Programmer's Edit Gear

Peggyシリーズ　Ver.4.6

ユーザーズマニュアル

株式会社エスコア

ご注意

●このソフトウエアの著作権は、株式会社エスコアにあります。

●このマニュアルに記載されている内容は、予告なく変更することがあります。

●このソフトウエアの機能は、予告なく変更することがあります。

●このソフトウエアおよびマニュアルの使用権は、本製品のライセンス契約に基づいた範囲でのみ認められます。

●このソフトウエアを使用して発生したお客様の損害に関して、弊社、および関連ツール各社にいかなる賠償責任をも請求することはできません。

●このソフトウエアおよびマニュアルの一部、または全部を株式会社エスコアの文書による許可無く、無断で複製、複写、転載することはできません。

本文中の商品名は、各社の商標または登録商標です。

株式会社エスコア
〒674-0082 兵庫県明石市魚住町中尾337
TEL 078-946-5551
FAX 078-946-5552
URL: http://www.anchorsystems.jp/
E-mail: anchor@anchorsystems.jp

次項は、本ソフトウエアの**エンドユーザー使用許諾契約書**です。
マニュアルから切り離して保管される場合は、切り取り線で切り離してください。

エンドユーザー使用許諾契約書

ライセンス内容

きりとりせん

〒674-0082 兵庫県明石市魚住町中尾337

株式会社エスコア

適用

以下の条項は、お客様が今回インストールされた製品に適用されます。

使用範囲

1. 本製品は、御登録いただいたお客様に対し使用許諾されます。

2. お客様は、いかなる事由に因らず製品を譲渡、販売することはできません。

3. お客様は、株式会社エスコアの許諾なくして、製品の2次的著作権を創作、譲渡、販売することはできません。

著作権

1. 本製品は、日本国内の著作権法ならびに国際条約の保護を受けています。

2. 本製品の著作権は、株式会社エスコアにあります。

3. 製品のリバースエンジニア、デコンパイル、ディスアセンブルその他の方法でソースコードを解明しようと試みることはできません。

4. インストールされたファイルの中で、<bin>ディレクトリ内のファイルに関する権利・配布条件は、同じ<bin>ディレクトリ内のファイル<COPYING>に記載されています。

保証範囲

1. 株式会社エスコアは製品の仕様を予告なく変更することがありますが、お客様が本契約に同意し、偽りなくユーザー登録を完了した後、180日以内に株式会社エスコアが製品の改訂版を販売する場合、改訂版に関する情報をお客様に提供します。

2. 株式会社エスコアは、上記第1項を唯一の保証責任とし、製品の品質、性能に関し発生する問題は、お客様の費用負担をもって処理します。

3. 株式会社エスコアは、お客様が製品を使用した結果生じたいかなる問題に対しても一切の責任を負うものではありません。また、製品内に表記されるいかなる関連各社も責任を負うものではありません。

契約期間

1. 本契約は、お客様が製品を受領した日より発効します。

2. お客様が本契約のいかなる条項にも違反した場合、本契約は解除されます。

目次

はじめに

Peggy シリーズの概要 2
ライセンス 3
ライセンスポリシー 4
マニュアルの構成と利用方法 5
マニュアルの表記規則 5

第 1 章 セットアップ

セットアップの前に確認しておくこと 8
セットアップ 8
アンインストール 9

第 2 章 Peggy の使い方

Peggy の特徴 12
基本操作 1 - ファイルを登録しよう 13
基本操作 2 - 開く・閉じる 14
基本操作 3 - 編集ウインドウ 15
基本操作 4 - 画面のカスタマイズ 17
基本操作 5 - 言語を設定しよう 18
基本操作 6 - 取り消しとやり直し 20
効果的な編集 1 - いろいろなジャンプ 21
効果的な編集 2 - 検索するには 23
効果的な編集 3 - 文字列の置換 25
効果的な編集 4 - 罫線 26
効果的な編集 5 - キーボードマクロ 27
効果的な編集 6 - 補完入力 28
効果的な編集 7 - 漢字コードと改行コード 29
新旧バージョンの違いを比べる 31
印刷する 33
テキストの一部分を印刷するには？ 35
ビルトイン「MocaScript」を実行してみよう 36
プロジェクトファイル 37
プロジェクトの応用 1 - テキスト以外も登録しよう 38
プロジェクトの応用 2 - 外部スクリプトを実行させよう 40
プロジェクトの応用 3 - ツールボタンを活用しよう 42
プロジェクトの応用 4 - 整理しよう 44
統合ツールとの組み合わせ 45
Peggy を中心とした作業環境を作ろう 46
好みに合わせてカスタマイズ 48

第 3 章 編集

ファイルメニュー 50
新規作成 51
テンプレートから新規作成 52
ファイルを開く 52
ファイルを開き直す 53
ファイルを閉じる 54

ファイルを保存する 54
Peggy の終了 56
編集ファイルの排他制御 56
ファイルヒストリ 56
最近使ったファイル、最近使ったフォルダ、最近使ったプロジェクト 58
URL をブラウズする 58
スクリプト実行 58
関連アプリケーション起動 59
ダイレクトオープン 59
関連ファイルを開く 59
インクルードファイルを開く 59
印刷 60
印刷 60
印刷プレビュー 61
印刷オプション 61
プリンタの設定 61
編集メニュー 61
カーソル移動 62
スクロール 63
インテリマウス 64
選択 65
選択・行選択・矩形選択 65
編集 66
文字の入力 66
切り取り・コピー・貼り付け 67
ドラッグ & ドロップ 68
取り消しとやり直し 69
再変換コマンド 70
挿入・削除 70
行挿入 70
制御コード挿入 70
日付と時刻の挿入 71
メール引用符の挿入と削除 71
改行コード挿入 72
改行コード削除 72
RGB 値挿入 72
削除 72
ジャンプ 73
指定行ジャンプ 73
対応括弧へジャンプ 74
タグジャンプ 74
ラベル行ジャンプ 75
次 / 前の対応する条件コンパイル文へジャンプ 75
変更行へジャンプ 76
ブックマーク 76
[ブックマーク一覧] ダイアログボックス 78
キーボードマクロ 79
アウトライン 80
罫線機能 81
罫線の種類 82
罫線の描画 82
罫線の消去 82
旧 NEC 罫線文字の変換 82
補完入力 83
単語補完 83
短縮入力 84
キーワード補完入力 85

コメント 85
インデント 86
HTML 編集 87
文字変換 89
[半角から全角へ変換]、[全角から半角へ変換]ダイアログボックス 92
選択範囲の漢字コード 93
ファイルの漢字コード 94
ファイルの改行コード 94
行順序 95
ウインドウメニュー 95
ヘルプメニュー 99
エディタプロパティ 101
コマンドライン引数 102
コマンドラインからの起動 102

第 4 章 検索・比較・置換

検索 106
共通の検索条件の切り替え 106
検索 107
検索ツール 109
再検索 111
クイック検索 111
ファイルから検索 112
プロジェクトから検索・グループ以下から検索・グループから検索 116
検索強調表示 117
シンボル検索 117
[定義位置ブラウザ]ダイアログボックス 120
シンボル検索とプロジェクト 122
TAGS のインポート 124
比較 124
クイック比較 125
ファイル比較 125
相違箇所のコピー 126
左から右へコピー・右から左へコピー 127
フォルダ比較 127
置換 131
条件保存 134
正規表現 134

第 5 章 言語モード

カラーコーディング 142
カラーコーディングとは 142
カラーコーディングを使う 142
言語モードと拡張子マップ 143
言語モードとは 143
拡張子マップ 144
新規言語モード作成・CREC キーワード定義ファイル 145
プロジェクトとの連携 146

第 6 章 プロジェクト機能

プロジェクト機能 148
プロジェクトとは 149
プロジェクトを作る 149

プロジェクトを保存する 150
プロジェクトを開く 151
プロジェクトを閉じる 151
プロジェクトにファイルを登録する 151
プロジェクトにグループを作る 152
複数のファイルやフォルダを一度に登録する 152
ファイルを開く 153
プロジェクトを移動する 153
アイテムの順番を入れ替える 154
アイテムを浅くソートする 154
アイテムを深くソートする 154
アイテムのコピー・切り取り・貼り付け 155
登録したアイテムを削除する 155
ごみ箱から削除する 156
ごみ箱を空にする 156
実体のないファイルアイテムを削除する 157
フルパスで表示する 157
プロジェクト名を変更する 157
アイテムの名前を変更する 157
位置の確認 158
実行中止 158
キーボードマクロを登録する 158
キーボードマクロのコピー、移動、名前の変更 158
キーボードマクロを再生する 158
キーボードマクロを呼び出す 159
検索、比較条件を登録する 159
検索、比較条件のコピー、移動、名前の変更 160
検索、比較を実行する 160
検索、比較条件を設定する・リストアする 160
マウスアクションを設定する 161
ツールアクションを設定する 161
スクリプトファイルの実行 162
タグジャンプ 163
プロジェクトの言語モード 163
統合環境作り 165
プロパティ 166
[プロジェクト]プロパティページ 167
[グループ]プロパティページ 167
[PHP デバッガの設定]ダイアログボックス 169
[ファイル]プロパティページ 170
[実行方法]プロパティページ 172
アドホックパラメタ 177
[アドホックパラメタ設定]ダイアログボックス 178
[アドホックパラメタ入力]ダイアログボックス 179
[キーボードマクロ]プロパティページ 180
[文字列検索]プロパティページ 181
[ファイルから検索]プロパティページ 182
[検索方法]プロパティページ 183
[ファイル比較]プロパティページ 184
[比較方法]プロパティページ 185
[ツールリスト]ダイアログボックス 186

第 7 章 ビルトインスクリプト

Hello, world 188
スクリプトファイルの作成 188
スクリプトの実行 188
操作編 189

コマンド宣言 189
コマンドの識別 190
コマンドの分類 191
スクリプトファイルを開く 192
コーディング 192
コンパイル 192
デバッグ 193
スクリプトの実行 193
ボタンの割り当て 194
ショートカットの割り当て 194

第 8 章 バージョン管理

バージョン管理 198
RCS、CVS、Subversion 198
RCS によるバージョン管理..... 199
RCS へチェックイン 200
編集して再び RCS へチェックイン 200
RCS ログを見る 201
RCS 古いリビジョンを見る 201
RCS 変更箇所を確認する 202
RCS コマンド 202
[RCS チェックイン] ダイアログボックス 203
CVS によるバージョン管理..... 204
CVS コマンド 204
RCS/CVS/Subversion 共通コマンド 205
[CVS ステータス] ダイアログボックス 206
[CVS リビジョンログ]・[RCS リビジョンログ] ダイアログボックス 210
[CVS ログフィルタ] ダイアログボックス 212
[CVS アップデート] ダイアログボックス 214
[CVS コミット] ダイアログボックス 215
[CVS ファイル追加] ダイアログボックス 217
[CVS ファイル削除] ダイアログボックス 217
[CVS タグ設定] ダイアログボックス 218
[CVS フォルダ選択]・[Subversion フォルダ選択] ダイアログボックス 218
Subversion によるバージョン管理..... 219
Subversion コマンド 219
[Subversion ステータス] ダイアログボックス 220
[Subversion リビジョンログ] ダイアログボックス 224
[Subversion ログフィルタ] ダイアログボックス 226
[Subversion リポジトリブラウザ] ダイアログボックス 228
[Subversion チェックアウト] ダイアログボックス 229
[Subversion アップデート] ダイアログボックス 230
[Subversion コミット] ダイアログボックス 231
[Subversion 追加] ダイアログボックス 232
[Subversion 削除] ダイアログボックス 232
[Subversion コピー] ダイアログボックス 233
[Subversion 戻す] ダイアログボックス 234
[Subversion 衝突解除] ダイアログボックス 234
[Subversion ロック] ダイアログボックス 235
[Subversion アンロック] ダイアログボックス 235
VSS (Visual SourceSafe) によるバージョン管理 237
VSS の設定 239
VSS コマンド 242
[VSS チェックイン] ダイアログボックス 243
[VSS チェックアウト] ダイアログボックス 244

[VSS チェックアウトの取り消し] ダイアログボックス ... 245
[VSS 最新のバージョンを取得] ダイアログボックス 246
[VSS ソース管理へ追加] ダイアログボックス 247
[VSS ソースコード管理から削除] ダイアログボックス ... 248
[VSS ファイルの履歴表示オプション] ダイアログボックス 249
[VSS 相違点表示オプション] ダイアログボックス 249
[VSS プロパティ] ダイアログボックス 251

第 9 章 DB ツール

DB ツールとは 254
Oracle クライアントのインストール 254
ODBC ドライバとデータソース 254
DB ツールの設定 254
DB ツールボタンをツールバーに追加する。 257
拡張子・言語に付いて 258
DB ツールの実行 258
範囲指定での実行 258
ファイル全体の実行 258
実行結果 259
タグジャンプ 259
文法の注意点 260
PL/SQL に関して 260
応用例 261

第 10 章 PHP プログラムのデバッグ

PHP プログラムのデバッグ 264
デバッグの概要 264
デバッグの準備 265
Web サーバーの設定 266
Peggy Pro for PHP の設定 269
デバッグ開始 271
デバッグの開始方法 271
Web ブラウザのキャッシュ 271
サンプルプログラム 272
コマンドライン版の PHP プログラムのデバッグ 274
デバッグコマンド 276
実行と中止 277
ブレークポイント 277
[ブレークポイントリスト] ダイアログボックス 278
[変数] ウインドウ 279
[文字列の編集] ダイアログボックス 281
[スタック] ウインドウ 281
[デバッグレイアウト] 281

第 11 章 カスタマイズ

ツールバー 284
表示と消去 284
フローティングにする 284
ドッキングする 285
簡単カスタマイズ 285
ツールボタンの追加と削除 285
新規ツールバーの作成と削除 286
デフォルトに戻す 287
スクリプトボタンのカスタマイズ 288
[ボタンギャラリー] ダイアログボックス 289

ユーザ定義メニューのカスタマイズ 290
[カスタマイズ]ダイアログボックス[ビットマップ]ページ 292
ショートカットキー 293
[ショートカット]ダイアログボックス 293
新しいショートカットキーを割り当てる 294
ショートカットキーを削除 295
ショートカットキー定義の保存、読み込み 295
[オプション設定]ダイアログボックス 296
[全般]ページ 297
[プロジェクト]ページ 299
[移動]ページ 302
[編集]ページ 303
[補完]ページ 321
[レイアウト]ページ 324
[言語]ページ 326
[拡張子]ページ 329
[アウトライン]ページ 330
[印刷]ページ 332
[表示色]ページ 337
[バージョン管理]ページ 338
[デバッグ]ページ 343

APPENDIX

付録A コマンド一覧 348
ファイル 348
編集 348
選択 349
挿入 350
削除 350
文字変換 351
補完入力 352
ジャンプ 352
カーソル移動 352
スクロール 353
罫線 354
キーボードマクロ 354
HTML 354
ブックマーク 355
検索 355
プロジェクト 356
デバッグ 356
ユーザ定義ツール 357
スクリプト 358
アウトライン 358
バージョン管理 358
表示 359
ウインドウ 360
ヘルプ 361
その他 361

索引

..... 363

はじめに

Peggy シリーズの概要

Peggyシリーズは、すべてのPCユーザのためのテキストエディタです。Peggyはそのフルネーム Programmer's Edit Gear が表しているとおり、もともとソフトウェア開発者向けのテキストエディタとして開発されましたが、日本語文章やメールの編集、ホームページを作成するためのエディタとしても快適にご利用いただけます。

Peggy シリーズの構成

(1)	Peggy Pad ノートパッド代わりに使えるフリーの簡易テキストエディタです。機能的には制限がありますが、文法認識カラー表示のためのライブラリは、そのまま利用できます。
(2)	Peggy 一般的なテキスト編集や小規模のプログラミングをするのに適しています。豊富な編集機能とプロジェクト機能により、複数のファイルを効率よく扱えます。JavaScript準拠のスクリプト言語「MocaScript」により、欲しい機能を簡単に作成可能です。RCSによる簡単なバージョン管理も利用できます。
(3)	Peggy Pro 加えてSubversion、CVS、Microsoft® Visual SourceSafeによる高度なバージョン管理をサポートします。開発チームメンバ間で大量のファイルを共有し、変更や履歴を管理できるようになります。個人で本格的なプログラミングを行う場合にも適しています。
(4)	Peggy Pro for PHP さらにPHPプログラムのデバッグ機能とデータベースへの接続機能が追加されたものです。Oracleを始めMicrosoft SQL ServerやAccessなどのデータベース開発に便利です。

スクリプト言語で自由に拡張

JavaScript準拠のスクリプト言語「MocaScript」を搭載しました。シンプルでわかり易い文法と豊富な組み込み関数により、欲しい機能が簡単に作成できます。

PHP プログラムのデバッグ (Peggy Pro for PHP のみ)

Webサーバーで実行されているPHPプログラムのリモートデバッグ機能を搭載しました。ステップ実行、ブレークポイント、スタック表示、変数値の参照と値の変更、変数値のチップ表示など、いままで難しかったPHPのデバッグが快適に行えます。

Oracle と ODBC サポート (Peggy Pro for PHP のみ)

編集中のSQL、PL/SQL文をOracleデータベースに接続して実行させ、結果を表示したり、ファイルへ保存することがボタン1つで可能です。エラー行へのタグジャンプも行えるので、データベースアプリケーションの開発効率を大幅にアップできます。ODBC経由で各種データベースへ接続することも可能です。

カラーコーディング

Peggyは、プログラミング言語の予約語、文字列、コメント、数値などの文法を正確に認識し色分け表示します。一般的なプログラミング言語に加え、HTML、スクリプト言語、アセンブリ言語、ハードウェア記述言語までも個別にサポートしていま

す。PHP/ASP/JSPなど、1つのファイルに複数の言語が含まれているファイルの色づけ表示もサポートしました。また、CREC(コンテキスト正規表現カラー表示)を搭載し、新しい言語の追加も簡単です。対応言語は随時増えていきますが、すべてアンカーシステムズのホームページより自由にダウンロードしていただけます。

バージョン管理

世界標準のバージョン管理システムであるSubversion/CVS/RCS、さらにVSSへのユーザインターフェイスを搭載しました(PeggyはRCSのみ)。趣味のプログラミングからチームによる大規模な開発プロジェクトまで、大切なソースファイルの履歴管理はもちろん、開発チームメンバ間でのソースファイル共有と共同作業を強力に支援します。

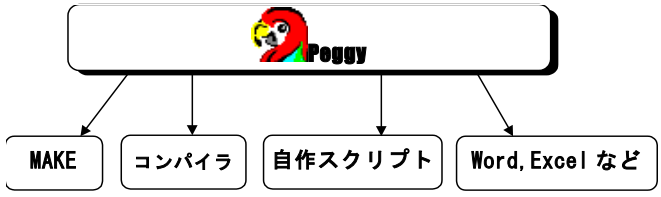
強力な検索と比較

エディタとしての基本機能はもちろんですが、それに加えて強力な検索・ファイル比較 ・フォルダ比較関連コマンドを数多く用意しました。複数ファイル対応の文字列置換機能もサポートしました。高度な機能が簡単な操作で使えます。特に比較機能は、バージョン管理システムと連携することで威力を発揮します。

プロジェクト機能

Peggyは単なるエディタとして使えることはもちろんですが、コンパイラ、デバッガなどの各種ツールと協調動作をさせることができます。またWord、Excel、Acrobatなどで書かれた仕様書やスケジュールなどの関連文書も同時にプロジェクトとして登録、呼び出しができます。

下図はPeggyを中心に開発ツールを統合したイメージです。プロジェクトにソースファイルやMakefile、仕様書などを登録し、Makefileやバッチファイル、コンパイラ、デバッガなどをボタンやマウスクリックなどのアクションと連動させることで統合環境ができあがります。各種ツールからの出力はすべてPeggyのアウトプットウインドウ内に表示されますから、タグジャンプなども自在にできます。



ライセンス

お試し版から、製品版への移行には、ライセンスキーとシリアル番号の入力が必要です。ライセンスキー、シリアル番号の入力は、メニューの[ヘルプ]-[バージョン情報]のダイアログボックスで行ってください。

ライセンスポリシー

1　本製品は、御登録いただいたお客様に対し使用許諾されます。別の人が使用することはできません。同じ人が利用する限りにおいては、何台のコンピュータにインストールしても、幾つ同時に起動しても構いません。なお、お試し版の場合は製品版ソフトウエア購入検討を前提とした場合に限り、使用許諾します。

2　お客様は、いかなる事由に因らず製品を譲渡、販売することはできません。

3　お客様は、株式会社エスコアの許諾なくして、製品の２次的著作権を創作、譲渡、販売することはできません。

マニュアルの構成と利用方法

この『Peggyユーザーズマニュアル』は、全般的な解説と、各機能ごとのリファレンスで構成されています。大まかな使い方は『第2章Peggyの使い方』を、すべての機能については第3章以降の機能別リファレンスから該当項目を読んでいただけるように構成されています。

【注意】このマニュアルは、Peggyシリーズすべてに共通です。製品によっては動作しない機能があります。また、特に断りのないかぎり、Peggyシリーズをまとめて Peggyと表記しています。

マニュアルの表記規則

このマニュアルは、次の表記規則に従って記述されています。

[ファイル]-[開く]	メニュー、ダイアログボックスなど、画面に表示されるユーザインターフェースの文字列は角カッコ([])で囲みます。また、複数段のメニューなどは、ハイフン(-)でつないで表現します。
Enterキー	キーボード上のキーの名前は先頭の文字を大文字で表します。キーの名前は、日本語106キーボードに準拠しています。
Ctrl+F1キー	キー名の間のプラス記号(+)は、キーの組み合わせを示します。Ctrl+F1キーの場合、Ctrlキーを押しながらF1キーを押すことを意味します。 マニュアルに記述したキー操作はデフォルトのキーの組み合わせです。ユーザがカスタマイズした場合、キーの組み合わせは記述のものと一致しません。
<ファイル名>	本文中、ファイル名は山カッコ(< >)で囲みます。
『本/章の名前』	本文中、他の本や章への参照は、二重鍵カッコ(『 』)で囲みます。
【注意】 **【ヒント】** **【参照】** **【プログラム】** **【メモ】**	本文の補足に使われている記号です。より深い理解のために記述されています。

第１章　セットアップ

Peggyをコンピュータにインストールするための手順を説明します。Peggyを使用するには、PeggyのCD-ROMからコンピュータに必要なファイルをコピーしたり、種々の設定をする必要がありますが、セットアッププログラムの指示に従って操作することにより、スムーズに作業を進めることができます。

セットアップの前に確認しておくこと

Peggyをインストールするコンピュータの必要条件は次の通りです。

●CPU	Pentium以上を推奨
●オペレーティングシステム	Windows 98/Me/NT4.0/2000/XP/Vista
●メモリ	32MB以上を推奨(OS、編集テキストによる)
●ハードディスク	25MB以上の空き容量
●ディスプレイ	1024x768、256色以上を推奨
●その他	マウス

以上の要件を満たしていることを確認してください。
それ以外に、必要に応じて次の点を確認してください。

●バージョン管理システムSubversion、CVS、RCSを使う場合は、それらのシステムがインストールされていることが必要です。Subversion以外は、Peggyのセットアップでもインストールすることができます。

●バージョン管理システムVisual SourceSafeを使う場合は、Visual SourceSafeが正しくインストールされていることが必要です。

●Oracleデータベースに接続する機能を使う場合は、OracleのSQL*NET V2以上がインストール済みでOracleデータベースサーバーと接続可能な状態であることが必要です。

●ODBC接続機能を使う場合は、目的のデータベースのODBCドライバがインストールされていることが必要です。

セットアップ

セットアッププログラムの起動

1. Windows のすべてのプログラムを終了させてください。
2. CD-ROM ドライブに Peggy の CD-ROM を挿入します。
3. [スタート] メニューの [ファイル名を指定して実行] で、CD-ROM のルートフォルダにある <setup.exe> を起動します。
4. セットアップ画面になりますので、[次へ] ボタンをクリックし、セットアップを進めてください。

ライセンス契約を確認する

Peggyの使用許諾契約書が表示されるので、最後まで読み同意されるなら、[はい]ボタンをクリックし、セットアップを進めます。

インストール先の選択

Peggyをインストールするフォルダを決定します。表示されたフォルダをそのまま使うことも、[参照]ボタンをクリックし、別のフォルダを指定することもできます。通常は、表示されたフォルダをそのまま使うことをお勧めします。

ユーザーの情報

お名前と会社名を入力してください。シリアル番号は、弊社からお知らせしている番号を入力します。シリアル番号を間違って入力すると、Peggyは動作しませんので、よく確認してください。

すべての項目を入力しないと[次へ]ボタンは有効になりません。個人で購入された場合は、会社名の欄にも名前を入れるなどしてください。

ライセンスキーの入力

弊社からお知らせしているライセンスキーを入力してください。ライセンスキーは半角文字の英数字16桁と半角文字のハイフン‘-’で構成されています。ライセンスキーを間違って入力すると、Peggyは動作しません。

よく確認してから、[次へ]ボタンをクリックします。

言語サポート DLL の選択 / キーワード定義ファイルの選択

Peggyのカラーコーディングや言語モードに対応した操作を行うには、言語ごとの言語サポートDLL/キーワード定義ファイルをインストールする必要があります。

使わないコンパイラ・アセンブラなどのチェックをはずすとハードディスクの消費量を減らすことができます。なお、チェックボックスの右の文字列をクリックすると、下のボックスに詳しい対応表が表示されます。

バージョン管理システムの選択

バージョン管理システムCVSとRCSをインストールするかどうかを決定します。

確認

今までに入力した内容が表示されるので、確認してください。[次へ]ボタンをクリックすると、ファイルのコピーが始まります。

セットアップの終了

以上でセットアッププログラムの終了です。Peggyを使う準備がすべて整いました。CD-ROMドライブからCD-ROMを取り出し、[終了]ボタンをクリックしてください。

アンインストール

インストールしたPeggyをコンピュータのハードディスクから削除することを、「アンインストール」と言います。アンインストールをするときは必ず次の手順で行ってください。

Windows 98/Me/NT4.0 の場合

1. Windows のすべてのプログラムを終了させてください。
2. [スタート] メニューの [設定]-[コントロールパネル]-[アプリケーションの追加と削除] を選択します。
3. [インストールと削除] または [セットアップと削除] ページから「Peggy」を探し、それを選択して [追加と削除] ボタンをクリックします。
4. [ファイル削除の確認] ダイアログボックスが表示されたら、[はい] ボタンをクリックします。

5. アンインストールが始まります。作業が終わるまで待ちます。以上でアンインストールは終了です。

Windows 2000 の場合

1. Windows のすべてのプログラムを終了させてください。
2. [スタート] メニューの [設定]-[コントロールパネル]-[アプリケーションの追加と削除] を選択します。
3. 左端の [プログラムの変更と削除] ボタンを押し、右の [現在インストールされているプログラム] の中から「Peggy」を探し、それを選択して [変更 / 削除] ボタンをクリックします。
4. [ファイル削除の確認] ダイアログボックスが表示されたら、[はい] ボタンをクリックします。
5. アンインストールが始まります。作業が終わるまで待ちます。以上でアンインストールは終了です。

Windows XP の場合

1. Windows のすべてのプログラムを終了させてください。
2. [スタート] メニューの [設定]-[コントロールパネル]-[プログラムの追加と削除] を選択します。
3. [インストールと削除] または [セットアップと削除] ページから「Peggy」を探し、それを選択して [変更と削除] ボタンをクリックします。
4. [ファイル削除の確認] ダイアログボックスが表示されたら、[はい] ボタンをクリックします。
5. アンインストールが始まります。作業が終わるまで待ちます。以上でアンインストールは終了です。

Windows Vista の場合

1. Windows のすべてのプログラムを終了させてください。
2. [スタート] メニューから [コントロールパネル] を選択します。
3. [プログラムと機能] を開き [Peggy] を選択してから、[アンインストールと変更] をクリックしてください。
4. [ファイル削除の確認] ダイアログボックスが表示されたら、[はい] ボタンをクリックします。
5. アンインストールが始まります。作業が終わるまで待ちます。以上でアンインストールは終了です。

第 2 章　Peggy の使い方

この章では、Peggyらしい使い方と、その基本的な操作方法を紹介します。より詳しい説明は、『第3章 編集』以降のリファレンスを参照してください。

Peggy の特徴

Programmer's Edit Gear - 愛称Peggyは、新しいアイデアに基づき、コンピュータのソフトウエア開発を強力にサポートするテキストエディタです。一般的なテキストファイル編集機能の他に、以下の特徴を備えています。

●文法認識 - プログラミング言語の文法に沿った色分け表示や、マウスでポイントしているキーワードの意味を表示する「キーワードチップ」、文法を認識してきれいな印刷を行なうプリティプリンタなどの機能を特別な設定なしで利用できます。

●プロジェクト - 「プロジェクト」と呼ぶ、ファイルをグループ分けして管理する機能があり、よく使うファイル、関連するファイルを分類・整理することができます。ユーザツールを定義したり、Makefileなどのスクリプトや各種コマンドのランチャーとしても機能するので、Peggyを中心とした作業環境を構築することが可能です。

●文字列検索 - 文字列の検索が強力、かつ簡単に行なえます。特にプロジェクトへ登録、グループ分けしてあるファイル群からの文字列検索が、ボタン一つで可能です。もともとプロジェクトには関連の深いファイルがグループ別に登録されているので、最も効率的な検索方法と言えるでしょう。

●データベース接続機能 - Oracleデータベース接続機能・ODBC接続機能を有し、SQL文や、PL/SQL文を実行、クエリー結果をアウトプットウインドウに表示します。エラー行へのタグジャンプもサポートしています。

●バージョン管理機能 - Peggyには、バージョン管理ソフトウェアとして全世界で広く使われているSubversion、RCS、CVS、さらにVSSのユーザインターフェイスを組み込みました。これにより、強力なバージョン管理機能を簡単に利用することが可能になりました。ソフトウェア開発に限らず、作りかけのホームページや完成後のメインテナンス、メモ書き、メールのログ、書きかけの原稿など何でも記録できます。

●スクリプト言語 - JavaScriptの文法をベースにしたシンプルで柔軟な言語「MocaScript」を搭載しました。Peggyの機能をユーザ自身が拡張することができるようになりました。

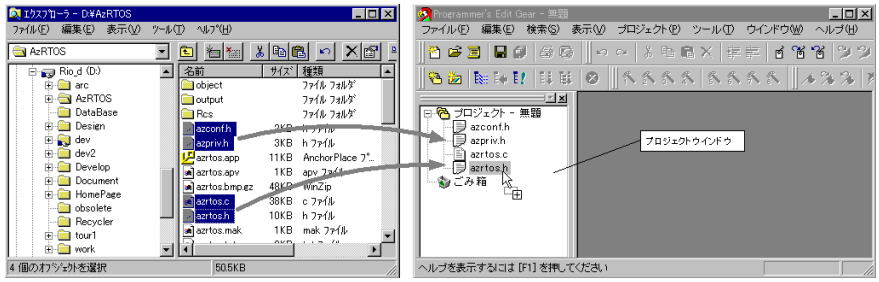
【注意】Oracleデータベース接続機能・ODBC接続機能をサポートしているのはPeggy Pro for PHPだけです。

【注意】バージョン管理ツールSubversion、CVS、RCS、VSSすべてをサポートしているのは、Peggy Pro (for PHP)です。Peggyは、RCSのみのサポートとなります。

基本操作１ － ファイルを登録しよう

テキストの編集を始める前に、まずファイルをPeggyに登録しましょう。これを「プロジェクトへ登録する」と言います。プロジェクトへ登録しておくと、いろいろな操作がより便利に、そして簡単に行なえます。

ファイルを登録するには、エクスプローラからファイルをドラッグし、Peggyのプロジェクトウインドウへドロップするだけです。1つだけでも、2つ以上のファイルやフォルダごとドロップしてもかまいません。何回かに分けてドラッグ＆ドロップすることもできます。登録済みのファイルは、プロジェクトウインドウへ表示されます。



フォルダごとドロップすると、プロジェクトウインドウに「グループ」が作られます。グループは登録してあるファイルをまとめる役目がありますが、必ずしもディスク上のフォルダと1対1に対応している必要はありません。

	全体を表わす「プロジェクト」
	ファイルをまとめる「グループ」
	削除したファイルが入る「ごみ箱」

【ヒント】すでに開いているファイルをプロジェクトに登録する最も簡単な方法は、ツールバーの[]ボタンをクリックすることです。

【ヒント】開発中プログラムのソースやインクルードファイルを登録するには、それらの入っているフォルダごとドラッグ＆ドロップすると簡単です。多少不必要なファイルが混じっていても、登録後に削除できます。

