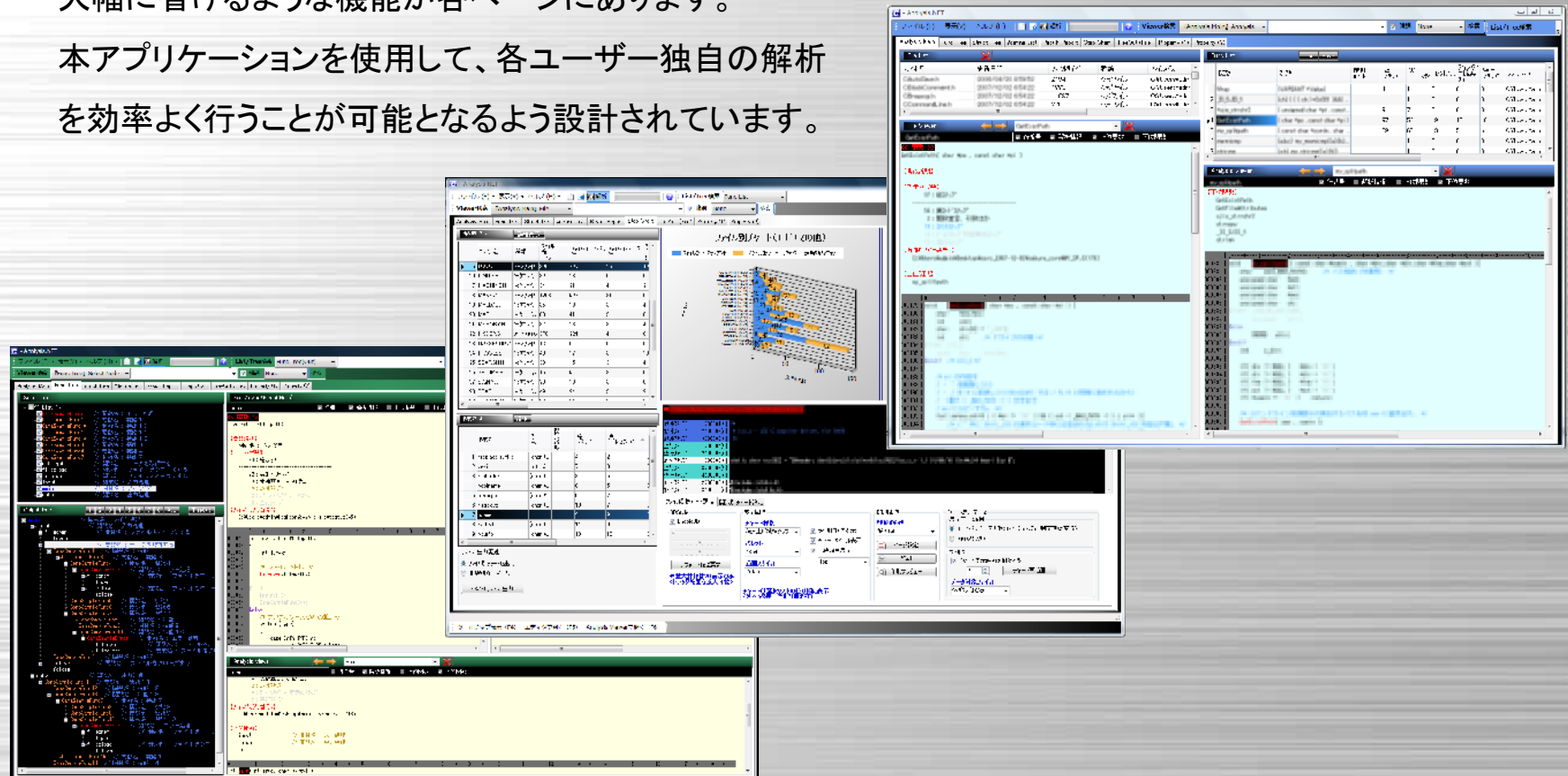
## ■ C-Analysis.NETとは？

C-Analysis.NETは、C言語のソース・ヘッダファイルの解析を支援するためのソフトウェアです。

タブ型のアプリケーションで、カーソルを置くだけで解析データがツールチップで表示される機能、

関数ツリーにより呼び出し順にたどれる「Func Tree」ページなど、関数や構造体などを参照する手間が大幅に省けるような機能が各ページにあります。

本アプリケーションを使用して、各ユーザー独自の解析を効率よく行うことが可能となるよう設計されています。



## ■ 必須要件

- 動作には、.NET Framework 4.0以上がインストールされていること。

<http://msdn.microsoft.com/ja-jp/netframework/default.aspx>

- 必須ではないが、Agentファイルがあれば要所で説明してくれる秘書Agentを起動することができる。  
XP/2000/Vistaでは、標準でインストールされています。(C:\WINDOWS\msagent\chars 等)  
また、下記URLなどでダウンロードが可能です。(Ver.1.00 リリース現在)

