

ソースコードを解析する
賢いプログラム エディタ

参照ツリー、クラス階層、コールツリーも表示できる
革新的なエディタ / ブラウザ



Version 3.5

スタート ガイド



エクセルソフト株式会社

はじめに

このたびは、Source Insight をご利用いただき誠にありがとうございます。

Source Insight は、C/C++、C#、Java プログラム、およびその他の言語を解析できる革新的な、プロジェクト志向のプログラム コード エディタ / ブラウザです。ソース コードを解析し、作業中に独自のシンボリック情報のデータベースをダイナミックに管理して、参照ツリー、クラス階層ダイアグラム、コール ツリーを表示、役立つコンテキスト情報を自動的に提供します。

大きなソースコードを理解する必要がある場合、Source Insight はソースコード ブラウザとして効果を存分に発揮します。

このスタート ガイドでは初めてこの製品を使用される方を対象に、インストールの手順および製品の起動方法について説明しています。

目次

1. インストール	1
1.1. インストール手順	1
■ インストールに必要な環境	1
■ プログラムのインストール	1
2. プロジェクトの設定	3
2.1. コモン プロジェクトの設定	3
2.2. プロジェクトの作成	8
2.3. 設定済みの日本語スタイルの読み込み	12
3. Source Insight の機能	14
3.1. SOURCE INSIGHT の機能概要	14
3.2. メイン ウィンドウ	15
3.3. リレーション ウィンドウ	16
3.4. リレーション ウィンドウの操作	19
4. カスタム コマンド	20
4.1. カスタム コマンド	20
■ カスタム コマンド ダイアログボックス	21
■ RUN フィールドの形式	23
■ コマンド シェルの実行	23
■ コマンドライン代替	24
5. 日本語環境での注意	28
5.1. 日本語フォントのスタイル変更	28
■ スタイルの変更	28
■ 設定済みの日本語スタイルの読み込み	31
5.2. 日本語環境での制限	32
■ 日本語の入力	32
■ UNICODE のソース	32
6. テクニカル サポート	33
6.1. テクニカル サポート	33
■ テクニカル サポートへのお問い合わせ	33
■ 製品のお問い合わせ	33

1. インストール

1.1. インストール手順

■ インストールに必要な環境

Source Insight 3.5 をインストールするには、下記の環境が必要です。

- OS: Microsoft Windows 2000、XP、Vista、NT 4.0 SP3+、98/Me、または 95(Internet Explorer 4.0 以上必須)
- CPU: Pentium クラス (Pentium II 266 MHz またはそれ以上を推奨)
- メモリ: 最低 64MB (128MB またはそれ以上を推奨)
- 空きディスク容量: 最低 4MB、フルインストール時は 12MB。

■ プログラムのインストール

ダウンロードしたファイルを解凍すると、以下の3つのファイルが展開されます。

- **Si35xxSetup.exe** : Source Insight 3.5 のインストールファイル (xx は、ビルド番号)
- **Source Insight 3.5_Start_Guide.pdf** : スタートガイド
- **JPN_Style.CF3**: 日本語表示用のスタイル Configuration ファイル (第5章を参照)

Source Insight 3.5 のインストールファイル (Si35xxSetup.exe) をダブルクリックします。

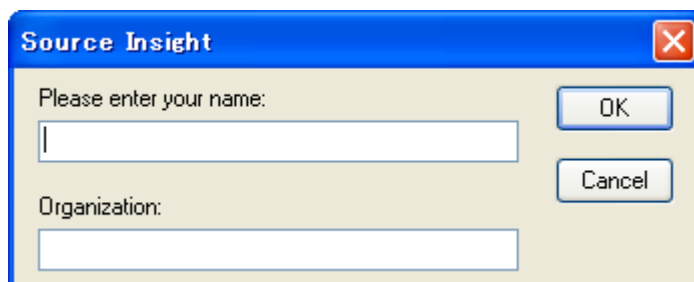
[End-User License Agreement (ライセンス許諾契約書)] ダイアログが表示されます。[Agree (同意する)] ボタンをクリックしてインストールを開始します。

インストール画面の指示に従いインストール操作を続行します。シリアル番号の入力画面が表示されたら、30日間評価版としてご利用のお客様は、空白のまま [Try It] ボタンをクリックします。





次に、お名前および会社名を入力して、[OK] をクリックします。

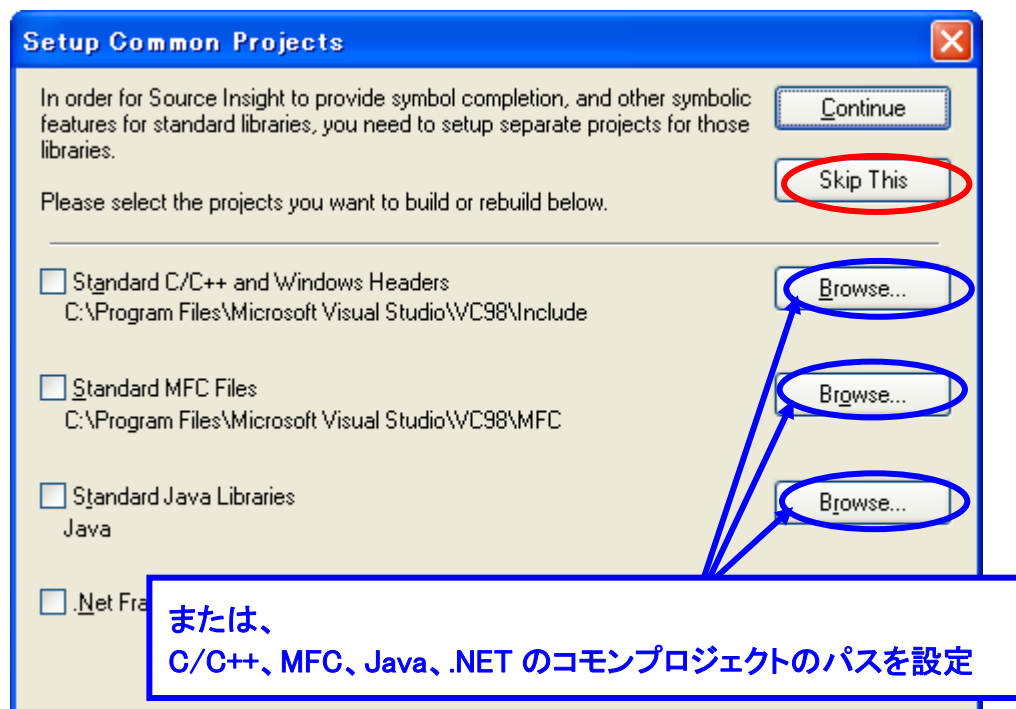


ユーザー情報の入力完了すると、Source Insight のコモンプロジェクトの設定画面に自動的に進みます。

2. プロジェクトの設定

2.1. コモン プロジェクトの設定

[Setup Common Project(コモン プロジェクトの設定)] ダイアログが表示されます。



コモンプロジェクトは、あとで **[Options]**メニューを選択し、**[Preferences] - [Symbol Lookups]** で **[Create Common Projects]** ボタンをクリックして作成または変更することができますので、作成しない場合は、**[Skip This]** を選択し、「2.2 プロジェクトの作成」に進みます。

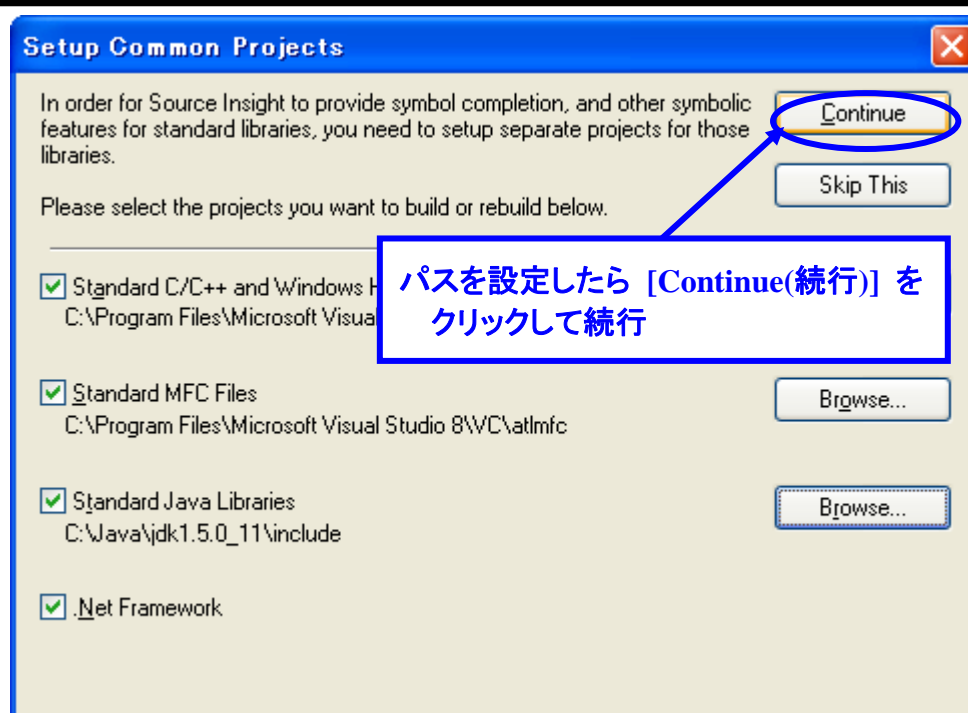
シンボル コンプリート機能、および C ランタイムまたは Java 標準パッケージなどの標準ライブラリ用の他のシンボリック機能を使用するには、これらのライブラリ用にそれぞれプロジェクトを設定する必要があります。カレント プロジェクトでシンボルを見つけれない場合に、これらのプロジェクトを検索します。

各コモンプロジェクト用に対応するディレクトリを指定してください。ライブラリのソースコードをインストールしている場合は、Source Insight からプロジェクトでソースコードを使用することができます。たとえば、C ランタイム ライブラリにある `strtok` 関数をコールすると、Source Insight は `strtok` のソースコードを見つけます。

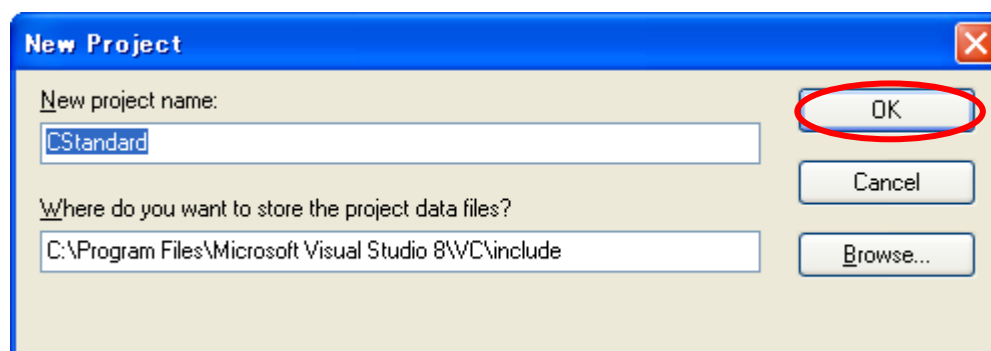
この時点で作成したプロジェクトは、自動的にプロジェクト シンボル パスに追加されます。