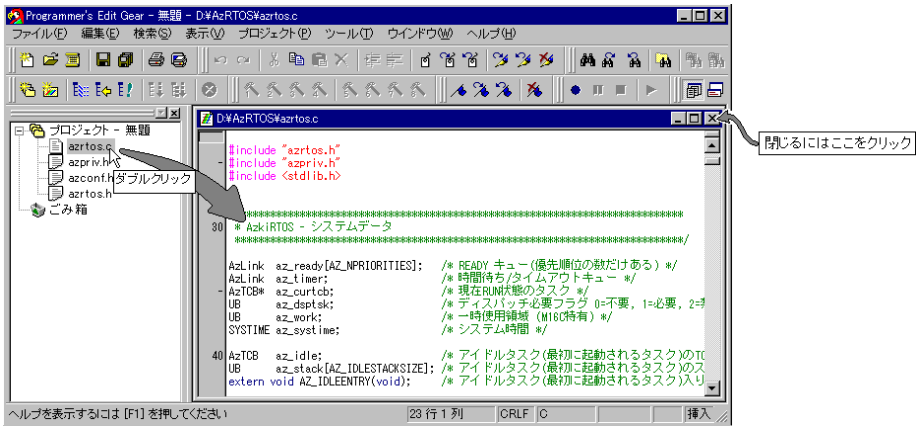
【ヒント】単なるエディタとして使うこともできますが、複数のファイルを扱うときは、プロジェクト機能を使った方が便利です。

【注意】巨大なフォルダをドロップすると、登録に時間がかかります。登録を中止するには、Escキーを押してください。

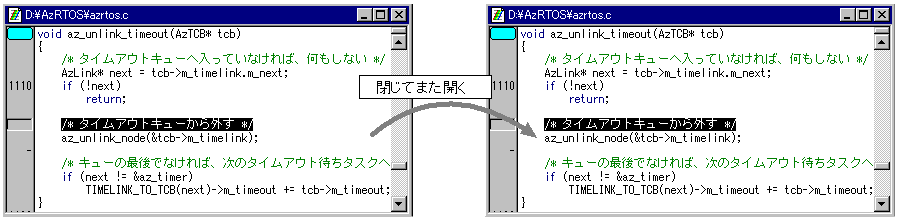
【参照】『第6章　プロジェクト機能/プロジェクトとは』、『第6章　プロジェクト機能/プロジェクトにファイルを登録する』

基本操作 2 - 開く・閉じる

テキストファイルを開くには、プロジェクトウインドウに表示されているファイルのアイコンをダブルクリックします。これだけでウインドウが開き、編集準備完了となります。



プロジェクトウインドウにファイルを登録しておくと、閉じた後でもウインドウの大きさ、スクロールやカーソルの位置、ブックマークなど、編集中の状態を記憶しています。次に開くときには以前の状態を再現するので、そのまま開いていたかのように編集を続けることができます。



【ヒント】他にもいろいろな方法でファイルを開くことができます。もちろんプロジェクトに登録してないファイルを開くことも可能です。

【注意】プロジェクトウインドウのテキストファイル以外のアイコンをダブルクリックすると、関連付けされているアプリケーションを起動する場合があります。直接Peggyで編集できるのは、テキストファイルだけです。

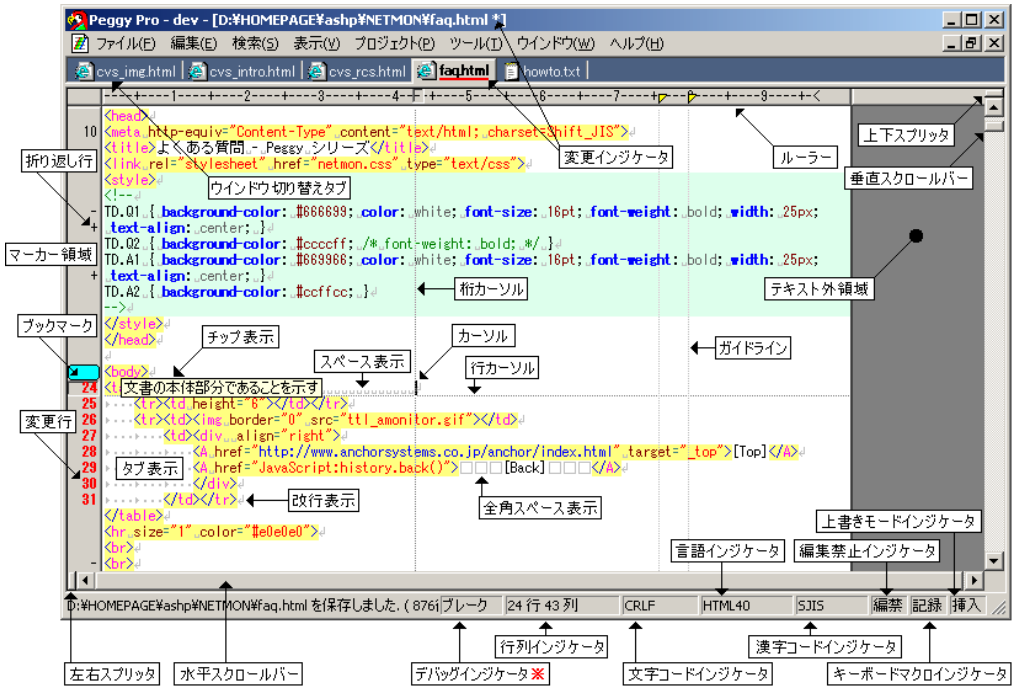
【注意】プロジェクトに登録していないファイルの編集中の状態は、保存されません。

【参照】『第3章 編集/ファイル/ファイルを開く』、『第3章 編集/ファイル/ファイルを閉じる』

基本操作 3 - 編集ウインドウ

基本的な使い方は、一般のWindowsアプリケーションと共通です。すでにWindowsアプリケーションの操作に慣れているならば、すぐに使い始めることができるでしょう。編集操作の詳しい説明は、『第3章 編集』を参照してください。

下図は、テキスト編集中の編集ウインドウと各部分の説明です。



ファイル名	編集しているファイルのパスです。
変更インジケータ	テキストを変更すると、アスタリスク(*)をファイル名のすぐ右に表示します。
ルーラー	折れ曲がり位置や、現在のカーソルのカラム位置を表示します。マウスでドラッグし、折れ曲がり位置を調整することもできます。
カーソル	入力するテキストの挿入位置を示しています。
行カーソル	カーソルのある行を分かりやすく表示します。
桁カーソル	カーソルのある桁を分かりやすく表示します。
キーワードチップ	プログラミング言語のキーワードの説明を簡素にチップ表示します。
マーカー領域	ブックマークや行番号を表示する領域です。
テキスト領域	プログラムやテキストを表示する領域です。
テキスト外領域	折れ曲がりカラムより右の領域です。
上下スプリッタ	ウインドウを上下に分割します。
垂直スクロールバー	ウインドウを縦にスクロールします。

左右スプリッタ	ウインドウを左右に分割します。
水平スクロールバー	ウインドウを横にスクロールします。
デバッグインジケータ	PHPデバッガの状態を表示します。
行列インジケータ	カーソルのある行の行番号と桁を表示します。
文字コードインジケータ	カーソル位置の文字コードを表示します。行末では改行文字の種類によって、[CRLF]、[LF]、[CR]と表示します。またファイルの終わりでは、[EOF](End-of-File)となります。【メモ】参照
言語インジケータ	編集中テキストの言語を表示します。
漢字コードインジケータ	編集中のテキストの漢字コードを表示します。
編集禁止インジケータ	書き込み禁止のファイルを開いているときに[編禁]と表示します。
キーボードマクロインジケータ	キーボードマクロの記録中に[記録]と表示します。
上書きモードインジケータ	上書きモードのときは[上書]、挿入モードのときは[挿入]、矩形選択中は[矩形]と表示します。

【注意】デバッグインジケータは、Peggy Pro for PHP のみに表示されます。

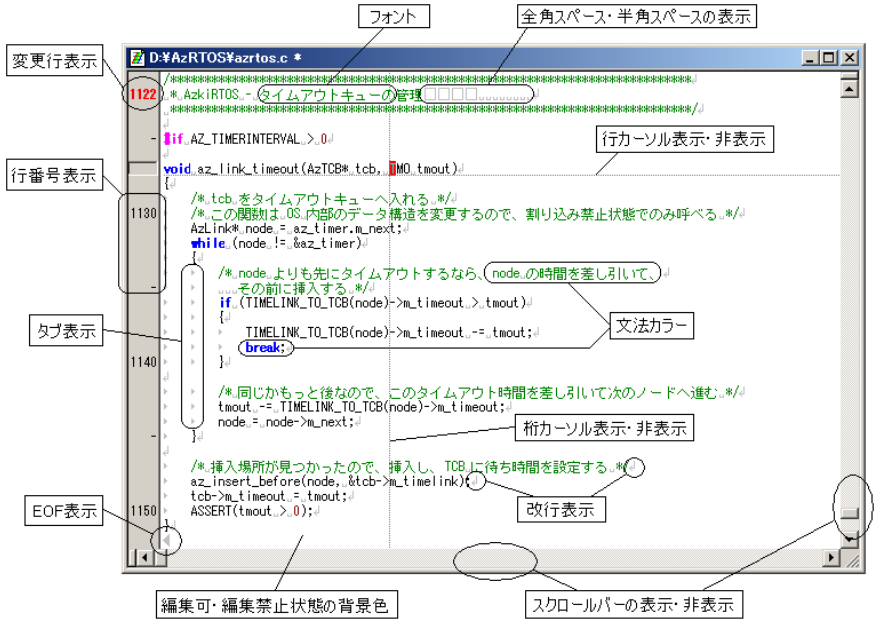
【ヒント】万が一画面が乱れてしまったら、メニューの[表示]-[リフレッシュ]を選択するか、Ctrl+Lとタイプしてみてください。ウインドウをリフレッシュし、乱れを修復します。

【メモ】文字コードインジケータに表示する文字コードは、そのファイルの漢字コードで表示します。例えば、EUCファイルならEUCの漢字コードを表示します。[オプション設定]ダイアログボックス[編集]ページにある[文字コードインジケータに常にShift JISコードを表示する]をONにすると、常にShift JISコードで表示することもできます。

【参照】『第3章 編集』

基本操作 4 － 画面のカスタマイズ

編集ウインドウは、フォント、文字の色、背景色、行番号表示、カーソルの形、行カーソル、改行/ファイルの終わり(EOF)/タブ/半角や全角スペースの表示・非表示など、多くの部分をカスタマイズできます。好みに合わせて変えてみてください。下図に示した部分がカスタマイズできます。



画面カスタマイズは、メニューの[表示]か、[ツール]-[オプション]を選択し、[オプション設定]ダイアログボックスから行ないます。

【ヒント】カスタマイズした設定は、レジストリに保存されます。

【ヒント】編集ウインドウ以外に、ツールバーボタンの種類や配置もカスタマイズできます。

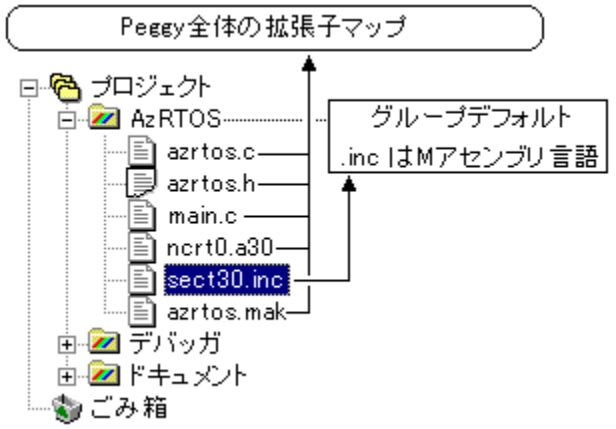
【参照】『第11章 カスタマイズ』

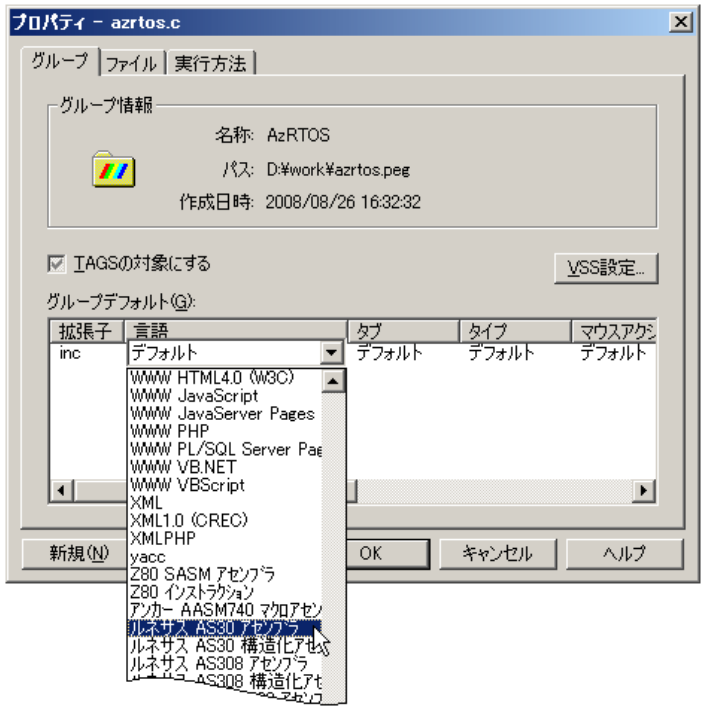
基本操作５ - 言語を設定しよう

Peggyは、ファイル名の拡張子からそのファイルのプログラミング言語を判断し、文法に従った色分けやキーワードの説明を表示しています。一般的な拡張子の場合、何もしなくてもPeggyが判断します。

しかし、意図しない言語が選ばれてしまったり、特殊な拡張子、または同じ拡張子を持った異なる言語のファイルを正しく表示できるようにするには、プロパティを設定します。以下は、拡張子が<.inc>であるファイルをアセンブリ言語のソースとして設定する手順です。

1. プロジェクトウインドウのファイルアイコンを右マウスボタンでクリックし、ポップアップメニューから［プロパティ］を選択してください。
2. ［グループ］ページを選択します。
3. ［新規］ボタンをクリックすると、［新規拡張子］ダイアログボックスが現れます。
4. 拡張子<inc>を入力して、[OK]をクリックしてください。これで［グループデフォルト］リストに、新しい拡張子<.inc>が登録されました。
5. 拡張子<inc>の［言語］欄で、［デフォルト］と表示されている部分を左マウスボタンでクリックします。
6. 現れた言語リストから、<.inc>に対応させたい言語を選択してください。
7. [OK]をクリックし、ダイアログボックスを閉じます。





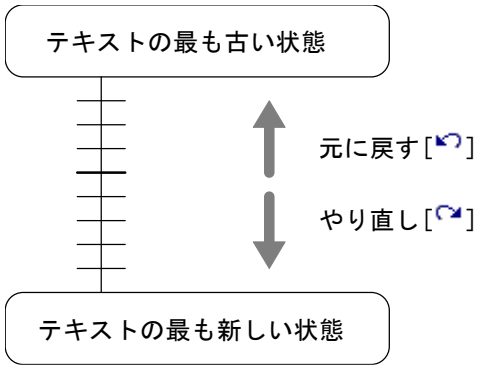
【ヒント】この方法では、同じグループにある拡張子<.inc>のファイルすべての言語を設定したことになります。他に、個々のファイルやPeggy全体での設定を変えることも可能です。プロジェクトウインドウの中で、ファイルにより近いところの言語設定が優先します。

【ヒント】編集中のテキストに対してどの言語が選ばれているかは、常にステータスバーの[言語インジケータ]に表示されています。

【参照】『第5章　言語モード』、『第6章　プロジェクト機能/プロパティ /[ファイル]プロパティページ』、『第11章　カスタマイズ/オプション設定/[拡張子]ページ』

基本操作 6 - 取り消しとやり直し

テキストを誤って変更してしまったら、メニューの[編集]-[元に戻す]で元へ戻すことができます。おおよそ10,000回程度前まで戻れます。戻り過ぎてしまったら、今度は[編集]-[やり直し]で再実行する、つまり[元に戻す]を取り消すことができます。



【ヒント】ボタン[ ]、[ ]やショートカットキーからも使えます。

【ヒント】回数と取り消す方法を記憶しているメモリ容量の両方から、[元に戻す]操作の上限を設定しています。テキストあたり10,000回、メモリ容量は5メガバイトまでです。

【注意】一回で限度を超える大きな変更操作は、取り消しできない場合があります。あえて取り消すには、そのままファイルを保存せずに閉じてください。ただし、最後に保存したときから後のすべての変更も失われます。

【参照】『第3章 編集/編集/取り消しとやり直し』

効果的な編集 1 - いろいろなジャンプ

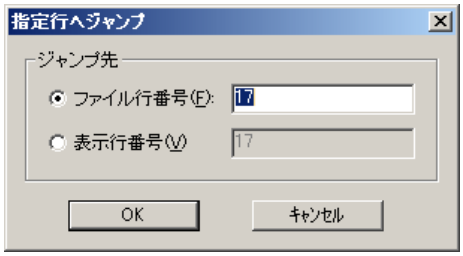
編集中にはテキストのあちこちを飛び回らなければならないことがあります。Peggyでは、下表にあるジャンプをサポートしています。これらを上手に使い分けることで、編集したい所へ素早く移動できます。詳細は、リファレンスのそれぞれの部分を参照してください。

指定行ジャンプ	行番号を指定してジャンプします。
対応括弧ジャンプ	カーソル位置の括弧と対になる括弧へジャンプします。
タグジャンプ	コンパイラからのエラーメッセージや、ファイルから検索の出力に含まれるファイルパスと行番号からジャンプします。
変更行ジャンプ	テキスト内の変更を加えた行へ順番にジャンプします。
ブックマークジャンプ	行に印(ブックマーク)を付けておけば、それらへ順にジャンプできます。
前後関数ジャンプ	前後の関数の先頭へカーソル移動できます。
シンボル検索	色々な言語の関数や変数の定義位置を抽出したファイル<TAGS>を作成し、それらをリスト表示したり、その定義位置へ簡単にジャンプすることができます。

【参照】『第3章 編集/ジャンプ』

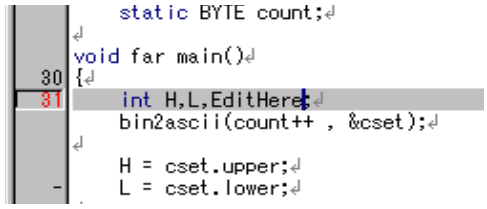
指定行ジャンプ

17行目へジャンプ



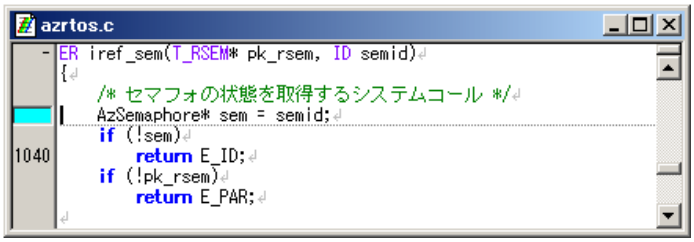
変更行ジャンプ

[ ]ボタン、Ctrl+Shift+9キーで31行目へジャンプ



ブックマークジャンプ

[ ]ボタン、F2キーで1038行目へジャンプ



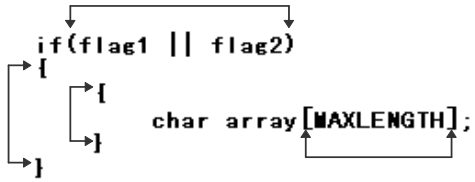
タグジャンプ

[ ]ボタン、Ctrl+Tキーで11行目、19行目へジャンプ

```
D:¥dev¥projct¥main.c 11  : Syntax Error No 5
D:¥dev¥projct¥main.c 19  : Warning No 12
```

対応括弧ジャンプ

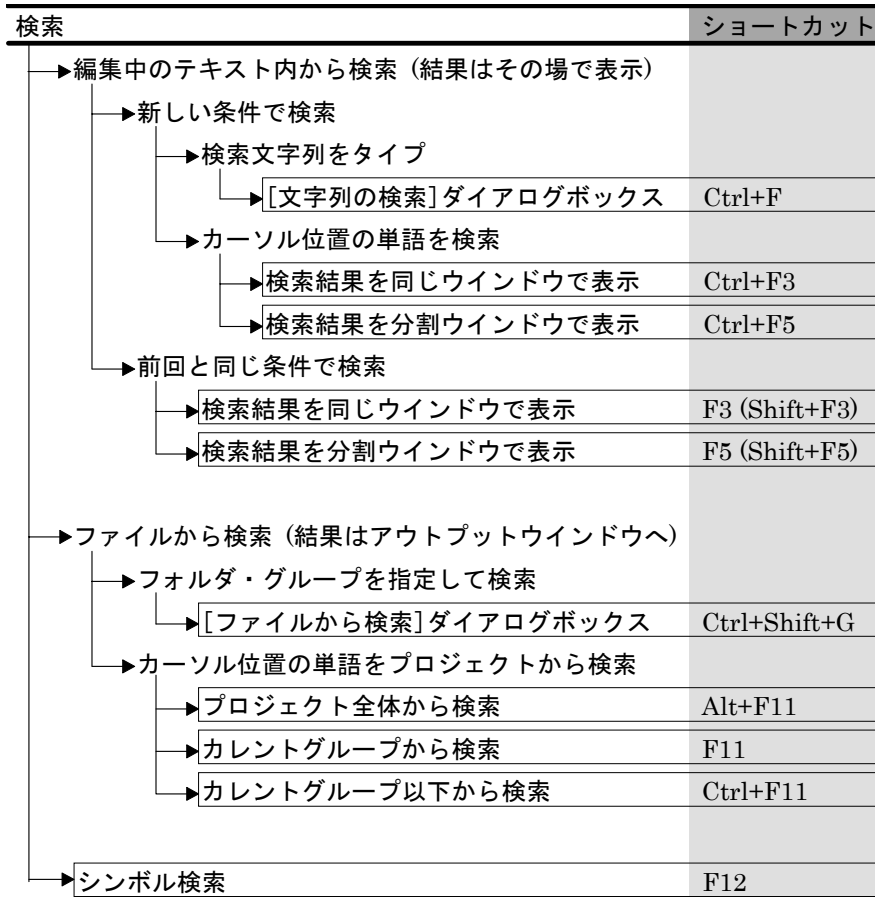
Ctrl+]キー（図中矢印に示すような対応括弧間でジャンプ）



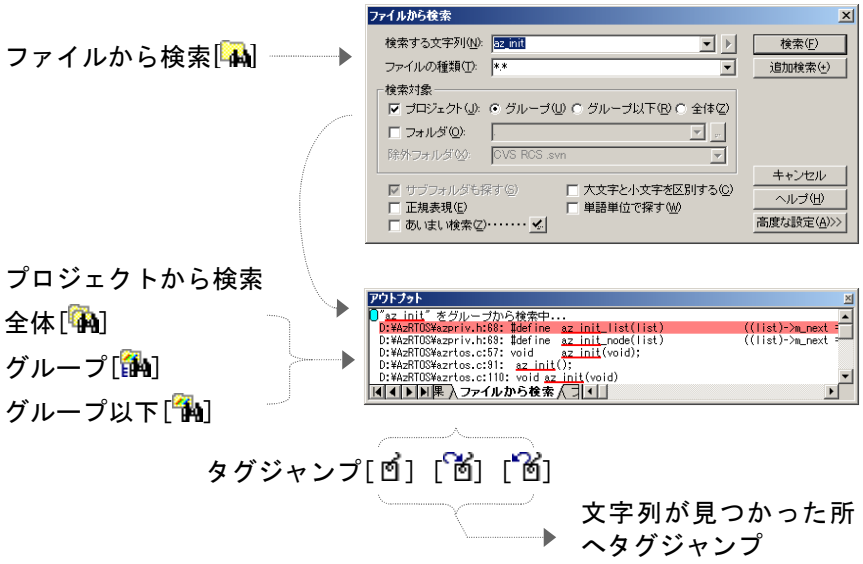
効果的な編集 2 – 検索するには

Peggyの検索機能は非常に強力で、状況に応じたいろいろな方法で文字列を探すことができます。例えば、新しい条件で探すか、それとも以前の検索を繰り返すのか、編集しているテキスト内から探すのか、ファイルから探すのか、同じウインドウか別のウインドウか、...などです。

図は、Peggyの検索コマンドマップです。これらの検索コマンドを使い分けられるようになれば、いつでもすぐに目的の文字列が見つかるでしょう。



編集しているテキスト内での検索は、見つかった文字列を直接表示しますが、ファイルやプロジェクトからの検索は、いったんすべての検索結果がアウトプットウインドウ内へ出力されてから、タグジャンプと同じ操作で順番にジャンプできます。



[文字列の検索]や[ファイルから検索]ダイアログボックスには、検索条件を変えるオプションがあります。大文字小文字の区別など基本的な検索条件は、[検索]メニューやツールバーのボタンからも選ぶことができます。

【ヒント】単語をプロジェクトから検索するには、その単語を右マウスボタンでクリックし、ポップアップメニューから[...から検索]を選択すると便利です。単語と認識されない文字が含まれているときは、文字列を選択してから右マウスボタンでクリックします。そして何か見つかったら、F4キーを押してタグジャンプしてください。

【ヒント】正規表現パターンで検索することもできます。正規表現の文法をはっきり覚えていないときは、[文字列の検索]や[ファイルから検索]ダイアログボックスの[▶]ボタンをクリックしましょう。メニューから選んで入力できます。

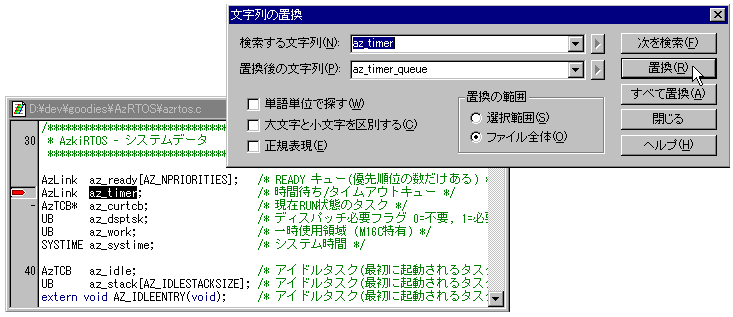
【ヒント】ファイルから検索している途中でもタグジャンプできます。

【参照】『第4章 検索・比較/検索』

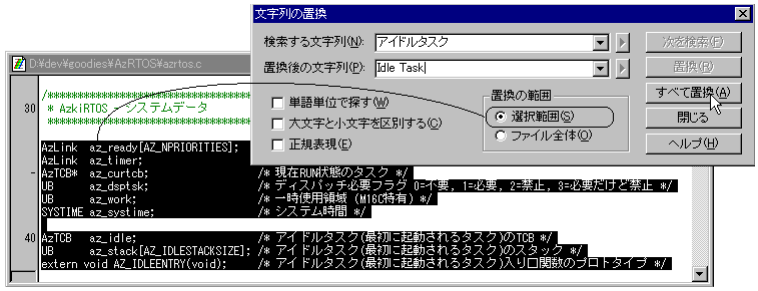
効果的な編集 3 － 文字列の置換

編集中テキスト内の特定の文字列を別のものに置き換えることができます。1つ1つ確認しながら置き換えたり、テキスト全体、または指定した範囲を一括して置き換えることも可能です。

文字列を置換するには、まずテキストを開き、次にメニューの[検索]-[置換]を選んで、[文字列の置換]ダイアログボックスを開きます。必要な項目を指定してから、[次を検索]、[置換]を状況に合わせてクリックしてください。



テキストの特定範囲にある文字列に限定して置換するには、まずその部分を選択し、その後で[文字列の置換]ダイアログボックスを開きます。[すべてを置換]をクリックすると、指定範囲のすべての文字列を一度に置き換えます。



【ヒント】正規表現を使った高度な置換もサポートしています。正規表現は、『第4章 検索・比較・置換』に詳しい解説があります。

【注意】編集禁止テキスト内の文字列は置換できません。

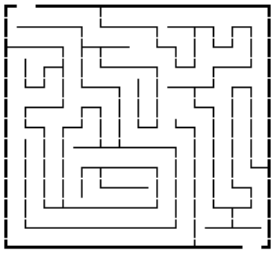
【参照】『第4章 検索・比較・置換』

効果的な編集 4 - 罫線

罫線文字を使ってある部分を囲んだり、簡単な図形を描くことができます。また部分罫線を消すこともできます。

Alt+矢印キー	矢印の方向に罫線を引きます。
Shift+Alt+矢印キー	矢印の方向に罫線を消去します。
Alt+Home、[┣]、[┼]、[┼]	罫線の種類を変更します。

罫線の例



引数1	argc	コマンドラインパラメタの数
引数2	argv	各パラメタへのポインタ配列
戻り値	int	終了ステータスコード

【ヒント】間違って引いた罫線は、[編集]-[元に戻す]で消すことができます。

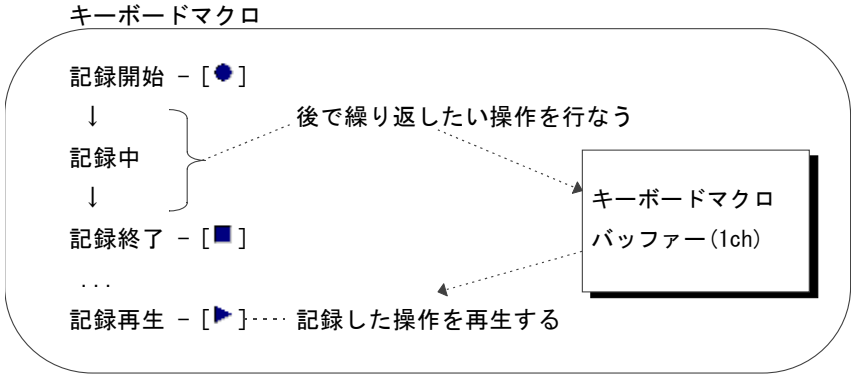
【ヒント】罫線の入ったテキストを印刷するときは、固定ピッチのフォントを使うようにしましょう。プロポーショナルピッチのフォントでは桁位置が揃わないため、罫線がガタガタになります。

【参照】『第3章 編集/罫線機能』

効果的な編集 5 - キーボードマクロ

繰り返し行なう一連のキー操作を記録しておき、後から再生する機能をキーボードマクロと呼びます。ダイアログボックスを表示せずに行なえるほとんどのテキスト編集操作を記録・再生できます。同じ修正をする行がいくつもあるような場合に便利です。

記録開始[●]から記録終了[■]までの間に行なったキー操作が記録され、[▶]で再生します。



Peggy内部には、一連のキーストロークを記憶するバッファが1チャンネルしかないため、新しいキーストロークを記録すると、古いものは消えてしまいます。しかし、プロジェクトへ保存することができるので、いくつでも制限なく記録できます。

【ヒント】キーボードマクロをプロジェクトへ保存するには、メニューから[ツール]-[キーボードマクロ]-[プロジェクトへ保存]を選択します。プロジェクトへ保存すると、複数のキーボードマクロを登録して使い分けることができます。

【注意】記録できるのはテキストを編集する操作だけなので、アウトプットウインドウやプロジェクトに対する操作は残せません。

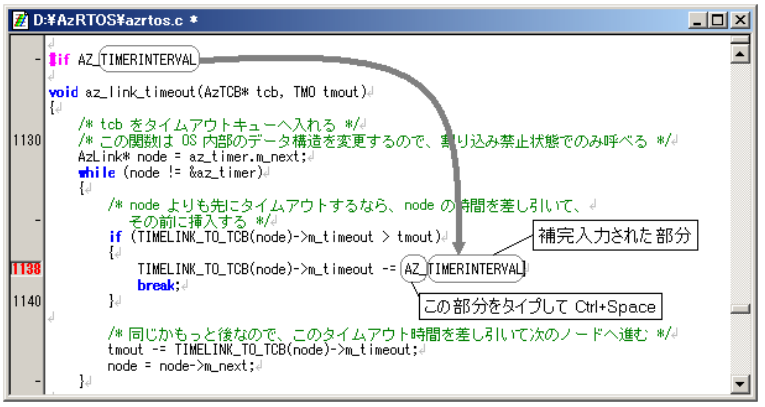
【参照】『第3章 編集/キーボードマクロ』、『付録A　コマンド一覧』

効果的な編集 6 - 補完入力

補完入力とは、文字列を途中まで入力し、後の部分をPeggyに補わせる機能です。すべての文字をタイプしなくてもよいので、入力スピードをアップすることができます。Peggyでは、以下3通りの補完入力をサポートしています。