<http://www.microsoft.com/products/msagent/main.aspx>

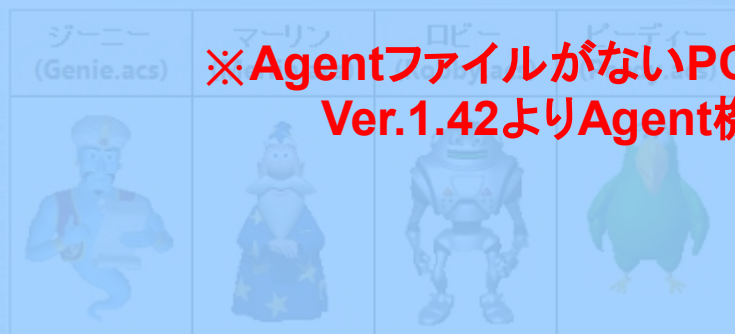


図. MSDNで公開されているキャラクターファイル

## ■ 公式ホームページ

本ソフトウェアのダウンロードやヘルプなど。

<http://canalysis.web.fc2.com/>

## ■ 使用手順

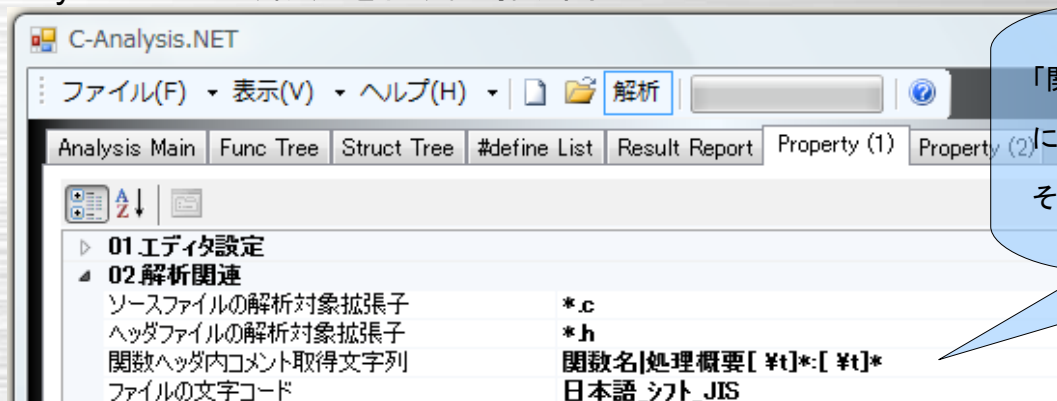
以下に、解析ファイル登録～解析完了～解析後の手順について記載します。

(※本ツール起動後、秘書Agentが使用手順について各所でナビゲートしてくれます。)