Visual C++ 6.0 がインストールされている場合、自動的に Include フォルダを検出します。

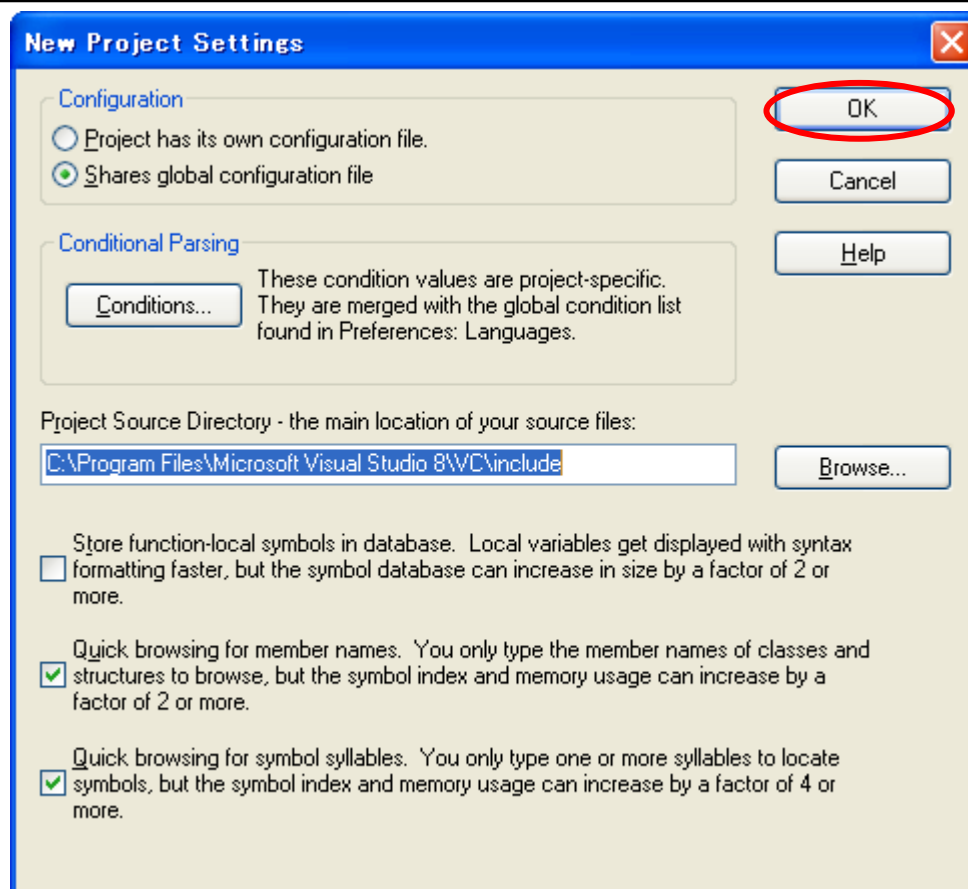
Visual Studio 2005 の場合、**[Browse]** ボタンをクリックし、対応するフォルダを指定します。



各コモンプロジェクトのパスを指定したら、**[Continue]** ボタンをクリックします。[New Project] ダイアログが表示されます。C 標準ライブラリのパスを指定します。

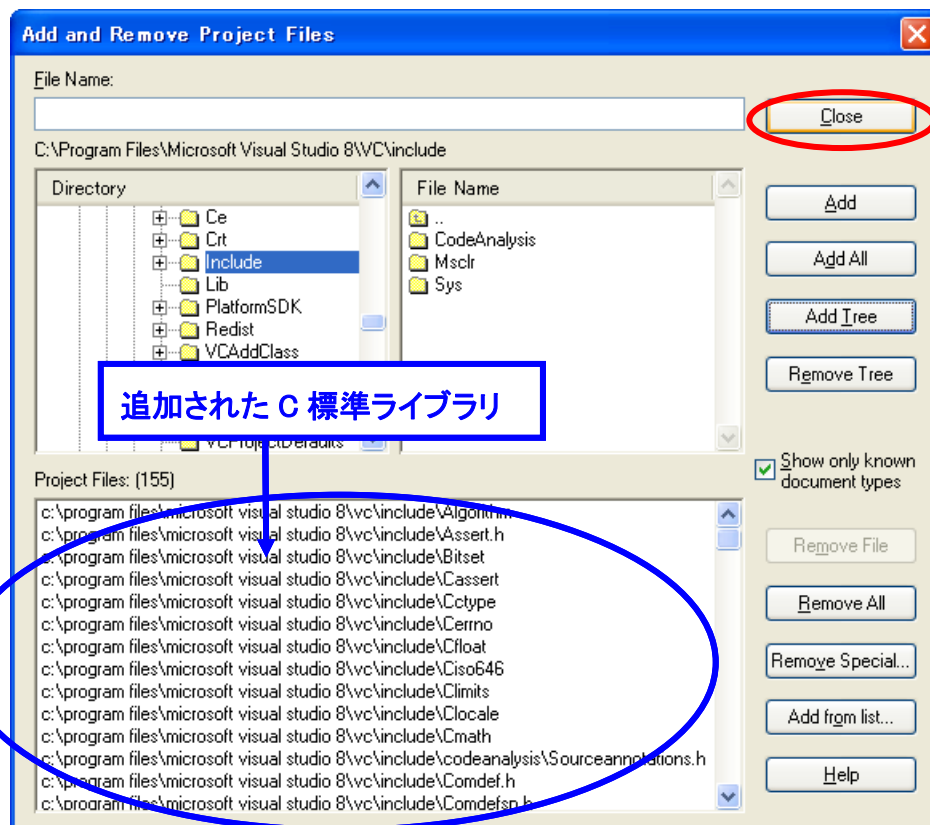
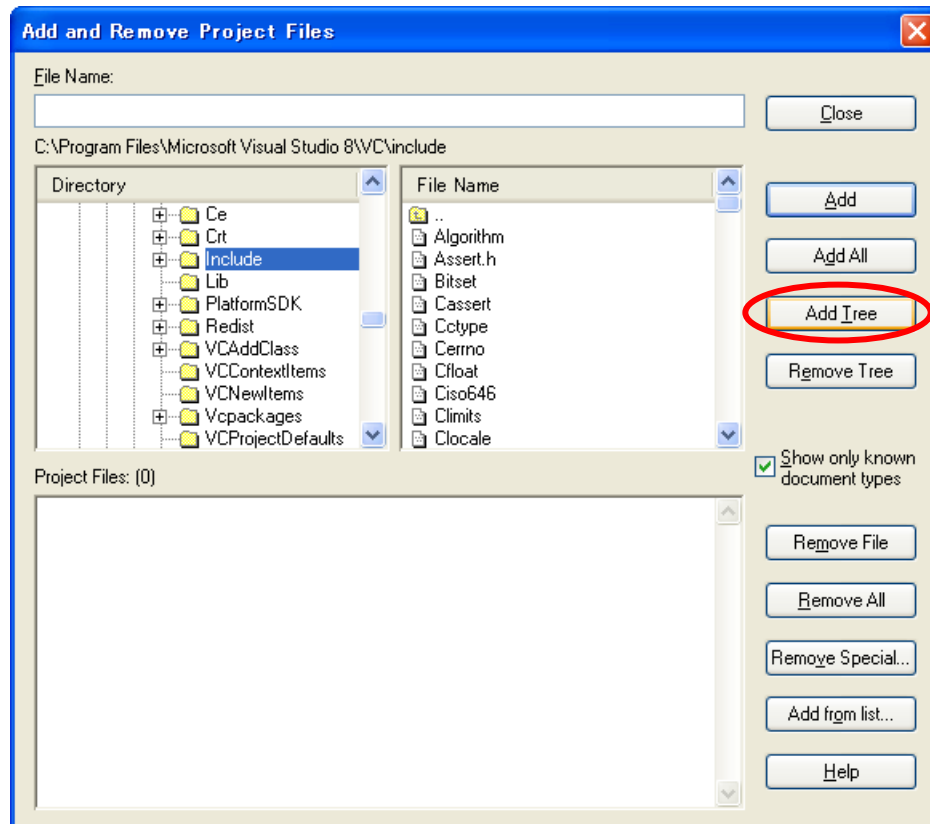


[OK] ボタンをクリックします。[New Project Settings] ダイアログが表示されます。

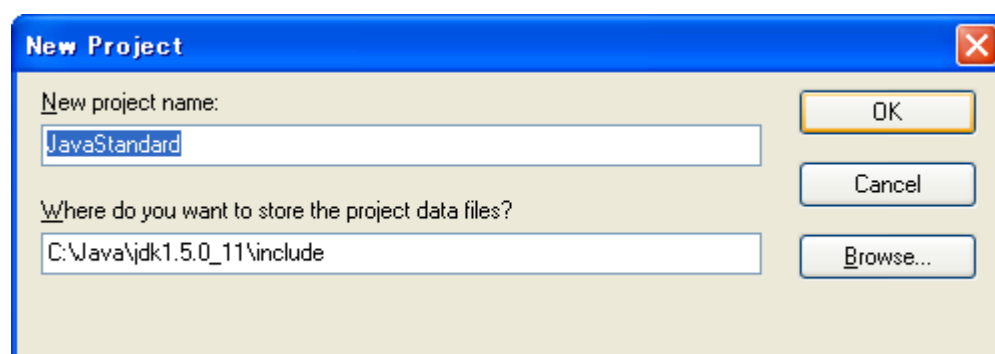
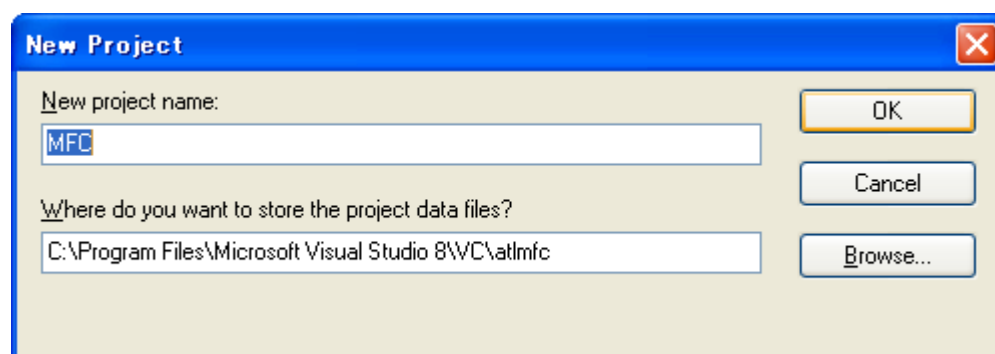


[OK] ボタンをクリックすると、[Add and Remove Project Files] ダイアログが表示されます。

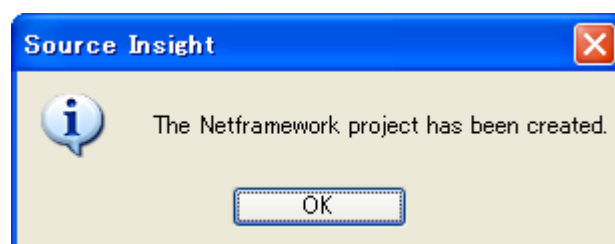
[Add Tree] ボタンをクリックし、Include フォルダ以下のファイルを読み込みます。