単語補完	途中まで入力した文字と同じ綴りで始まる単語をカーソルの前後から探して、足りない部分を補います。
キーワード補完	プログラミング言語のキーワードを途中まで入力して、残りの部分を補います。
短縮入力	予めキーとなる文字列と文章を定義しておき、キー文字列を文章に変換する機能です。

短縮入力は、予めキー文字列と変換する文章を登録しておく必要がありますが、他は前準備なしですぐに使えます。次の図は、単語補完の動作を示しています。



【ヒント】単語補完では、[オプション設定]ダイアログボックスの[補完]ページで、補完候補を探す範囲や方向を指定できます。デフォルトでは、ウインドウ上に見えている範囲から探します。

【ヒント】[オプション設定]ダイアログボックスの[補完]ページで、[日本語も補完]をONにすると、日本語文章も補完入力できるようになります。

【参照】『第3章 編集/補完入力』、『第11章 カスタマイズ/オプション設定/[補完]ページ』

効果的な編集 7 - 漢字コードと改行コード

いろいろな漢字コードと改行コード

複数のオペレーティングシステムが混在するネットワーク環境では、いろいろな漢字コードを含んだテキストファイルを読み書きしなければなりません。また、行の終わりを表わす改行コードも、使用環境によって異なっています。以下に主なテキストファイルの漢字コードと改行コードを示します。

用途/OS	漢字コード	改行コード
Windows	Shift JIS、Unicode (NT)	CR-LF
MS-DOS	Shift JIS	CR-LF
UNIX	EUC、Shift JIS	LF
マッキントッシュ	Shift JIS	CR
インターネット	JIS、Unicode、UTF-7、UTF-8	CR-LF、他

Peggyではこれらすべてのテキストファイルをほとんど意識することなく編集できます。これらすべての漢字コードと改行コードの組み合わせを自動認識し、それに従って読み込み、保存するからです。いま編集しているファイルの漢字コードが何であるかは、漢字コードインジケータで確認できます。

しかし、100%正確に漢字コードを自動判定する方法がないため、まれに正しくない漢字コードとしてファイルを開いてしまう可能性があります。特に短いテキストファイルだと、判定材料となる文字が少ないため適中率が低下します。もしも正しくない漢字コードでファイルが開かれた場合、メニューの[ファイル]-[開き直す]コマンドで明示的に漢字コードを指定し、ファイルを読み直してください。

改行コード

改行コードは、行単位で記憶しています。各行の改行コードを知るには、[表示]-[改行記号]コマンドで改行記号表示をONにしてください。それぞれの行の最後に、その行の改行コードに従った記号が表示されます。また、行末にカーソルを移動させると、文字コードインジケータにも表示されます。

改行コード	改行記号
CR-LF	↵
LF	↓
CR	←

【メモ】Peggy内部では漢字コードと改行コードを別々に管理しているので、上表の組み合わせに限らず、任意の漢字コードと改行コードを組み合わせることも可能です。

【ヒント】Peggyは、ファイルの漢字コードからShift JISに変換して読み込み、保存のときにまたファイルの漢字コードへ戻して書き出します。

漢字コードの学習

漢字コードの自動認識は便利ですが、そのため多少ファイルを開くのが遅くなったり、誤判定の可能性もあります。そこでPeggyでは、ファイルの漢字コードを学習する機能をサポートしました。プロジェクトに登録してあるファイルの漢字コード

を記憶し、それを使って2回目以降の自動判定を省略します。学習機能をOFFにする(毎回自動判定する)ことも可能です。

この学習機能は、誤判定の防止にも役立ちます。最初にファイルを開いたとき漢字コードが誤って判定されてしまっても、[ファイル]-[開き直す]コマンドで開き直すと、その漢字コードを学習するため、以後は正しい漢字コードでファイルを開くようになります。

【ヒント】改行コードの種類は学習しません。ファイルを開くときに内部で使われている改行コードを種類別に集計し、最も多く使われているものをそのファイルの改行コードとします。

漢字コードの変換

ある漢字コードのファイルを開き、別の漢字コードで保存することができます。その際、改行コードも選択可能です。漢字コードを変えて保存するには、[ファイル]-[名前を付けて保存]コマンドを使います。

漢字コードの部分変換

1つのファイルの中にいろいろな漢字コードが混ざってしまっているような場合、ファイルのある部分だけの漢字コードを変換し、正しく表示できるようにする機能です。[編集]-[文字変換]-[選択範囲の漢字コード]コマンドを参照してください。

漢字コードとファイルから検索

ファイルから検索で日本語を含んだ文字列を探す場合、同じフォルダ中にいろいろな漢字コードのファイルが混在していても検索できます。ただし、検索できる文字は、Shift JIS文字セットに含まれているものに限ります。第3水準の漢字やUnicode独自の文字などは検索できないので注意してください。

プロジェクトに登録してあるファイルを、[検索]-[グループから検索]、[検索]-[グループ以下から検索]、[検索]-[プロジェクトから検索]コマンドを使って検索する場合には、プロジェクト中に学習されている漢字コード情報を使って検索するため、漢字コード自動判定の必要がなくなり、より高速で正確な検索が行えます。

プロジェクトに登録されていないファイルの漢字コードは、毎回自動判定します。自動判定には誤差があり、100%正確に判定できるとは限りません。特に短いファイルの場合は判断材料である文字が少ないため、誤判定してしまう可能性があります。

漢字コード	ASCII文字の検索	日本語(漢字・かな)の検索
Shift JIS	○	○
EUC	○	○
JIS	○	○
UTF-7	○	○注1
UTF-8/UTF-8N	○	○
Unicode	○	○

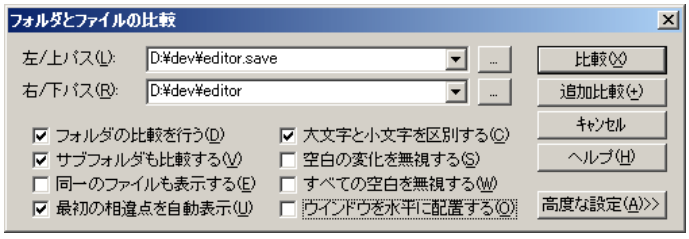
【注1】UTF-7は、編集オプションの[UTF-7を自動判定の対象外とする]をチェックしていないときだけ、漢字コード自動判定の対象となります。

【メモ】[文字列の検索]ダイアログボックスから[複数ファイル]を選択し、Peggyに読み込みながら検索する方法もあります。

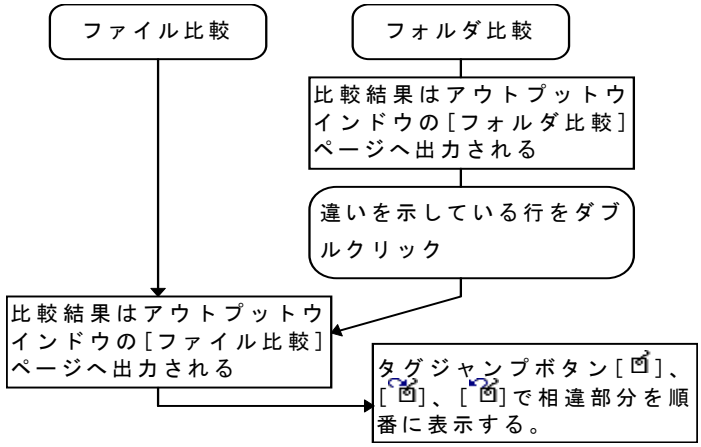
新旧バージョンの違いを比べる

2つのファイルの相違点を比べることができます。2つのフォルダ内のファイルをまとめて比較したり、さらにその中のサブフォルダまで深く比較することも可能です。別フォルダに保存している旧バージョンからの変更箇所を調べるのに威力を発揮します。

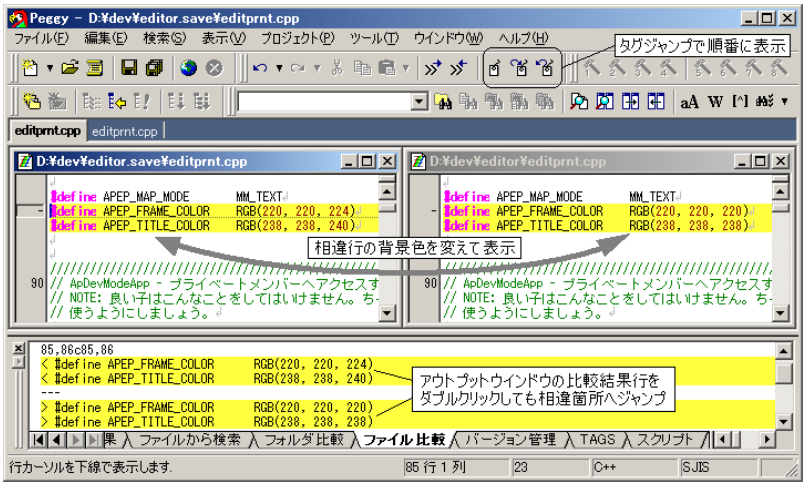
ファイルやフォルダを比較するには、[ ]ボタンを押して[フォルダとファイルの比較]ダイアログボックスを開いてください。必要な項目を設定して[比較]ボタンをクリックすると、違いを調べて結果をアウトプットウインドウへ出力します。



ファイルを比較したのかフォルダを比較したのかによって、実際の相違点を表示させるまでの手順が違ってきます。下図がその手順です。



タグジャンプボタン[ ]、[ ]、[ ]をクリックするだけで、相違点を順番に見て行くことができます。これらは編集ウインドウなので、そのままテキストを編集することも可能です。



【ヒント】比較するファイルやフォルダは、エクスプローラからドラッグ＆ドロップで指定すると簡単です。

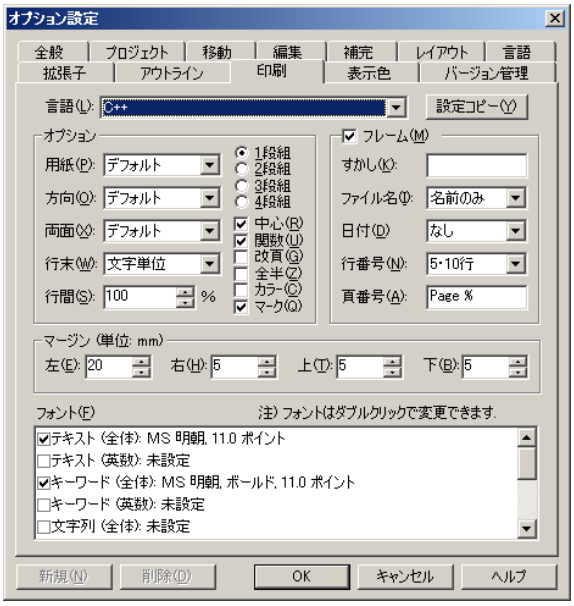
【ヒント】何度か繰り返し行なう比較条件は、プロジェクトに保存しておきましょう。頻繁に比較するときは、ツールボタンへ割り当てることもできます。

【注意】異なる漢字コードのファイルを比較することはできません。

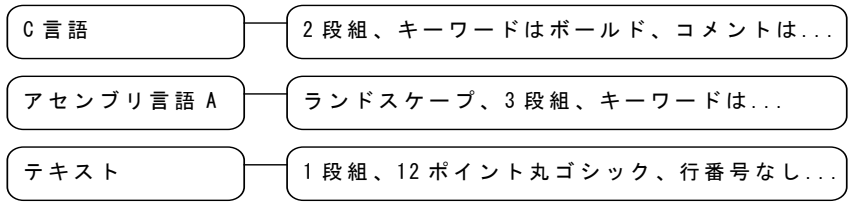
【参照】『第4章　検索・比較・置換/ファイル比較』、『第6章　プロジェクト機能/プロパティ /[ファイル比較]ページ』

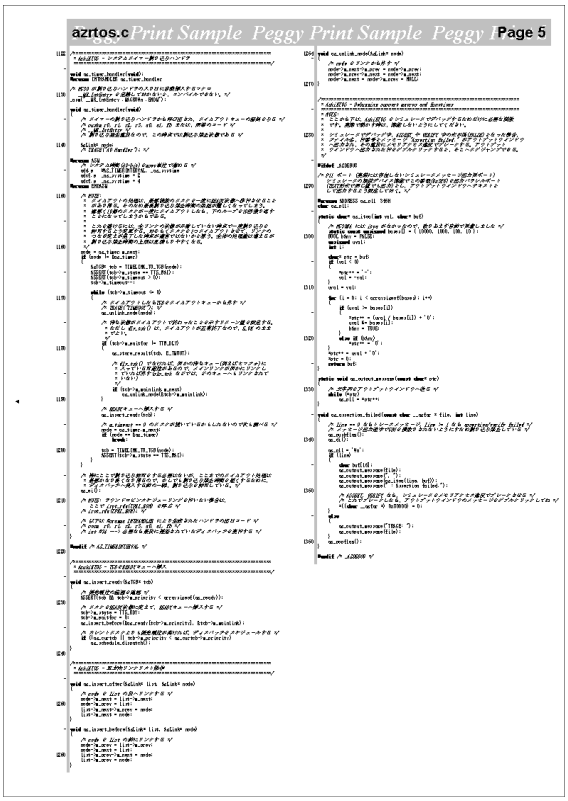
印刷する

テキストを印刷するには、メニューの[ファイル]-[印刷]を選択してください。しかしその前に印刷オプションを設定しておくと、お好みのスタイルで印刷することができます。印刷オプションを設定するためのダイアログボックスは、[ファイル]-[印刷オプション]で表示できます。



印刷オプションは、言語毎に設定できます。つまり普通の日本語テキスト、C言語のソースファイル、A社のアセンブリ言語、... それぞれにフォントや行間、段組数、紙の方向など、多くの項目が選択できるのです。さらにソースファイルであれば、キーワード、名前、数値、コメントなど、文法に従ってフォントを切り替えながら印刷できます。





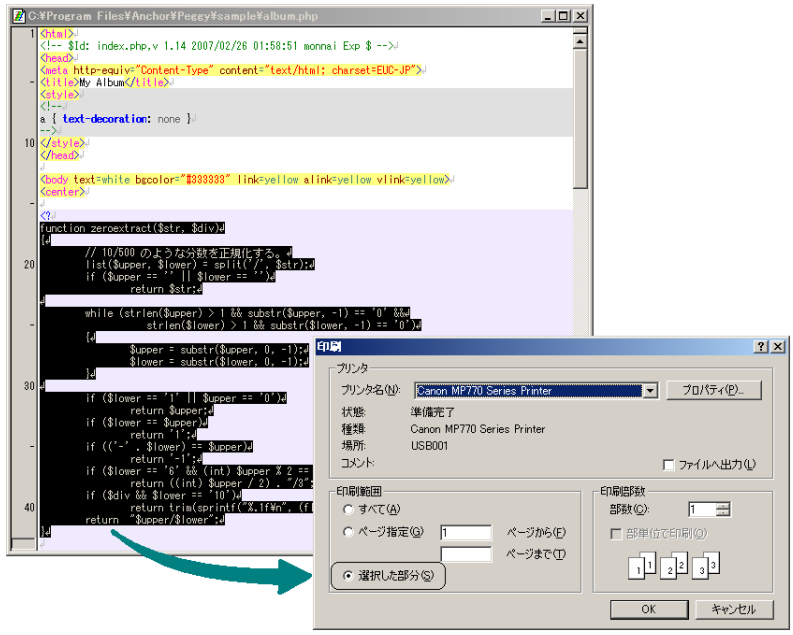
【ヒント】印刷イメージを前もって確認できます。印刷オプションを変更したときなど、実際の印刷前に[ファイル]-[印刷プレビュー]で確認すると失敗がありません。

【ヒント】印刷オプションの説明を見るには、[オプション設定]ダイアログボックスの[印刷]ページを開いている状態で、[ヘルプ]ボタンをクリックしてください。

【参照】『第11章 カスタマイズ/オプション設定/オプション[印刷]ページ』

テキストの一部分を印刷するには？

長いファイルの一部分だけを印刷するには、まずテキストの印刷したい部分を選択します。その後でメニューから[ファイル]-[印刷]で[印刷]ダイアログボックスを開き、[OK]をクリックしてください。



【ヒント】プレビューでも選択部分の印刷イメージを確認することができます。

【ヒント】[印刷]ダイアログボックスを表示しない[ダイレクト印刷]でも、選択範囲だけを部分印刷できます。

【参照】『第11章 カスタマイズ/オプション設定/オプション[印刷]ページ』

ビルトイン「MocaScript」を実行してみよう

Peggyにはマクロ言語の実行環境が組み込まれています。JavaScriptに準じた「MocaScript」というスクリプト言語です。これを使えばエディタを自由に拡張したり、思いの動作をさせることができるようになります。

スクリプトの置き場所

Peggyに実行させるスクリプトファイルは、<share¥script>フォルダに入れます。

例えば、インストールしたフォルダの中の<share¥sample>フォルダの中を覗いてください。いくつかの<***.ms>ファイルがあります。その中から、<挿入.ms>を<share¥script>フォルダへコピーしてください。

Peggy に読み込み

スリプトファイルをPeggyで実行できるようにするには、メニューの[ツール]-[スクリプト]-[再ロード]を実行します。または、Peggyを再起動すれば自動的に読み込まれます。その後プロジェクトウインドウの[スクリプト]ページを表示してください。

コマンドが実行単位

[ ]はスクリプトファイルを表します。[ ] がコマンドグループで、[ ] がコマンドです。

スクリプトコマンドを実行してみる

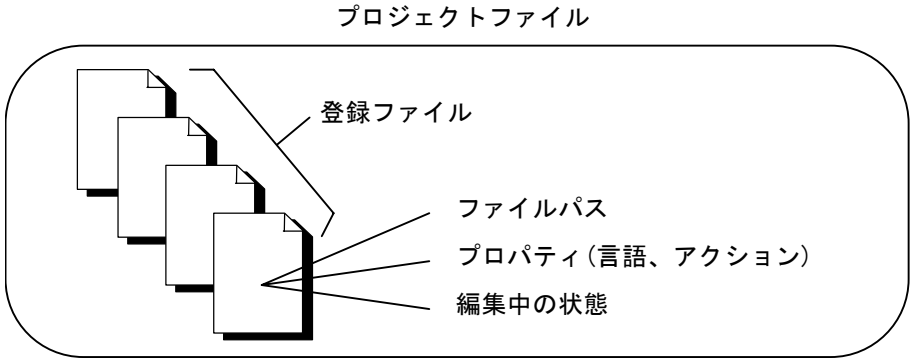
いずれかのコマンド[ ]をダブルクリックしてください。上図のコマンドなら顔文字が編集画面のカーソル位置に入力されます。続いて、他のコマンドもダブルクリックしてみてください。

【参照】最新の提供マクロは、WEBをご覧ください。

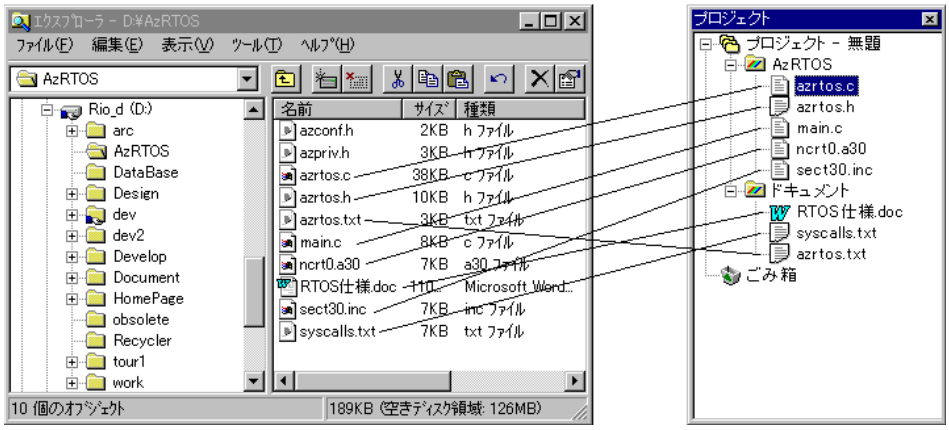
【参照】『第7章 ビルトインスクリプト』

プロジェクトファイル

登録したファイルや言語の設定などは、まとめてプロジェクトファイルに保存します。プロジェクトファイルの拡張子は、<.peg>です。あるまとまった仕事や開発プロジェクトで必要となるファイル群をまとめておけば、効率よく作業を進めることができるでしょう。プロジェクトファイルを開くと、以前編集していた状態をほとんどそのまま復元します。



プロジェクト内部のグループ階層と、実際のファイルがあるディスク上のフォルダ階層とが同じである必要はありません。ディスク上ではフォルダAにあるファイルを、プロジェクトのグループBに登録したり、複数のグループX、Y、Zに繰り返し登録することも可能です。ディスク上のファイルとは関係なく、使い勝手が良くなるよう自由に構成してください。



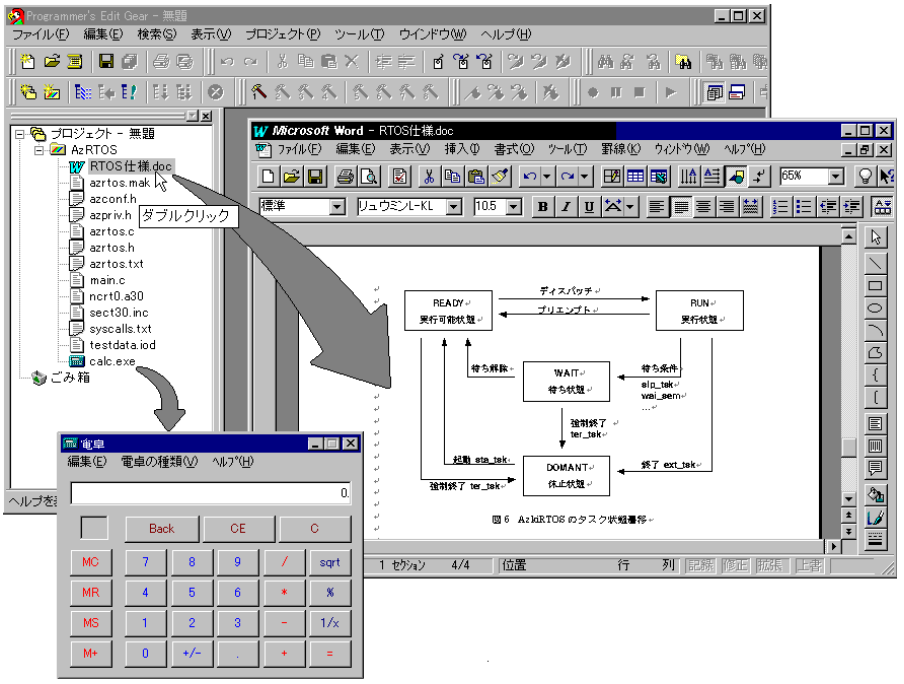
【ヒント】なんでもかんでも登録して巨大なプロジェクトファイルになってしまうと、全体の処理速度が低下することがあります。プロジェクトは、ほどほどの大きさに収まるよう分割しておいた方が効率的です。

【注意】2つのプロジェクトファイルを同時に開くことはできません。プロジェクトを交互に開くか、Peggyを2つ起動してください。

【参照】『第6章　プロジェクト機能/プロジェクトとは』

プロジェクトの応用１ - テキスト以外も登録しよう

プロジェクトには、テキストファイル以外になんでも登録できます。登録したファイルがあるアプリケーションのデータファイルならば、プロジェクトウインドウのアイコンをダブルクリックすることで、そのデータファイルに関連するアプリケーションを起動します。ちょうどエクスプローラのような感じです。



登録ファイルが<.exe>や<.com>など、そのまま実行できるものであれば、ダブルクリックでそのコマンドを実行します。例えば、<calc.exe>を登録しておくと、Peggyから電卓を起動できます。

プロジェクトに登録できるものと、それに対してどんな操作ができるかの一覧表です。◎は、マウスでダブルクリックしたときのデフォルト動作です。

種類（ファイル関連）	編集	実行	アプリケーション起動
テキスト	◎	×	○
スクリプト	◎	○	○
アプリケーションデータ	○	×	◎
実行可能（<.exe>、<.com>）	×	◎	×

【ヒント】<.com>や<.exe>ファイルのプロパティには[実行方法]ページがあり、コマンドを実行するときのパラメタや実行フォルダなどを設定できます。

【注意】Peggyから実行中のコマンドやスクリプトへキーボード入力を送ることはできません。ユーザからの入力を要求するコンソールアプリケーションは、[実行方法]ページの[出力をアウトプットウインドウへリダイレクトする]をOFFにし、独立したコンソールウインドウ上で実行させてください。

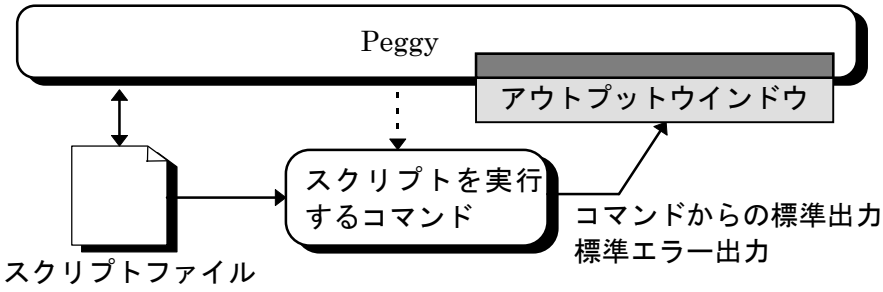
以下の4つはファイルと関連なく、プロジェクト内だけに存在するもので、キーボードマクロや検索条件をプロジェクトへ保存すると作られます。それぞれ独自のプロパティや動作を持っています。

	種類（Peggy独自）	動作
	文字列検索	編集中のテキストから文字列を検索する
	ファイルから検索	ファイルから文字列を検索する
	ファイル比較	ファイルやフォルダを比較する
	キーボードマクロ	キーボードマクロを再生する

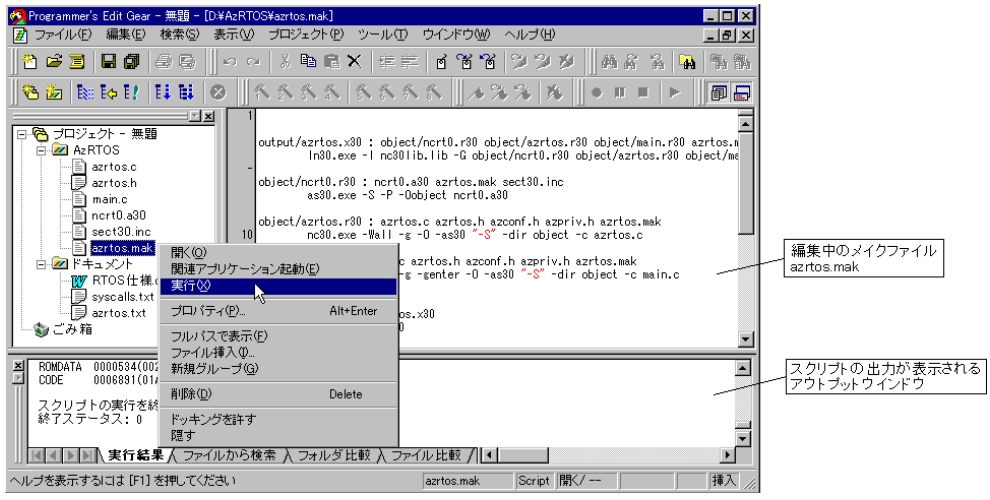
【参照】『第6章　プロジェクト機能/プロジェクトにファイルを登録する』、『第6章　プロジェクト機能/プロパティ /[ファイル]プロパティページ』

プロジェクトの応用 2 - 外部スクリプトを実行させよう

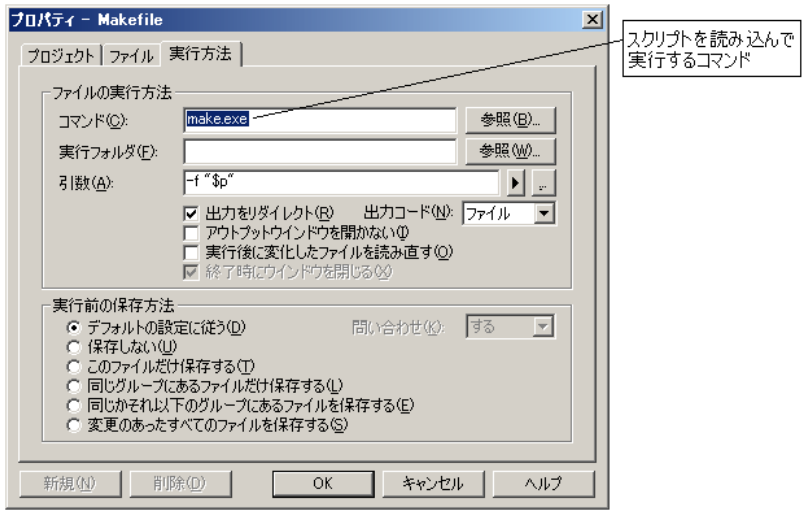
テキストファイルの中には､特定のコマンドの入力となって実行することができるものがあります。バッチファイル<.bat>は、<cmd.exe>が読み込んで解釈・実行します。メイクファイル<.mak>は、<make.exe>の入力となり、ソースファイルをコンパイル・リンクします。Peggyでは、これら別のコマンドの力を借りて実行できるテキストを「スクリプト」と呼びます。スクリプトを実行するのは簡単です。



1. 登録してあるスクリプトファイルのアイコンを右マウスボタンでクリックし、現れたポップアップメニューから［実行］を選んでください。
2. スクリプトの実行が始まり、出力メッセージがアウトプットウインドウへ表示されます。



【ヒント】あまり一般的ではないスクリプトでは、実行させるためのコマンドが自動設定されません。プロパティの[実行方法]ページを開いて、コマンドのパスやパラメータの渡し方を設定してください。



【ヒント】スクリプトはテキストでもあるので、登録したファイルのアイコンをダブルクリックすると、そのファイルを開いて編集できます。スクリプトの編集と実行が簡単に行なえます。

【注意】Peggyから実行中のコマンドやスクリプトへキーボード入力を送ることはできません。ユーザからの入力を要求するスクリプトは、[実行方法]ページの[出力をアウトプットウインドウへリダイレクトする]をOFFにし、独立したコンソールウインドウ上で実行させてください。

【注意】このプロジェクト機能における「スクリプト」とは、一般的な外部プログラムのスクリプト言語を意味します。Peggyのビルトインスクリプト「MocaScript」の登録、実行とは関係ありません。

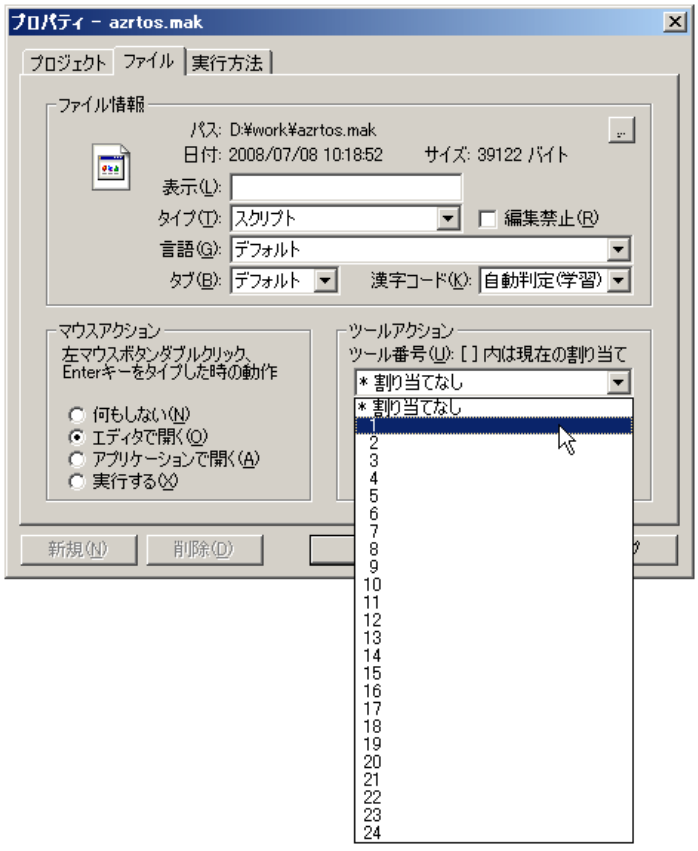
【参照】『第6章　プロジェクト機能/スクリプトファイルの実行』、『第6章　プロジェクト機能/[実行方法]プロパティページ』