**Ver.1.42よりAgent機能はなくなりました**

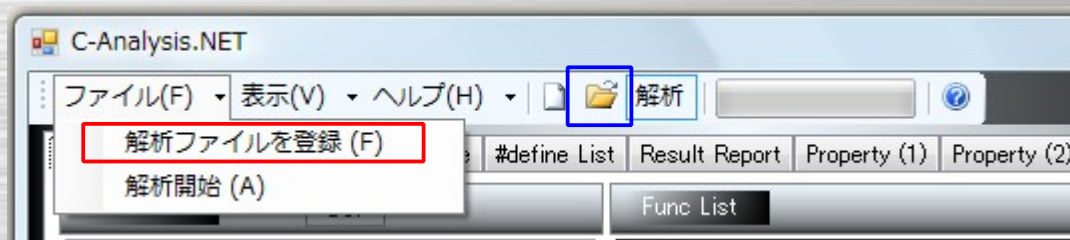
### ◆ 解析ファイルの登録

1. Propertyページにて、設定を行う。(解析関連など)

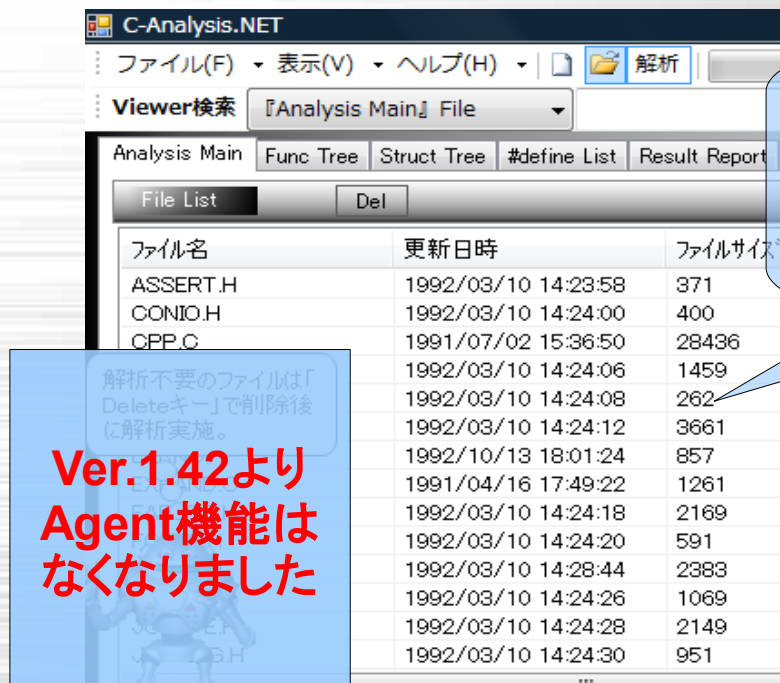


「関数ヘッダ内コメント取得文字列」  
については、別途説明がありますので  
そちらを参照下さい。

2. メニューの「ファイル」－解析ファイルを登録 を選択。(※青枠で示しているボタンでも同様)



3. 解析対象フォルダを選択することで、サブフォルダ以下のファイルを取得。



解析不要ファイルは  
選択状態で「Deleteキー」または「Del」ボタン  
を押下すると削除されます。

解析対象ファイルは必要最小限、より少なく  
することが重要です。  
本ツールはソースコードを詳細に解析しますので、  
解析スコープを狭めることで、より効果的に  
使用することができます。

4. 登録されたファイルのうち、解析不要のファイルをFile Listから削除する。
5. これで、解析対象ファイルの登録が完了です。

## ◆ 解析開始～完了

1. メニューの「ファイル」→「解析開始」を選択
2. 解析が完了すると、解析データが各所に表示されます。

## ◆ プリプロセッサ定義の確認

解析完了後、「Result Report」ページの【検出プリプロセッサ定義】には、解析したファイル内で検出したプリプロセッサ定義が表示されます。

解析時のディレクティブ判定に使用する【プリプロセッサ定義設定】に、検出プリプロセッサ定義を追加することができます。

「Menu」ボタン、またはセル上で右クリック押下によりコンテキストメニューが表示され、追加することができます。

(※「項目全てを無効状態で追加」など)

検出プリプロセッサ定義

プリプロセッサ定義	ディレクティブ判定結果	備考	ファイルパス	行数
135	<input type="checkbox"/>		C:\Users\admin...	178
136 MSKANJI	<input type="checkbox"/>		C:\Users\admin...	40
137 NDEBUG	<input type="checkbox"/>		C:\Users\admin...	10
138 NULL	<input checked="" type="checkbox"/>		C:\Users\admin...	12
139	<input checked="" type="checkbox"/>		C:\Users\admin...	12
140	<input checked="" type="checkbox"/>		C:\Users\admin...	12
141	<input checked="" type="checkbox"/>		C:\Users\admin...	12
142	<input checked="" type="checkbox"/>		C:\Users\admin...	12
143	<input checked="" type="checkbox"/>		C:\Users\admin...	18
144	<input checked="" type="checkbox"/>		C:\Users\admin...	8
145	<input checked="" type="checkbox"/>		C:\Users\admin...	8
146 OLDLSIFASHION	<input type="checkbox"/>		C:\Users\admin...	603
147	<input type="checkbox"/>		C:\Users\admin...	630

138～145までは「NULL」で同定義のため表示がグループ化されています。

【プリプロセッサ定義設定】に追加

プリプロセッサ定義設定

有効/無効	プリプロセッサ定義	備考
<input checked="" type="checkbox"/>	_ASSERT_H	
<input type="checkbox"/>	CLOCK_T_DEFINED	
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>	_FONTL_H	
<input type="checkbox"/>	_FILE_DEFINED	
<input type="checkbox"/>	_FLOAT_H	