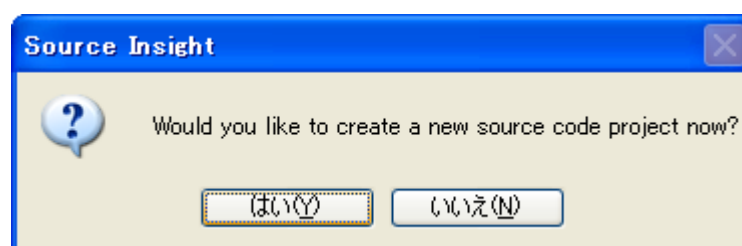
同様に、MFC プロジェクト、Java プロジェクトを設定します。



.NET Framework は、自動的に作成されます。



コモン プロジェクトの設定が完了すると、新規プロジェクトの作成に進みます。

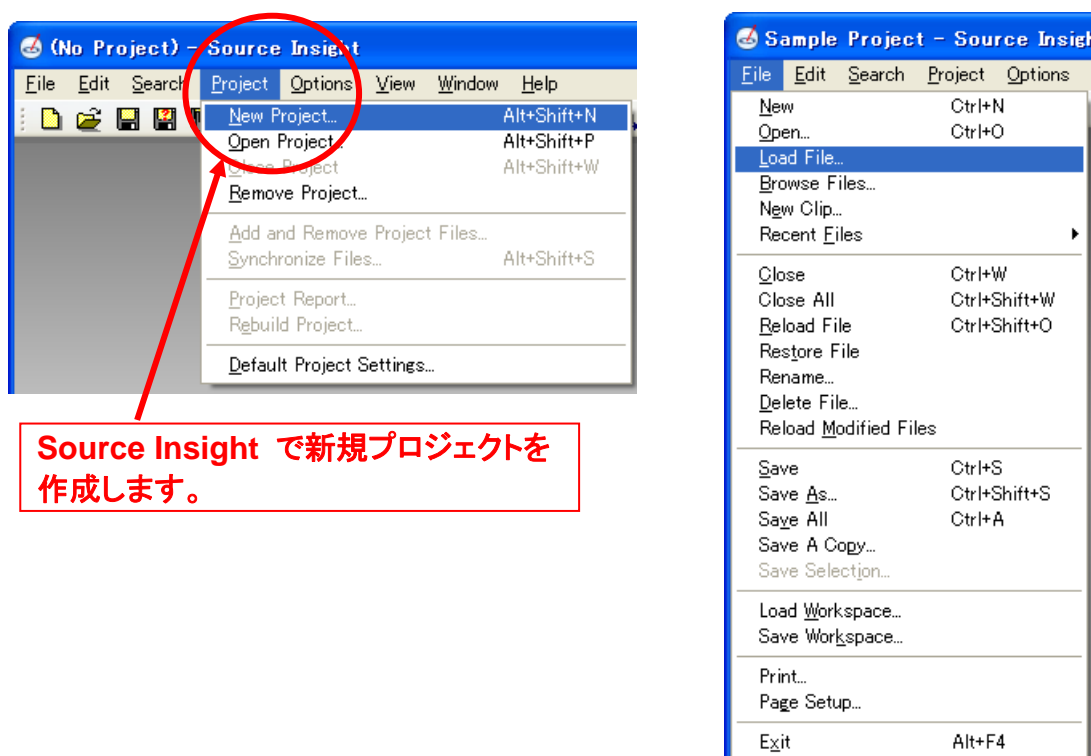


2.2. プロジェクトの作成

プロジェクトは、ソース ファイルの集まりです。プロジェクトを作成するときは、そこにファイルを追加する必要があります。Source Insight は、含まれているファイルを記録するためにプロジェクト用に簡単なファイル データベースを管理します。

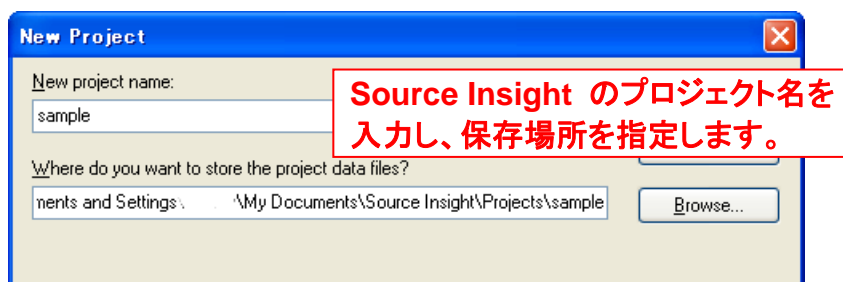
このスタートガイドでは、Visual Studio 2005 のウィザードで作成した簡単な C++言語の MFC アプリケーションサンプルを使用します。

新規のプロジェクトを作成するには、[Project] - [New Project] を選択します。既存の Source Insight のプロジェクトを開くには、[Project] - [Open Project] を選択します。

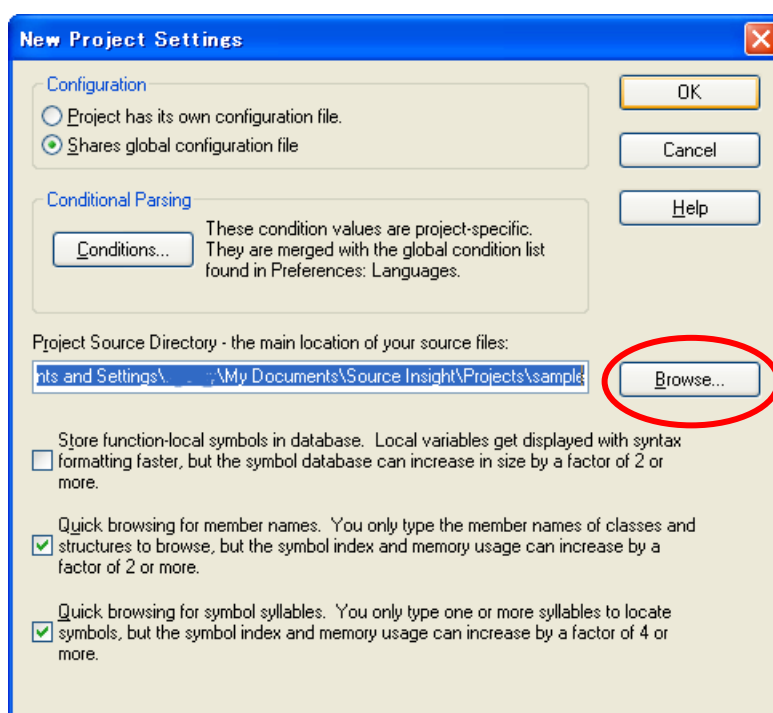


プロジェクトを作成しないで、個別のソースファイルだけを編集する場合、[File] - [Load File] を選択し、ソースファイルを読み込むこともできます。

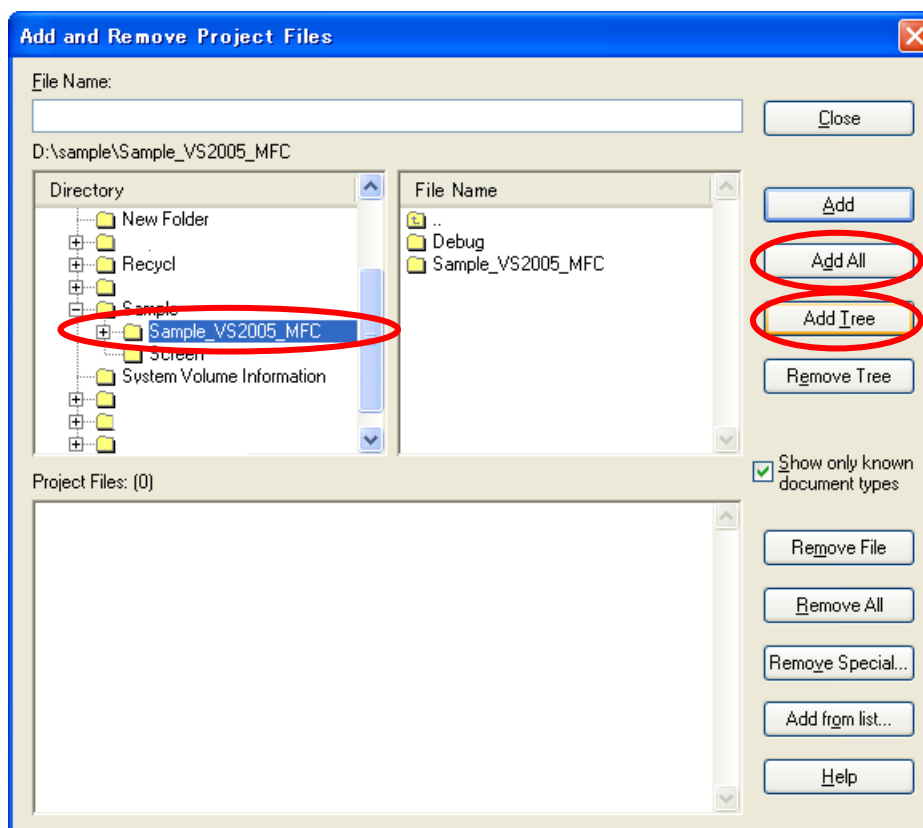
上記で新規プロジェクトの作成を選択すると、[New Project] ダイアログが表示されます。このダイアログで、プロジェクト名とその保存場所を指定します。



ソースファイルの場所を [Browse] ボタンで指定します。



ソースファイルのフォルダを指定し、[Add All] または [Add Tree] ボタンをクリックし、プロジェクトに必要なソースファイルをすべて追加します。ここで個別にファイルを追加することもできます。

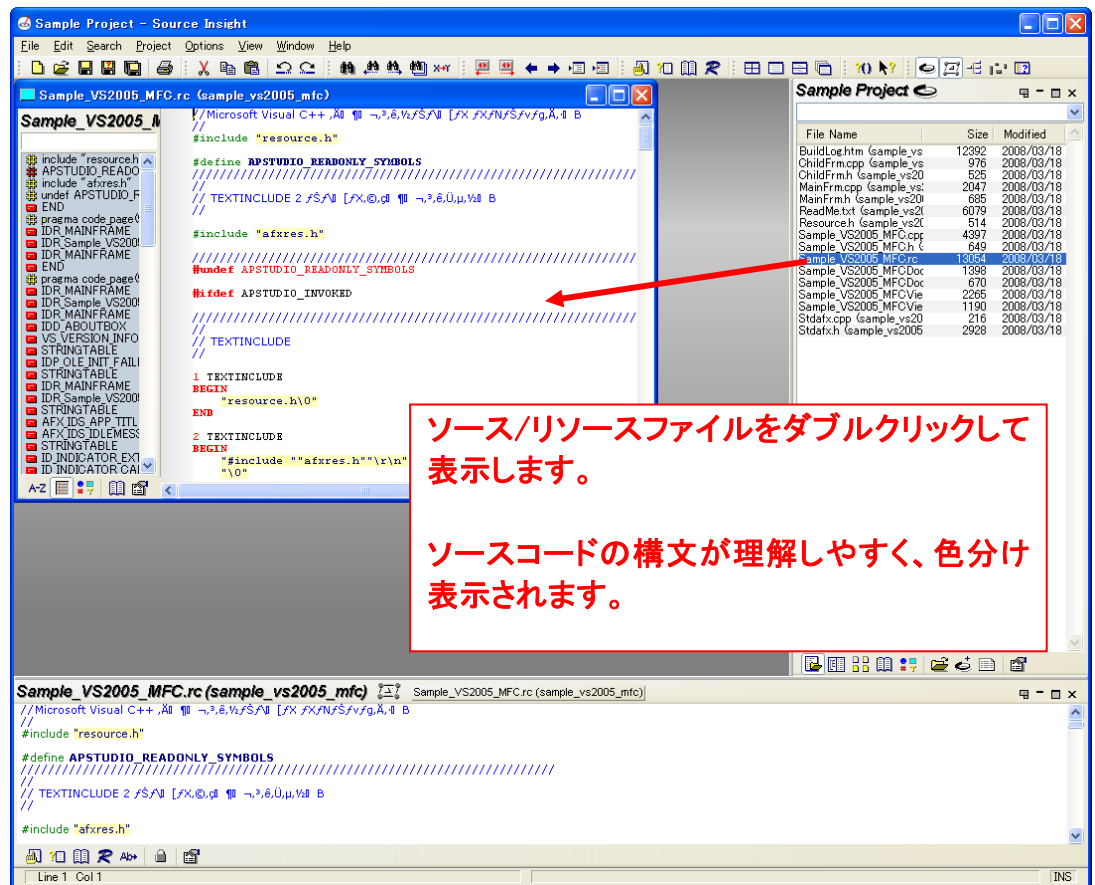


[Add and Remove Project Files (プロジェクト ファイルの追加と削除)] ダイアログで個別のファイルを追加、ディレクトリツリー全体をプロジェクトに追加することができます。

プログラム ソース ファイルとテキストファイルだけを Source Insight のプロジェクトに含めてください。EXE ファイルなどのバイナリ ファイルを含めても効果はありません。