プロジェクトの応用3 - ツールボタンを活用しよう

Peggyには24個のツールボタン[1]～[24]があります。これらのボタンには、「ファイルを開く」、「スクリプトを実行する」などの動作を結び付けることができます。この動作を「ツールアクション」と呼びます。いったんボタンへアクションを登録すると、あとはマウスクリック1つでコンパイルしたり、デバッガを起動できるようになります。

以下は、例としてボタン1にMakefileを実行するツールアクションを割り当てる手順です。

1. プロジェクトウインドウの Makefile を右マウスボタンでクリックして、ポップアップメニューから [プロパティ] を選択します。
2. [プロパティ] ダイアログボックスの [ファイル] ページを選択します。
3. [ツール番号] リストから、[1] 番を選んでください。
4. [ツールアクション] が [実行する] になっていることを確認します。
5. [OK] をクリックして、ダイアログボックスを閉じます。



ボタン[ ]をクリックするだけで、Makefileに従いコンパイルできるようになりました。ボタンに割り当てた機能を忘れてしまったときは、マウスを近づけることで簡単な説明をチップ表示します。

【ヒント】マウスアクションとは、プロジェクトウインドウのアイコンをダブルクリックしたときに行なう動作のことで、これもファイル毎にカスタマイズできます。

【ヒント】ボタンに割り当てた動作は、メニューの[ツール]-[ユーザ定義ツール]からも実行できます。

【ヒント】ボタン1～9、10に割り当てたツールアクションは、ショートカットキーのCtrl+1～9、Ctrl+0でも実行できます。

【参照】『第6章 プロジェクト機能/プロパティ /[ファイル]プロパティページ』、『第6章 プロジェクト機能/ツールアクションを設定する』

プロジェクトの応用４ － 整理しよう

プロジェクトにファイルを登録して作業をしていると、ファイルが必要なくなったり、最初思っていたのとは別のグループに分類したくなることがあります。そんなときは、ファイルやグループを自由に削除・移動させてください。名前順に並べ替えることもできます。

グループを作って分類	新しいグループを作って関連するものを集めましょう。
いらなくなったら削除	Deleteキーでファイルやグループを削除し、[ごみ箱]へ捨てましょう。
別の場所へ移動するには、	ファイルやグループは、ドラッグ＆ドロップでプロジェクトのどこへでも移動できます。
順番を揃えるには、	名前順、逆順、グループ内、グループ以下などの並べ替えができます。
存在しないファイルの削除	ディスク上にもう存在しないファイルを、まとめてプロジェクトから削除します。

【ヒント】プロジェクトからの削除は、プロジェクトへのファイル登録を抹消するだけなので、実際のファイルがなくなることはありません。Peggyの[ごみ箱]とWindowsの[ごみ箱]は別物です。プロジェクトの中で移動させても実際のファイルはそのまま元の場所から動きません。

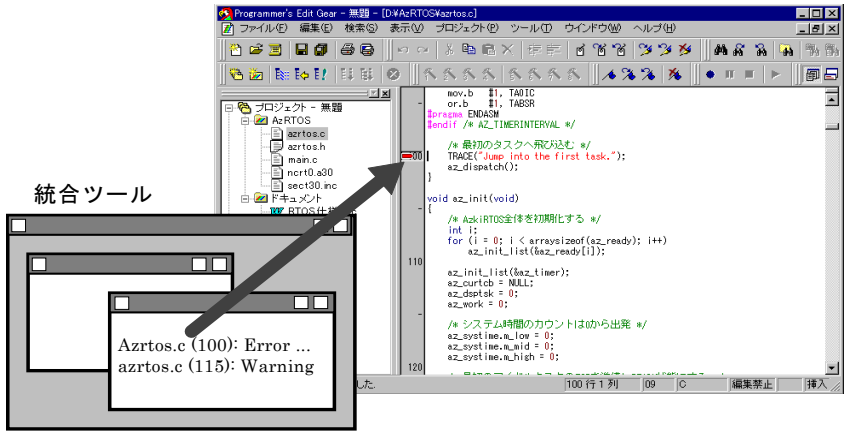
【ヒント】間違って削除してしまったら、ごみ箱からドラッグ＆ドロップで拾ってくることが可能です。

【注意】ごみ箱から削除したものは、もう戻ってきません。

【参照】『第6章 プロジェクト機能/プロジェクトにグループを作る～アイテムの名前を変更する』

統合ツールとの組み合わせ

各種統合ツールのエディタとしてPeggyを組み合わせ、連携動作させることができます。例えば、統合ツールがコンパイル中に見つけた文法エラーを修正するためにPeggyを起動し、問題のあった行へジャンプさせるような使い方です。



統合ツールと組み合わせるには、統合ツールとPeggy、それぞれに設定が必要です。

Peggy の設定

メニューの[ツール]-[オプション]で[オプション設定]ダイアログボックスを呼び出し、[全般]ページにある[二重起動を許す]チェックをOFFにします。

統合ツールの設定

エディタが必要なときにPeggyを起動するよう設定します。Peggyを起動すると同時にファイルを開いてエラー行へジャンプするには、コマンドラインへファイルパスと行番号を指定します。ファイルパスはそのまま、行番号は-Lオプションで指定します。-Lと行番号の間にスペースを入れないでください。

```
Peggy　〈ファイルパス〉　[-L〈行番号〉]
```

二重起動を禁止しているので、統合ツールがファイルパスと行番号を指定してPeggyを起動しても、いくつもPeggyが起動することはありません。いま実行中のPeggyの中でファイルを開き、ジャンプします。

【ヒント】統合ツールから呼び出される前に、予めPeggyを起動し、関連ソースファイルの登録してあるプロジェクトを開いておきましょう。そうすれば、言語が正しく設定された状態で統合ツールからジャンプできます。

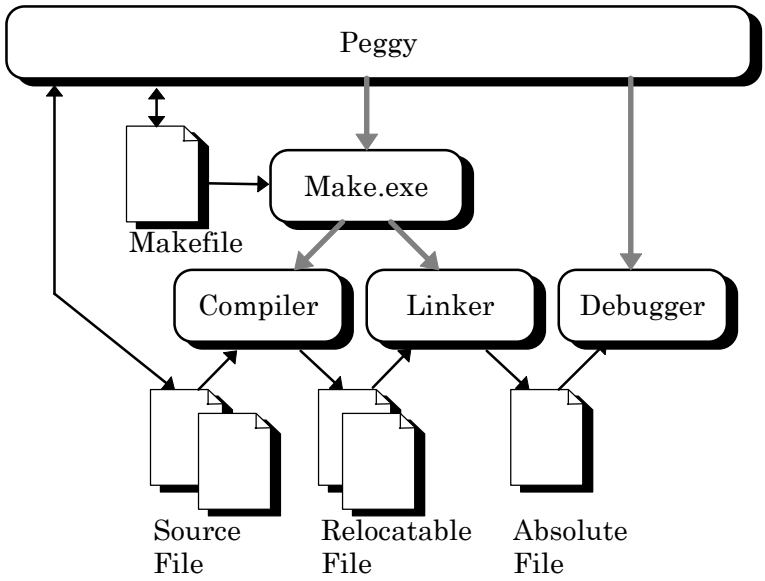
【ヒント】統合ツールから開くファイルを指定してPeggyを起動する場合、できるだけフルパスでファイルを指定するようにしましょう。より確実にファイルを開くことができます。

【注意】Peggyをインストールしたフォルダにシステムのパスが通っていない場合は、統合ツールへPeggyのフルパスを指定してください。必ずしもPeggyのインストールフォルダへパスを通す必要はありません。

【参照】『第3章　編集/コマンドライン引数』、『第11章　カスタマイズ/オプション設定/オプション[全般]ページ』

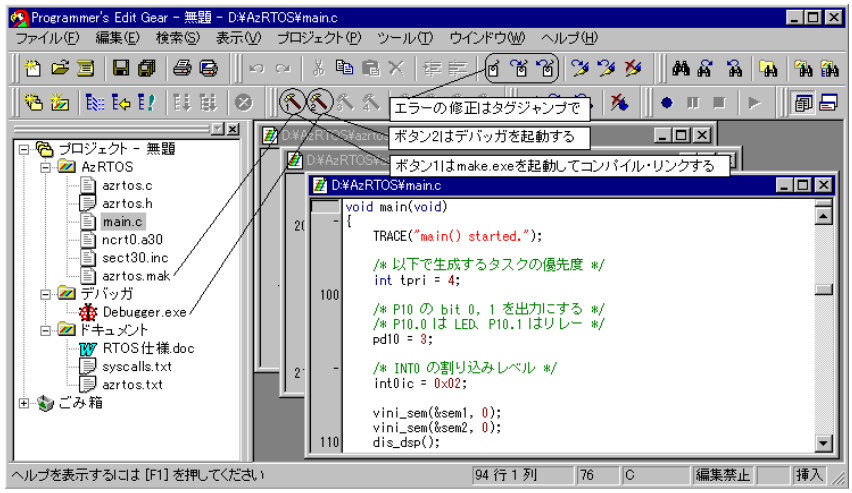
Peggy を中心とした作業環境を作ろう

Peggyのプロジェクトへソースやインクルードファイル、Makefile、デバッガの実行ファイルなどを登録することで、編集、コンパイル、デバッグがボタン一つで行なえるようになります。以下は、Peggyを中心とした作業環境を設定する一例です。



1. 開発中のソースファイルがあるフォルダで Peggy のプロジェクトを作成し、ソースやインクルードファイルなど、必要なファイルをすべて登録します。
2. コンパイル、リンクするための Makefile を作ってプロジェクトに登録します。
3. Makefile の［プロパティ］ダイアログボックスで、［実行する］アクションをツールボタン［1］へ割り当てます。
4. デバッガの実行ファイルをプロジェクトに登録します。デバッガの実行ファイルは、どこのフォルダにあってもかまいません。
5. 登録したデバッガの［プロパティ］ダイアログボックスを開き、［ファイル］ページでツールアクションをツールボタン［2］へ割り当てます。
6. 次に、［実行方法］ページの［実行フォルダ］と［引数］を設定します。一般的には、［実行フォルダ］を $h、［引数］にはコンパイル / リンクしてできるロードモジュールのファイル名やパスを指定することになると思います。$h は、プロジェクトファイルのあるフォルダを意味します。
7. ［OK］をクリックして、ダイアログボックスを閉じます。

これで、編集、コンパイル/リンク、デバッガ起動といった一連の作業をまとめることができました。ソースファイルを編集し終わったら、ボタン[ ]をクリックしてコンパイル/リンクします。エラーがあれば、F4キーでタグジャンプできます。コンパイルに成功したら、[ ]ボタンでデバッグしてください。



【ヒント】SubversionやCVSバージョン管理機能を使えば、コンパイルやデバッグだけでなくソースファイルの履歴管理とプロジェクトチームメンバ間でのソースファイル共有も可能になります。

【ヒント】$h 以外にも特別なパスやファイルを指定する記述方法があります。

【ヒント】いろいろな言語のソースファイル、インクルードファイルが混在している場合、それらを別々のグループに分けて登録しておくと、拡張子と言語を対応させやすくなります。

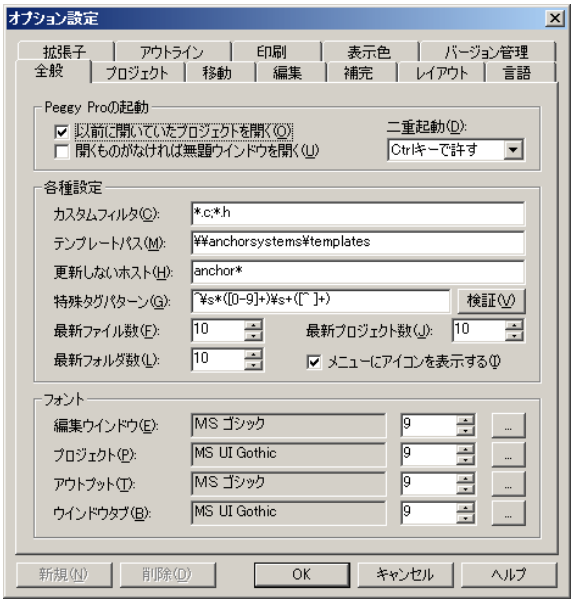
【ヒント】プロジェクトへドキュメントなども登録しておけば、簡単に参照できるようになります。

【参照】『第6章　プロジェクト機能/プロパティ /[ファイル]プロパティページ』、『第6章　プロジェクト機能/プロパティ /[実行方法]プロパティページ』

好みに合わせてカスタマイズ

Peggyを好みに合わせてカスタマイズするための豊富なオプションが用意されています。メニューから[ツール]-[オプション]を選んで、[オプション設定]ダイアログボックスを開きます。

全体の動作、プロジェクトの扱い、各種編集オプション、補完入力、言語と拡張子の関係、印刷オプション、各部分の色などを選ぶことができます。



【注意】オプション設定は、レジストリに記憶されます。Peggyを複数起動しているときは、後から終了した方が有効になり、先に終了したPeggyでのオプション変更は上書きされ、消えてしまいます。

【参照】『第11章 カスタマイズ/オプション設定/[全般]～[バージョン管理]ページ』

第 3 章　編集

このリファレンスでは、編集に関するコマンドを分類し、そのコマンド一覧とより詳細な使い方を記載しています。

ファイルメニュー

[ファイル]メニューには、以下のコマンドがあります。

コマンド一覧

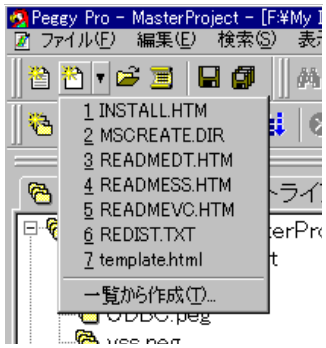
ボタン	メニュー操作	キー操作	機能
	[新規作成]-[ブランク]	Ctrl+N	新しいテキストを作成するためのウインドウが開きます。
	テンプレートから新規作成		基になるテンプレートファイルを読み込み、新しいテキストを作成します。
	[開く]	Ctrl+O	既存の文書を開きます。
	[開き直す]-[Shift JIS]		作業中のテキストをShift JISファイルとして開き直します。
	[開き直す]-[EUC]		作業中のテキストをEUCファイルとして開き直します。
	[開き直す]-[JIS]		作業中のテキストをJISファイルとして開き直します。
	[開き直す]-[UTF-7]		作業中のテキストをUTF-7ファイルとして開き直します。
	[開き直す]-[UTF-8]		作業中のテキストをUTF-8ファイルとして開き直します。
	[開き直す]-[Unicode]		作業中のテキストをUnicodeファイルとして開き直します。
	[閉じる]		現在開いているテキストを閉じます。
	[上書き保存]	Ctrl+S	作業中のテキストを上書きして保存します。
	[名前を付けて保存]		作業中のテキストに新しい名前を付けて保存します。
	[すべて保存]		開いているテキストをすべて保存します。
	[印刷]	Ctrl+P	ファイルを印刷します。
	[印刷プレビュー]		ファイルを印刷時のイメージで表示します。
	[印刷オプション]		ファイルを印刷する方法を指定します。
	[プリンタの設定]		プリンタの選択や接続を設定します。
	[ヒストリ]	Ctrl+H	過去に作業したファイルを記憶しています。
	[最近使ったファイル]		最近開いたファイルを記憶しています。

	[最近使ったフォルダ]		最近使ったフォルダを記憶しています。
	[最近使ったプロジェクト]		最近開いたプロジェクトを記憶しています。
	[URLをブラウズする]	Ctrl+B	カーソル位置にあるURLやメールアドレスに従い、アプリケーションを起動します。
	[スクリプト実行]		編集中のテキストファイルがスクリプトのとき、そのファイルを実行します。
	[関連アプリケーション起動]		ファイルに関連するアプリケーションを起動します。
	[ダイレクトオープン]	Ctrl+Shift+O	カーソル位置のパスを使ってそのファイルを開きます。
	[関連ファイルを開く]		同じフォルダにある現在編集しているファイルと関連のあるファイルを開きます。
	[インクルードファイルを開く]		編集しているファイルにインクルードされているファイルの一覧を表示し、選択して開きます。
	[Peggyの終了]	Alt+F4	Peggyを終了します。

新規作成

現在編集中のテキストはそのままで、新しいテキストを作成するためのウインドウが開きます。テキストには仮の名前がついていますので、保存するときには適当な名前に付け直してください。新規に作成したテキストの漢字コード、改行コード、タブ間隔は、[オプション設定]ダイアログボックスの[編集]ページで設定できます。初めて保存するときに、漢字コードと改行コードを指定することも可能です。

ツールバーの[新規ファイル]ボタンの右横の[▼]をクリックすると、テンプレート一覧が表示されます。メニューから選択すると、そのファイルをテンプレートにして新しいテキストを作成できます。

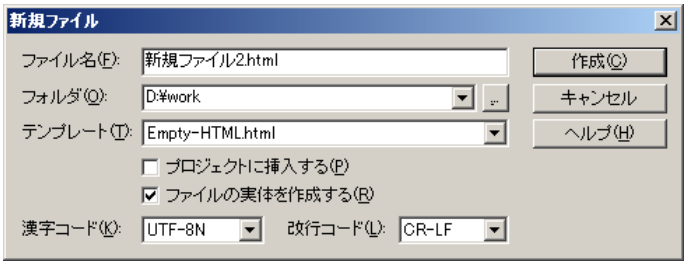


[一覧から作成]を選ぶと、[新規ファイル]ダイアログボックスが表示されます。

【メモ】新規テキストの作成で、Shiftキーを押しながらこのコマンドを実行すると、ウインドウを大きく表示することができます。また、[編集]ページの[できるだけ大きなウインドウを開く]オプションにより、常に大きく開くよう設定することも可能です。

テンプレートから新規作成

指定したテンプレートを読み込み、新しいテキストのウインドウが開きます。テンプレートの指定などは[新規ファイル]ダイアログボックスで行います。



[ファイル名]

作成するテキストのファイル名を指定します。

[フォルダ]

ファイルを作成するフォルダを指定します。

[テンプレート]

リストボックスからテンプレートを選択します。テンプレートは次の２つのフォルダに用意しておきます。

- Peggyの実行ファイルが格納されているフォルダ<Peggy>と同じレベルのフォルダ<share>の下の<template>
- [オプション設定]ダイアログボックスの[全般]ページの[テンプレートパス]で指定したフォルダ。

テンプレートを[なし]にすると、空のテキストを開きます。

[プロジェクトに挿入する]

新規作成するファイルをプロジェクトに追加します。

[ファイルの実体を作成する]

新しいテキストのウインドウを開くと同時にファイルの実体を作ります。

[漢字コード]、[改行コード]

新しいテキストの漢字コード、改行コードを指定します。

[作成]

テンプレートをベースにした新しい編集ウインドウを作成します。

ファイルを開く

プロジェクトに登録されているファイルの場合。

プロジェクトウインドウ中のアイコンを左マウスボタンでダブルクリックします。もしくは、アイコンのポップアップメニューから[～を開く]を選択します。アイテムを選択してから、Enterキーを押しても開くことができます。ただし、これらの方法は[ファイル]プロパティページで[マウスアクション]が[エディタで開く]になっていなければなりません。

プロジェクトウインドウからドラッグし、プロジェクトウインドウ以外の場所でドロップしてもファイルを開くことができます。

プロジェクトに登録していないファイルの場合。

メニューの[ファイル]-[開く]を選択すると現れる[開く]ダイアログボックスから、編集したいテキストファイルを指定します。Ctrl+Oをタイプするか、[編集]ツールバーの[]ボタンをクリックしても同じです。Windowsでは最も一般的な方法です。[開く]ダイアログボックスでは、開こうとしているファイルの漢字コードを指定することもできます。

開きたいファイルがエクスプローラで見えている場合。

ファイルをドラッグし、Peggy上へドロップします。複数のファイルを一度に開くこともできます。

最近開いたことがあるファイルの場合。

[ファイル]メニューに目的のファイルがあれば、それを選択します。

過去に開いたことがあるファイルの場合。

[ファイル]メニューの[ヒストリ]を実行すると過去に開いたファイルのリストが表示されます。目的のファイルがあれば、それを選択します。

この他にも、ファイルから検索などでファイルが必要になると、Peggyが自動的に開いたり閉じたりすることがあります。

【メモ】Shiftキーを押しながらファイルを開くと、ウインドウを大きく表示することができます。また、[編集]ページの[できるだけ大きなウインドウを開く]オプションにより、常に大きく開くよう設定することも可能です。Ctrlキーと組み合わせることもできます。

【メモ】Ctrlキーを押しながらファイルを開くと、プロジェクトに登録してあるファイルでも、以前作業していたウインドウの状態を復元ぜず、はじめて開いたかのようなウインドウとなります。Shiftキーと組み合わせることも可能です。

【参照】『第6章 プロジェクト機能/プロジェクトとは』

ファイルを開き直す

作業中のドキュメントの漢字コードが正しく自動判定されなかった場合など、[開き直す]コマンドにより、漢字コードを明示的に指定してファイルを再度開きます。指定できる漢字コードは、以下の6種類があります。

メニュー操作	漢字コード
[ファイル]-[開き直す]-[Shift JIS]	Shift JIS
[ファイル]-[開き直す]-[EUC]	EUC
[ファイル]-[開き直す]-[JIS]	JIS
[ファイル]-[開き直す]-[UTF-7]	UTF-7
[ファイル]-[開き直す]-[UTF-8]	UTF-8
[ファイル]-[開き直す]-[Unicode]	Unicode

プロジェクトに登録されているファイルをこの方法で開き直すと、プロジェクト内にユーザが明示的に指定した漢字コードが記憶され、次回からは正しい漢字コードで開けるようになります。ただし、[ファイル]プロパティの[漢字コード]が[自動判定(学習)]となっていない場合は、学習機能が働きません。

【注意】開き直そうとしているファイルを既に編集してしまっている場合、漢字コードを変えて読み直すと、いままでの変更がすべて失われます。何かを失う可能性のあるときは確認のダイアログボックスが表示されます。よく確かめてから開き直してください。

【参照】『［開く］コマンド』、『効果的な編集7 - 漢字コードと改行コード』、『第6章 プロジェクト機能/プロパティ /［ファイル］プロパティ』

ファイルを閉じる

閉じたいファイルの編集ウインドウにカーソルがあることを確認してください。もしカーソルが他のファイルのウインドウにあるときは、閉じたいファイルの編集ウインドウをマウスでクリックします。

●メニューの［ファイル］-［閉じる］を選択する。

●編集ウインドウ右上にある［ ］ボタンをクリックする。

●Ctrl+F4キーをタイプする。

●メニュー［ウインドウ］-［すべて閉じる］を選択すると、すべてのファイルを一度に閉じることができます。

●プロジェクトを使っている場合、メニュー［プロジェクト］-［プロジェクトを閉じる］を選択すると、すべてのファイルとプロジェクトを一度に閉じることができます。

ファイルを閉じるとき、ファイルが変更されていれば、保存するかどうかを問い合わせてきます。

ファイルを保存する

保存の方法は大きく分けて３通りあります。

●上書き保存

●名前を付けて保存

●すべて保存

上書き保存

現在編集中のテキストを、前回保存時と同じファイル名で保存します。前回保存してあったファイルの内容は失われます。

●メニューの［ファイル］-［上書き保存］を選択する。

●ショートカットキー Ctrl+Sをタイプする。

●［編集］ツールバーの［ ］ボタンをクリックする。

名前を付けて保存

新しく作成したテキストに、ファイル名をつけて保存します。保存されていたテキストの場合は、別のファイル名を付けたり、別の場所に保存するときに使います。漢字コードや改行コードを変更して保存することも可能です。

●メニューの［ファイル］-［名前を付けて保存］を選択する。

●［ ］ボタンをクリックする。

以下は、［名前を付けて保存］ダイアログボックスの説明です。

[保存する場所]

ファイルを保存するフォルダを指定します。

[ファイル名]

保存するファイル名を指定します。

[ファイルの種類]

ファイルの種類を指定します。

[漢字コード]

保存するテキストの漢字コードを選択します。漢字コードを変更して保存したい場合は、ここから指定できます。Shift JIS、EUC、JIS、UTF-7、UTF-8、UTF-8N、Unicode が選択できます。

[改行コード]

改行コードを変更して保存したい場合は、新しい改行コードを選択してください。

選択	改行コード
[そのまま]	ファイルの各行の改行コードを一切変更せず、そのまま出力します。
[CR-LF]	すべての行の改行コードをCR-LFに変更して保存します。
[LF]	すべての行の改行コードをLFに変更して保存します。
[CR]	すべての行の改行コードをCRに変更して保存します。

[強制保存]

選択したファイルが書き込み禁止であっても、それを解除して強制的に上書きします。

すべて保存

開いているすべてのテキストを上書き保存します。前回保存してあったファイルの内容は失われます。

●メニューの[ファイル]-[すべて保存]を選択する。

●[編集]ツールバーの[]ボタンをクリックする。

この他にもPeggyを終了するときや、ファイルから検索を始めるとき、プロジェクトからの[実行]を選択したときに変更されたファイルがあると、保存するかどうかを問い合わせてきます。

【注意】ファイルを保存するとき、Peggyはまず編集内容をテンポラリファイルへ保存してから、オリジナルのファイル名へリネームします。従って保存しようとしているファイルと同じか、それ以上の空きディスクスペースがないと保存できません。参考までにファイルを保存するときの手順を示します。

1. 編集中のテキストをテンポラリファイルへ保存します。
2. オリジナルのテキストファイルをバックアップファイルの名前に変更します。(オリジナルの名前 +.bak)
3. テンポラリファイルを正規の名前に変更します。
4. バックアップファイルを残さない場合は、バックアップファイルを削除します。

Peggy の終了

すべての編集ウインドウを閉じて、Peggyを終了します。編集を加えたテキストがある場合は、保存するかどうかを確認するメッセージが表示されます。

●メニューの[ファイル]-[Peggyの終了]を選択する。

●ショートカットキー Alt+F4をタイプする。

●Peggyのウインドウ右上にある[×]ボタンをクリックする。

編集ファイルの排他制御

ネットワーク環境などでお使いの場合、現在編集中のファイルを他人が同時に開き、編集結果が保存されていないというような事故を防ぎます。

[ファイル]-[ツール]-[オプション]-[編集]ページの[編集オプション]リストにある、[開いている間はファイルを書き込み禁止にする]をチェックすることで、編集している間ファイルに書き込み禁止属性を付け、他からの不用意な書き込みを防止することができます。

その逆に、[ファイル]-[開く]-[書き込み禁止ファイルは編集禁止にする]にチェックをすることで、開いたファイルが書き込み禁止属性のファイルであったり、外部で使われていて書き込み禁止属性が設定されているときに、編集できなくすることができます。またこのとき、[表示色]ページの[基本 編集禁止ウインドウの背景]を設定しておけば、書き込み禁止ファイルを開いた瞬間にエディタ全体の背景色が変化しますから、書き込み禁止ファイルであることがわかります。

【ヒント】チームでファイルを編集しているときは、バージョン管理システムを使った共同作業を推奨します。書き込み禁止によるファイルの保護より安全かつ効率的に作業が進められ、個々のファイルの変更履歴も管理できます。Peggyは、Subversion/CVS/RCSやVSSバージョン管理システムのユーザインターフェイスを備えているので、高度な機能が簡単に利用できます。

【ヒント】Peggyの排他制御は、ファイルに書き込み禁止属性を付ける方法により実現しています。それ以上の強いアクセス制御(ファイルのロック、他からのアクセスを禁止してファイルを開いたままにする...など)は採用していません。従って、エクスプローラなどで意識的に書き込み禁止属性を解除すれば、排他制御を破ることが可能です。しかし場合によってはアクセスしたいこともあり、敢えて強い排他制御は採用していません。

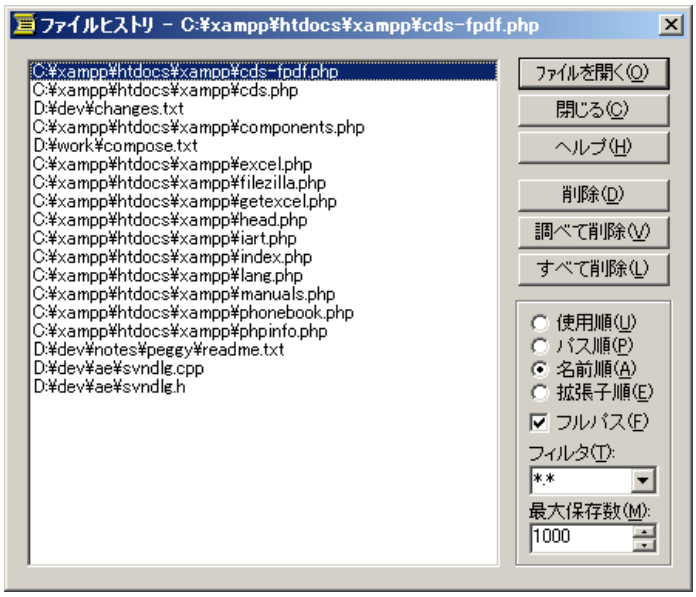
【注意】書き込み禁止属性をON/OFFする他のアプリケーションとPeggyの排他制御を組み合わせて使うと、お互いが書き込み禁止属性をON/OFFするため混乱を生じることがあります。

ファイルヒストリ

以前に開いたファイルを再度開きたい場合には、[ファイル]メニューの[最近使ったファイル]、[最近使ったプロジェクト]から選択するのが最も簡単な方法です。しかし、そこに表示できるファイルの数が16個までという制限のため、それ以前のファイルを開くには、[開く]ダイアログボックスでフォルダを上がったり下がったりしなければならず面倒です。

そのようなときにはファイルヒストリを使います。ファイルヒストリは、最近開いたかなりの数のファイルを覚えているので、ヒストリリストからファイルを選んで

簡単に開くことができます。ファイルヒストリは、メニューの[ファイル]-[ヒストリ]か、Ctrl+Hで呼び出せます。



[ファイルヒストリ]ダイアログボックスが現れたら、リストから開きたいファイルを選択し、[ファイルを開く]ボタンをクリックするか、リスト中のそのファイルをダブルクリックしてください。

【注意】フルパス表示しているときにファイルパス名が長くなりすぎてダイアログボックスにすべて表示できないときは、[フルパス]のチェックをはずしてください。その際、[名前順]でソートされているときは表示順序が替わります。

[ファイルを開く]

リストで選択しているファイルを開きます。

[閉じる]

このダイアログボックスを閉じます。

[削除]

リストで選択しているファイルをヒストリから削除します。

[調べて削除]

ヒストリ中のファイルがディスク上に存在しているかどうか調べ、存在しないファイルをヒストリから削除します。

[すべて削除]

すべてのファイルをヒストリから削除します。

[使用順]

ファイルを開いた順に並べてリストに表示します。最後に開いたファイルが一番上になります。

[パス順]

ファイルをフルパス順に並べてリストへ表示します。

[名前順]

ファイルを名前順に並べてリストへ表示します。

[拡張子順]

ファイルを拡張子順に並べてリストへ表示します。

[フルパス]

ファイルをフルパスで表示します。チェックを外すとファイル名だけになります。

[フィルタ]

ワイルドカードを含んだファイル名のパターンを指定し、一致したものだけを表示します。空白で区切って最大10個までのフィルタを同時に指定することができます。例: *.h *.c

[最大保存数]

ファイルヒストリに保存するファイルの最大数を指定します。この数より多くなると、古いものから順に削除します。デフォルトは、100です。

ダブルクリック

リスト中のファイル上をマウスの左ボタンでダブルクリックすると、そのファイルを開いて、[ファイルヒストリ]ダイアログボックスを閉じます。

【ヒント】間違ってすべてを削除してしまったような場合、Escキーか、ダイアログボックスのタイトルバー右端にある[×]ボタンを使って閉じてください。ファイルヒストリの状態を元へ戻すことができます。

最近使ったファイル、最近使ったフォルダ、最近使ったプロジェクト

[最近使ったファイル]コマンドは、直近に開いたファイルを最大16個まで記憶しています。選択すると、すぐにそのファイルを開くことができます。

[最近使ったフォルダ]コマンドは、直近に開いたファイルのフォルダを最大16個まで記憶しています。選択すると、そのフォルダから素早くファイルを選択して開くことができます。