登録完了しました。

OK

登録完了後、有効/無効を設定して再解析することで、設定内容が解析結果に反映されます。



## ◆ 排除関数設定

解析完了後、「Func Tree」ページにて関数ツリーを展開した場合、解析には不要な関数も出力されている場合があります、排除関数設定にて排除することができます。

The screenshot displays the C-Analysis.NET interface with two panels. The left panel, titled 'Output Tree', shows a hierarchical tree of functions. A blue callout bubble points to the 'printf' and 'fprintf' functions under the 'main' node, stating: '下記ツリーで "printf" と "fprintf" が解析には不要で排除関数に設定した場合 (※選択状態で右クリックより追加可)'. A red box highlights the 'CanSampleError' node, which contains 'memset', 'memcpy', 'printf', 'fileopen', 'fclose', and 'fileclose'. A blue arrow points from this box to the right panel. The right panel, titled 'Select Tree', shows a list of functions with checkboxes. A blue callout bubble points to the 'CanSampleError' entry, stating: 'CanSampleError 配下の関数を出力しない場合はチェックを外す。'. Below this, another blue callout bubble points to the 'CanSampleError' entry, stating: '解析に不要な関数が排除された結果、見通しが良くなります。 (※Tree展開時は特に効果あり)'. The right panel also shows a tree view of the selected functions, with 'CanSampleError' and its sub-functions excluded.

Output Tree

1 2 3 4 5 6 7 8 ALL

main // メイン処理  
input // 入力処理  
fileopen // ファイルをオープンする  
fopen  
printf  
data\_get  
CanSampleFunc1 // 機能3を実行する処理  
fprintf  
CanSampleFunc3 // 機能4～7を実行する処理  
fprintf  
CanSampleFunc4 // 機能4  
fprintf  
CanSampleError // エラー処理  
memset  
memcpy  
printf  
fileopen  
fclose  
fileclose  
CanSampleFunc5 // 機能5  
fprintf  
CanSampleFunc6 // 機能6  
fprintf  
CanSampleFunc7 // 機能8、9、10  
CanSampleFunc8 // 機能8  
CanSampleFunc9 // 機能9  
CanSampleFunc10 // 機能10  
CanSampleError // エラー処理  
memset  
memcpy  
printf  
fileopen  
fclose  
fileclose  
CanSampleFunc2 // ファイルに書き込む処理  
fprintf  
fclose  
fileclose  
output // 出力処理  
CanSampleFunc11 // 機能12～14の実行  
CanSampleFunc12 // 機能12  
CanSampleFunc13 // 機能7と機能15を呼び出す処理  
CanSampleFunc14 // 機能8、9、10を呼び出す処理

Select Tree

<< List >>

CanSampleError // エラー処理  
CanSampleFunc1 // 機能3を実行する処理  
CanSampleFunc3 // 機能4～7を実行する処理  
CanSampleFunc4 // 機能4  
CanSampleError // エラー処理  
CanSampleFunc5 // 機能5  
CanSampleFunc6 // 機能6  
CanSampleFunc7 // 機能8、9、10を呼び出す処理  
CanSampleFunc8 // 機能8  
CanSampleFunc9 // 機能9  
CanSampleFunc10 // 機能10  
CanSampleError // エラー処理  
CanSampleFunc2 // ファイルに書き込む処理  
fileclose // ファイルをクローズする  
output // 出力処理  
CanSampleFunc11 // 機能12～14の実行  
CanSampleFunc12 // 機能12  
CanSampleFunc13 // 機能7と機能15を呼び出す処理  
CanSampleFunc14 // 機能8、9、10を呼び出す処理  
exit

## ◆ Treeのファイル出力

関数ツリー、構造体ツリーは「TreeOut (File)」ページよりファイルに保存することができます。  
各ページ(「Func Tree」「Struct Tree」)にてOutput Treeへツリーを出力後、  
「Tree Out」ボタン押下で「TreeOut (File)」ページにTree出力することができます。  
「TreeOut (File)」ページでは、下記のファイル形式でファイルに保存することができます。

保存形式	内容
エクセルファイル(.xls)	エクセルファイル形式(Microsoft Excelで開くことができる形式)で保存します。 この形式で出力すると、階層構造をグループ化して出力されますので、Microsoft Excel上でも階層レベルで展開することが可能です。
テキストファイル(.txt)	ASCIIテキストファイルとして保存します。
リッチテキスト(.rtf)	リッチテキスト形式(RTF)で保存します。 表示文字色がそのまま出力され、Microsoft Word やワードパットなどで開くことができます。 Viewer背景色を変更することで文字色の変更が可能です。 <b>※注意</b> Viewer背景色が「Black」の場合は、文字色が白になります。 ファイル出力後は背景色を黒にするなど変更して下さい。

## ■ Analysis Mainページ

関数単位で解析を行うページです。

The screenshot displays the C-Analysis.NET application window. The interface includes a menu bar (File, Edit, View, etc.), a toolbar, and several panels. The 'File List' panel on the left shows a list of files. The 'Func List' panel on the right shows a list of functions. The 'File Viewer' panel at the bottom left shows the content of a selected file. The 'Analysis Viewer' panel at the bottom right shows the analysis results.