ファイルの追加が完了したら、[Close] ボタンをクリックします。Source Insight のメイン画面に Project Window が表示され、ファイル一覧がリストされます。

このリストからソースコード ファイルをダブルクリックして開きます。この例では、日本語サンプルの RC ファイルを開きます。



2.3. 設定済みの日本語スタイルの読み込み

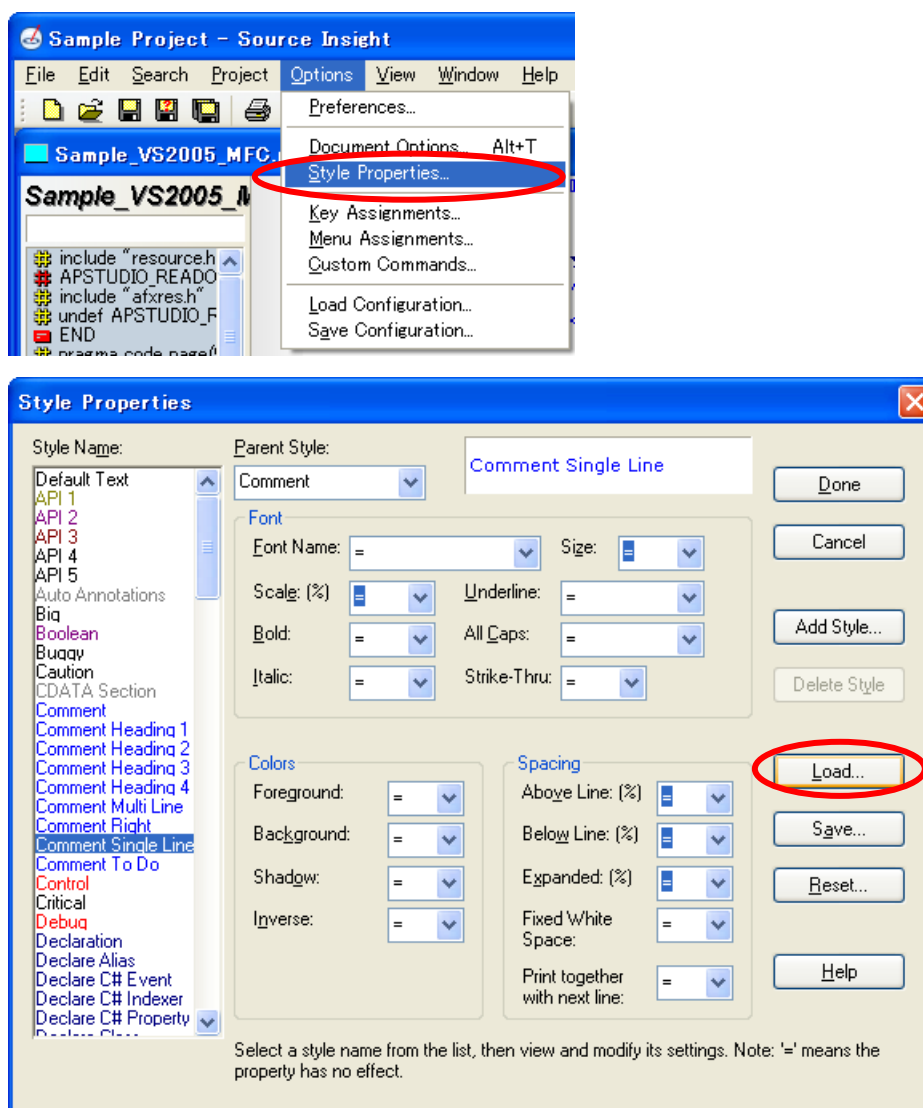
Source Insight のデフォルト設定では、欧文フォントがスタイルに適用されているため、日本語テキストが正しく表示されません。日本語スタイルを適用することで、リソースおよびソースコードの日本語が正しく表示されます。

ダウンロードしたファイルを解凍すると、インストールファイルといっしょに日本語スタイル用の設定ファイルが展開されます。

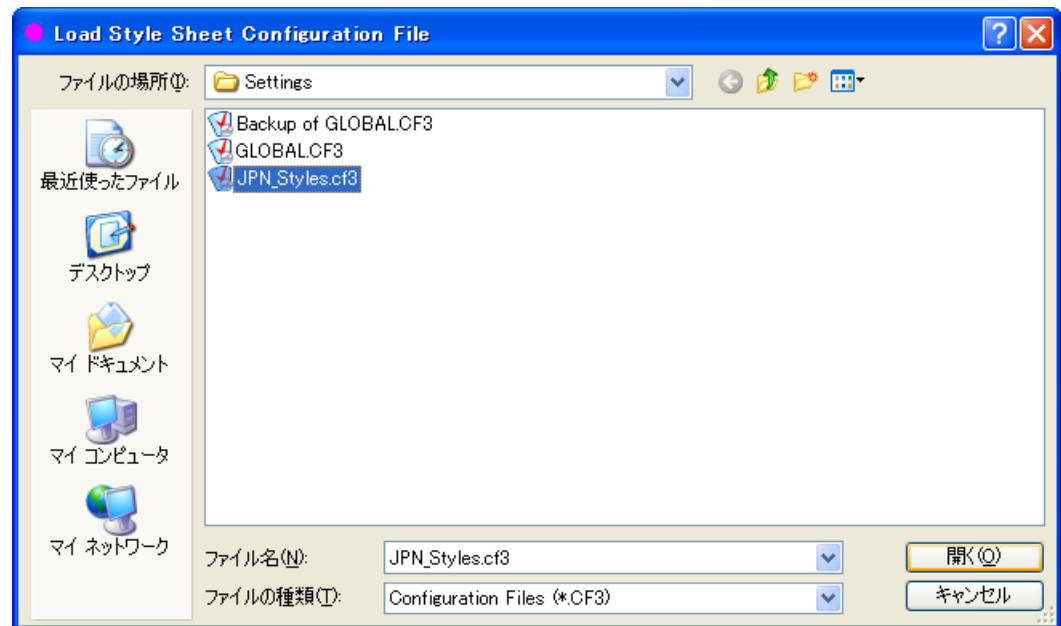
日本語フォントを設定した Configuration ファイル : **JPN_Style.CF3**

このファイルを [マイ ドキュメント] フォルダの Source Insight¥Setting フォルダにコピーします。
(C:¥Documents and Settings¥(ログインユーザー名)¥My Documents¥Source Insight¥Settings フォルダ)

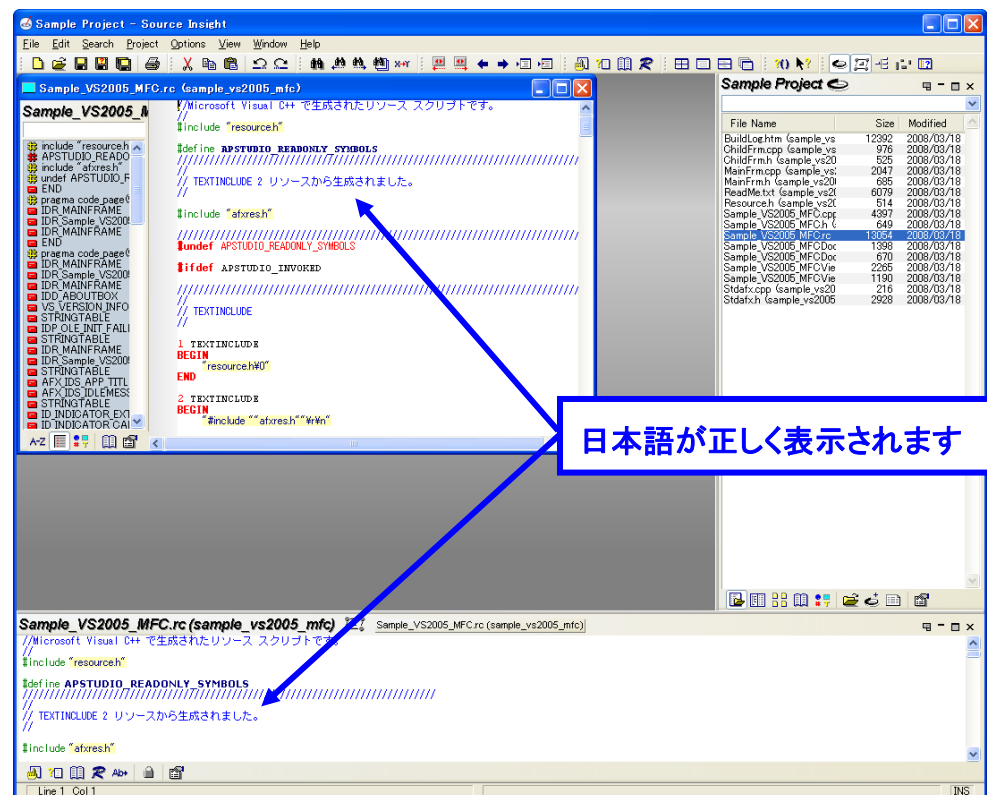
[Options] - [Style Properties] を選択し、[Load] ボタンでプロジェクトに読み込んでください。



[Load Style Sheet Configuration File] ダイアログで JPN_Style.CF3 を選択し、[開く] をクリックします。[Done(完了)] ボタンをクリックして、[Style Properties] ダイアログを閉じます。



[Load Style Sheet Configuration File] ダイアログで JPN_Style.CF3 を選択し、[開く] をクリックします。[Done(完了)] ボタンをクリックして、[Style Properties] ダイアログを閉じます。



日本語テキストを正しく表示させる方法は、第 4 章を参照してください。

Source Insight は、ソースコードを解析し、理解しやすいように構文を色分けして表示します。

3. Source Insight の機能

3.1. Source Insight の機能概要

Source Insight には、次のような機能が含まれています。

- コール グラフとコール ツリー ダイアグラム
- 状況に応じてダイナミックにソースを分類
- ファイルごとにシンボル ウィンドウを表示
- コンテキスト ウィンドウに宣言を自動的に表示
- 構文フォーマット - 構文の色分け など
- 状況に応じて名前を変更
- 参照を素早く検索
- 混在言語の編集
- コード ベースでインターネット検索のようなキーワード検索
- シンボルの自動補完
- すべてのシンボルとファイルに素早くアクセス
- チーム プログラミングのサポート
- リンクやコンパイラ エラーへのハイパー ソース リンク
- プロジェクト全体の高速な検索と置換
- 複数のビューで構成されたプロジェクト ウィンドウ
- 外部コンパイラおよびツールとの統合
- クリップボードと定型コード格納用のクリップ ウィンドウ
- 2 段階の行改訂マークと選択的な行の復元
- ドキュメントの種類および言語を拡張可能
- クラッシュ リカバリによりフルタイムの保護を実現
- カスタマイズ可能なメニューとキーボード
- リモート ターミナル サーバー セッションの特別なサポート
- フル機能のエディタ
- ドラッグ アンド ドロップ編集