[最近使ったプロジェクト]コマンドは、直近に開いたプロジェクトファイルを最大16個まで記憶しています。選択すると、すぐにそのプロジェクトを開くことができます。

記憶する数は[オプション設定]ダイアログボックスの[全般]ページの、[最新ファイル数] 、[最新フォルダ数]と[最新プロジェクト数]でそれぞれ指定します。

【注意】プロジェクトに登録されているファイルを開いた場合、[プロジェクト]ページの[最新ファイルリストへ追加する]がOFFになっていると、このリストへは追加されません。

URL をブラウズする

カーソル位置にある文字列がURLやメールアドレスの場合、関連するアプリケーションを起動します。[ ]ボタンをクリックしてもブラウズできます。

スクリプト実行

編集中のテキストファイルがスクリプトのとき、そのファイルを実行します。ポップアップメニューや、[ ]ボタンをクリックしても実行できます。

【参照】『第2章 Peggyの使い方/外部スクリプトを実行させよう』

関連アプリケーション起動

ファイルがあるアプリケーションのデータファイルならば、そのデータファイルに関連しているアプリケーションを起動します。ポップアップメニューや、[]ボタンをクリックしても起動できます。

【参照】『第2章 Peggyの使い方/テキスト以外も登録しよう』

ダイレクトオープン

編集ウインドウのカーソル位置にある文字列を解析し、それがファイルパスであれば開きます。ポップアップメニューの[...を開く]を選択するか、[]ボタンをクリックしてください。

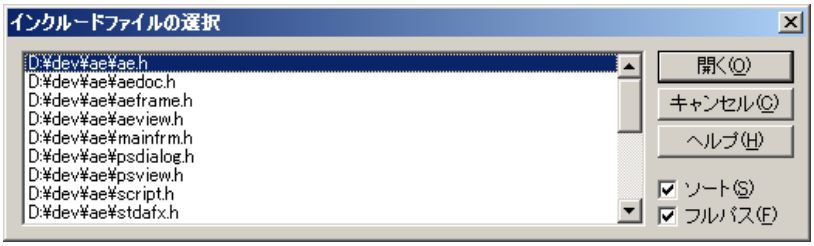
関連ファイルを開く

現在編集しているファイルと同じフォルダにある「関連ファイル」を開きます。関連ファイルとは、同じ名前で拡張子だけが異なっているファイルです。例えば<foo.c>という名前のC言語のソースファイルを編集しているとき、このコマンドを実行するとヘッダーファイル<foo.h>を開きます。[]ボタンでも開くことができます。

拡張子の種類は、言語モードにより決まっています。その言語でどの拡張子を関連ファイルと判定するかは、言語リスト<langlist.pdf>ファイルを参照してください。関連する拡張子が3つ以上ある場合は、それらを順番に探し、最初に見つけたファイルを開きます。

インクルードファイルを開く

編集しているファイルにインクルードされているファイルの一覧を表示し、そこから1つのファイルを選択して開きます。[]ボタンからでも実行できます。



インクルードファイルのリストに表示されるのは、実際にファイルが存在することが確認できたものだけです。ファイルインクルードの記述があっても実際にファイルが見つからなければ、リストに表示されません。

インクルードファイルを見つけるためにパスを設定する必要がある場合は、プロジェクトの[プロパティ]ページにある[ファイル検索パス]を設定します。

[開く]

リスト中で選択されているインクルードファイルを開きます。リスト中のファイルパスをダブルクリックして開くこともできます。

[ソート]

リスト中のインクルードファイルをソートして表示します。OFFにすると、元の編集中のファイルに記述されている順番で表示します。

[フルパス]

インクルードファイルのパスをフルパスで表示します。

印刷

Peggyには、ソースファイルの印刷に便利な機能が数多く備わっています。Peggyがサポートしているプログラミング言語のテキストを印刷するときは、その言語の文法を理解して印刷するので、例えばキーワードはボールド体で、コメントはイタリック体でといった指定が可能です。それ以外にも、C/C++言語の関数がページをまたがないように見やすくしたり、段組印刷、両面印刷など紙を節約できるオプションもあります。

ボタン	メニュー操作/機能	キー操作
	[ファイル]-[印刷]	Ctrl+P
	[ファイル]-[印刷プレビュー]	
	[ファイル]-[印刷オプション]	
	[ファイル]-[プリンタの設定]	

印刷

部分印刷

[印刷]ダイアログボックスの[印刷範囲]を指定すると、10ページから20ページというようにページ単位で部分印刷を行うことができます。しかし、テキストファイルにはもともと「ページ」という概念がありません。例えば、5ページ目に何行目から何行目までが印刷されるかは、やってみなければ分かりません。

テキストの特定の部分だけを印刷したいときは、前もってその部分を選択してから[印刷]ダイアログボックスを呼び出します。そうすると[印刷範囲]の[選択した部分]がデフォルトで選ばれ、そのまま[OK]ボタンをクリックすれば、選択範囲だけを部分印刷できます。

ダイレクト印刷

[印刷]ダイアログボックスを呼び出さずに、直接プリンタへテキストを送る印刷方法です。テキストの一部分が選択されていればその選択範囲だけを、選択されていなければ全体を印刷します。ダイレクト印刷ではコピー数を1に戻すので、常に1部しか印刷しません。ソースコードの一部分を仮に印刷したいときなどに便利です。

【注意】ダイレクト印刷はメニューにないので、ショートカットキーから使います。しかし、デフォルトではショートカットキーも割り当てられていません。まず、メニューの[ツール]-[ショートカット]で[ショートカット]ダイアログボックスを呼び出し、[ファイル-ダイレクト印刷]コマンドに適当なキーを割り当ててください。

印刷プレビュー

印刷プレビューとは、紙に印刷するイメージを前もってスクリーン上で確認する機能です。印刷プレビューで印刷イメージを確かめてから実際の印刷を行うことで、失敗を防ぐことができます。

印刷オプション

印刷オプションの設定でまず最初に理解しておかなければならないのは、言語モード毎に印刷オプションを設定できるということです。つまり、C言語、HTML、Delphi、Perl、テキストなど、それぞれに異なった指定ができることを意味しています。[オプション設定]ダイアログボックスから各言語の印刷オプションを設定できます。

【参照】『第11章 カスタマイズ/オプション設定/オプション[印刷]』

プリンタの設定

Peggyから印刷を行う場合のプリンタのデフォルトを設定/変更します。デフォルトのプリンタ、用紙サイズなどが選べます。Windowsの標準的なプリンタ設定です。

編集メニュー

[編集]メニューには、以下のコマンドがあります。

ボタン	メニュー操作	キー操作	機能
	[元に戻す]	Ctrl+Z	操作を間違えたときに、前の状態に戻します。
	[やり直し]	Ctrl+Y	[元に戻す]で戻り過ぎてしまった部分をやり直します。
	再変換	Ctrl+X	確定している日本語文字列に対して再変換を開始します。
	[切り取り]	Ctrl+X Shift+Delete	選択した範囲をテキストから削除し、クリップボードに保存します。
	[コピー]	Ctrl+C/ Ctrl+Insert	選択した範囲をクリップボードにコピーします。
	[貼り付け]	Ctrl+V/ Shift+Insert	クリップボードの内容をカーソル位置に挿入します。範囲が選択されている場合は、その範囲がクリップボードに保存されている内容と置き換えられます。
	[削除]	Delete	選択した文字列を削除します。
	[すべてを選択]	Ctrl+A	編集中のテキスト全体を選択します。
	[挿入と削除]		制御コード、カラーコード、日付などいろいろな文字列を挿入削除します。

	[コメント]		選択した範囲を言語モードに従いコメントにしたり、コメントを取り除いたりします。
	[インデント]		インデントや文字の位置を揃えます。
	[HTML]		HTMLタグの前後や削除、エンティティの変換などを行います。
	[文字変換]		大文字、小文字の変換およびタブ、スペースの変換をします。
	[文字揃え]		選択した範囲の右端などを揃えます。
	[補完入力]		長い単語や決まり文句を何度も入力するとき、タイピング量を軽減する機能です。
	[行順序]		選択範囲の行をソートしたり順序を逆転させます。
	[指定行ジャンプ]	Ctrl+G	行番号がわかっている行へジャンプするときに使います。
	[対応括弧ジャンプ]	Ctrl+]	カーソルが括弧の前後にあるとき、対応する括弧へジャンプします。
	[タグジャンプ]	F4/Shift+F4/Ctrl+T	ファイル名と行番号の情報を元に、ファイルを自動的に開き、指定された行にジャンプします。プロジェクト機能を使えば、コンパイラのエラー情報を元にジャンプすることもできます。
	[変更行ジャンプ]	Ctrl+Shift+8/Ctrl+Shift+9	変更が加えられた行には変更行マークが付き、そのマーク行へ順にジャンプすることができます。
	編集禁止モード切り替え		編集禁止モードをON/OFFします。
	[プロパティ]	Alt+Enter	プロパティダイアログボックスを表示します。プロパティの編集ができます。

カーソル移動

カーソルを移動するには、進めたい方向の矢印キーを使います。今カーソルが何処にあるかは、ステータスバーの[行列インジケータ]に、カーソルのすぐ右の文字コードが[文字コードインジケータ]へ常に表示されています。

キー操作	機能
←	1文字左へ
→	1文字右へ
↑	1行上へ

↓	1行下へ
Shift+←	選択範囲を1文字左へ
Shift+→	選択範囲を1文字右へ
Shift+↑	選択範囲を1行上へ
Shift+↓	選択範囲を1行下へ
Ctrl+←	1単語左へ
Ctrl+→	1単語右へ
Ctrl+Shift+←	選択範囲を1単語左へ
Ctrl+Shift+→	選択範囲を1単語右へ
	左の単語境界へ
	右の単語境界へ
	選択範囲を左の単語境界へ
	選択範囲を右の単語境界へ
Home	行の先頭へ※
End	行の最後へ※
Shift+Home	選択範囲を行の先頭へ※
Shift+End	選択範囲を行の最後へ※
PageUp	1ページ上へ
PageDown	1ページ下へ
Shift+PageUp	選択範囲を1ページ上へ
Shift+PageDown	選択範囲を1ページ下へ
Ctrl+Alt+PageUp	2つのウインドウを同時に1ページ上へ
Ctrl+Alt+PageDown	2つのウインドウを同時に1ページ下へ
Ctrl+Home	テキストの先頭へ
Ctrl+End	テキストの最後へ
Ctrl+Shift+Home	選択範囲をテキストの先頭へ
Ctrl+Shift+End	選択範囲をテキストの最後へ
Ctrl+Shift+A	アンカー /カレント入れ替え

【ヒント】※のコマンドは、行番号のモードがファイル行単位か表示行単位かによって、ファイル行または表示行を基準にしてカーソルを移動させます。

【ヒント】アンカー /カレント入れ替えは、選択範囲の初めと終わりを交換します。

【ヒント】[編集]ページのオプション設定により、Page Up、Page Downキーの動作を半ページ単位に行うよう選択することもできます。

スクロール

カーソルを移動させずに画面がスクロールします。

キー操作	機能
Ctrl+PageUp	左スクロール
Ctrl+PageDown	右スクロール
Ctrl+↓	スクロールアップ
Ctrl+↑	スクロールダウン
Ctrl+Alt+→	2ウインドウ同時に右スクロール
Ctrl+Alt+←	2ウインドウ同時に左スクロール
Ctrl+Alt+↓	2ウインドウ同時にスクロールアップ
Ctrl+Alt+↑	2ウインドウ同時にスクロールダウン
Ctrl+Shift+↓	高速スクロールアップ
Ctrl+Shift+↑	高速スクロールダウン

クイックスクロール機能

マウスで縦スクロールバーの上下端にある矢印部分をクリックしスクロールさせているとき、Ctrlキー、Shiftキー、または両方を押し下げることにより、それぞれ2、4、6倍のスピードでスクロールします。Ctrl+↑、↓キーを押してスクロールさせている場合でも、Shiftキーを押すことで約4倍のスピードでスクロールします。

インテリマウス

Peggyでは、ホイールの付いたインテリマウスをサポートしています。インテリマウスを使うことで、テキストのスクロールをより簡単に行うことができます。以下は、インテリマウスでサポートしている機能一覧です。

ホイール回転

[オプション設定]ダイアログボックスの[編集]ページの[スクロール行数]で指定してある行数だけ上下にスムーススクロールします。デフォルトは、5行です。

Ctrl+ホイール回転

半ページ単位で上下にスムーススクロールします。

ホイールを押し下げたまま回転

ホイールの回転量に応じたスピードで連続上下スクロールします。

Shift+ホイール回転

Peggy内のウインドウを順番にアクティブにします。アウトプットウインドウにカーソルがあるときは、アウトプットウインドウのページを順番に切り替えます。

スムーススクロールを止める

スムーススクロールを止めたい場合は、[オプション設定]ダイアログボックスの[編集]ページにある[編集オプション]リストの[インテリマウススムーススクロール]のチェックを外してください。マウスホイールの回転で、指定行数だけジャンプスクロールするようになります。

【ヒント】ここで説明したマウスホイールの機能は、編集ウインドウ以外にも、アウトプットウインドウやプロジェクトウインドウでも有効です。

選択

複数の文字やある範囲のテキストに対して削除、移動、コピーなどの操作を行うには、あらかじめ操作したい範囲を「選択」してから行います。以下、操作の対象として選択した部分を「選択範囲」と呼ぶことにします。選択範囲は画面上で反転表示されます。

メニュー操作/機能	キー操作
文字選択モード	F8
矩形選択モード	Ctrl+Shift+F8
カーソル位置の単語を選択	Shift+F8
カーソル位置の行全体を選択	Ctrl+F8
括弧内部の選択	
括弧と内部の選択	
[編集]-[すべてを選択]	Ctrl+A

選択・行選択・矩形選択

テキスト上のある部分を選択するには、マウスによる方法と、キーボードによる方法、それにメニューによる方法があります。マウスとキーボードを併用してもかまいません。選択範囲を解除するには、マウスを別の場所でクリックするか、Escキーを押してください。反転表示が消え、カーソルが点滅します。

マウスによる選択

●選択したい範囲の始めで左マウスボタンを押し下げ、そのまま終点までドラッグしてからボタンを離します。

●マーカー領域で左マウスボタンを押し下げ、そのまま上下に移動させると、行単位で選択できます。

●マーカー領域を左マウスボタンでダブルクリックすると、ファイル行全体を選択します。また、そのままマウスを移動させると、ファイル行単位で選択範囲を調整します。

●数字、英単語、C/C++/アセンブリ言語のシンボル上をダブルクリックすると、そのシンボルを選択します(※1)。また、そのままマウスを移動させると、単語単位で選択範囲を調整できます。

●日本語文字列上でダブルクリックすると、漢字またはカタカナとそれに続くひらがな部分を選択します(※2)。

●スペースやタブなどの空白文字上をダブルクリックすると、空白部分全体を選択します。

●括弧の上をダブルクリックすると、括弧を含んでその内側を選択します。また、括弧のすぐ内側をダブルクリックすると、括弧を含まずに内側の文字だけを選択します。この方法で選択できる括弧は、()、[]、{}、「」、〈〉、（）、｛｝、《》、【】、〔〕、［］、「」、『』、“”です。なお、HTMLとJavaScriptのように複数言語が混在している言語の場合や、コメントや文字列の中の括弧なども文法を認識しているので、正しく対応する括弧の内側を選択します。

●引用符のすぐ内側をダブルクリックすると、引用符の内側の文字を選択できます。

●トリプルクリックをするとカーソル位置のファイル行を選択します。また、トリプルクリックでそのままドラッグする３回目のクリックを押し下げたままマウスを動かす)と、ファイル行単位で選択範囲を上下させることができます※3)

●Altキーを押しながら左マウスボタンで範囲指定を行うと、矩形選択ができます。矩形選択を使うと、四角い部分をクリップボードへコピーしたり、削除や移動が行えます。矩形選択したテキストのカット&ペーストは、Microsoft Visual C++や秀丸エディタとの間でも可能です。矩形選択中は、ステータスバーの[上書きモードインジケータ]に[矩形]と表示されています。

【※1】[オプション設定]ダイアログボックスの[言語]ページの[単語の一部と判定する記号文字]を指定すると、その文字も含めて単語やシンボルを選択します。

【※2】[オプション設定]ダイアログボックスの[編集]ページ、[編集オプション]の[ダブルクリックで日本語文字列も選択する]がONになっている必要があります。

【※3】[オプション設定]ダイアログボックスの[編集]ページ、[編集オプション]の[トリプルクリックでファイル行を選択する]がONになっている必要があります。

キーボードによる選択

●Shiftキーを押しながらカーソルを移動させると、移動した部分が選択範囲になります。

●Ctrl+Shift+F8をタイプした後、Shiftキーを押しながらカーソルを移動させると、矩形領域の選択もできます。

メニューによる選択

●メニューの[編集]-[すべてを選択]を選ぶと、テキスト全体を一度に選択することができます。

編集

文字の入力

文字をタイプすると、そのままカーソル位置に挿入されます。ただし上書きモードになっていると、カーソル位置の文字と置き換えます。挿入モードと上書きモードの切り替えは、Insertキーで行ってください。また、[オプション設定]ダイアログボックスの[編集]ページの設定により、挿入モードか上書きモードに固定することも可能です。ステータスバーの[上書きモードインジケータ]で現在のモードを確認できます。

テキストに変更を加えると、ファイル名の右端に[変更インジケータ]のアスタリスク(*)が付きます。

書き込み禁止のファイルを開くと、編集禁止状態となり、文字を入力できません。書き込み禁止のファイルの見分け方は、

●ステータスバーの[編集禁止インジケータ]が点灯している。

●編集ウインドウの色が違う。これは、[オプション設定]ダイアログボックスの[表示色]ページで、[基本-編集禁止ウインドウの背景]が有効で、[基本-背景]と違う色が指定されている場合に限ります。

編集禁止状態では文字入力だけでなく、テキストを変更するすべての操作が禁止されます。

書き込み禁止のファイルを編集したい場合は、メニューの[編集]-[プロパティ]を選択するか、Alt+Enterキーをタイプすると現れる[プロパティ]ダイアログボックスの[編集禁止]チェックを外してください。ただし、編集したテキストを別の名前で保存しなければなりません。

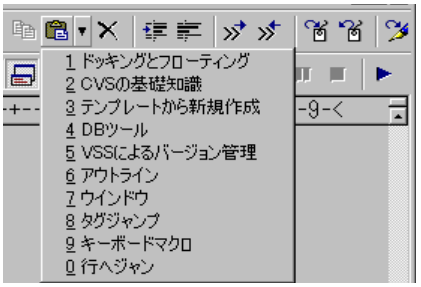
切り取り・コピー・貼り付け

クリップボードを経由して、他のテキストや、Microsoft Wordなど別のアプリケーションとデータをやりとりします。

ボタン	メニュー操作/機能	キー操作
	[編集]-[切り取り]	Ctrl+X/Shift+Delete
	切り取って追加	Ctrl+Shift+X
	単語切り取り	
	単語を切り取って追加	
	行切り取り※	
	ファイル行切り取り	
	表示行切り取り	
	行を切り取って追加※	
	ファイル行を切り取って追加	
	表示行を切り取って追加	
	[編集]-[コピー]	Ctrl+C/Ctrl+Insert
	コピーして追加	Ctrl+Shift+C
	単語コピー	
	単語をコピーして追加	
	行コピー※	
	ファイル行コピー	
	表示行コピー	
	行をコピーして追加※	
	ファイル行をコピーして追加	
	表示行をコピーして追加	
	[編集]-[貼り付け]	Ctrl+V/Shift+Insert
	履歴から貼り付け	
	行として貼り付け	Ctrl+Shift+V

[切り取り]は、選択範囲をテキストから削除し、クリップボードに移動します。[コピー]は、選択範囲をクリップボードにコピーします。[貼り付け]は、クリップ

ボードの内容をカーソルの位置に挿入します。範囲が選択されている場合は、その範囲がクリップボードに保存されている内容と置き換えられます。ツールバーの[貼り付け]ボタンの右横の[▼]をクリックすると、クリップボードに入った文字列の履歴(最大10個)が表示されます。選択すると貼り付けることができます。



[行として貼り付け]コマンドは、カーソル位置が行の途中にあっても適当に改行を追加して、貼り付けたテキストが1つの行となるように調整します。

【メモ】行切り取り、行を切り取って追加、行コピー、行をコピーして追加など※の付いたコマンドは、行番号のモードがファイル行か表示行かによって、ファイル行、表示行を単位として動作します。また、ファイル行、表示行を明示的に扱うコマンドも別に用意してあります。

【ヒント】Ctrl+Shift+X、Ctrl+Shift+Cのようにクリップボードへ選択範囲を追加するコマンドは、テキストのあちこちに散らばっている文字列を集めたいときに便利です。

【ヒント】他のWindowsアプリケーションとの間でスムースなデータ交換をするため、カット、コピーでは改行コードがCR-LFとなるよう必要に応じて変換してからクリップボードへ登録します。貼り付けでは、クリップボードから取り出したテキストデータの改行コードを、編集中のテキストの改行コードに変換してから貼り付けます。これによりWindows以外のテキストファイルでもクリップボード経由でのデータ交換が可能になりました。この改行コード自動変換は、[オプション設定]ダイアログボックスの[編集]ページでON/OFFを切り替えられます。

【ヒント】カット、コピー、貼り付けでクリップボードとのデータ交換に使う漢字コードも、常にShift JISです。従って、編集中のEUCやJIS漢字のファイルから他のWindowsアプリケーションと漢字を含んだデータ交換が可能です。

ドラッグ & ドロップ

マウスのドラッグ&ドロップ操作を使って、選択範囲を直接移動したりコピーすることも可能です。

ドラッグ & ドロップによる移動

マウスを使って選択範囲を別の所へ移動させるには、

1. ドラッグ & ドロップできる選択範囲の上では、マウスの形が矢印に変わります。選択範囲の上を左マウスボタンでドラッグしてください。
2. ドラッグ中はドロップする場所を示す「ドロップポイントマーカー」が表示されます。ドロップポイントマーカーを移動させたい個所へ合わせてください。ドロップ先が画面の外にあるときは、ウインドウの上下左右端にマウスを近づけるとスクロールします。
3. 左マウスボタンを離してテキストをドロップしてください。

同じ編集ウインドウ内のテキストに限らず、別のウインドウ、ドラッグ&ドロップをサポートしている別のアプリケーション間でも移動できます。逆に、他のアプリケーションからドラッグ&ドロップで文字列を持ってくることも可能です。矩形選択範囲もドラッグ&ドロップで移動、コピーできます。

ドラッグ & ドロップによるコピー

ドラッグ&ドロップで文字列をコピーするには、Ctrlキーを押しながらドロップしてください。Ctrlキーを押し下げると、コピーであることを示すためマウスポインタに[+]印が付きます。

ドラッグ & ドロップの中止

ドラッグ&ドロップを中止するには、ドロップできないところでマウスボタンを離すか、Escキーを押します。

【ヒント】ドラッグ&ドロップで他のアプリケーションとテキストデータの移動やコピーを行う場合、編集しているテキストファイルの漢字コードや改行コードにかかわらず、常にShift JIS、CR-LFに変換してデータ交換を行います。これにより他のWindowsアプリケーションとスムースにデータの共有が可能になりました。

【注意】編集禁止のテキスト上にはドロップできません。マウスがドロップできない個所を指しているときは、マウスが駐車禁止のような形に変化します。反対に、編集禁止テキストからのドラッグ&ドロップは、Ctrlキーを押していなくてもコピーとして扱います。

取り消しとやり直し

間違った編集操作を取り消します。または再実行します。

ボタン	メニュー操作/機能	キー操作
	[編集]-[元に戻す]	Ctrl+Z/Alt+Backspace
	[編集]-[やり直し]	Ctrl+Y

間違った操作をしたときは、メニューから[編集]-[元に戻す]を選択してください。

前の状態に戻ります。[編集]ツールバーの[]ボタンか、Ctrl+Zをタイプしても同じです。変更の規模にもよりますが、最大で10,000回程度まで前に戻ることができます。

戻り過ぎてしまった場合は、メニューから[編集]-[やり直し]を選択します。[やり直し]は、[元に戻す]のちょうど逆の操作です。戻り過ぎてしまった部分を先に進めることができます。[編集]ツールバーの[]ボタンか、Ctrl+Yをタイプしても同じです。

[元に戻す]と[やり直し]は、文字の入力だけでなくあらゆる編集操作に対しても機能します。間違ったらとにかくCtrl+Zをタイプしてみましょう。

[]、[]ボタンの横にある[▼]をクリックすると、編集操作の履歴メニューが表示されます。このメニューから選択することで、複数の操作をまとめて元に戻したり、やり直ししたりできます。

また、メニューの最後の[編集操作リスト]を選択すると、[編集操作リスト]ダイアログボックスが表示されます。このダイアログボックスは、前後の編集操作を最大

1000個表示します。任意の状態を選択して[戻る・進む]ボタンをクリックすると、そのときの状態に戻ります、または進みます。

再変換コマンド

すでに確定している日本語文字列に対して、再度仮名漢字変換を開始します。選択範囲があるときは、その選択範囲を再変換します。選択範囲がないときは、カーソル位置の文節(※メモ参照)に対して再変換を開始します。

再変換を中止するには、編集ウインドウの違う場所をクリックしてください。

【メモ】ここで言う文節とは、漢字・カタカナの後にひらかなが続いた部分のことです。文法上の文節とは異なる場合があります。

【注意】複数ファイル行を同時に再変換することはできません。選択範囲を再変換するときは、1行以内で指定してください。

挿入・削除

行挿入

カーソル行の前後に新しい行を挿入します。

メニュー操作/機能	キー操作
上に空白行挿入	Ctrl+Shift+N
下に空白行挿入	
カーソル行を複製して下に挿入	

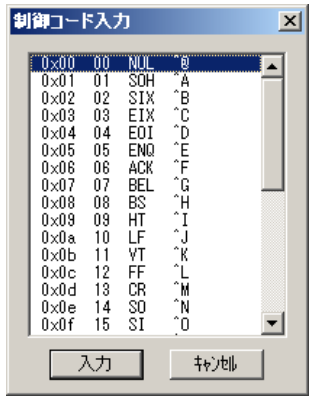
制御コード挿入

キーボードから直接入力できない制御コードを入力します。ASCIIコードの0x00から0x1fまでと0x7fの文字コードです。制御コードにキャリッジリターン、ラインフィードやタブコードなども含まれます。

ボタン	メニュー操作/機能	キー操作
^C	[編集]-[挿入と削除]-[制御コード挿入]	Ctrl+Q

文字入力するには、制御コード入力ダイアログボックスのリストから目的のコードを指定して[入力]ボタンを押すか、目的のコードを左マウスボタンでダブルクリックすることで入力できます。また対応する英文字や記号文字をタイプしても入力できます。

【メモ】ただし、タブコードに関しては、通常のキーボード上のTabキーで入力されます。また、^I、^J、^M、に関しては対応文字が表示されるのではなく表示上はそれぞれ、タブ、ラインフィード、キャリッジリターンが機能します。



【例】制御コード0x00は、^@ が対応文字であり、エディタ上は、オレンジ色の背景色表示になります。

日付と時刻の挿入

カーソル位置に今の時刻や日付を挿入します。

メニュー操作	キー動作	機能
[編集]-[挿入と削除] -[日付]	Ctrl+D	カーソル位置に今日の日付を挿入します。
[編集]-[挿入と削除] -[時間]		カーソル位置に現在の時間を挿入します。
[編集]-[挿入と削除] -[日付と時間]		カーソル位置に今日の日付と現在時刻を挿入します。

挿入される日付のフォーマットは、Windowsコントロールパネルの[地球]の[日付]の[短い形式]で選択しているものを使います。時間のフォーマットは、同じく[地球]の[時刻]で設定されているものに従います。

メール引用符の挿入と削除

選択範囲の行の先頭に、メールの引用符を挿入したり、取り除いたりします。

ボタン	メニュー操作	機能
	[編集]-[挿入と削除] -[メール引用符挿入]	選択範囲の各行の先頭にメール引用符を挿入します。
	[編集]-[挿入と削除] -[メール引用符削除]	選択範囲の各行の先頭にあるメール引用符を1レベル削除します。

[メール引用符挿入]コマンドで挿入する引用符は、[オプション設定]ダイアログボックスの[編集]ページの[メール引用]で変更できます。

[メール引用符削除]コマンドは、選択範囲の行の先頭が、>(半角)、＞(全角)、|(半角)、｜(全角)か、[オプション設定]ダイアログボックスの[編集]ページにある[メール引用]で指定した文字列の先頭の1文字と一致した場合、それを削除します。

改行コード挿入

折り返し表示をしているとき、選択範囲の折り返し位置に改行コードを挿入します。

改行コード削除

カーソルのある行、または選択範囲の改行コードを削除し、行を連結します。

RGB 値挿入

[RGB値挿入]ダイアログボックスを表示して、そこで選択したカラーのRGB値をカーソル位置に挿入します。

RGB値は、R・G・Bの順番(HTMLファイルに適した順)ですが、[オプション設定]ダイアログボックス[編集]ページの[編集オプション]の[編集]-[RGB値挿入のときのRGB順序を逆転させる(BGRの順)]をONにすると、WindowsのCOLORREF値を指定するのに便利なB・G・R順となります。編集しているファイルの言語が、HTMLで始まっている場合はオプション設定に関わらず、#RRGGBB形式で挿入します。

ボタン	メニュー操作	機能
	[編集]-[挿入と削除]-[RGB値挿入]	現在のカーソル位置にRGB値を挿入、または修正します。

【ヒント】[RGB値挿入]コマンドを実行する前に、テキストの#FFFFFFのような部分を選択しておくと、これをHTML形式の色情報と判断して、[RGB値挿入]ダイアログボックスの初期値にします。これにより、すでにテキスト中にあるRGB値の修正が可能です。

削除

文字を削除します。削除した文字列はクリップボードに入りません。

ボタン	メニュー操作/機能	キー操作
	バックスペース	Shift+Backspace
	[編集]-[削除]、1文字削除	Delete
	すべて削除	
	単語削除	Shift+Delete
	右へ単語削除	Ctrl+Delete
	左へ単語削除	Ctrl+Backspace
	行末まで削除※	Ctrl+K
	ファイル行の行末まで削除	
	表示行の行末まで削除	
	行頭まで削除※	Ctrl+Shift+K
	ファイル行の行頭まで削除	
	表示行の行頭まで削除	
	複数の空白を1つにまとめる	
	現在行を削除	

	連続した空白行削除	
	行末の空白を削除	Ctrl+Shift+S

【メモ】※のコマンドは、行番号表示モードがファイル行単位か表示行単位かによって、ファイル行または表示行を基準に削除します。

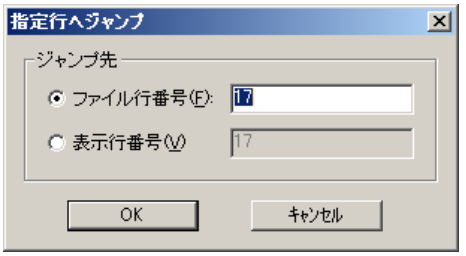
ジャンプ

編集しているテキストのいろいろな部分へジャンプするコマンドです。

ボタン	メニュー操作/機能	キー操作
	[編集]-[指定行ジャンプ]	Ctrl+G
	[編集]-[対応括弧ジャンプ]	Ctrl+]
	対応括弧へジャンプして範囲選択	Ctrl+Shift+]
	[編集]-[タグジャンプ]-[次のタグ]	F4
	[編集]-[タグジャンプ]-[前のタグ]	Shift+F4
	[編集]-[タグジャンプ]-[カーソル行のタグ]	Ctrl+T
	[編集]-[タグジャンプ]-[次のブロック]	Ctrl+.
	[編集]-[タグジャンプ]-[前のブロック]	Ctrl+,
	対応する次の条件コンパイル文へジャンプ	Alt+]
	対応する前の条件コンパイル文へジャンプ	Alt+[
	[編集]-[変更行ジャンプ]-[次の変更行]	Ctrl+Shift+9
	[編集]-[変更行ジャンプ]-[前の変更行]	Ctrl+Shift+8
	[編集]-[変更行ジャンプ]-[最終変更個所]	Ctrl+Shift+L
	[編集]-[変更行ジャンプ]-[変更行情報クリア]	Ctrl+Shift+0
	[ツール]-[ブックマーク]-[次へジャンプ]	F2
	[ツール]-[ブックマーク]-[前へジャンプ]	Shift+F2