Callouts provide instructions and information:

- File Viewerにて、解析データのある位置(関数など)にカーソルを置いて、「F6キー」でAnalysis Viewerに表示します。** (In the File Viewer, place the cursor on the location of the analysis data (function, etc.) and press the F6 key to display it in the Analysis Viewer.)
- 関数を選択すると、File Viewerに関数の内容をファイルから読み込み、表示します。** (When a function is selected, the content of the function is loaded from the file into the File Viewer and displayed.)
- Analysis Viewerはライセンスがない場合は、無効化されます。(選択・スクロール不可)** (The Analysis Viewer is disabled if there is no license. (Selection/scrolling is not possible))
- ライセンスがある場合は、File Viewerと同様に解析データのある位置(関数など)にカーソルを置いた場合の機能が動作します。** (If there is a license, the function operates when the cursor is placed on the location of the analysis data (function, etc.) in the File Viewer, similar to the File Viewer.)



## ■ Func Treeページ(※Struct Treeページも同等)

関数ツリー(※構造体ツリー)による解析を行うページです。

The screenshot shows the C-Analysis.NET interface with the Func Tree page selected. The interface includes a menu bar, a toolbar, and several panels: 'Func Tree (Out)', 'File Viewer (Parent Node)', 'File Viewer (Child Node)', and 'Analysis Viewer'. The 'Func Tree (Out)' panel displays a tree structure of functions. The 'File Viewer (Parent Node)' and 'File Viewer (Child Node)' panels show the source code of the selected function. The 'Analysis Viewer' panel shows the analysis results.

関数を選択し、ダブルクリック、または「ENTERキー」で関数ツリーをOutput Treeに展開表示します。

関数を選択すると、File Viewerに関数の内容をファイルから読み込み、表示します。

Output Treeにて選択された項目の親ノードを表示します。

Output Treeにて選択された項目を表示します。

Analysis Viewerはライセンスがない場合は、無効化されます。(選択・スクロール不可)

ライセンスがある場合は、File Viewerと同様に解析データのある位置(関数など)にカーソルを置いた場合の機能が動作します。

## ■ File Viewer、Analysis Viewerの機能

ここでは、ソース・ヘッダファイルの表示を行います。また、カーソル位置の文字列が本ツールの解析データと一致する場合は解析データを表示することが可能です。(関数、構造体、#defineなど)

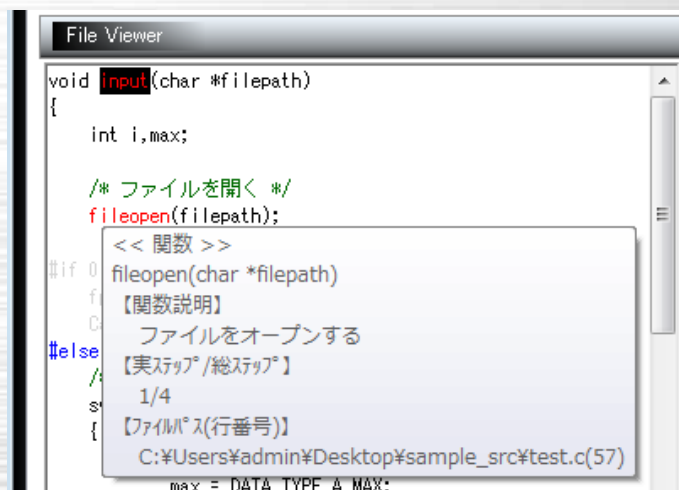


図. カーソル位置の文字列が関数の解析データと一致した場合のツールチップ表示

ツールチップにファイルパス・行番号が表示される状態であれば、下記の機能が使用できます。

「F4キー」で、ツールチップ再表示。

「F5キー」で、エディタを開く。(※ライセンス機能制限)

「F6キー」で、Analysis Viewer(同ページ内)で開く。(※ライセンス機能制限)