詳細は、オンラインヘルプまたはユーザズガイドをご参照ください。

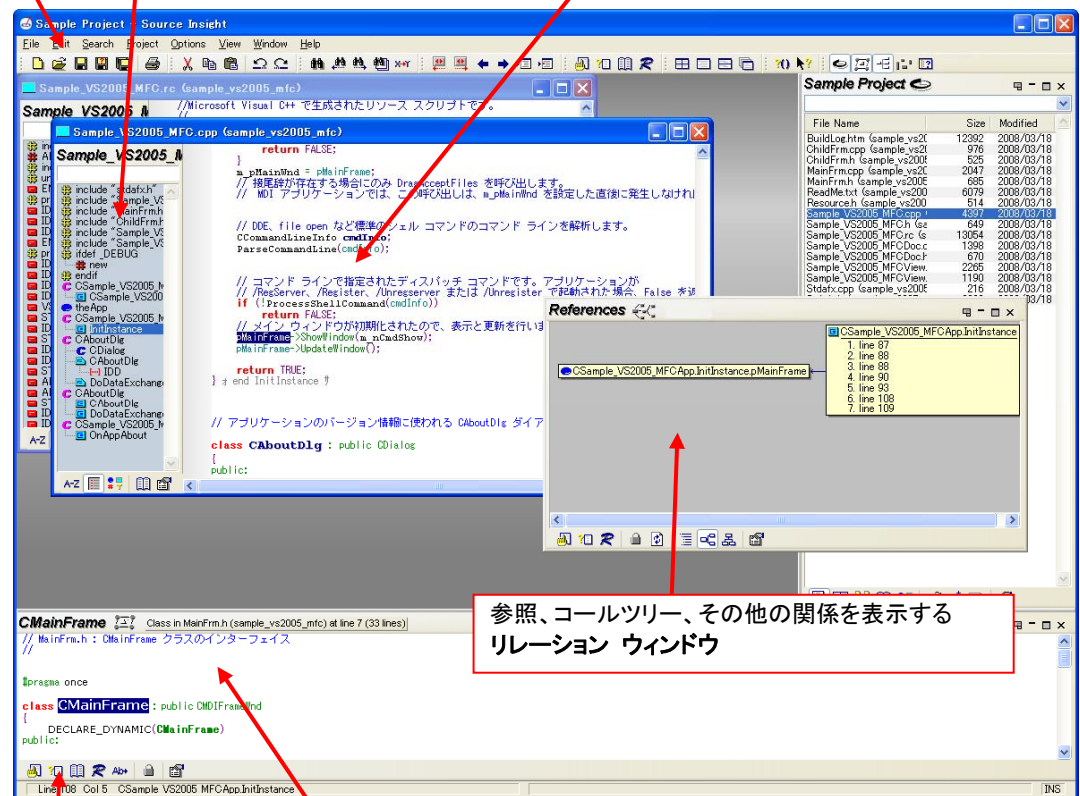
3.2. メイン ウィンドウ

Source Insight のメイン画面は、上部にメイン ツールバーが表示され、シンボル ウィンドウ、ソース ファイル ウィンドウ、リレーション ウィンドウ、コンテキスト ウィンドウが表示されます。

メインメニューおよびツールバー

各ファイルで定義されたシンボルを表示する シンボル ウィンドウ

編集および表示用のソースファイル ウィンドウ



参照、コールツリー、その他の関係を表示するリレーション ウィンドウ

ステータスバー

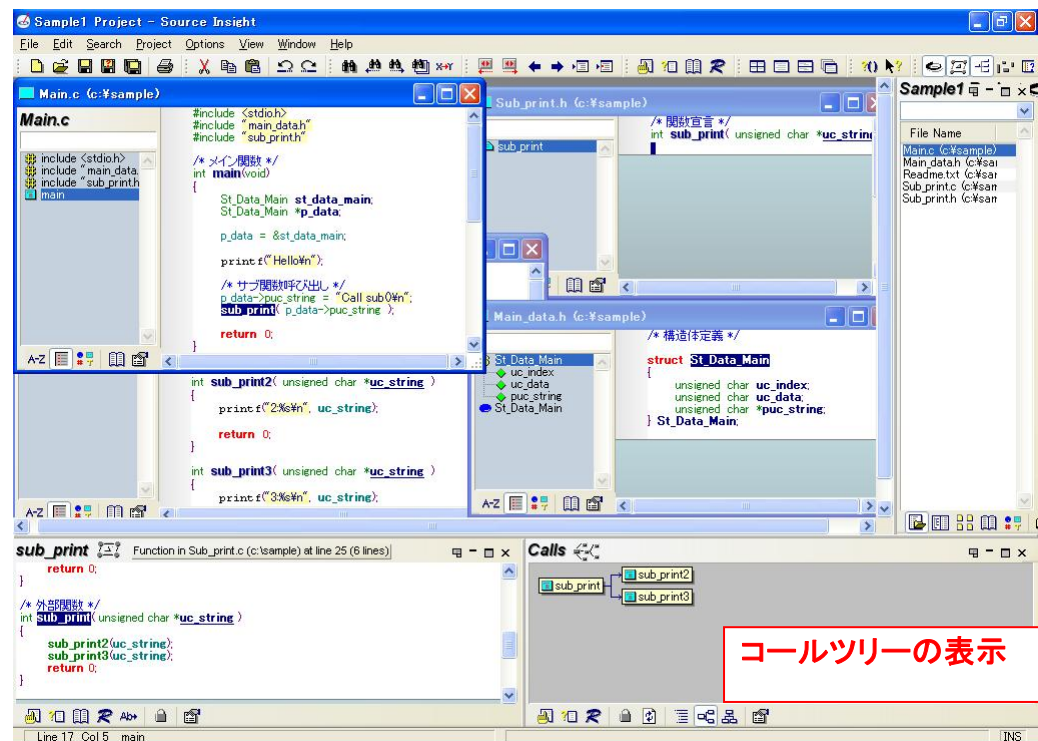
クリックまたは入力中に宣言やその他の情報を自動的に表示する コンテキスト ウィンドウ

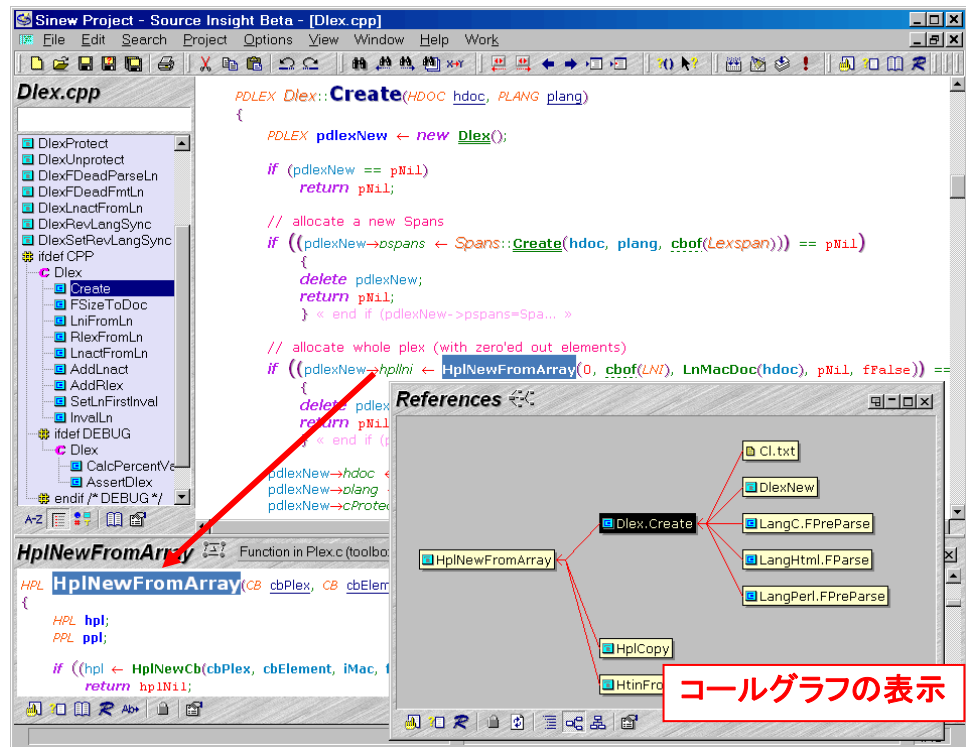
3.3. リレーション ウィンドウ

リレーション ウィンドウは、シンボル間の関係を示す Source Insight の革新的な機能です。バックグラウンドで動作し、ユーザーが選択したシンボルを追跡します。クラス階層、コール ツリー、参照 ツリー、その他を参照できます。

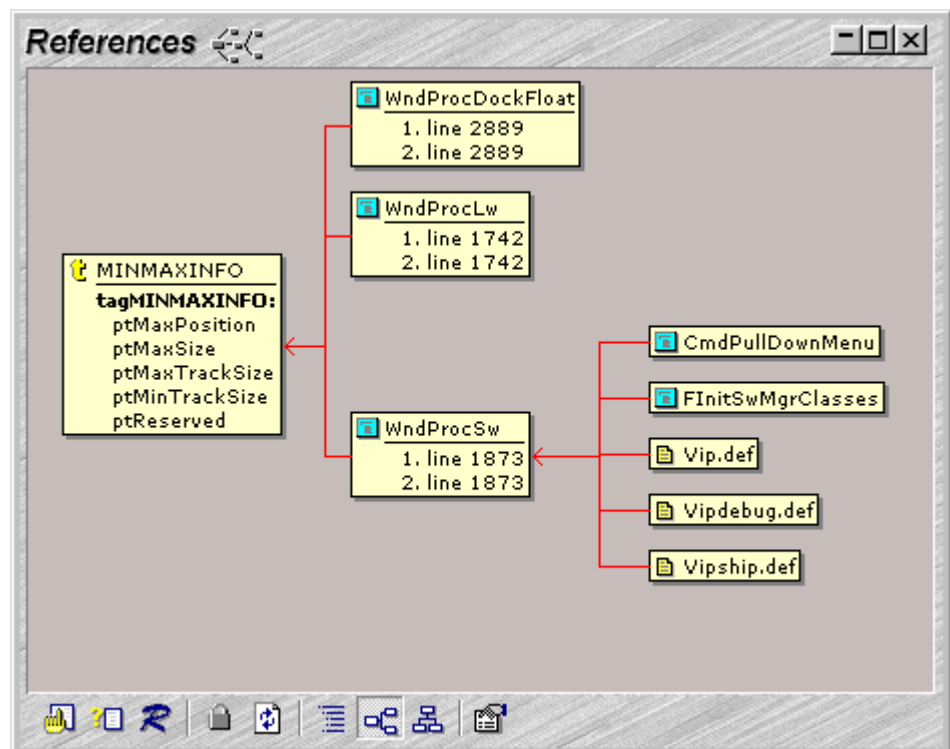
リレーション ウィンドウの利点は、何も特別なことを行う必要がないことです。作業中はバックグラウンドで動作していますが、必要なときに呼び出すことができます。

リレーション ウィンドウは、グラフィカルまたはアウトライン形式で表示することができます。複数のリレーション ウィンドウを開いて、異なる種類の情報を表示することもできます。