指定行ジャンプ

行番号がわかっている行へジャンプするときに使います。ファイル行または表示行を選択してから行番号をタイプし、[OK]ボタンをクリックしてください。



対応括弧へジャンプ

カーソルが括弧の前後にあるとき、対応する括弧へジャンプします。次の括弧が有効です。()、[]、{}、「」、〈〉、（）、｛｝、《》、【】、〔〕、［］、「」、『』、“”

[対応括弧ジャンプ]コマンドでは、HTMLとJavaScriptのように複数言語が混在している言語の場合、それぞれの言語を区別して、同じ言語の対応括弧へジャンプします。また、個々の言語の中でも、コメントや文字列の中の括弧は、同じくコメントや文字列の中の括弧と対応付けます。

この反対に、[見かけの対応括弧]コマンドでは文法や混在している言語を無視し、テキストとして対応括弧を探し、カーソルを移動させます。

【注意】文法を認識して対応括弧へジャンプするには、言語モードが正しく設定されている必要があります。

タグジャンプ

ファイル名と行番号の情報を元に、ファイルを自動的に開き、指定された行にジャンプします。Peggyのプロジェクト機能を使えば、コンパイラのエラー情報を元にジャンプすることもできます。

ボタン	メニュー操作	キー操作	機能
	[編集]-[タグジャンプ]-[次のタグ]	F4	アウトプットウインドウ内にある、次のタグジャンプ形式の情報を元にジャンプします。
	[編集]-[タグジャンプ]-[前のタグ]	Shift+F4	アウトプットウインドウ内にある、ひとつ前のタグジャンプ形式の情報を元にジャンプします
	[編集]-[タグジャンプ]-[カーソル行のタグ]		カーソルのある行を解析してタグジャンプします。編集ウインドウ内でも使えます。 コンパイラなどの出力からエラータグファイルを作成し、そのファイルをPeggyで開いてエラー行にカーソルを移動し、このコマンドを実行してください。カーソル行のタグ情報を解析し、そのファイルを開いて問題のあった行へジャンプします。

編集ウインドウからの比較タグジャンプ

編集ウインドウにはdiffコマンドの出力を解析する機能が備わっています。これによって、アウトプットウインドウを使ったファイル比較、フォルダ比較と同じこと

が任意の編集ウインドウから可能になります。diffコマンドの比較結果をファイルにリダイレクトし、それをPeggyで開いて[タグジャンプ]コマンドを実行します。比較した2つのファイルを開いて、相違点の背景色を変え、見やすい形で並べて表示します。

```
C:¥>diff file1.txt file2.txt > diff.out
```

【ヒント】タグジャンプできる文字列形式

行先頭からファイル名、区切り、行番号、区切り、その他文字列...の形式であればタグジャンプ可能です。

【例】D:¥dev¥projct¥main.c(11) : Syntax Error No 5

D:¥dev¥projct¥main.c 11 = Warning No 5

特殊タグパターン

[ツール]-[オプション]の[全般]ページにある、[特殊パターン]にオリジナルタグパターンを追加することができます。

Syntax Error No 5 D:¥dev¥projct¥main.c(11)

Warning No 5 Line(11) D:¥dev¥projct¥main.c

などの場合もタグジャンプすることが可能になります。

【参照】『第11章 カスタマイズ/オプション設定/オプション[全般]』

ラベル行ジャンプ

編集しているテキストの言語によってジャンプ動作が変わります。Cなどのプログラミング言語では、前後の関数へ素早くカーソル移動できるので便利です。

ボタン	メニュー操作	キー操作	機能
	[編集]-[タグジャンプ]-[次のブロック]	Ctrl+.	編集中のテキストの言語に従い、次のブロックにジャンプします。
	[編集]-[タグジャンプ]-[前のブロック]	Ctrl+,	編集中のテキストの言語に従い、次のブロックにジャンプします。

言語	ジャンプ先
テキスト/メール	前後の行頭が見出しのように見える行
C、Pascalなどブロック構造を持ったプログラミング言語	前後の関数の先頭、または最も外側のブロック構造の先頭行
アセンブリ言語	前後のラベル

【注意】ラベル行として認識するのは、行の先頭文字列がプログラミング言語の識別子に半角コロン(:)が続いている場合です。

次 / 前の対応する条件コンパイル文へジャンプ

カーソル行から、それぞれ前/後ろの対応する条件コンパイル文、または条件アセンブル文にジャンプします。

ボタン	メニュー操作	キー操作	機能
		Alt+]	次の対応する条件コンパイル文へジャンプします。
		Alt+[	前の対応する条件コンパイル文へジャンプします。

変更行へジャンプ

変更が加えられた行には変更行マークが付き、そのマーク行へ順にジャンプすることができます。

ボタン	メニュー操作	キー操作	機能
	[編集]-[変更行ジャンプ]-[次の変更行]	Ctrl+Shift+9	テキストの終わりに向かって変更行マークのある行を探しジャンプします。
	[編集]-[変更行ジャンプ]-[前の変更行]	Ctrl+Shift+8	テキストの先頭に向かって変更行マークのある行を探しジャンプします。
	[編集]-[変更行ジャンプ]-[最終変更個所]	Ctrl+Shift+L	最後に変更した行へジャンプします。
	[編集]-[変更行ジャンプ]-[変更行情報クリア]	Ctrl+Shift+0	変更行情報をクリアします。

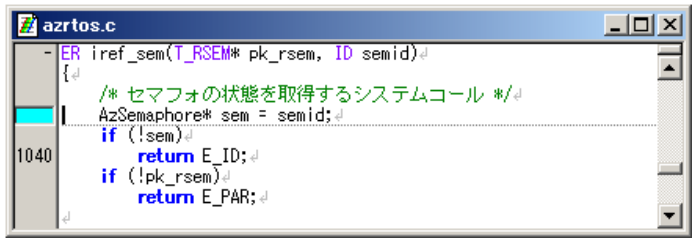
ブックマーク

編集中テキストの任意行に印（ブックマーク、しおり）を付ける機能です。ブックマークの付いた行のマーカー領域には、四角形のブックマークが付くので一目で確認できます。

ボタン	メニュー操作	キー操作
	[ツール]-[ブックマーク]-[一覧表示]	Alt+F2
	[ツール]-[ブックマーク]-[設定/解除]	Ctrl+F2
	[ツール]-[ブックマーク]-[１を設定/解除]	
	[ツール]-[ブックマーク]-[２を設定/解除]	
	[ツール]-[ブックマーク]-[３を設定/解除]	
	[ツール]-[ブックマーク]-[次へジャンプ]	F2
	[ツール]-[ブックマーク]-[前へジャンプ]	Shift+F2
	[ツール]-[ブックマーク]-[すべてクリア]	Ctrl+Shift+F2

ブックマークは3種類あります。[ ]、[ ]、[ ]ボタンでそれぞれON/OFFできる他、[ ]ボタン、またはCtrl+F2キーで順番に切り替えることもできます。マーカー領域のポップアップメニューからも設定可能です。それぞれのブックマークの色は、[オプション設定]ダイアログボックスの[表示色]ページで変えられます。

ブックマークを付けておくと、F2、Shift+F2キーを使って簡単にそこにジャンプできます。[ ]ボタンにより、プロジェクト中のすべてのブックマークを一覧表示できるので、他のファイルに付けられているブックマークにジャンプすることも簡単にできます。



[ブックマークの設定・解除]

カーソルのある行にブックマークを付けたり、クリアします。

[ブックマーク１・２・３を設定 / 解除]

それぞれカーソルのある行にブックマーク1、2、3を設定、または解除します。

[次へジャンプ]

テキストの終わりに向かってブックマークが付いている行を探します。

[前へジャンプ]

テキストの先頭に向かってブックマークが付いている行を探します。

[リスト表示]

プロジェクトに登録されているファイル、および編集中のファイルに付けられているすべてのブックマークの一覧を表示します。

[すべてクリア]

すべてのブックマークをクリアします。

【注意】プロジェクトに登録してあるファイルに付けたブックマークは、プロジェクト内部に保存されているので、ファイルを閉じてもそのまま残ります。しかしプロジェクトを開いていなかったり、プロジェクトとは無関係のファイルに付けたブックマークは、ファイルを閉じると失われてしまいます。また、変更したファイルを保存せずに閉じたり、外部エディタで変更したような場合、ブックマークの位置がずれてしまうことがあります。

[ブックマーク一覧] ダイアログボックス

プロジェクトに登録されているファイルと、現在開いているファイルに設定されているブックマーク一覧を表示します。各ブックマークには、その行のサンプルテキストも合わせて表示されるので、すばやく目的の行を探しだすことが可能です。

[ジャンプ]

選択しているブックマークへジャンプし、[ブックマーク一覧]ダイアログボックスを閉じます。

[開く]

選択しているブックマークのファイルを開き、その行を表示します。[ブックマーク一覧]ダイアログボックスは閉じません。

[終了]

[ブックマーク一覧]ダイアログボックスを閉じます。

[種類変更]

選択しているブックマークの種類を変更します。ファイルを選択しているときは、ファイル中のすべてのブックマークの種類を変更します。

[削除・復活]

選択しているブックマークに削除印[X]を付けます。または、すでに付いている削除印[X]を取り消します。ファイルを選択しているときは、ファイルのすべてのブックマークに削除印を付けたり消したりします。

削除印の付いているブックマークは、[ブックマーク一覧]ダイアログボックスを閉じたときに削除されます。削除せずにダイアログボックスを閉じるには、Escキーをタイプしてください。

[すべて削除]

すべてのブックマークに削除印[X]を付けます。

フィルタ [1]、[2]、[3]

3種類のブックマークそれぞれに、表示するかどうかを選択します。

[展開]

チェックすると､すべてのファイルアイコンを開いて設定されているブックマークを表示します。

[行番号]

ブックマークの付いているファイル行番号を表示します。

[フルパス]

ファイルのパスをフルパスで表示します。

[自動閉じ]

リスト中のブックマークをダブルクリックすることによりブックマーク位置を表示させたあと､[ブックマーク一覧]ダイアログボックスを自動的に閉じるかどうかを選択します。

[リサイクル]

このダイアログボックスからファイルを開くとき､すでに開いているウインドウを再利用するかどうかを選択します。(1)リサイクルする、(2)リサイクルしない、(3)Peggy全体のオプションに従う、から選択できます。

キーボードマクロ

何度も繰り返し行う一連のキー操作を一時的に記憶して､再生する機能をキーボードマクロと言います。カーソル移動、文字の入力、削除など、ダイアログボックスを開かずに行うほとんどの編集コマンドを記憶できます｡同じ編集操作を何度も繰り返して行う場合に便利です。どのコマンドが記録できて何ができないかは、『APPENDIX/付録Aコマンド一覧』を参照してください。

ボタン	メニュー操作/機能	キー操作
	[ツール]-[キーボードマクロ]-[記録開始]	Ctrl+Shift+R
	[ツール]-[キーボードマクロ]-[記録一時停止]	
	[ツール]-[キーボードマクロ]-[記録終了]	
	[ツール]-[キーボードマクロ]-[再生]	Ctrl+Shift+P
	[ツール]-[キーボードマクロ]-[プロジェクトへ保存]	

キーストロークの記録中は､ステータスバーの[キーボードマクロインジケータ]に[記録]と表示されるので確認できます。また、記録を一時停止しているときには、[停止]表示に変わります。

[記録開始]

キーストロークの記録を開始します。すでにキーストロークを記録しているときは、記録を終了します。

[記録一時停止]

キーストロークの記録を一時停止します。もう一度押すと記録を再開します。

[記録終了]

キーストロークの記録を終了します。

[再生]

記録したキーストロークを再生します。

[プロジェクトへ保存]

記録したキーストロークをプロジェクトに名前を付けて登録します。プロジェクトに登録されたキーボードマクロは、マウスアクションかツールアクションで再生できます。

【参照】『第6章 プロジェクト機能/プロジェクト機能/キーボードマクロを再生する』

アウトライン

アウトラインコマンドは、編集中のテキストを解析し、プロジェクトウィンドウのアウトラインページにその結果を表示します。

ボタン	メニュー操作/機能	キー操作
	[ツール]-[アウトライン]-[解析]	
	[ツール]-[アウトライン]-[対応項目]	
	[ツール]-[アウトライン]-[プログラム解析]	
	[ツール]-[アウトライン]-[見出し解析]	
	[ツール]-[アウトライン]-[HTML解析]	

関数、見出し行、HTML/XMLタグなどをツリー表示し、テキスト入力を支援します。表示されたツリーアイテムをダブルクリックするとテキストの位置を表示します。解析の仕方は編集中のテキストの種類により、３つに分けられます。

種類	概要
プログラム	関数、外部変数など、ctags/etagsが生成するシンボルをツリー表示します。

文章	見出しをツリー表示します。特に文法のないテキストファイルを対象にしますので正確に見出しを抽出することはできません。しかし、編集中のテキストに合った表示にする方法としては、<share>フォルダ内の<ac_item.kwd>を基にして新しいキーワード定義ファイルを作り、テキストの言語をそれに設定します。すると、その中のHeadingPatternの正規表現にマッチする行が見出しになります。 【例】HeadingPattern: /^第¥d+部/ また、複数行記述することで大見出し、中見出しなどの階層表示も可能です。 【例】HeadingPattern: /^第¥d+章/ HeadingPattern: /^第¥d+節/
HTML / XML	デフォルトではHTML解析を行います。開始タグ[ ]、や終了タグ[ ]、警告タグ[ ]、エラータグ[ ]、[ ]、などタグを行番号順に表示をします。入れ子になっているタグは階層表示します。見出し解析では、H1からH6までのタグをそのレベルで階層表示します。

【ヒント】HTML/XML解析でエラーが見つかったときは、アウトプットウインドウにエラーとなった行の情報が出力されます。解析後F4、Shift+F4キーによりエラー行へ順番にジャンプすることができます。

表示の順序やHTMLタグのフィルタの設定などは[オプション設定]ダイアログボックスの[アウトライン]ページを参照してください。

[解析]

アウトライン解析を行い、プロジェクトウィンドウのアウトラインページに表示します。

[対応項目]

編集ウィンドウのカーソル位置に対応するアウトラインページのアイテムを表示します。

[プログラム解析]、[見出し解析]、[HTML 解析]

通常は[解析]コマンドでPeggyがテキストの種類を判別します。ただし、HTML/XMLファイルで見出し解析をさせたい場合や自動判別がうまくいかなかった場合、ユーザがその種類を指定して解析させるときに使用します。

罫線機能

罫線文字を使ってテキスト上に罫線を描いたり消したりすることができます。文字でテキスト上に図形を描きたいときなどに便利です。

ボタン	メニュー操作/機能	キー
	罫線を左へ	Alt+←
	罫線を右へ	Alt+→
	罫線を上へ	Alt+↑

	罫線を下へ	Alt+↓
	罫線を左へ向かって消す	Alt+Shift+←
	罫線を右へ向かって消す	Alt+Shift+→
	罫線を上へ向かって消す	Alt+Shift+↑
	罫線を下へ向かって消す	Alt+Shift+↓
	罫線の種類を変更する	Alt+Home
	[ツール]-[罫線の種類]-[文字罫線]	
	[ツール]-［罫線の種類]-[細い罫線]	
	[ツール]-［罫線の種類]-[太い罫線]	
	NECの旧罫線文字を変換	

罫線の種類

以下の3種類の罫線をサポートしています。罫線の種類を変えるには、Alt+Homeをタイプしてください。新しく選択した罫線の種類が、ステータスバーに表示されます。

●細い全角罫線(─、 │、┼、┬、┐ など)

●太い全角罫線(**━、 ┃、╋、┳、┓** など)

●半角文字罫線(アスキー文字+、-、|を使った罫線)

罫線の描画

罫線を描くには、Alt+↑、Alt+↓、Alt+←、Alt+→キーを使います。それぞれの方向に罫線を引きます。罫線を引き過ぎたときは、[編集]-[元に戻す]で取り消すことができます。

罫線を引こうとしている個所にまだ文字がない場合､その行に必要数の空白文字を付け足してから罫線を引きます。その際、スペースだけを使うのか、それともスペースとタブを組み合わせて使うのかを選択できます｡タブとスペースを組み合わせて使った方がテキストサイズを小さくできる反面､タブ間隔を変えると罫線がガタガタに崩れてしまう欠点があります。

【参照】『第11章 カスタマイズ/[オプション設定]ダイアログボックス/[編集]ページ』

罫線の消去

すでに引いてある罫線を部分的に消したいようなときは、Alt+Shift+ ↑、Alt+Shift+↓、Alt+Shift+←、Alt+Shift+→を使ってください。罫線の他の部分を壊さないようにしながら、カーソルを移動させた部分だけ罫線を消去できます。

旧 NEC 罫線文字の変換

何年も前にコーディングされた古いソースファイルの中には、NEC PC-9800シリーズ独特の罫線文字を使っている場合があります｡このNEC 9800シリーズ独自の罫線文字は、Windowsのフォントにはないため罫線として表示されなかったり、ときには画面を乱す原因になったりもします。

Peggyでは､この独自罫線文字をWindowsフォントにある罫線文字に一括変換するコマンドを用意しました。しかしあまり一般的な機能とはいえないため、キーに割り

当てられていません。この機能を使いたい場合は、まず旧NEC罫線変換コマンドを任意のショートカットキーに割り当なければなりません。コマンドの名前は、[罫線 - NECの旧罫線文字を変換]です。ショートカットキーの割り当て方法は、『第11章 カスタマイズ/ショートカットキー』を参照してください。

仮にこの変換コマンドをショートカットキーのCtrl+9に割り当てたとします。以下の手順で、NEC 9800シリーズの独自罫線文字を変換できます。

1. 旧罫線文字を含んだファイルを開きます。このとき、ファイルが書き込み禁止でないことを確認してください。
2. 罫線文字を変換する範囲を選択します。ファイル全体を変換するには、メニューから [編集]-[すべてを選択] を実行します。
3. Ctrl+9 とタイプします。
4. 必要ならスクロールして、正しく変換されていることを確認してください。
5. メニューの [ファイル]-[上書き保存] を選択して、変換したファイルを保存します。

以上でWindowsの罫線文字に変換できました。

補完入力

長い単語や決まり文句を何度も入力しなければならないことがあります。大規模なプログラムになってくると、ラベルや関数名も長くなりがちです。Peggyには補完によりタイピング量を軽減する機能が備わっています。

ボタン	メニュー操作/機能	キー操作
	[編集]-[補完入力]-[単語補完]	Ctrl+Space
	[編集]-[補完入力]-[以前から単語補完]	
	[編集]-[補完入力]-[以降から単語補完]	
	[編集]-[補完入力]- [キーワード補完]	Ctrl+/
	[編集]-[補完入力]- [短縮入力]	

単語補完

「単語補完」は簡単かつ便利な補完入力機能です。 1 文字以上をタイプして「単語補完」を実行すると、その文字列で始まる単語をカーソルの前後から探し、見つかるとその単語をカーソル位置にコピーします。

上図はif文の中を記述しようとして“L”をタイプしたところです。カーソルは“L”の右にあります。ここで[単語補完]コマンドを実行すると、“L”で始まる単語が2つ表示されます。いずれかを選択すると、右図のように単語が補完されます。な

お、候補となる単語が１つしかないときは選択画面は表示されず、直ちに単語が補完されます。

単語補完には次の3種類があります。

[単語補完]

カーソル位置の文字列で始まる単語を上下方向に探します。

[以前から単語補完]

カーソル位置の文字列で始まる単語を上方向にだけ探します。

[以降から単語補完]

カーソル位置の文字列で始まる単語を下方向にだけ探します。

【ヒント】単語を探すとき、大文字と小文字は区別します。また、上方向、下方向それぞれどの範囲まで単語を探すかを指定できます。[オプション設定]ダイアログボックスの[補完]ページで設定してください。

【ヒント】日本語文字列の補完入力も可能です。[オプション設定]ダイアログボックスの[補完]ページの[日本語も補完]をONにしてください。英単語と同じように日本語文字列を補完します。日本語文字列を補完する単位は、漢字またはカタカナの後にひらがなが続いた部分です。

短縮入力

「短縮入力」は文字列の事前登録が必要ですが、強力な補完入力機能です。

ユーザが定義した短い文字列(キー文字列)を入力してメニューから[短縮入力]を選択すると、それに対応する任意の文字列に置換する機能が「短縮入力」です。あらかじめ定義しておけば､少ないキーストロークで長い文字列の入力が可能になります。

短縮入力を有効にする

1. [オプション設定] ダイアログボックスの [補完] ページにある [短縮入力] をチェックします。これで短縮入力が有効になりました。
2. 次にキーとなる文字列と、変換キーで変換する文字列をその下へ入力します。１つの短縮入力は、１行に記述します。キーとなる文字列は、半角英文字で始まりその後へ英数文字が続いたものです。さらにその後へ１つ以上スペースを空けて変換する文字列をタイプしてください。

短縮入力は、いくつでも定義できます。同じキー文字列が複数重なった場合は、後にある方が有効となります。

【例】 howdy　　拝啓 ますますご清祥のこととお喜び申しあげます。

変換する文字列中に改行やタブを入力したいときは､¥nや¥tとしてください。円記号自身は、¥¥とします。変換文字列の先頭にスペースを入れたいときは、スペースの前に¥を置きます。

タブ	¥t
改行	¥n
最初のスペース	¥を先頭に付ける
円記号(¥自身)	¥¥

【例】hello　int main(int argc, char* argv[])¥n{¥n¥tprintf("%s¥¥n", "Hello, world);¥n}¥n

短縮入力する

短縮入力を行うには、テキスト入力中にキー文字列をタイプし、その後[編集]-[補完入力]-[短縮入力]を選択します。

短縮入力を無効にする

短縮入力を無効にするには、[オプション設定]ダイアログボックスの[補完]ページにある[短縮入力]のチェックを外してください。設定してある文字列を削除する必要はありません。またチェックすれば、いつでも使えるようになります。

キーワード補完入力

現在編集しているテキストがプログラミング言語のソースコードである場合、キーワードを途中まで入力して、その後の部分を補完させる機能が「キーワード補完入力」です。キーワードを数文字入力した時点で、メニューの[編集]-[補完入力]-[キーワード補完]を選択してください。

入力済みの文字からキーワードを1つだけ特定できた場合は、そのキーワードに変換します。特定できなかった場合は、候補となるキーワードが表示されるので、そこから選んで入力できます。

非常に多くの補完候補がある場合は、さらに数文字タイプすることで、候補を絞り込むことができます。補完中の動作は、[オプション設定]ダイアログボックスの[補完]ページでカスタマイズしてください。

【例】言語モードがC言語のときに、con と入力して、[キーワード補完]を選択すると、2つの候補const、continueがメニュー表示されます。continueをマウスでクリックすると、tinueが補完されます。

【メモ】[オプション設定]ダイアログボックスの[補完]ページの[ヒント無しでも補完]をONにしておくと、1文字もキーをタイプしなくても候補リストを表示させることができます。この場合、候補リストはすべてのキーワードを含んでいます。

コメント

ファイルの言語モードに従い、選択範囲をコメント化したり、コメント記号を取り除いたりします。

ボタン	メニュー操作/機能	キー操作
	[編集]-[コメント]-[行コメント挿入]	F10
	行コメント(2)挿入	
	[編集]-[コメント]-[ブロックコメント挿入]	Ctrl+F10
	[編集]-[コメント]-[コメント解除]	Shift+F10
	[編集]-[コメント]-[コメント削除]	Ctrl+Shift+F10
	[編集]-[コメント]-[コメント内部を削除]	
	単語のコメント化	Alt+F10
	行をコメント化	Ctrl+Alt+F10

	コメントを選択	Alt+F8
	コメント内部を選択	Ctrl+Alt+F8

[行コメント挿入]

選択範囲の文字を、行単位のコメントにするため、コメント記号を挿入します。矩形選択範囲でも実行できます。

行コメント (2) 挿入

行コメントの形式が2種類あるような言語では、2種類目の行コメントを挿入します。行コメントが1種類の言語では、[行コメント(1)挿入]コマンドと同じ動作をします。

[ブロックコメント挿入]

選択範囲の文字を、ブロック単位のコメントにするため、コメント記号を挿入します。矩形選択範囲でも実行できます。

[コメント解除]

選択範囲内にコメントがあると、そのコメントのコメント記号だけを取り除きます。

[コメント削除]

選択範囲内にコメントがあると、コメント本体とコメント記号の両方とも削除します。

[コメント内部を削除]

選択範囲内にコメントがあると、コメント本体のみを削除します。

単語のコメント化

カーソル位置の単語をコメントにします。

行をコメント化

カーソルのある行全体をコメントにします。

コメントを選択

カーソル位置を含んでいるコメント全体を選択します。

コメント内部を選択

カーソル位置を含んでいるコメントの内部を選択します。

インデント

選択している行のインデントや揃え位置を調整します。

ボタン	メニュー操作/機能	キー操作
	[編集]-[インデント]-[深く]	選択+Tab
	[編集]-[インデント]-[浅く]	選択 +Shift+Ta
	[編集]-[インデント]-[選択範囲]	Ctrl+I

	[編集]-[インデント]-[ファイル全体]	
	[編集]-[インデント]-[左揃え]	F9
	[編集]-[インデント]-[右揃え]	Shift+F9
	[編集]-[インデント]-[中央揃え]	Ctrl+F9

[深く]

選択している範囲のテキストのインデントを1レベル深くします。

[浅く]

選択している範囲のテキストのインデントを1レベル浅くします。

[選択範囲]

選択範囲を言語のルールに従ってスマートインデントします。選択範囲がないときは、カーソルのある行をスマートインデントします。

[ファイル全体]

ファイル全体をスマートインデントします。

[左揃え]

選択している範囲のテキストを左に揃えます。

[右揃え]

選択している範囲のテキストを右に揃えます。

[中央揃え]

選択している範囲のテキストを中央に揃えます。

HTML 編集

HTMLファイル特有の編集操作をサポートするコマンドです。

ボタン	メニュー操作/機能	キー操作
	 と改行を挿入する	
	[編集]-[HTML]-[次のタグ]	Ctrl+Shift+.
	[編集]-[HTML]-[前のタグ]	Ctrl+Shift+,
	[編集]-[HTML]-[対応するタグ]	Ctrl+Shift+/
	タグを選択する	
	タグ間のテキストを選択する	
	タグペア間のテキストを選択する	
	タグペアと間のテキストを選択する	Ctrl+F7
	[編集]-[HTML]-[タグ削除]	Shift+F7
	[編集]-[HTML]-[タグペア削除]	Ctrl+Shift+F7
	エレメント名とアトリビュート名を大文字化する	

	エレメント名とアトリビュート名を小文字化する	
	[編集]-[HTML]-[文字からエンティティへ変換]	Alt+F7
	[編集]-[HTML]-[エンティティから文字へ変換]	Alt+Shift+F7
	[編集]-[HTML]-[テキストへ変換]	

【注意】これらのコマンドを利用するには、ファイルの言語モードがHTMLベースのもの(例えばHTML、ASP、JSPなど)に設定されている必要があります。

**
と改行を挿入する**

カーソル位置に
と改行文字を挿入します。[編集オプション]の[HTMLタグ挿入は大文字で行う]により、
か
を選ぶことができます。

[次のタグ]、[前のタグ]

次、前のHTMLタグへカーソルを移動します。[編集オプション]の[前後や対となるHTMLタグへジャンプ後にタグを選択する]をONにすると、ジャンプ後にタグを選択します。

[対応するタグ]

現在カーソルのあるHTMLと対になるHTMLタグへカーソル移動します。[編集オプション]の[前後や対となるHTMLタグへジャンプ後にタグを選択する]をONにすると、ジャンプ後にタグを選択します。

タグを選択する

カーソルのあるHTMLタグ全体を選択します。

タグ間のテキストを選択する

カーソルがHTMLタグの間のテキストにある場合、そのテキスト部分を選択します。

タグペア間のテキストを選択する

ペアとなるHTMLタグ間のテキストを選択します。

タグペアと間のテキストを選択する

ペアとなるHTMLとその間のテキストを選択します。

[タグ削除]

カーソ位置のHTMLタグを削除します。

[タグペア削除]

カーソル位置のHTMLタグと、対になっているHTMLの両方を削除します。タグ間のテキストは、そのまま残ります。[編集オプション] の[ペアHTMLタグ削除で片方しかない場合でも削除する]により、対になるHTMLタグが見つからない場合でも、削除を実行するかどうかを選択できます。

エレメント名とアトリビュート名を大文字化する

選択範囲にあるHTMLタグのエレメント名とアトリビュート名を大文字化します。

エレメント名とアトリビュート名を小文字化する

選択範囲にあるHTMLタグのエレメント名とアトリビュート名を小文字化します。

[文字からエンティティへ変換]

選択範囲のHTML/XMLタグ外にある以下のエンティティを文字に変換します。

文字	エンティティ
<	<
>	>
&	&

[エンティティから文字へ変換]

選択範囲の以下のエンティティを文字に変換します。

エンティティ	文字
&lt、<	<
&gt、>	>
&quot、"	″
&amp、&	&
、	（半角スペース）
&copy、©	(C)
&reg、®	(R)

[HTML をテキストに変換する]

選択範囲のHTMLタグ、HTMLコメント、スクリプトをすべて削除し、エンティティを文字に変換します。

文字変換

各種の文字変換を行うコマンドです。

ボタン	メニュー操作/機能	キー操作
	[編集]-[文字変換]-[大文字化]	Ctrl+Shift+U
	[編集]-[文字変換]-[小文字化]	Ctrl+U
	[編集]-[文字変換]-[頭文字化]	
	大文字小文字を反転	
	単語を大文字に変換	
	単語を小文字に変換	
	単語の頭文字だけ大文字に変換	
	[編集]-[文字変換]-[カタカナへ変換]	
	[編集]-[文字変換]-[ひらがなへ変換]	
	[編集]-[文字変換]-[全角に変換]	
	[編集]-[文字変換]-[半角に変換]	
	[編集]-[文字変換]-[行頭スペースをタブに変換]	
	[編集]-[文字変換]-[行頭タブをスペースに変換]	Ctrl+Shift+Space
	[編集]-[文字変換]-[スペースをタブに変換]	

	[編集]-[文字変換]-[タブをスペースに変換]	
	改行をスペースに変換	
	[編集]-[文字変換]-[選択範囲の漢字コード]	
	[編集]-[文字変換]-[ファイルの漢字コード]	
	[編集]-[文字変換]-[ファイルの改行コード]	

[大文字化]

カーソル位置の文字または選択範囲の中の文字について、半角英小文字があれば、すべて半角英大文字にします。矩形選択範囲の変換も行えます。

[小文字化]

カーソル位置の文字または選択範囲の中の文字について、半角英大文字があれば、すべて半角英小文字にします。矩形選択範囲の変換も行えます。

[頭文字化]

カーソル位置の文字または選択範囲の中の文字について、半角英大文字の単語の頭文字を大文字に、それ以外をすべて半角英小文字にします。

大文字小文字を反転

カーソル位置の文字または選択範囲の中の文字について、半角英文字の大文字と小文字を逆転させます。

単語を大文字に変換

カーソル位置の単語をすべて大文字に変換します。

単語を小文字に変換

カーソル位置の単語をすべて小文字に変換します。

単語の頭文字だけ大文字に変換

カーソル位置の単語の頭文字を大文字に、それ以外を小文字に変換します。

[カタカナへ変換]

カーソル位置または選択範囲の全角ひらがな文字を全角カタカナに変換します。矩形選択範囲の変換も行えます。

[ひらがなへ変換]

カーソル位置または選択範囲の全角カタカナ文字を全角ひらがなに変換します。矩形選択範囲の変換も行えます。

[行頭スペースをタブに変換]

カーソル位置の行、または複数行を選択している場合はそのすべての行について、行頭にあるスペースをタブに変換します。

[行頭タブをスペースに変換]

カーソル位置の行、または複数行を選択している場合はそのすべての行について、行頭にあるタブをスペースに変換します。

[スペースをタブに変換]

カーソル位置の行、または複数行を選択している場合はそのすべての行について、スペースをタブに変換します。

[タブをスペースに変換]

カーソル位置の行、または複数行を選択している場合はそのすべての行について、タブをスペースに変換します。

[改行をスペースに変換]

改行コードを削除し、行を連結します。その際、元の行と行の間に半角スペースを挿入します。

[全角に変換]