## ■ List/Tree検索、Viewer検索の機能

各ページにて表示している解析データの検索を行う機能です。

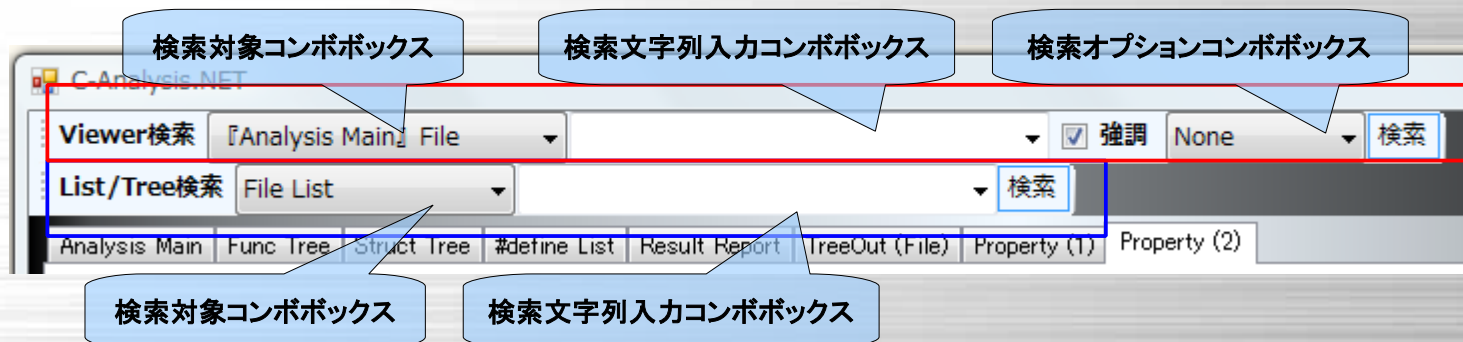


図. List/Tree検索とViewer検索

- ◆ List/Tree検索: 上記図の青枠がList/Tree検索で使う部分となります。
- ◆ Viewer検索: 上記図の赤枠がViewer検索で使う部分となります。

### ● 使用方法

検索対象のコントロール(ListやTree、Viewer)にフォーカスが当たると、上記枠内の検索対象コンボボックスが自動で更新されます。

検索文字列を入力し、検索ボタンを押下(検索文字列入力コンボボックス内でENTERでも可)すると検索を実行します。

検索対象コンボボックスは手動でも変更可能ですので、他ページを検索することも可能です。他ページの検索実行時は、そのページに切り替わります。

- 検索オプションコンボボックス

- 【検索オプション】

- MatchCase:

- 大文字と小文字が正確である検索文字列のインスタンスだけを検索します。

- None:

- 検索で見つかるインスタンスが検索文字列の語句に完全一致であるかどうかにかかわらず、検索文字列に一致するインスタンスをすべて検索します。

- WholeWord:

- 検索文字列の語句に完全一致するインスタンスだけを検索します。

- 結果表示

- List検索 (Func List、#define List) の場合:

- List全体を検索し、結果を選択状態で表示します。

- 選択列が検索対象となります。(Func Listの関数説明、など)

- Tree検索 (Select Tree、Output Tree) の場合:

- 選択位置から下方向に検索し、一致した場合は選択状態となります。

- Viewer検索 (File Viewer、Analysis Viewer) の場合:

- caret位置 (カーソル) から下方向に検索し、一致した場合は選択状態となります。

- 便利な使い方

- 検索対象のコントロール (ListやTree、Viewer) にフォーカスを当て、Ctrl + Fキーを押下すると検索文字列入力コンボボックスにフォーカスが移動します。

- そのまま検索文字列を入力し、ENTERキーで検索実行が可能です。

- 検索実行後、F3で下方向に検索、SHIFT+F3で上方向に検索が可能です。

- (フォーカスが検索対象のコントロールにある場合 (Listは除く))