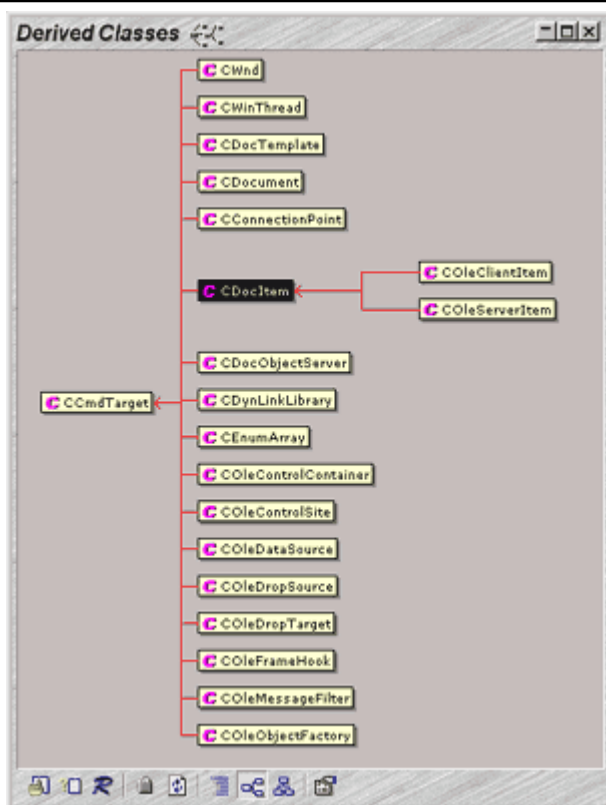




ダイナミックな関係表示で、ソース コードをわかりやすく表示



リレーション ウィンドウで間接参照を含む参照情報を表示

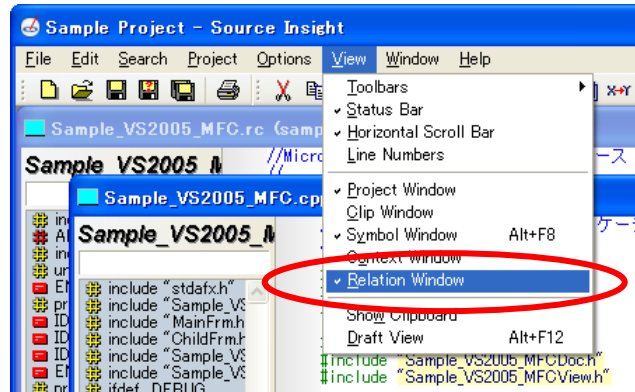


リレーション ウィンドウで派生クラスを表示

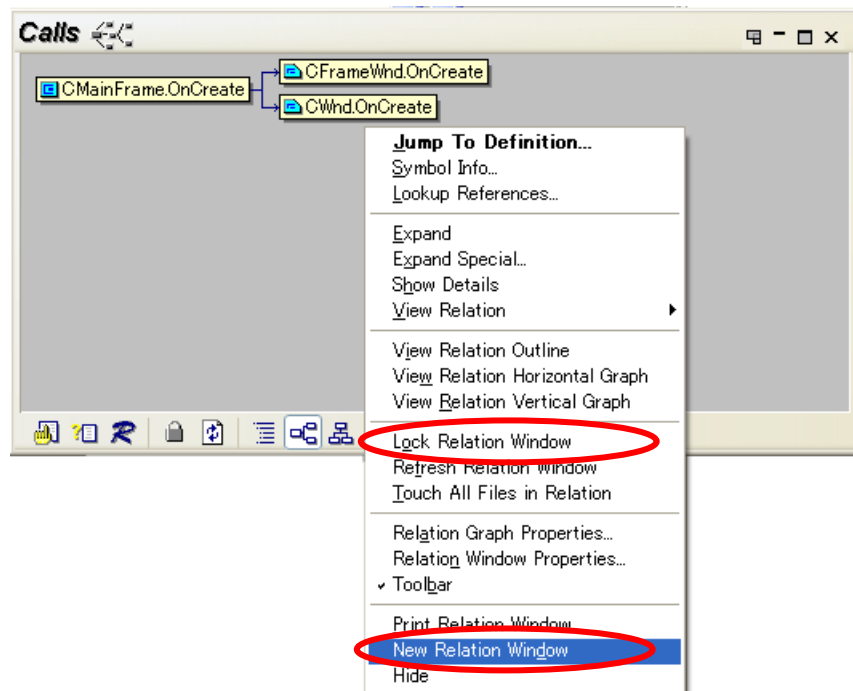
3.4. リレーション ウィンドウの操作

リレーション ウィンドウの表示内容は、ソースコード行のカーソル位置、選択範囲により動的に変化します。

リレーション ウィンドウを表示するには、[View] - [Relation Window] を選択し、メニュー項目の左にチェックマークが付いた状態にします。



Source Insight のメインウィンドウおよびリレーション ウィンドウなどにあるツールバー、または各ウィンドウを右クリックして表示される コンテキスト メニューを選択すると、新規のリレーション ウィンドウを表示し複数のリレーション ウィンドウの表示したり、リレーション画面のダイナミックな変更をロック(停止)することもできます。



4. カスタム コマンド

4.1. カスタム コマンド

カスタムコマンドは、Source Insight 内から起動する shell コマンドです。

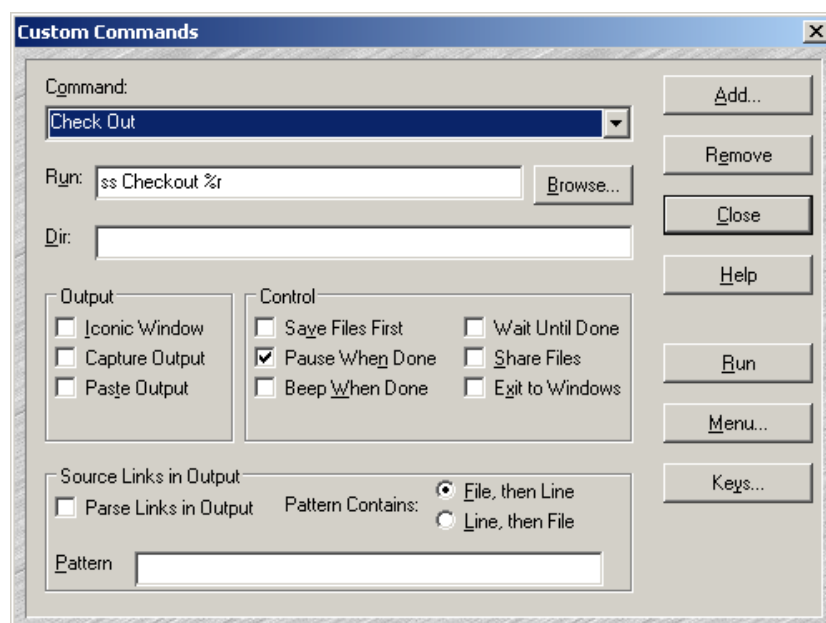
Source Insight に装備されている標準コマンドに加えて、カスタムコマンドを定義できます。カスタムコマンドは Shell バッチコマンドに似ていて、外部のコマンドライン プログラムと Windows GUI プログラムを実行します。Source Insight はそれをバックグラウンドで実行できます。カスタムコマンドの出力は、ファイルに保存できます、または現在の選択位置に貼り付けできます。

[Options]-[Custom Command] を選択して、コマンドを定義できます。一度定義したら、他のコマンドと同様に使用できます。メニューに割り当て、キーストロークの割り当てができます。

カスタムコマンドは、コンパイルを起動するときに役に立ちます。コンパイラや make プログラムを実行するカスタムコマンドを定義すると、コンパイラ エラーメッセージをキャプチャー、エラーの分析、間違ったソースコードへのリンクを自動的に取得できます。

カスタムコマンドを使って、テキスト フィルタを実装することもできます。たとえば、ソート(並べ替え) フィルタを実行し、現在の選択箇所を上書きで戻すカスタムコマンドを定義できます。

■ カスタム コマンド ダイアログボックス



Command : 現在選択しているコマンド名を表示します。

Run : カスタムコマンドが呼ばれたときに実行されるコマンドラインです。Run テキストボックスには、特別なメタ キャラクタを含めることができます。

Dir : Run テキストボックスで指定されたスクリプトが実行するときを使用される作業ディレクトリです。空白の場合、プロジェクトのソース ディレクトリが現在の作業ディレクトリに設定されます。

Output

コマンドのアウトプット時のオプションです。

Iconic Window : 実行されるプログラムは最小化されます。オフの場合は、通常通り起動します。

Capture Output : コマンドの標準アウトプットがキャプチャーされ、コマンドが完了したときに新しいコマンド アウトプット ウィンドウが表示されます。コマンド アウトプットのウィンドウタイトルは、カスタムコマンドの前後名になります。オフの場合は、アウトプットがキャプチャーされません。

Paste Output : コマンドの標準アウトプットが現在の選択位置に貼り付けられます。

Control

コマンド実行の前後に Source Insight が実行することを指定できます。

Save Files First : コマンドを実行する前にすべての[Save All(すべて保存)]コマンドを実行します。Save All(すべて保存) コマンドは、編集した各ファイルを保存するか確認します。

オフの場合、変更したファイルを保存せずにコマンドを実行します。保存されていない変更は、コマンドが完了しコントロールが Source Insight に戻ったときに維持されます。コマンドがクラッシュした場合、リカバリを実行でき、すべての変更は損なわれません。

Pause When Done : Source Insight は、コマンドが完了したときに、DOS ボックスでこのメッセージ

を表示します。

Press any key to return to Source Insight...(Source Insight に戻るにはどれかのキーを押してください…)

オフの場合、DOS ボックスはコマンドが完了したときに終了します。

Beep When Done : Source Insight は、コマンドが終了したときにビーブ音を鳴らします。

Wait Until Done : Source Insight は、コマンドが終了するまで中断します。オフの場合、コマンドを実行し、続行します。バックグラウンドでコマンドを実行している間、Source Insight にスイッチして戻ることができます。

Exit Source Insight : Source Insight は、プログラムを起動後に終了します。

Source Links in Output

これらのオプションは、コマンドが終了した後で、アウトプットをどのように扱うかを指定します。特定の警告やエラーメッセージを検索するためにキャプチャーしたアウトプットを分析すること指示できます。

Parse Source Links : ソースリンクパターンをコマンド アウトプットで検索します。パターンは通常、警告やエラーメッセージと一致します。パターン一致が検出されると Source Insight はその行にソースリンクを挿入します。ソースリンクは警告やエラーメッセージをターゲット ソース行へリンクするために使用されます。このオプションが有効な場合、[Pattern] テキストボックスに有効な検索表現を指定しなければなりません。

Pattern contains : [ファイル - 行の順]、[行 - ファイルの順]。パターン表現のグループの順序を示します。

[ファイル - 行の順]は、パターン表現の最初のグループがファイル名、次のグループが行番号の場合に選択します。この設定では、2 番目のグループ(行番号)はオプションです。

[行 - ファイルの順] は、パターン表現の最初のグループが行番号、2 番目のグループがファイル名の場合に選択します。

Pattern : アウトプット コマンドでファイル名と行番号を検索するために使用された正規表現を含んでいます。[Parse Source Links]オプションが無効の場合、これは無視されます。オプションが有効な場合、このテキストボックスにファイル名と行番号のグループを含む正規表現が指定されていなければなりません。

Add :名前ボックスで新規のコマンドを定義します。コマンドが既に存在する場合、再定義されません。

Remove : コマンドを削除します。

Run : コマンドを定義して、実行します。

Cancel : キャンセルします。このダイアログでの定義は保持されます。

Menu... : 現在のコマンドを定義し、メニューの割り当てダイアログにジャンプします。

Keys... : 現在のコマンドを定義し、キーの割り当てダイアログにジャンプします。

■ Run フィールドの形式

カスタムコマンドが呼び出されたときに実行されるコマンドラインが含まれます。複数のコマンドを含めることができます。

各コマンドは、セミコロンで区切ります。たとえば、

```
cat make.log;echotime
```

このストリングは、“cat make.log”、続いて “echotime” を実行します。

■ コマンド シェルの実行

“type” または “dir” のような shell コマンドを実行したい場合、または、バッチ ファイルを実行したい場合、最初に `cmd.exe` を実行する必要があります。たとえば

```
cmd /c mybat.bat または
```

```
cmd /c type foo.txt
```

注意: Windows 9x/Me の場合、`cmd.exe` の代わりに `command.com` を使用します。

Run ストリングに複数のコマンドが含まれる場合、セミコロンで区切ります。Source Insight が run ストリングからバッチファイルを作成し、自動的に `command.com` を実行するので、`cmd.exe` を実行する必要はありません。たとえば、

```
cat readme.txt;dir
```

この場合、shell 内でバッチファイルがすでに実行しているため正常に動作します。

`cmd.exe` で作成された shell は十分な環境スペースを持っていないことがあります。その場合、`/e` スイッチを `cmd.exe` で使用してください。たとえば、

```
cmd /e:1024 /c mybat.bat
```