カーソル位置の文字、または選択範囲の半角文字を全角に変換します。変換する文字の種類によって、以下の7コマンドを使い分けてください。矩形選択範囲の変換も行えます。

メニュー操作	全角に変換する文字
[編集]-[文字変換]-[全角に変換]-[すべて]	すべての半角文字を全角に変換します。
[編集]-[文字変換]-[全角に変換]-[英文字]	半角英文字(A～Z、a～z)を全角(Ａ～Ｚ、ａ～ｚ)に変換します。
[編集]-[文字変換]-[全角に変換]-[数字]	半角数字(0～9)を全角(０～９)に変換します。
[編集]-[文字変換]-[全角に変換]-[英数字]	半角の英文字と数字を全角に変換します。
[編集]-[文字変換]-[全角に変換]-[スペース]	半角スペースを全角スペースに変換します。
[編集]-[文字変換]-[全角に変換]-[スペース以外]	半角スペース以外のすべての文字を全角に変換します。
[編集]-[文字変換]-[全角に変換]-[文字種指定]	[半角から全角へ変換]ダイアログボックスを表示し、詳細な文字種の選択や変換する範囲を指定することができます。

[半角に変換]

カーソル位置の文字、または選択範囲の全角文字を半角に変換します。変換する文字の種類によって、以下の7コマンドを使い分けてください。矩形選択範囲の変換も行えます。

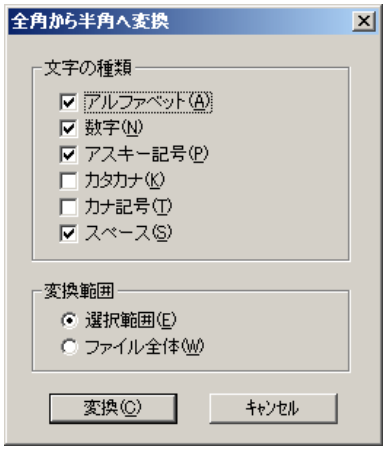
メニュー操作	半角に変換する文字
[編集]-[文字変換]-[半角に変換]-[すべて]	すべての全角文字を半角に変換します。
[編集]-[文字変換]-[半角に変換]-[英文字]	全角英文字(Ａ～Ｚ、ａ～ｚ)を半角(A～Z、a～z)に変換します。
[編集]-[文字変換]-[半角に変換]-[数字]	全角数字(０～９)を半角(0～9)に変換します。
[編集]-[文字変換]-[半角に変換]-[英数字]	全角の英文字と数字を半角に変換します。
[編集]-[文字変換]-[半角に変換]-[スペース]	全角スペースを半角スペースに変換します。

[編集]-[文字変換]-[半角に変換]-[スペース以外]	全角スペース以外のすべての文字を半角に変換します。
[編集]-[文字変換]-[半角に変換]-[文字種指定]	[全角から半角へ変換]ダイアログボックスを表示し、詳細な文字種の選択や変換する範囲を指定することができます。

【メモ】漢字など対応する半角文字がないような場合は変換されません。全角ひらがなを半角カタカナに変換したいときは、予め全角ひらがなから全角カタカナへ変換してから、このコマンドを実行してください。

[半角から全角へ変換]、[全角から半角へ変換] ダイアログボックス

半角から全角、またはその逆で全角から半角へ変換する文字の種類や範囲を指定します。このダイアログボックスは、[編集]-[文字変換]-[全角に変換]-[文字種指定]、[編集]-[文字変換]-[半角に変換]-[文字種指定]で呼び出せます。



[文字の種類]

アルファベット	A～Z、a～z
数字	0～9
アスキー記号	アスキー文字セットの記号を変換します。 !、″、#、$、%、&、’、(、)、*、+、,、−、.、/、:、;、<、=、>、?、@、[、¥、]、^、_、`、{、|、}、~
カタカナ	カタカナを変換します。
カナ記号	日本語の記号を変換します。
スペース	スペースを変換します。

[変換範囲]

選択範囲	選択範囲の文字だけを対象に変換します。矩形選択範囲の変換も行えます。
ファイル全体	ファイル全体を対象として変換します。

選択範囲の漢字コード

選択範囲の漢字コードをShift JISに変換するコマンド群です。Peggyはファイルを開くときに使われている漢字コードを自動判定し、内部コードであるShift JISに変換して読み込みます。しかし、ファイルの一部に異なる漢字コードが混在していた場合、それらを正しく表示することができません。

漢字が正しく表示されていない部分を選択し、以下の7つのコマンドを使い分けることでShift JISに変換し、正しく表示できるようになることがあります。

メニュー操作	機能
[編集]-[文字変換]-[選択範囲の漢字コード]-[自動判定して変換]	まず選択範囲の漢字コードを推測し、自動推測した漢字コードからShift JISへ変換します。もしも誤った漢字コードと判定してしまった場合は正しい変換ができません。そのときは、以下、6つのコマンドで漢字コードを明示して変換を試みてください。
[編集]-[文字変換]-[選択範囲の漢字コード]-[EUCから変換]	選択範囲の漢字コードをEUCであると仮定し、Shift JISへ変換します。
[編集]-[文字変換]-[選択範囲の漢字コード]-[JISから変換]	選択範囲の漢字コードをJISであると仮定し、Shift JISへ変換します。
[編集]-[文字変換]-[選択範囲の漢字コード]-[UTF-7から変換]	選択範囲の漢字コードをUTF-7であると仮定し、Shift-JISへ変換します。
[編集]-[文字変換]-[選択範囲の漢字コード]-[UTF-8から変換]	選択範囲の漢字コードをUTF-8であると仮定し、Shift-JISへ変換します。
[編集]-[文字変換]-[選択範囲の漢字コード]-[Unicode(LH)から変換]	選択範囲の漢字コードをUnicodeであると仮定し、Shift-JISへ変換します。Unicodeは、1文字が下位バイト/上位バイトの順で並んでいるとして解析します。
[編集]-[文字変換]-[選択範囲の漢字コード]-[Unicode(HL)から変換]	選択範囲の漢字コードをUnicodeであると仮定し、Shift-JISへ変換します。Unicodeは、1文字が上位バイト/下位バイトの順で並んでいるとして解析します。

【メモ】これらのコマンドはテキストの一部分の漢字コードを変換するだけです。ファイル全体の漢字コードを設定/変更するものではありません。ファイル全体の漢字コードを変更するには、[ファイルの漢字コード]コマンドで行います。

【ヒント】Unicodeらしい部分がなかなか正しく変換できないときは、変換を開始する位置を1文字前後させてみると上手く行くことがあります。Unicodeは常に2バイトで1文字となるため、どこが区切りか、また上位バイトが先か下位バイトが先かを考慮する必要があります。

【注意】Unicodeのバイト順(HL、LH)を考慮し別々のコマンドとしていますが、変換する文字列にUnicodeのバイト順マーカー(FFFE、またはFEFF)があると、それ以降はそのバイト順に切り替えて変換します。

【注意】漢字コードの自動判定が常に正しい結果を出すとは限りません。特に判定の元になる文字列が非常に短いと判断材料が足りなくなり、誤判定をする可能性が高くなります。多くの文字を選択した方が精度のよい変換が可能です。

ファイルの漢字コード

ファイル全体の漢字コードを指定します。このコマンドによりテキストファイルの漢字コードを変更すると、以後の保存、読み出しはその漢字コードで行います。以下の6つの漢字コードを選ぶことができます。

メニュー操作	漢字コード
[編集]-[文字変換]-[ファイルの漢字コード]-[Shift JISに変更]	Windows、MS-DOS、マッキントッシュで使われている漢字コードです。
[編集]-[文字変換]-[ファイルの漢字コード]-[EUCに変更]	主にUNIXオペレーティングシステム上で使われている漢字コードです。
[編集]-[文字変換]-[ファイルの漢字コード]-[JISに変更]	インターネットメールなどの通信関係で使われる漢字コードです。
[編集]-[文字変換]-[ファイルの漢字コード]-[UTF-7に変更]	Unicodeをインターネットメールとして送るときに使われる文字コードです。
[編集]-[文字変換]-[ファイルの漢字コード]-[UTF-8に変更]	Unicodeを変形した文字コードです。Javaなどで使われています。保存するとき、ファイルの先頭にBOM(バイトオーダーマーク)が付きます。
[編集]-[文字変換]-[ファイルの漢字コード]-[UTF-8Nに変更]	Unicodeを変形した文字コードです。保存するとき、ファイルの先頭にBOM(バイトオーダーマーク)は付きません。
[編集]-[文字変換]-[ファイルの漢字コード]-[Unicode Little Endianに変更]	1文字16-bitのコード体系です。下位バイト/上位バイトの順でファイルに保存します。
[編集]-[文字変換]-[ファイルの漢字コード]-[Unicode Big Endianに変更]	1文字16-bitのコード体系です。上位バイト/下位バイトの順でファイルに保存します。

【ヒント】[ファイル]-[名前を付けて保存]コマンドで保存するときに、漢字コードや改行コードを変換する方法もあります。

【メモ】[ファイルの漢字コード]コマンドは、編集中のテキストを保存する場合の漢字コードを変更するだけなので、テキストの内容を変更するものではありません。しかし、[編集]-[元に戻す]コマンドによって取り消すことは可能です。

ファイルの改行コード

作業中のファイル全体の改行コードを変更します。改行コードを変えると、以降Enterキーをタイプしたときもその改行コードが入力されるようになります。以下の3種類の改行コードが選択できます。

メニュー操作	改行コード
[編集]-[文字変換]-[ファイルの改行コード]-[CR-LFに変更]	Windows、MS-DOS、インターネットメールなどで使われている改行コードです。
[編集]-[文字変換]-[ファイルの改行コード]-[LFに変更]	UNIXオペレーティングシステム上のテキストファイルに使われている改行コードです。
[編集]-[文字変換]-[ファイルの改行コード]-[CRに変更]	マッキントッシュで使われている改行コードです。

【ヒント】[ファイル]-[名前を付けて保存]コマンドで保存するときに漢字コードや改行コードを変換する方法もあります。

【メモ】Peggyでは、各行の改行コードを個別に記憶しています。あまりないことですが、LF改行とCR改行、それにCR-LF改行の混じったファイルもそのままの状態で編集/保存が可能です。しかし逆に、そのような状態のファイルの改行コードを揃えたいときに、このコマンドを使ってください。誤って改行コードを変更してしまった場合は、[編集]-[元に戻す]で変更を取り消すことができます。

【注意】これらのコマンドは、ファイル全体の改行コードを変更します。部分的に改行コードを変更するコマンドはありません。

行順序

選択範囲の行を並べ替えるコマンドです。

ボタン	メニュー操作/機能	キー操作
	[編集]-[行順序]-[昇順にソート]	
	[編集]-[行順序]-[降順にソート]	
	[編集]-[行順序]-[先頭の空白を無視]	
	[編集]-[行順序]-[逆転]	

[昇順にソート]・[降順にソート]・[先頭の空白を無視]

選択範囲の行を文字コード順、または文字コードの逆順にソートします。[行順序]-[先頭の空白を無視]コマンドを実行しておくと、行の先頭にある空白文字を無視してソートします。また、[大小文字の区別]コマンドの設定状態により、半角英大文字を区別するかしないかが決まります。

[逆転]

選択範囲の行の順番を逆転させます。

ウインドウメニュー

[ウインドウ]メニューには、複数のウインドウの表示方法を操作する以下のコマンドがあります。

ボタン	メニュー操作/機能	キー操作
	[ウインドウ]-[新しいウインドウを開く]	
	[ウインドウ]-[表示位置]	
	[ウインドウ]-[分割]	
	[ウインドウ]-[上下に分割]	Ctrl+Shift+H
	[ウインドウ]-[左右に分割]	

	[ウインドウ]-[分割解除]	
	[ウインドウ]-[リサイクル]	
	[ウインドウ]-[重ねて表示]	
	[ウインドウ]-[上下に並べて表示]	
	[ウインドウ]-[左右に並べて表示]	
	[ウインドウ]-[アイコンの整列]	
	[ウインドウ]-[すべて閉じる]	
	[ウインドウ]-[ウインドウ選択]	Ctrl+Enter
	次のウインドウ	Ctrl+F6
	前のウインドウ	Ctrl+Shift+F6
	次のペイン	F6
	前のペイン	Shift+F6
	左のタブを選択	
	右のタブを選択	
	左端のタブを選択	
	右端のタブを選択	
	左隣のタブと入れ替え	
	右隣のタブと入れ替え	
	ペインを縦に拡大	Alt+F6
	ペインを縦に縮小	Alt+Shift+F6
	ペインを横に拡大	Ctrl+Alt+F6
	ペインを横に縮小	Ctrl+Alt+Shift+F6
	ウインドウへ保存	Ctrl+Alt+Shift+F6

[新しいウインドウを開く]

作業中のウインドウと同じ内容を表示する別のウインドウを開きます。1つの文書に対して複数のウインドウを開くことができるので、同時に1つの文書のいろいろな部分をそれぞれのウインドウに表示することができます。1つのウインドウで文書の内容を変更すると、その文書を表示する他のすべてのウインドウにもその変更が反映されます。新規ウインドウを開くと、そのウインドウがアクティブになり、他のウインドウの前面に表示されます。

[表示位置]

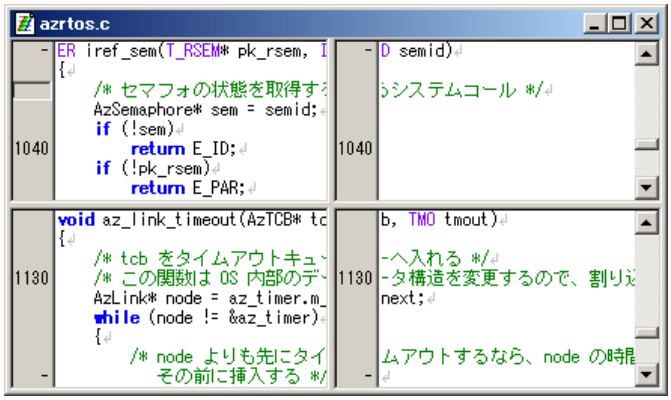
作業中のウインドウの位置や大きさを調整します。これらコマンドをショートカットキーに割り当てておけば、キーボードから手を離さずにウインドウを移動できます。

メニュー操作	機能
[ウインドウ]-[表示位置]-[大きく表示]	作業中のウインドウをできるだけ大きく表示します。
[ウインドウ]-[表示位置]-[上下に大きく表示]	作業中のウインドウの上下をできるだけ大きく表示します。
[ウインドウ]-[表示位置]-[左右に大きく表示]	作業中のウインドウの左右をできるだけ大きく表示します。
[ウインドウ]-[表示位置]-[上半分に表示]	作業中のウインドウを上半分の大きさで表示します。
[ウインドウ]-[表示位置]-[下半分に表示]	作業中のウインドウを下半分の大きさで表示します。
[ウインドウ]-[表示位置]-[左半分に表示]	作業中のウインドウを左半分の大きさで表示します。
[ウインドウ]-[表示位置]-[右半分に表示]	作業中のウインドウを右半分の大きさで表示します。
[ウインドウ]-[表示位置]-[2つを上下に並べて表示]	作業中のウインドウと1つ下のウインドウを上下に並べて表示します。
[ウインドウ]-[表示位置]-[2つを左右に並べて表示]	作業中のウインドウと1つ下のウインドウを左右に並べて表示します。

【メモ】[上下に並べて表示]や[左右に並べて表示]コマンドは、開いているすべてのウインドウを縦や横に配置しようとしますが、[2つを上下に並べて表示]、[2つを左右に並べて表示]コマンドは、重なっているウインドウの上から2つだけを対象に上下、左右に配置します。

[分割]

ひとつのテキストの離れた部分を同時に見たり編集したいことはよくあります。そのような場合は、スクロールバーの端に付いているスプリッタを左マウスボタンでドラッグし、適当なところで離してウインドウを分割します。ウインドウは縦にも横にも分割できます。



ウインドウを分割すると、それぞれがテキスト中の任意の部分を表示できるようになります。分割したウインドウはまったく対等です。どこから入力や削除を行っても、違いはありません。

分割したウインドウの間でカーソルを移動させるには、マウスでクリックするか、F6、Shift+F6キーをタイプします。

分割ウインドウの大きさを変えるには、スプリッタを左マウスボタンでドラッグして移動します。またキーボードからは、Alt+F6、Alt+Shift+F6、Alt+Ctrl+F6、Alt+Ctrl+Shift+F6キーで、分割ウインドウを大きくしたり小さくしたりできます。

分割をやめるには、スプリッタ上をダブルクリックするか、スプリッタをウインドウの端までドラッグして離します。

さらに多くの個所を一度に見たいときは、メニューから[ウインドウ]-[新しいウインドウを開く]を選択してください。同じテキストを複数のウインドウから参照できるようになります。新しく開いたウインドウも元のウインドウもまったく対等です。どちらのウインドウから編集を行っても違いはありません。

[上下に分割]

編集中のテキストを上下に分割表示します。

[左右に分割]

編集中のテキストを左右に分割表示します。

[分割解除]

分割表示をやめます。

[リサイクル]

ウインドウリサイクルの対象とするかどうかを交互に切り替えます。

[重ねて表示]

テキストタイトルが見える状態で、開かれているウインドウを重ねて表示します。

[上下に並べて表示]

複数のウインドウを上下に並べて表示します。

[左右に並べて表示]

複数のウインドウを左右に並べて表示します。

[アイコンの整列]

アプリケーションのメインウインドウの一番下にアイコン化されているウインドウを整列させます。もし文書のウインドウが開いていて、メインウインドウの一番下の部分にかかっている場合、このウインドウの下になっているアイコンは見えません。

[すべて閉じる]

開かれているすべての編集ウインドウを閉じます。

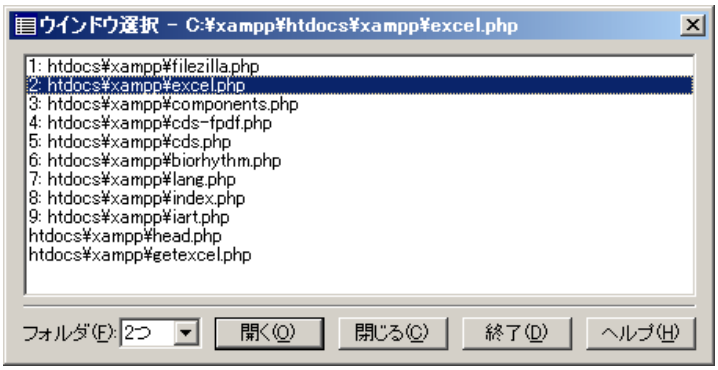
このとき変更のあったテキストに関しては、保存するかどうかを確認します。

ウインドウに保存

新しい編集ウインドウを作成し、現在表示されているアウトプットウインドウの内容をコピーします。

[ウインドウ選択]

リストから開いている編集ウインドウを選択します。



[フォルダ]

ファイル名でだけでなくパスをどのくらい表示するかを選択できます。

[ウインドウ一覧]

現在開いている編集ウインドウのリストをウインドウの重なり順に表示します。1～9のキーを押すと、先頭に数字の付いているウインドウを選択します。他のA～Zなど数字以外のキーを押すと、ファイル名の頭文字と一致するウインドウを選択します。

選択後にEnterキーを押すことで、その編集ウインドウを最前面に置きます。また、左マウスボタンでダブルクリックしても同じです。

[開く]

ウインドウリストで選択している編集ウインドウを最前面に配置し、[ウインドウ選択]ダイアログボックスを閉じます。

[閉じる]

ウインドウリストで選択している編集ウインドウを閉じます。

[終了]

[ウインドウ選択]ダイアログボックスを閉じます。

ヘルプメニュー

オンラインヘルプの表示と、Peggyバージョンの表示を行います。

ボタン	メニュー操作/機能	キー操作
	[ヘルプ]-[ヘルプ]	F1
	状況ヘルプ	Shift+F1
	[ヘルプ]-[目次]	
	[ヘルプ]-[トピックの検索]	
	[ヘルプ]-[キーワードヘルプ]	Ctrl+F1

	[ヘルプ]-[ユーザーズマニュアル]	
	[ヘルプ]-[バージョン情報]	

[ヘルプ]

F1キーを押すと、その状況にあったヘルプトピックを表示します。MocaScript やキーワード定義(CREC)ファイル編集中は、キーワードや関数の上にカーソルを置いて F1 キーを押すと、その関数やキーワードのトピックが表示されます。

[状況ヘルプ]

このコマンドを選択すると、マウスカーソルが［ ］に変化します。そのままヘルプを見たいボタンをクリックするか、メニューからコマンドを選択すると、対応するトピックが表示されます。

[目次]

ヘルプの最初の画面を表示します。この画面から始まってPeggyの操作手順やリファレンスなど関連情報の項目のヘルプを読むことができます。ヘルプ項目を参照した後、ヘルプの最初の画面に戻るには、ヘルプウインドウの[目次]ボタンを選択します。

[トピックの検索]

キーワードによるヘルプトピックの検索ができます。

[キーワードヘルプ]

カーソル位置の単語または選択範囲の文字列を、プロジェクトに登録しているヘルプファイルから直ちに検索し、ヘルプを表示します。ヘルプファイルは拡張子が.hlpか.chmであるものが対象です。このコマンドを効果的に使うには次のようにします。

1. 編集中のファイルと、関連するヘルプファイルをプロジェクトの同じグループに登録します。
   例えば、Ｃ言語のソースファイルを編集中の場合、Ｃコンパイラのヘルプファイルを同じグループに登録します。
2. カーソルを、調べたい単語に移動させます。
3. Ctrl+F1キーをタイプします。
4. カーソル位置の単語または選択範囲の文字列をキーワードとして、登録したヘルプファイルから検索し表示します。上の例ですと、Ｃコンパイラのヘルプファイルを検索します。

複数の言語を編集する場合、それぞれのファイルを別々のグループに登録し、Ｃ言語のグループには、Ｃコンパイラのヘルプファイル、SQLクエリーのグループにはデータベースのヘルプファイルというように登録しておけば、複数のヘルプファイルを使い分けることもできます。

【メモ】同じグループにヘルプファイルが登録されていない場合は、その上位グループ、そこにも無ければさらにその上というように、上位グループへ向かってヘルプファイルを探していきます。なお、プロジェクトに登録していないファイルの場合は、プロジェクトのルートだけを探します。

[ユーザーズマニュアル]

PDFマニュアル<peggy.pdf>がインストールフォルダにあれば、それを表示します。

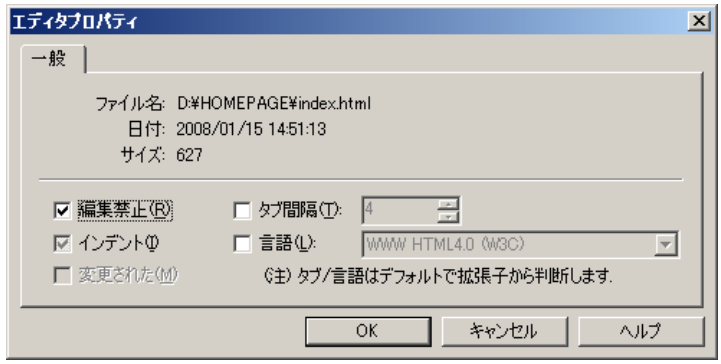
[バージョン情報]

Peggyのバージョン情報を表示します。シリアル番号とライセンスキーもこのダイアログボックスから設定します。

エディタプロパティ

プロジェクトに登録していないファイルのプロパティです。

メニュー操作/機能	ショートカットキー
[編集]-[プロパティ]	Alt+Enter



[ファイル名]

現在編集中のファイル名を表示します。

[日付]

ファイルの最終変更時間を表示します。

[サイズ]

現在編集中のファイルサイズを表示します。

[編集禁止]

ファイルが編集禁止属性のとき、チェックが付きます。強制的に編集を許可する場合、このチェックをOFFにし、編集後、[名前を付けて保存]することができます。

[インデント]

オートインデントの設定ができます。デフォルトでは[ツール]-[オプション]-[編集]ページでの設定が適用されます。

[変更された]

編集中のファイルが変更されたらチェックが付きます。

[タブ間隔]

タブ間隔の設定ができます。デフォルトでは、拡張子マップで指定された拡張子と言語の関係におけるタブ間隔が適用されます。

［言語］

言語を強制的に変更できます。デフォルトでは、拡張子マップで指定された拡張子と言語の関係が適用されます。

コマンドライン引数

コマンドラインからの起動

Peggyをコマンドラインから起動したり、他のアプリケーションから呼び出すように設定することができます。

Peggy のコマンドラインオプション

オプション	記述例	機能
ファイル名	Peggy sample.c main.asm	Peggy起動時に<sample.c>、<main.asm >を開きます。
-L /L	Peggy -L15.10 sample.c	Peggy起動時に<sample.c>を開き、カーソルを15行目に設定します。行番号に続けてドット(.)文字インデックスを指定すると、その文字の先頭へカーソル移動します。行頭の文字が1で次が2です。全角文字は2とカウントされます。文字位置は省略することができます。
-KS /KS	Peggy -KS sjistext.txt	<sjistext.txt>をShift JISファイルとして開きます。
-KE /KE	Peggy -KE euctext.txt	<euctext.txt>をEUCファイルとして開きます。
-KJ /KJ	Peggy -KJ jistext.txt	<jistext.txt>をJISファイルとして開きます。
-K7 /K7	Peggy -K7 utf7text.txt	<utf7text.txt>をUTF7ファイルとして開きます。
-K8 /K8	Peggy -K8 utf8text.txt	<utf8text.txt>をUTF8ファイルとして開きます。
-KU /KU	Peggy -KU unitext.txt	<unitext.txt>をUnicodeファイルとして開きます。
-RO /RO	Peggy -RO readme.txt	すぐ後のファイルを編集禁止モードで開きます。
-RW /RW	Peggy -RW readme.txt	すぐ後のファイルを編集可能モードで開きます。
-p /p	Peggy -p sample.c	Peggyを起動せずに<sample.c>を標準のプリンタに印刷出力します。
-d /d	Peggy -d file1 file2	Peggyを起動したあと<file1>と<file2>の比較を実行します。

-n /n	Peggy -n	新規ファイルを開きます。
-np /np	Peggy -np	プロジェクトウインドウを閉じた状態で起動します。
-sp /sp	Peggy -sp	プロジェクトウインドウを開いた状態で起動します。
-no /no	Peggy -no	アウトプットウインドウを閉じた状態で起動します。
-so /so	Peggy -so	アウトプットウインドウを開いた状態で起動します。
-nw /nw	Peggy -nw	ウインドウリストを閉じた状態で起動します。
-sw /sw	Peggy -sw	ウインドウリストを開いた状態で起動します。
-wm /wm	Peggy -wm	ウインドウを最大化した状態で起動します。
-wn /wn	Peggy -wn	ウインドウを最大化しない状態で起動します。
-mi /mi	Peggy -mi	二重起動を許していない場合でも、新しいPeggyを起動します。
-si /si	Peggy -si	二重起動を禁止します。オプションより優先します。
-nomru /nomru	Peggy -nomru sample.c	起動時に<sample.c>を開きますが、[最近使ったファイル]リストに追加しません。
-phplog /phplog	Peggyphp -phplog	標準出力と標準エラー出力のりダイレクトを除く PHP/Xdebug と Peggy Pro for PHP の通信ログを出力します。
-phplogall /phplogall	Peggyphp -phplogall	標準出力と標準エラー出力のりダイレクトを含んだ完全な PHP/Xdebug と Peggy Pro for PHP の通信ログを出力します。

【注意】Peggy Proの実行ファイル名は<peggypro.exe>、Peggy Pro for PHPの実行ファイル名は<peggyphp.exe>です。

【注意】-phplog、-phplogall オプションは、Peggy Pro for PHP 専用です。このオプションは、PHPデバッグ機能のトラブルシューティングに使います。通信ログは、<peggyphp.exe> と同じフォルダのファイル <peggyphp.log> へ追加モードで出力されます。詳しくは、PHPデバッグのトラブルシューティングを参照してください。

【ヒント】オプションフラグは、大文字、小文字とも使えます。

【例】コンパイラメーカーの統合ツールからPeggyでソースファイルを開き、エラータグジャンプしたい場合は、以下のように引数を設定してください。ファイル名が %F、行番号が %L で引き渡される場合。

```
Peggy %F -L%L
```

と登録することで、エラータグジャンプができます。また、Peggyのオプションで、[二重起動を許す]のチェックをはずしておけば、Peggyが複数起動されずに、編集ウインドウが順に開きます。

第 4 章　検索・比較・置換

このリファレンスでは、文字列検索やファイル比較に関するコマンドを分類し、そのコマンド一覧とより詳細な使い方を記載しています。

検索

Peggyは豊富な検索機能があります。

検索

現在編集中のテキスト、または複数のファイルから文字列を検索します。

再検索

前回と同じ検索条件で「検索」を再実行します。ダイアログボックスは表示されません。

クイック検索

カーソル位置の単語または選択範囲の文字列を、現在編集中のテキストから検索します。ダイアログボックスは表示されません。

ファイルから検索

プロジェクトに登録しているファイル、または指定したフォルダにあるファイルから文字列を検索します。一度に複数のファイルから検索することができます。

このファイルから検索

プロジェクトウインドウで現在選択しているファイルから検索します。

プロジェクトから検索・グループ以下から検索・グループから検索

プロジェクトに登録しているファイルから、カーソル位置の単語または選択範囲の文字列を検索します。ダイアログボックスは表示されません。「クイック検索」と「ファイルから検索」を組み合わせたものです。

シンボル検索

色々な言語の関数や変数の定義位置を抽出したファイル<TAGS>を作成し、それらをリスト表示したり、その定義位置へ簡単にジャンプできます。

共通の検索条件の切り替え

以下は、すべての検索機能に共通の検索条件を切り替えるコマンドです。設定された条件が以後、すべての検索機能で有効になります。

ボタン	メニュー操作	キー操作	機能
aA	[検索]-[大小文字の区別]	Ctrl+E	検索条件[大文字と小文字を区別する]のON/OFFを切り替えます。
W	[検索]-[単語として検索]	Ctrl+W	検索条件[単語単位で探す] のON/OFFを切り替えます。
[^]	[検索]-[正規表現]	Ctrl+Shift+E	検索条件[正規表現]のON/OFFを切り替えます。
	[検索]-[あいまい検索]		あいまい検索のON/OFFを切り替えます。

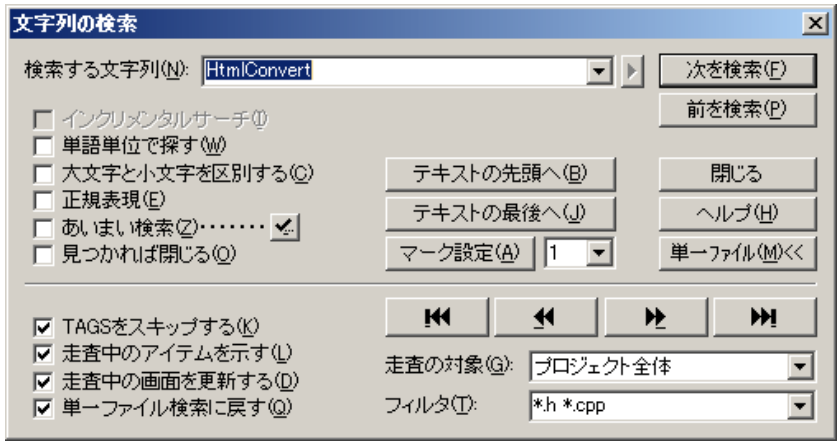
【注意】[大文字小文字の区別]、[単語として検索]は、正規表現検索においても意味を持ちます。

【注意】正規表現とあいまい検索は同時に指定できますが、一部制限があります。

検索

編集中のテキスト、または複数のファイルを対象に文字列を検索します。

ボタン	メニュー操作	キー操作
	[検索]-[検索]	Alt+F3
	検索ツール	Ctrl+F
	[検索]-[再検索]-[次を検索]	F3
	[検索]-[再検索]-[前を検索]	Shift+F3



[検索する文字列]

検索する文字列を入力します。以前に検索した文字列を覚えているので、リストから選ぶことも可能です。[正規表現]をチェックしているときは、正規表現パターンでも検索できます。右にある[▶]を押すと、正規表現のメニューがポップアップするので、選択するだけで正規表現を入力できます。

[次を検索]

テキストの終わりに向かって文字列を検索します。

[前を検索]

テキストの先頭に向かって文字列を検索します。

[閉じる]

このダイアログボックスを閉じます。Escキーでも閉じることができます。

[ヘルプ]

オンラインヘルプを表示します。

[テキストの先頭へ]

検索を始める位置をテキストの先頭へ移動します。ファイルの途中まで検索した後で、先頭からやり直したいときなどに便利です。

[テキストの最後へ]

検索を始める位置をテキストの最後へ移動します。

[マーク設定]