## ■ Grep検索の機能

ファイル中から、正規表現に一致する行を検索して出力を行う機能です。

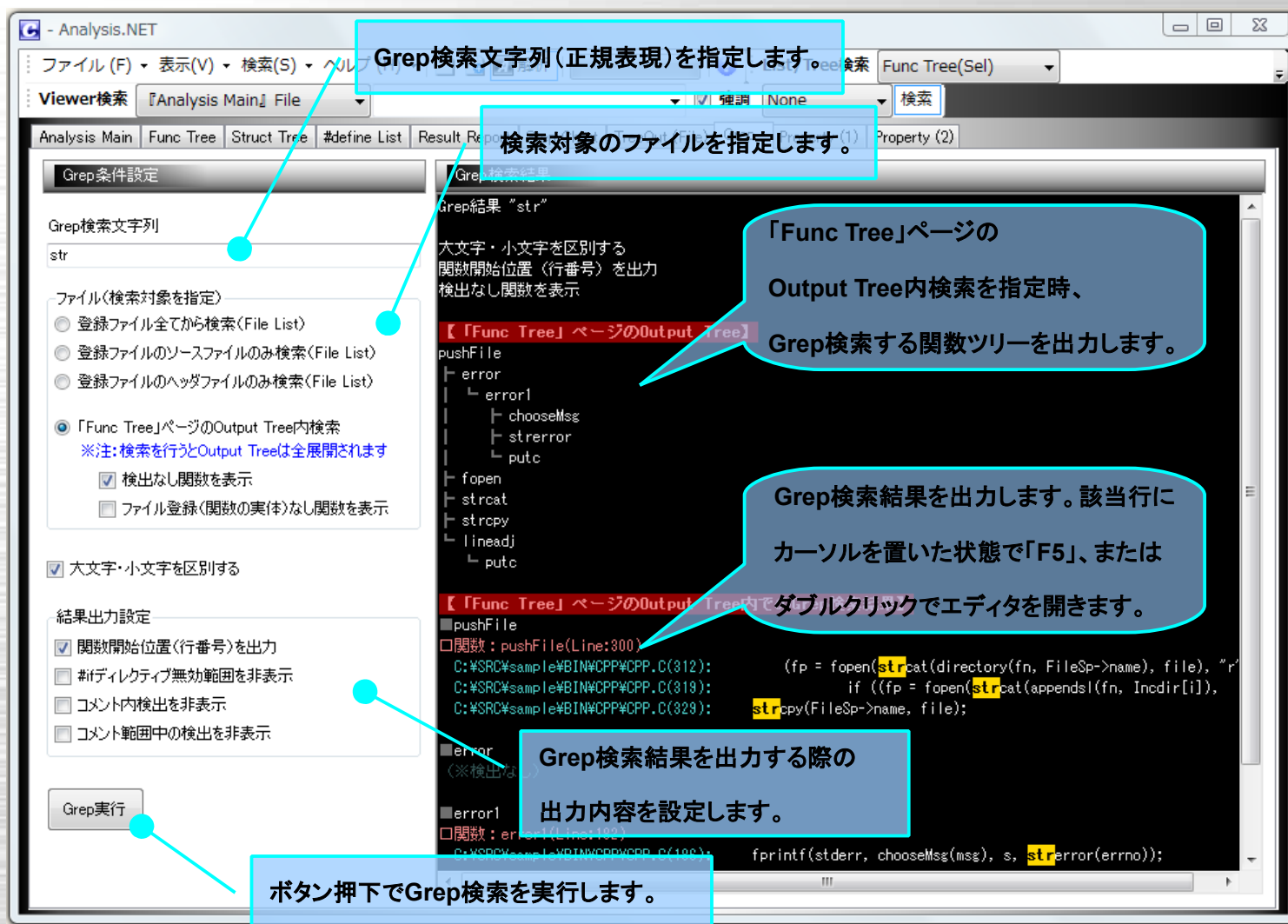


図. Grep検索



## ■ 解析範囲

各解析データの解析範囲は下記表の通りです。

解析範囲	ソースファイル	ヘッダファイル
関数	○	○
構造体	×	○
#define	×	○
検出プリプロセッサ定義	○	○

## ■ 条件付きコンパイル・ディレクティブ（#if、#ifdef、#ifndef...等）

条件付きコンパイル・ディレクティブとは、ソース内の一部のコンパイルを条件によって抑止させるものです。

条件付きコンパイル・ディレクティブについて、プリプロセッサ定義を判定し解析を行います。

（※プリプロセッサ定義は、「Property (2)」ページにて設定）

本ソフトでは下記に対応しています。

- ・#if、#ifdef、#ifndef
- ・#if defined、#if !defined
- ・#elif、#else、#endif

## ■ 関数ヘッダ内コメント取得文字列

関数の直前にコメントがあるような場合、その関数ヘッダから関数説明を取得することが可能です。

「Property (1)」ページにて設定できます。

設定条件(正規表現)にマッチした行を取得します。設定条件の文字列は除かれます。

正規表現については、下記URL参照。

<http://msdn.microsoft.com/ja-jp/library/hs600312.aspx>

複数指定時、検出順に複数行で使用されます。

メタキャラクタ	意味(メタキャラクタとして使用する場合)
[]	キャラクタクラス
()	パターンをグループ化する
^	行頭(またはキャラクタクラス 内での補集合)
.	改行を除く任意の1文字
*	直前のパターンの0回以上の繰り返し
+	直前のパターンの1回以上の繰り返し
	パターンの論理和
[ ¥t]*	半角スペース・タブ(0回以上)を検出する正規表現の例

## ■ ライセンス料金

本ソフトはシェアウェアです。試用期間は無制限です。

機能制限はありますが試用期間に制限はありませんので、充分試用して問題ないことを確認してからご購入下さい。

(動作保証や、将来的な機能拡張を保証するものではありません。)