これは、1K バイトの環境スペースをアロケートします。

■ コマンドライン代替

Run テキストボックスには、次の文字列を代替するメタ キャラクタを含めることができます。

表 カスタムコマンドのメタ キャラクタ

キャラクタ	説明	例
%f	現在のファイルのフルパス名 *	c:\¥myproj¥file.c
%r	現在のファイルのプロジェクト ソースディレクトリからの相対パス名 *	file.c
%n	現在のファイルのリーフ名(leaf name) *	file.c
%d	現在のファイルのディレクトリ パス	c:\¥myproj
%h	ドライブ文字なしの現在のファイルのディレクトリパス	¥myproj
%b	拡張子なしの現在のファイルのリーフ名	file
%e	現在のファイルの拡張子 *	c
%c	現在のファイルのドライブ文字	c:
%p	現在のプロジェクト名	c:\¥myproj¥myproj
%j	現在のプロジェクトのソースディレクトリ	c:\¥myproj
%J	現在のプロジェクトのデータディレクトリ	C:\¥Documents and Settings¥Jim Smith¥My Documents¥Source Insight¥Projects¥Base
%v	現在のプロジェクトのソースディレクトリのドライブ文字	c:
%o	パスなしのプロジェクトのリーフ名	myproj
%l	現在の行番号	任意の番号
%w	選択範囲の最初の単語、またはカーソル位置の単語	任意の単語
%s	カスタムコマンド実行中に現在の選択範囲が保存される一時ファイルの名前	d:\¥tmp¥vt0004.
%a	現在の日付	05-12-02 (mm-dd-yy)
%t	現在の時刻	08:23 (hh:mm)
%1 - %9	ユーザーに確認する引数	任意の文字列

上記の文字で * が付いているものには、次の修飾文字で接尾語をつけることができます。

キャラクタ	説明	例
%o	すべての開いているファイル	%f%o
%m	すべての修正したファイル	%f%m

ShellExecute コマンド

ShellExecute は Windows shell コマンドを呼び出します。

カスタムコマンドは、指定したファイルで操作を実行することを Windows shell に指示する “ShellExecute” 関数をサポートします。ShellExecute の良い点は、特定のファイルを処理するための登録されたアプリケーションを知る必要がないことです。詳細は、Windows Shell API ドキュメントの “ShellExecute” を参照してください。この機能を使用するには、カスタムコマンドの Run 文字列が “ShellExecute” で始まる必要があります。以下ようになります：

```
ShellExecute <verb> <filespec> <optional parameters>
```

たとえば、web サイトを参照するには：

```
ShellExecute open http://www.somedomain.com
```

verb は、以下のどれかの単語です：

- **edit** そのファイル用のエディタを開きます。
- **explore** 指定したフォルダを表示(エクスプローラ)します。
- **open** 指定したファイルを開きます。ファイルは実行ファイルまたはドキュメント ファイルです。フォルダの場合もあります。
- **print** 指定したドキュメントファイルを印刷します。ファイルがドキュメントでない場合、関数は失敗します。
- **properties** フォルダのプロパティを表示します。
- **find** スタートメニューで見つけたファイル検索アプリケーションを起動します。
- "" (空の文字列) は、ShellExecute へのパラメータをスキップします。

filespec パラメータは、有効なパスでなければなりません。スペースを含む場合、二重引用で囲ってください。%f (現在のファイル) などのメタ キャラクタを使用することもできます。実行ファイルの名前にすることもできます。

オプションのパラメータリストは、任意で filespec の右にあります。アプリケーションを実行するために渡すパラメータを指定します。形式は呼び出される verb によって決められます。カスタム コマンドのメタ キャラクタをここでも使用できます。

カスタムコマンドの作業ディレクトリ テキストボックスは、ShellExecute が呼び出される前に適用されます。しかし、ShellExecute を使用するとき、アウトプットはキャプチャーまたは分析されません。

ShellExecute の例

いくつかの役に立つ例を示します。

アクション	カスタムコマンド Run 文字列
web サイトの参照:	ShellExecute open http://www.someweb-site.com
Windows 2000 のドキュメントフォルダの表示	ShellExecute explore "C:\My Documents and Settings"
Windows 98 のドキュメントフォルダの表示	ShellExecute explore "C:\My Documents"
Internet Explorer の起動:	ShellExecute "" iexplore
Internet Explorer でファイルをプレビュー:	ShellExecute "" iexplore %f
現在のプロジェクトフォルダでファイルを検索:	ShellExecute find %j

バックグラウンドでカスタムコマンドを実行

Source Insight がカスタム コマンド shell プログラムを実行するとき、実際には Sihook3.exe と呼ばれるプログラムを実行します。Sihook3.exe は コマンドを生成し、アウトプットをキャプチャーします。カスタム コマンドを実行し、Source Insight ウィンドウをクリックすると、カスタムコマンドがバックグラウンドで実行している間に Source Insight での編集を継続できます。

コンパイルとビルド コマンドの作成

カスタムコマンドを使用して、Source Insight からコンパイラを起動でき、アウトプットをキャプチャーし、エラーメッセージを分析できます。次に、[Go To First Link] と [Go To Next Link] を使ってソースファイルの各エラーを表示することができます。

単純なコンパイル コマンドを作成するには

Microsoft C++ コンパイラを使った単純なコンパイル コマンドを作成するには:

1. Custom Command で [Add...] ボタンをクリックします。
2. [Name] ボックスに "Compile File" と入力します。
3. [Run] テキストボックスに "cl %f" と入力します。これは現在のファイルでコンパイラを呼び出します。ここで、代わりに "make" プログラムまたはバッチファイルを呼び出すこともできます。バッチファイルを使用する場合、command を最初に実行しなければなりません。たとえば、"cmd /c mybatch.bat" となります。
4. [Parse Links in Output] オプションをオンにします。デフォルトの分析パターンが、コンパイラアウトプットからのコンパイル エラーメッセージを分析するために設定されています。
5. [Save Files First] オプションをオンにして、コンパイル実行前にファイルを保存させます。

6. 新規のコマンドを保存するために **[Close]** をクリックします。別の方法として、**[Menu]** または **[Keys]** をクリックし、メニューまたはキーストロークをこの新規コマンドに割り当てます。

[Parse Links in Output] は、コンパイラのアウトプットを検索し、各エラーメッセージをソースにリンクさせるオプションです。この場合、“リンクソース” はコンパイラのアウトプットファイル内の各エラーメッセージです。“リンクターゲット” は、ファイルと行番号です。

Microsoft Developer Studio でプロジェクトをビルドするには

1. **[Option] - [Custom Command]** を選択します。
2. コマンドのドロップダウン リストから **[Build Project]** コマンドを選択します。これが既定義のカスタムコマンドです。
3. **[Run]** テキストボックスに、次のように入力します:
C:¥MsDevPath¥msdev project.dsp /make /rebuild
“MsDevPath” は、msdev.exe プログラムのパスです。“project.dsp” は、Developer Studio プロジェクトの名前です。
このラインは、与えられたプロジェクトをリビルドするために msdev.exe を呼び出します。
4. **[Parse Links in Output]** オプションをオンにします。デフォルトの分析パターンは、コンパイラのアウトプットからのコンパイラ エラーメッセージを分析するために設定されています。
5. **[Save Files First]** オプションをオンにして、コンパイル実行前にファイルを保存させます。

コンパイラ エラーを表示

エラーのあるソース行を表示するには:

1. 上記で定義した **Compile File** カスタムコマンドを実行します。
2. エラーがある場合、コンパイラが終了したときにエラーメッセージが "Compile File" ウィンドウに表示されます。Source Insight は、自動的にソースリンクを設定し、**[Go To First Link]** コマンドを実行します。最初のエラーメッセージとエラーのあるソース行が選択されて表示されます。
3. **[Go To Next Link]** コマンドを実行します。次のエラーメッセージが “Compile File” ウィンドウで選択されて、リンクのターゲットが表示されます。
4. すべてのリンクを確認するために **[Go To Next Link]** コマンドを続行します。リンクがなくなると Source Insight はビーブ音と “No links.” メッセージをステータスバーに表示します。

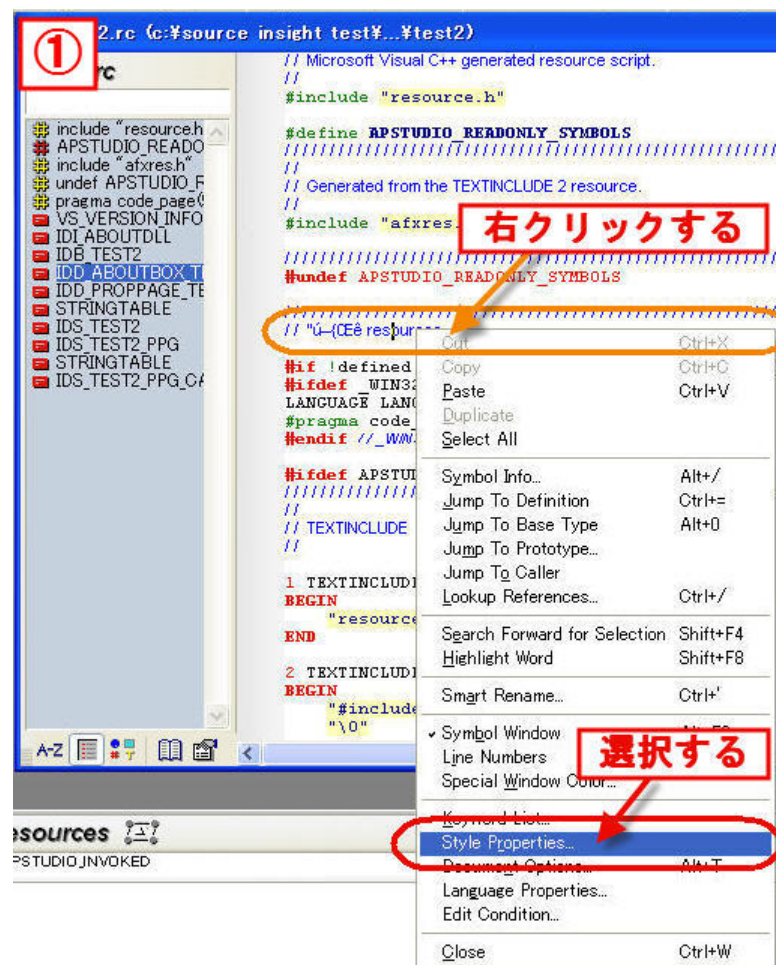
5. 日本語環境での注意

5.1. 日本語フォントのスタイル変更

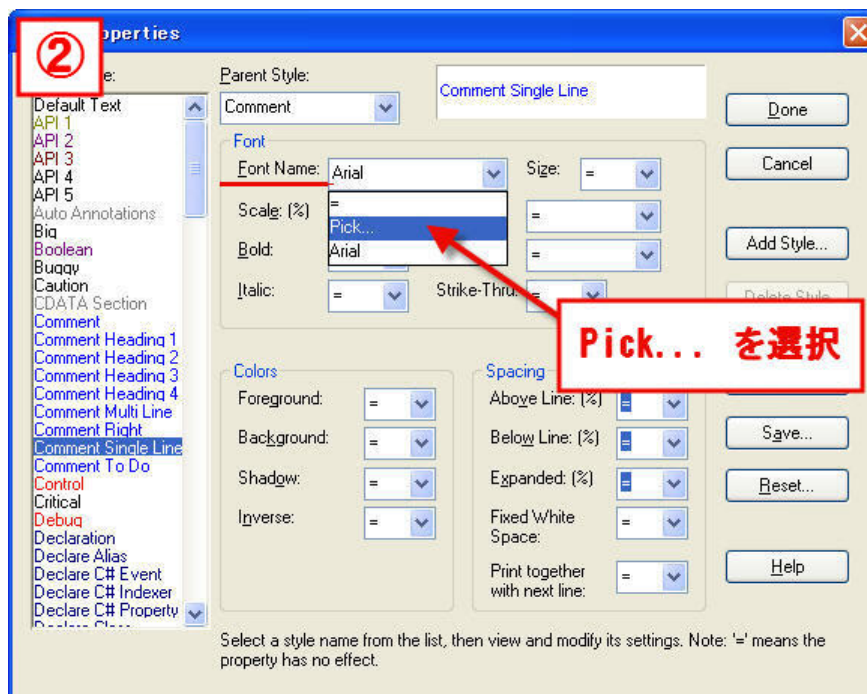
ソースファイルまたはリソースファイルに日本語が含まれていると、デフォルト設定では正しく表示されません。[Options] メニューから [Style Properties] を選択するか、日本語を含むソース行を右クリックして、ポップアップメニューから [Style Properties] を選択して、フォント名を日本語フォント (MS ゴシックなど) に変更してください。