ライセンス数による値引きはありません。

送金方法は、以下の2通りあります。

### 1. ベクターのシェアレジから送金

シェアレジ番号: SR293560

作品名: C-Analysis.NET

作品本体価格	2800円
消費税	140円
購入者負担手数料	100円
消費税	5円
計	3045円

※送金に関する手続きやサポートはすべてベクターが行ってくれることになります。

購入者負担手数料とは、その分と考えて下さい。

ライセンス・キー再送付サービスがあるのもシェアレジ利用の利点です。

## 2. 作者に直接メール後、指定口座に送金 ⇒ (※作者アドレスは「■免責・サポート」欄参照)

ご送金頂く口座情報(楽天銀行)を送付致します。

ライセンス・キーを取得したい旨のメールを送付下さい。

メールには、使用者氏名、ふりがな、メールアドレスをご記載下さい。

振込み確認後、作者からライセンス・キーを送付致します。

※会社での購入の際など、請求書、領収書の発行が必要な場合は、発行いたします。

(詳細はメールで連絡頂いた際に相談させていただきます)

※振込み手数料については、お客様ご負担とさせていただきます。

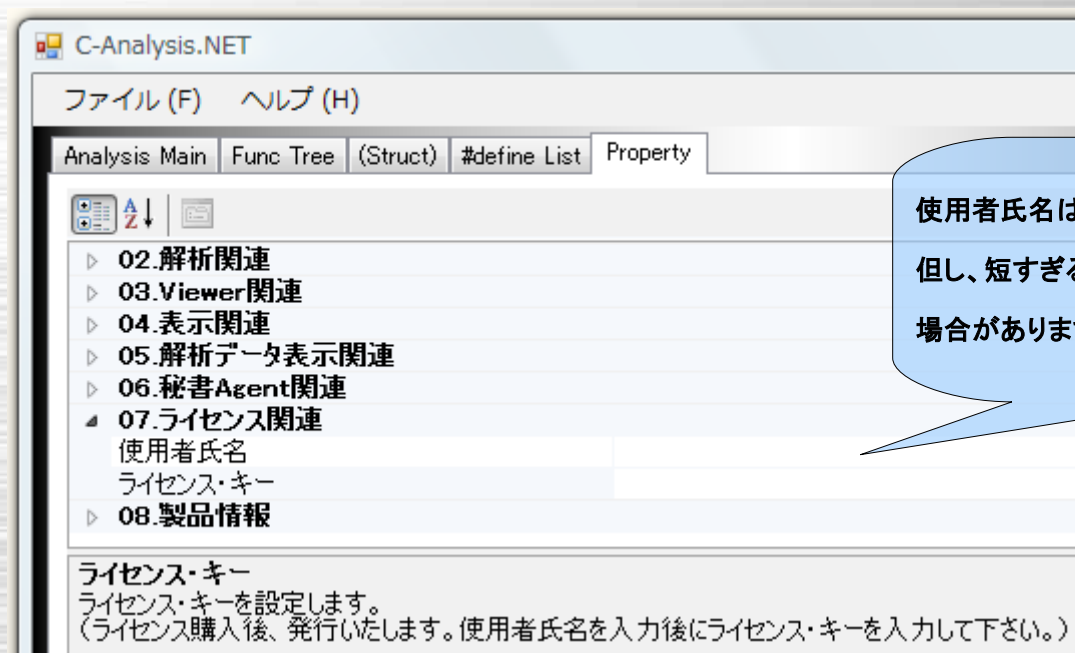
作品本体価格	2800円
割引	-200円
消費税	130円
計	2730円

※こちらでの購入の場合はベクター関連手数料がかからないため、上記の額を割引しています。

## ■ ライセンス・キー入力

ライセンスを取得された方は下記に従い、ライセンス・キーを入力して下さい。

- ◆ ライセンス・キー購入の際に、作者に連絡いただいた使用者氏名を入力。
- ◆ 作者が発行したライセンス・キーを入力。



- ◆ ライセンス・キーが解除されれば、ライセンス機能制限が解除されます。



## ■ 使用条件

ライセンスを取得された方は、本人使用の場合のみ複数台のコンピュータに本ソフトをインストールして使用しても構いません。

ライセンス取得の際は、本ソフト使用人数で取得して下さい。

※企業様でご購入の場合は本ソフトをインストールするPC台数分で取得して下さい。

## ■ 免責・サポート

本ソフト使用により発生した、いかなる損害についても開発者側は一切の責任を負わないものとします。

転載を希望される方は、事前にメールを下さい。

本ソフトに対するご質問、不具合などはメールで受け付けております。

「C-Analysis.NET」作者:

[c\\_ana11@yahoo.co.jp](mailto:c_ana11@yahoo.co.jp)

※メールの件名 (Subject) に”C-Analysis.NET”の文字を入れて下さい。

例). 「C-Analysis.NET」問い合わせ

「C-Analysis.NET」ライセンス購入