別の方法として、日本語フォントを設定した Configuration ファイル(JPN_Style.CF3) をダウンロードし、[Load] ボタンでプロジェクトに読み込むことができます。この方法は次のセクションで説明します。

■ スタイルの変更

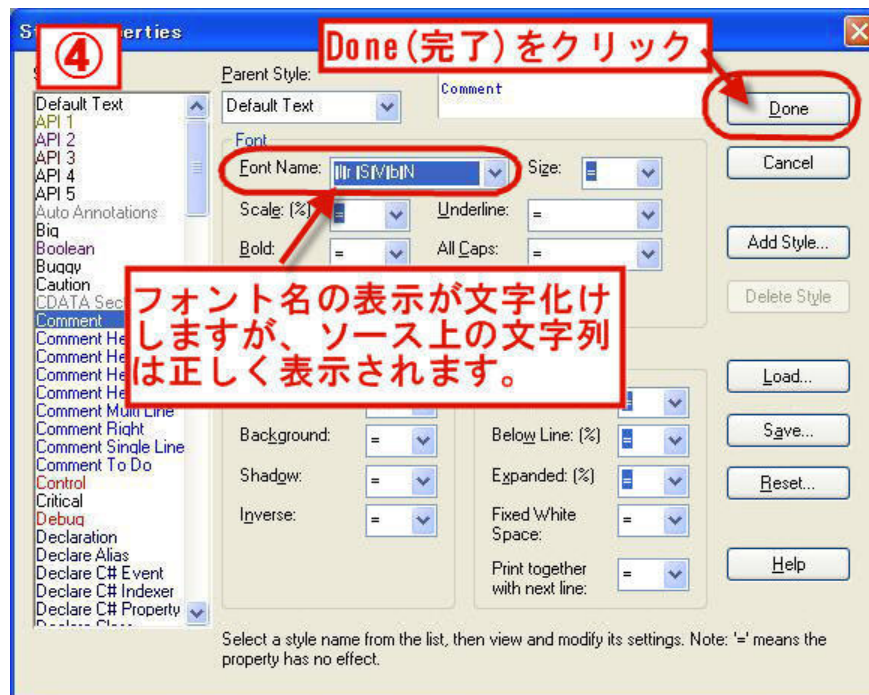


[Style properties]ダイアログの [Font Name] のドロップダウン リストから [Pick...] を選択します。
[フォント] ダイアログが表示されます。



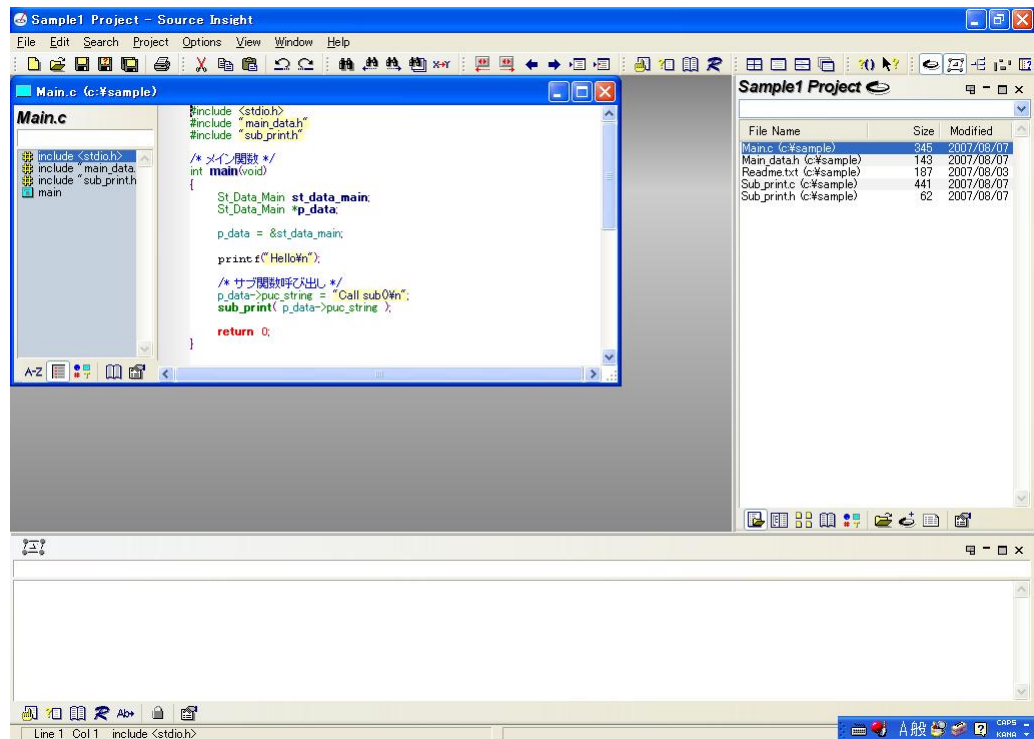
日本語フォントを選択します。"MS ゴシック" などの固定ピッチフォントを選択することを推奨します。





日本語を含むスタイルをすべて変更したら、[Done(完了)] ボタンをクリックして、[Style Properties] ダイアログを閉じます。

ソース画面で日本語が正しく表示されます。



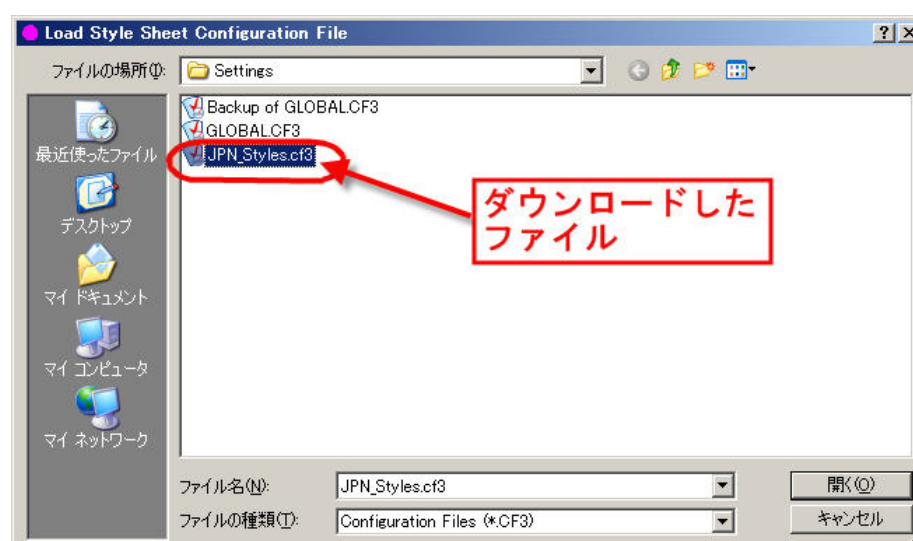
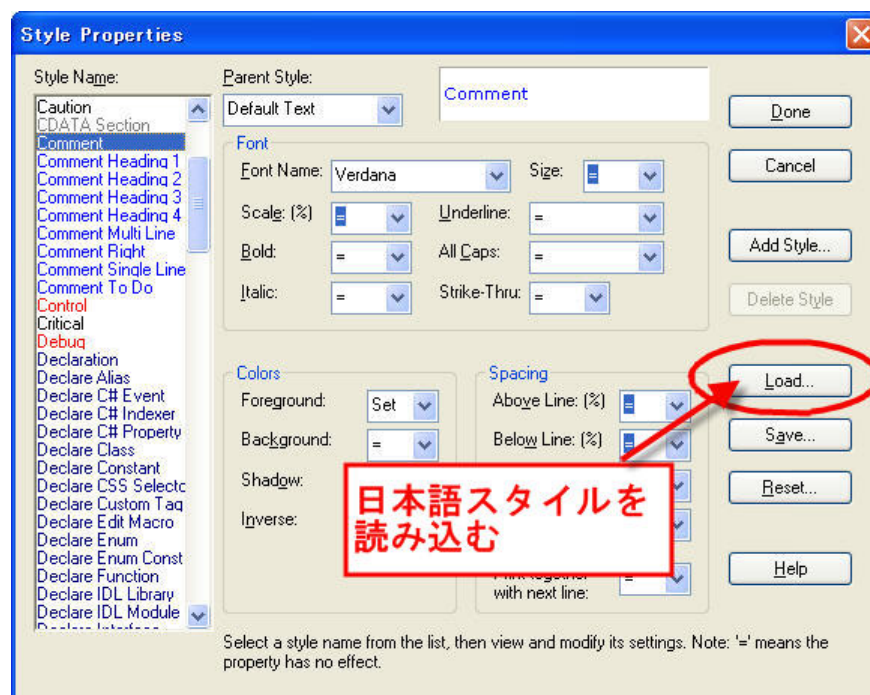
■ 設定済みの日本語スタイルの読み込み

ダウンロードしたファイルを解凍すると、インストールファイルといっしょに日本語スタイル用の設定ファイルが展開されます。

日本語フォントを設定した Configuration ファイル： **JPN_Style.CF3**

このファイルを [マイ ドキュメント] フォルダの Source Insight¥Setting フォルダにコピーします。
(C:¥Documents and Settings¥(ユーザー名)¥My Documents¥Source Insight¥Settings フォルダ)

[Options] - [Style Properties] を選択し、[Load] ボタンでプロジェクトに読み込んでください。



[Load Style Sheet Configuration File] ダイアログで JPN_Style.CF3 を選択し、[開く] をクリックします。[Done(完了)] ボタンをクリックして、[Style Properties] ダイアログを閉じます。

5.2. 日本語環境での制限

Source Insight は、英語版ソフトウェアをそのまま販売しています。現在、日本語環境での使用において一部制限が報告されています。

■ 日本語の入力

バージョン **3.50.0062** (2008 年 3 月 14 日リリース) では、日本語テキスト上での矢印キー、バックスペースキー、Delete キーなどの動作は、修正されましたが、一部の日本語が正しく入力できません。たとえば、"数" などを入力すると文字化けします。後ろに ¥ 記号を付けて"数¥" と入力すると正しく入力する場合があります。あるいは日本語文字列をコピー&ペーストで編集してください。

バージョン **3.50.0058** (2007 年 7 月 16 日リリース) では、一部の日本語が正しく入力できません。また、2 バイト文字上でカーソルを移動するには 2 回矢印キーを押す必要があります。Delete キーなども 2 回押さないと、2 バイト文字が正しく削除できません。日本語入力に対応されるバージョンがリリースされるまでは、直接入力を避けてください。

日本語文字列をコピー&ペーストで編集することは問題ありません。

■ Unicode のソース

現バージョンでは、Unicode に対応していませんので UTF-8 などで保存されたソースコードは、いったん ShiftJIS に変換してから Source Insight でご利用ください。

6. テクニカル サポート

6.1. テクニカル サポート

■ テクニカル サポートへのお問い合わせ

エクセルソフト株式会社ではテクニカル サポートを下記のお問い合わせ窓口より承っております。

http://www.xlsoft.com/jp/services/xl_form.html

■ 製品のお問い合わせ

エクセルソフト株式会社

〒108-0014 東京都港区芝 5 丁目 1 番 9 号 豊前屋ビル 4 階

Tel: 03-5440-7875

Fax: 03-5440-7876

<http://www.xlsoft.com>

E-Mail : xlsoftkk@xlsoft.com