文字列の見つかった行すべてにブックマークを付けます。このダイアログボックスを閉じた後で、F2、Shift+F2キーを押してブックマークを順番に見てゆくことができます。右隣の数字でブックマークの種別を指定します。

[インクリメンタルサーチ]

[次を検索]、[前を検索]ボタンをクリックしなくても、[検索する文字列]に1文字タイプする毎に、検索を実行します。

[単語単位で探す]

長い単語の一部ではなく、全体が一致する単語を検索するにはこれをチェックします。

[大文字と小文字を区別する]

これをチェックすると、半角アルファベットの大文字と小文字を区別して検索します。なお、全角文字の仮名、たとえば「あ」と「ぁ」は常に別の字として扱います。

[正規表現]

これをチェックすると、正規表現による文字列検索が有効になります。

[あいまい検索]

これをチェックすると、現在の「あいまい検索の設定」に従ったあいまい検索が有効になります。あいまい検索の設定を変更するには、右側にある[ ]ボタンをクリックしてください。[あいまい検索の設定]ダイアログボックスが表示されます。正規表現とあいまい検索は同時に指定できますが、一部制限があります。詳しくは、正規表現とあいまい検索を参照してください。

[見つかれば閉じる]

[次を検索]、[前を検索]で文字列が見つかれば、自動的にダイアログボックスを閉じます。ダイアログボックスが閉じられた後でも、F3、Shift+F3キーで続けて検索可能です。

[複数ファイル・単一ファイル]

複数のファイルから検索するときに押します。複数ファイル検索のためのオプションが表示されます。すでに表示されている場合は隠します。

[TAGS をスキップする]

<TAGS>ファイルを検索の対象から除外します。

[走査中のアイテムを示す]

検索中のテキストが、プロジェクトウィンドウのどの位置にあるかを表示します。

[走査中の画面を更新する]

検索中のテキストを編集ウィンドウに順に表示します。検索文字列が含まれないファイルも表示されるため、画面がちらつくことがあります。

[単一ファイル検索に戻す]

このダイアログボックスを閉じた後、再び検索を行うとき、「複数ファイル」検索のオプションが表示されず、通常の検索に戻ります。

[走査の対象]

検索の対象となるファイルの範囲を指定します。[開いているファイル]、[グループ内のファイル]、[グループ以下のファイル]、[プロジェクト全体]の４つから選択します。

【注意】プロジェクト中に同じファイルが何度の登録されている場合、そこを通過するたびに、そのファイルを検索します。

【注意】検索の途中で[走査の対象]を変更すると、その時点でカーソルのあるファイルを基準に新しく走査の対象を再設定したと見なします。例えば、[グループ内のファイル]を選択すると、現在カーソルのあるファイルが登録されているプロジェクトグループ内のファイルが新しい走査対象となります。

[フィルタ]

ワイルドカードを使って走査対象のファイルを絞り込みます。スペースで区切ることで複数のパターンを指定できます。

【注意】フィルタを変更した場合でも、現在カーソルのあるファイルは検索の対象となります。フィルタに一致するファイルまで移動するには、[ ]、または[ ]ボタンをクリックしてください。

[ ]

検索対象のファイルの中で、検索の順序が最初のファイルを表示し、検索開始位置もファイルの先頭に戻します。

[ ]

検索の順序が一つ前のファイルを表示し、検索開始位置をそのファイルの先頭に置きます。

[ ]

検索の順序が次のファイルを表示し、検索開始位置をそのファイルの先頭に置きます。

[ ]

検索対象のファイルの中で、検索の順序が最後のファイルを表示し、検索開始位置もファイルの最後に置きます。

【ヒント】任意のフォルダ内の複数ファイルを対象として検索する場合は、[ファイルから検索] コマンドを使います。

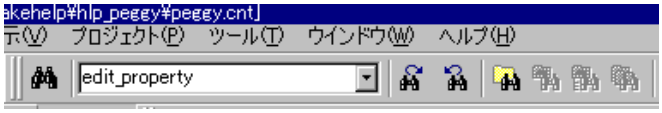
【ヒント】[ファイルから検索] コマンドの場合、検索の対象はファイルとして保存されていないといけません。しかしこの複数ファイル検索であれば、編集途中のテキストでも検索することができます。

【ヒント】検索対象のファイルが多いと、検索に時間がかかることがありますが、Esc キーで中断することができます。

検索ツール

ツールバーに配置した検索ツールへ文字列をタイプし、編集中のテキストを対象に検索します。検索する文字列の初期値は前回行った検索文字列です。20回分の履歴を覚えていますので、コンボボックスから選択することもできます。

ボタン	メニュー操作/機能	キー操作
	検索ツール	Ctrl+F



F3キーまたはEnterキーで検索を実行します。Shift+F3またはShift+Enterキーで上方向へ検索を行います。F3、Shift-F3キーを使った検索で文字列が見つかると、編集ウインドウへカーソルが移動します。その他にも、以下のような色々な検索が可能です。

検索ツールでのキー操作	動作
Enter	編集ウインドウを終りに向かって検索する。
Shift+Enter	編集ウインドウを先頭に向かって検索する。
F3	編集ウインドウを終りに向かって検索し、見つかれば編集ウインドウへカーソルを移す。
Shift+F3	編集ウインドウを先頭に向かって検索し、見つかれば編集ウインドウへカーソルを移す。
F5	編集ウインドウを上下に分割してから終りに向かって検索し、見つかれば編集ウインドウへカーソルを移す。
Shift+F5	編集ウインドウを上下に分割してから先頭に向かって検索し、見つかれば編集ウインドウへカーソルを移す。
Ctrl+F	[文字列の検索]ダイアログボックスを表示する。
Ctrl+R	[文字列の置換]ダイアログボックスを表示する。
F11、Shift+F11	現在のグループから文字列を(Shiftで追加)検索する。
Ctrl+F11、Ctrl+Shift+F11	現在のグループ以下から文字列を(Shiftで追加)検索する。
Alt+F11、Ctrl+Alt+F11	プロジェクト全体から文字列を(Shiftで追加)検索する。
F12	検索ツールの文字列が定義されている個所へジャンプする。
Ctrl+F1	検索ツールの文字列をヒントとして、キーワードヘルプを呼び出す。
F1	検索ツールのオンラインヘルプを表示します。
Esc	編集ウインドウへカーソルを移動させます。

再検索

現在編集中のテキストから前回と同じ検索条件で「検索」を再実行します。

ボタン	メニュー操作/機能	キー操作
	[検索]-[再検索]-[次を検索]	F3
	[検索]-[再検索]-[前を検索]	Shift+F3
	[検索]-[再検索]-[別ペインで次を検索]	F5
	[検索]-[再検索]-[別ペインで前を検索]	Shift+F5

[次を検索]

現在の条件のままで、カーソル位置から下方向へ検索を行います。

[前を検索]

現在の条件のままで、カーソル位置から上方向へ検索を行います。

[別ペインで次を検索]

まず編集中の編集ウインドウが上下に分割されます。上のペインは現在のカーソルがある位置を表示し、下のペインでは現在の検索条件で、カーソル位置から下方向へ検索を行います。

[別ペインで前を検索]

まず編集中の編集ウインドウが上下に分割されます。上のペインは現在のカーソルがある位置を表示し、下のペインでは現在の検索条件で、カーソル位置から上方向へ検索を行います。

クイック検索

[文字列の検索]ダイアログボックスを呼び出さずに、カーソル位置の単語または選択範囲の文字列を、検索することを「クイック検索」と言います。

ボタン	メニュー操作/機能	キー操作
	[検索]-[クイック検索]-[次の単語]	Ctrl+F3
	[検索]-[クイック検索]-[前の単語]	Ctrl+Shift+F3
	[検索]-[クイック検索]-[別ペインで次の単語を検索]	Ctrl+F5
	[検索]-[クイック検索]-[別ペインで前の単語を検索]	Ctrl+Shift+F5

[次の単語]

カーソル位置の単語を下方向へ検索します。

[前の単語]

カーソル位置の単語を上方向へ検索します。

[別ペインで次の単語]

まず編集中の編集ウインドウが上下に分割されます。上のペインは現在のカーソルがある位置を表示し、下のペインでカーソル位置の単語を下方向へ検索します。

[別ペインで前の単語]

まず編集中の編集ウインドウが上下に分割されます。上のペインは現在のカーソルがある位置を表示し、下のペインでカーソル位置の単語を上方向へ検索します。

【ヒント】検索したい文字列に記号や空白が含まれている場合は、まずその部分を選択してからキーを押します。

【注意】単語として探すか、英大文字小文字を区別するかは現時点の設定に従います。しかし、エディタのカーソル位置から取得した文字列が正規表現パターンであるとは考えにくいので、常に正規表現ではないとして検索します。

ファイルから検索

ファイルから文字列を検索します。

ボタン	メニュー操作/機能	キー操作
	[検索]-[ファイルから検索]	Ctrl+Shift+G

ファイルから検索の基本操作

3. [検索する文字列] に探したい文字列をタイプします。
4. 検索対象のファイル、検索のオプションなどを指定してから、
5. [検索] ボタンをクリックします。検索結果は、アウトプットウインドウの [ファイルから検索] ページに出力されます。
6. メニューの [編集]-[タグジャンプ]-[次のタグ]、または F4 キーで順番に見つかった文字列の場所にジャンプして行くことができます。

メニューの[編集]-[タグジャンプ]-[前のタグ]、またはShift+F4キーで逆順ジャンプも可能です。アウトプットウインドウに出力されている検索結果の上をマウスでダブルクリックするか、その行へカーソルを置いてEnterキーを押しても、そこへジャンプできます。

ファイルから検索の中止

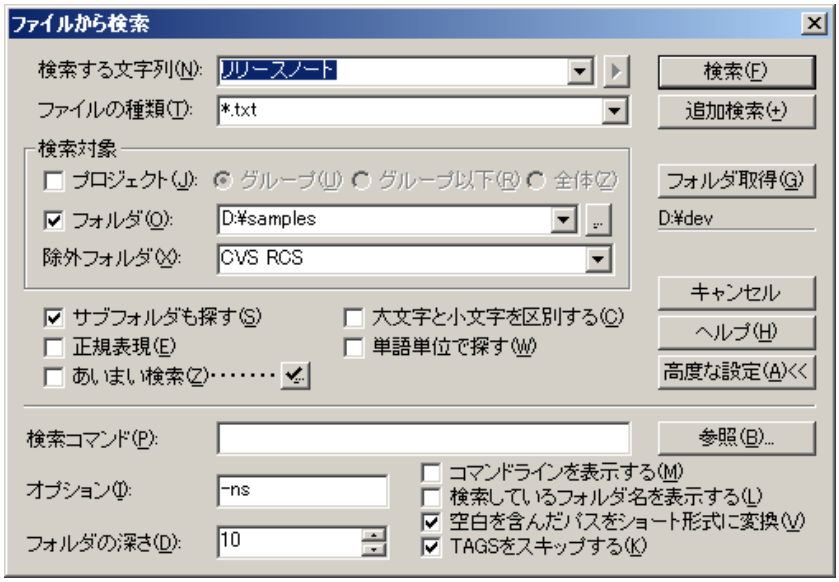
[プロジェクト]-[実行中止]を選択します。ツールバーの[ ]でも中止することができます。

検索対象ファイルの指定方法

[検索対象]に[プロジェクト]と[フォルダ]という２つのチェックボックスがあり、チェックの仕方で次の３つに分類できます。

指定方法	特徴
[プロジェクト]だけをチェック	プロジェクトに登録しているファイルの内、タイプが[テキスト]か[スクリプト]のものだけが対象。同じパスのファイルが複数登録されていても１度しか検索しないようにしています。
[フォルダ]だけをチェック	指定のフォルダ内にあるファイルが対象。 【注意】指定のフォルダ内に、実行可能ファイルなどのバイナリファイルがあるとそれも検索してしまいます。[ファイルの種類]と[除外フォルダ]を使って、テキストファイルだけ検索するようにしてください。
両方をチェック	【注意】<d:¥main.c>をプロジェクトに登録していて、[フォルダ]に<d:¥>を指定すると、<d:¥main.c>は２回検索され、検索結果も２回出力されます

【注意】バイナリファイルを検索すると、検索結果を表示する行が乱れてしまい、正しくタグジャンプできないことがあります。できるだけバイナリファイルを避けるようにしてください。[プロジェクト]を検索する場合は、その心配はいりません。



[検索する文字列]

検索する文字列を入力します。以前に検索した文字列も覚えているので、リストから選ぶことも可能です。[正規表現]をチェックしていれば、正規表現パターンでも検索できます。右にある[▶]ボタンを押すと、正規表現のメニューがポップアップするので、選択するだけで正規表現を入力できます。

[ファイルの種類]

検索対象のファイルをワイルドカードを使って、絞り込むときに使用します。この絞り込みは次に説明する[プロジェクト]と[フォルダ]両方に対して有効です。複数指定する場合は半角空白で区切ってください。指定しない場合は、*.*が入力されたと見なします。

【例】*.c *.h *.asm

[プロジェクト]

プロジェクトに登録しているファイルの中から文字列を検索する場合、ここをチェックします。さらに、３つのラジオボタンの中から対象範囲を選択します。

ボタンの名前	対象範囲
[グループ]	現在選択しているグループ内のファイル
[グループ以下]	現在選択しているグループ内とその下のサブグループ内すべてのファイル
[全体]	プロジェクトに登録しているすべてのファイル

Shift JIS、EUC、JIS、Unicode、UTF-7、UTF-8のファイルから文字列を検索できます。詳しくは、『効果的な編集7 - 漢字コードと改行コード』の表を参照してください。

[フォルダ]

ここをチェックすると、指定のフォルダ内のファイルを検索対象とします。[...]ボタンを押せば、フォルダを選択入力できます。エクスプローラからフォルダをドラッグ&ドロップすることも可能です。また、[フォルダ取得]ボタンを押せば、ボタンの下に表示しているフォルダが入ります。プロジェクトに登録していないファイルも検索します。

[除外フォルダ]

ここに指定したフォルダ内のファイルを検索対象から外します。その下のサブフォルダもすべて再帰的に検索対象から除外されます。フォルダは半角空白で区切っていくつでも指定できます。[サブフォルダも探す]をチェックしたが一部のサブフォルダは検索したくない場合に使います。

【例】object output RCS

[検索]

検索を開始します。

[追加検索]

[検索]と同様に検索を開始しますが、前回の検索結果を消さずに、検索結果をその後に追加表示します。

[フォルダ取得]

このボタンの下に現在編集中のファイルまたは選択されているプロジェクトのアイテムのフォルダが表示されています。このボタンを押すと、[フォルダ]欄にそのフォルダが入ります。

[サブフォルダも探す]

[フォルダ]で指定したフォルダ内のファイルだけでなく、その下にあるサブフォルダも再帰的に検索します。

[正規表現]

正規表現による文字列検索を有効にします。

[あいまい検索]

これをチェックすると、現在の「あいまい検索の設定」に従ったあいまい検索が有効になります。あいまい検索の設定を変更するには、右側にある[ ]ボタンをクリックしてください。

[大文字と小文字を区別する]

これをチェックすると、半角アルファベットの大文字と小文字を区別して検索します。なお、全角文字の仮名、たとえば「あ」と「ぁ」は常に別の字として扱います。

[単語単位で探す]

長い単語の一部ではなく、全体が一致する単語を検索するにはこれをチェックします。

[高度な設定]

より高度な設定のオプションを表示します。すでに表示されている場合は隠します。

[検索コマンド]

何も指定しないときは、標準の検索コマンド<pgrep.exe>を使ってファイルから検索します。GNU grepのような標準以外のコマンドを使ってファイルから検索を行いたい場合は、ここへ指定してください。

【注意】標準以外の検索コマンドでは、漢字コードの自動判定を行いません。

【注意】外部コマンドとしてGNU grepを起動し実際の検索を行う場合は、サポートしている正規表現の仕様が若干異なっています。

[オプション]

実際の検索コマンドへ与える追加のオプションを指定します。どのようなオプションが使えるかは検索コマンドによるため、詳しくはそのコマンドのマニュアルなどを参照してください。デフォルトは-nsです。行番号を出力させるnと、エラーメッセージの出力を抑止するsが指定されています。

[フォルダの深さ]

[サブフォルダも探す]をチェックしている場合、どれぐらい深くまで降りて行くかを指定します。ここで指定している深さ以上には検索を行いません。

[コマンドラインを表示する]

検索コマンドを実行するときのコマンドラインをアウトプットウインドウへ表示します。検索コマンドを他のものと入れ替えたような場合、これをチェックすることで、正しいオプションが渡されているか確認できます。

[検索しているフォルダ名を表示する]

検索しているフォルダ名をアウトプットウインドウへ表示します。検索文字列を含んだファイルが見つからなくても、どのフォルダを検索したかが分かります。

[空白を含んだパスをショート形式に変換]

検索しているパスの途中に空白が含まれていると、正しく動作しない検索コマンドがあります。これを回避するため、フォルダのパス中に空白文字があると、ショートパス形式に変換してから検索コマンドを起動します。

[TAGS をスキップする]

<TAGS>ファイルを検索の対象から除外します。

【注意】[大文字小文字の区別]、[単語として検索]は、正規表現検索においても意味を持ちます。

プロジェクトから検索・グループ以下から検索・グループから検索

[ファイルから検索]ダイアログボックスを呼び出さずに、カーソル位置の単語または選択範囲の文字列を、直ちに検索します。

ボタン	メニュー操作	キー操作
	このファイルから検索	
	このファイルから追加検索	
	[検索]-[グループから検索]	F11
	グループから追加検索	Shift+F11
	[検索]-[グループ以下から検索]	Ctrl+F11
	グループ以下から追加検索	Ctrl+Shift+F11
	[検索]-[プロジェクトから検索]	Alt+F11
	プロジェクトから追加検索	Alt+Shift+F11

このファイルから検索

プロジェクトウインドウで現在選択されているテキストファイルから検索して、結果をアウトプットウインドウへ出力します。

このファイルから追加検索

プロジェクトウインドウで現在選択されているテキストファイルから塚検索して、結果をアウトプットウインドウへ追加出力します。

[グループから検索]

プロジェクトウインドウで現在選択されているグループ内のファイルが検索対象です。

グループから追加検索

プロジェクトウインドウで現在選択されているグループ内のファイルを検索し、結果をアウトプットウインドウへ追加します。

[グループ以下から検索]

プロジェクトウインドウで現在選択されているグループとそのサブグループ内のファイルが検索対象です。

グループ以下から追加検索コマンド

プロジェクトウインドウで現在選択されているグループとそのサブグループ内のファイルを検索し、結果をアウトプットウインドウへ追加します。

[プロジェクトから検索]

プロジェクトに登録してあるファイルすべてが検索対象です。

プロジェクトから追加検索コマンド

プロジェクトに登録してあるすべてのファイルを検索し、結果をアウトプットウインドウへ追加します。

【メモ】Shift JIS、EUC、JIS、Unicode、UTF-7、UTF-8のファイルから検索できます。ただしUnicode独自の文字など、Shift JIS文字セットに含まれていない文字は検索できません。

【ヒント】検索したい文字列に記号や空白が含まれている場合は、まずその部分を選択してからキーを押します。

【注意】単語として探すか、英大文字小文字を区別するかは現時点の設定に従います。しかし、エディタのカーソル位置から取得した文字列が正規表現パターンであるとは考えにくいので、常に正規表現ではないとして検索します。

検索強調表示

検索強調表示とは、編集しているファイルで行った検索に一致した部分すべてにアンダーラインを引いて強調表示する機能です。検索に一致した部分をまとめて確認できます。

ボタン	メニュー操作/機能	キー操作
	[検索]-[検索強調表示]	Ctrl+Alt+F3
	検索強調クリア	
	[検索]-[検索強調全クリア]	Alt+Shift+F3

[検索強調表示]

検索強調表示をするかしないかを選択します。このコマンドを選択するたびに、強調表示する・しないを切り替えます。

検索強調クリア

現在編集しているファイルの検索強調表示をクリアします。

[検索強調全クリア]

開いているすべてのファイルの検索強調表示をクリアします。

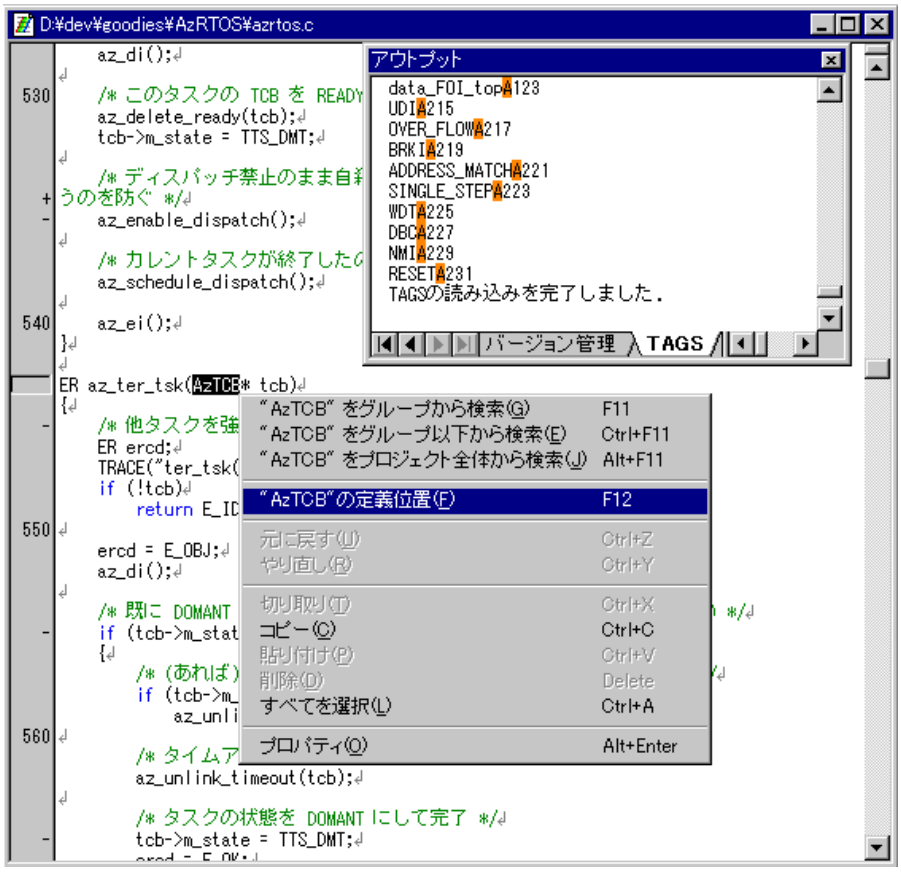
【ヒント】検索強調表示のアンダーラインの色と太さは、[表示色]ページの[基本-検索強調表示]でカスタマイズできます。

【ヒント】検索強調の動作は、[ツール]-[オプション]-[編集]ページの[編集オプション]リストにある[検索・比較・タグジャンプ]-[検索強調表示]でカスタマイズできます。

【注意】検索条件に行末の改行が含まれているときは、強調表示できません。

シンボル検索

シンボル検索では、色々な言語の関数や変数の定義位置を抽出したファイル<TAGS>を作成し、それらをリスト表示したり、その定義位置へ簡単にジャンプすることができます。



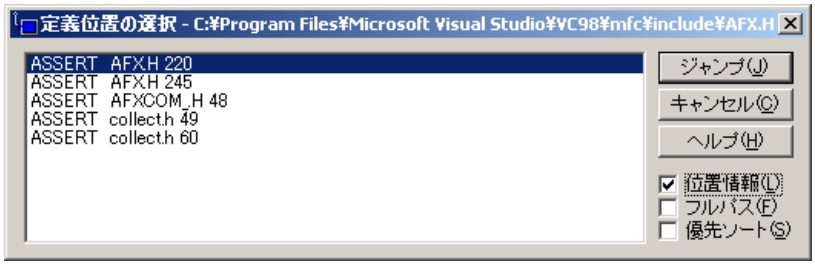
ボタン	メニュー操作	キー操作
	[検索]-[シンボル検索]-[定義位置]	F12
	[検索]-[シンボル検索]-[最初の定義位置]	
	[検索]-[シンボル検索]-[最後の定義位置]	
	[検索]-[シンボル検索]-[次の定義位置]	Ctrl+F12
	[検索]-[シンボル検索]-[前の定義位置]	Ctrl+Shift+F12
	[検索]-[シンボル検索]-[次のジャンプ個所]	Ctrl+Shift+F
	[検索]-[シンボル検索]-[前のジャンプ個所]	Ctrl+Shift+B
	[検索]-[シンボル検索]-[定義位置ブラウザ]	Alt+F12
	[検索]-[シンボル検索]-[TAGSの作成]	Shift+F12

【ヒント】このシンボル検索をするためのタグファイル<TAGS>を生成するために、etagsコマンド、またはctagsコマンドを使用しています。

シンボル定義を参照

ファイルを編集中に、カーソル位置のシンボルの定義を調べるのに便利です。以下の手順で検索できます。

1. 検索したいシンボル文字列の上にカーソルを移動します。
2. [検索]-[シンボル検索]-[定義位置]を実行します。[ ]ボタンでも同じです。マウス右ボタンのポップアップメニューから[“xxx”の定義位置]メニューを選択するか、キーボードからF12を押すことでも実行できます。
3. <TAGS>ファイルがない場合は生成されます。
4. 目的のシンボルが定義してあるファイルが開き、カーソルがその行を示します。
5. 同じ名前のシンボルが複数個所で定義されている場合は、下図のようなダイアログボックスが開きます。目的のシンボル定義と思われるものを選択し、[ジャンプ]ボタンを押します。[優先ソート]をチェックすると、現在編集しているファイルに含まれる行を最初に表示します。



【メモ】2回目以後の検索では<TAGS>ファイルは生成されません。また、<TAGS>ファイルには、表示不可能なASCIIコードが入っていますので、アウトプットウインドウで正しく表示できない文字があります。<TAGS>ファイルを直接ユーザが意識する必要はありません。

前のジャンプ個所・次のジャンプ個所

シンボルの定義位置を参照した後は、編集していたファイルへ戻りたいことがよくあります。そのようなときは、[ ]で簡単に元の場所へ戻ることができます。戻り過ぎたときは、[ ]で先へ進めます。

あいまいなシンボル定義を閲覧

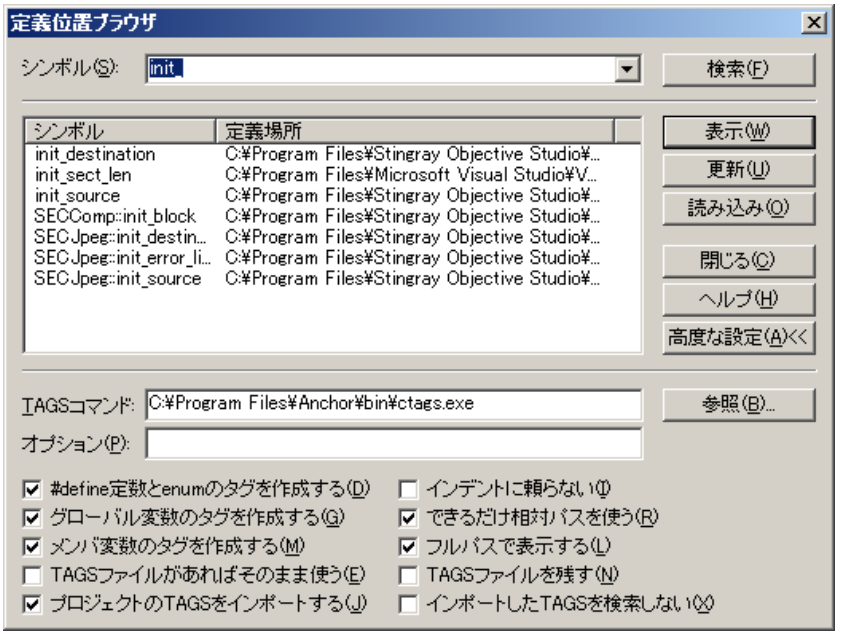
1. [検索]-[シンボル検索]-[定義位置ブラウザ]を実行します。[ ]ボタンでも同じです。[定義位置ブラウザ]ダイアログボックスが表示されます。
2. <TAGS>ファイルがない場合は、[更新]ボタンをクリックして作成します。
3. [定義位置ブラウザ]ダイアログボックスの[シンボル]に目的のシンボルの始めの何文字かを入力します。[検索]を押すと、文字列の一致するシンボル定義がすべて表示されます。

TAGS の作成

TAGSを作り直します。PeggyはTAGSを作成してからの各ファイルの行数の増減を追跡しているので、少しくらいの編集では正しくジャンプすることができます。しかし、プロジェクトへファイルを追加したり、ファイルに大幅な変更を加えた場合は、[ ]ボタンにより、TAGSを作り直してください。

[定義位置ブラウザ] ダイアログボックス

曖昧なシンボルの定義位置を検索したり、シンボル定義位置ファイル<TAGS>を作成するための条件を設定します。



[シンボル]

検索対象の文字列（シンボル）を入力します。曖昧なシンボルを検索するには、先頭の数文字を入力します。大文字小文字は区別します。

[検索]

現在のアウトプットウインドウの[TAGS]ページから検索を開始し、結果をリスト表示します。リスト項目をダブルクリックすれば、定義位置を編集ウインドウで確認できます。

[表示]

検索結果リストの選択されたファイルを表示し、定義位置を確認できます。リスト上をダブルクリックしたのと同じです。

[更新]

ファイルを大幅に編集した場合など、行数が大きく変化したときは<TAGS>ファイルを再生成する必要があります。小規模な編集であれば、Peggyが行数の増減を補正するので、<TAGS>ファイルをつくり直す必要はありません。

[読み込み]

[更新]のように新しく<TAGS>ファイルを作成するのではなく、すでにある<TAGS>ファイルを読み込みます。

高度な設定

[TAGS コマンド] [オプション]

TAGSを作成するコマンドを指定したいときに設定します。デフォルト(ctags)では、Peggyをインストールしたフォルダの<bin¥ctags.exe>が使われます。etags.exeに変更するには、パスを設定するか、空欄にしてください。

[#define 定数と enum のタグを作成する]

C言語の#defineマクロと、enum定数のタグを作成します。これをOFFにした場合は、#defineやenumの定義は、検索できなくなります。

[グローバル変数のタグを作成する]

グローバル変数も検索対象とするタグを作成するときはチェックします。

[メンバ変数のタグを作成する]

メンバ変数も検索対象とするタグを作成するときはチェックします。

[インデントに頼らない]

カラム1にある閉じ括弧を関数や定義の終わりとして扱わないようにします。

[できるだけ相対パスを使う]

これをチェックすると、TAGS作成コマンドに渡すファイル名を相対パスにします。フォルダ名、ファイル名に全角文字や空白文字を含んでいると正常に動作しない場合がありますが、相対パスにすることで、その問題を回避できる可能性が高くなります。

[フルパスで表示する]

検索結果のファイル名をフルパスでリスト表示します。

[TAGS ファイルがあればそれを使う]

<TAGS>ファイルが必要になったとき、プロジェクトファイルのあるフォルダに<TAGS>という名前のファイルがあれば、そのファイルを使います。このダイアログボックスの[更新]ボタンや、ツールバーの[ ]をクリックした場合は、常に新しく作成します。

[TAGS ファイルを残す]

<TAGS>ファイルを作成してアウトプットウインドウの[TAGS]ページに読み込んだ後も、<TAGS>ファイルを削除せずに残します。<TAGS>は、プロジェクトファイルのあるフォルダ内に作成します。

【メモ】<TAGS>ファイルを残さない設定の場合、Peggyはテンポラリファイルを作成して結果を書き込み、使い終わると削除します。従って、同じフォルダにある<TAGS>という名前のファイルは、そのまま残ります。

[プロジェクトの TAGS をインポートする]

プロジェクトに登録してある<TAGS>ファイルを追加読み込みします。これはライブラリなど、変化しない部分の<TAGS>を予め作成しておくことで、<TAGS>作成の時間を節約できます。アウトプットウインドウへ読み込んだ各<TAGS>ファイルの先頭には、ブックマークが付きます。

[インポートした TAGS を検索しない]

プロジェクトに登録してある<TAGS>ファイルをインポートした部分を検索対象から外し、プロジェクトに登録されているファイルから作成した部分だけを検索しま

す。ただし、[検索]-[シンボル検索]-[最初の定義位置]コマンドや、[ ]ボタンによる検索では、常にすべての部分から検索します。

シンボル検索とプロジェクト

シンボル検索で参照する<TAGS>ファイルをどのように作成するかは、プロジェクトに登録されているファイルの設定に従います。プロジェクトにソースファイルを登録するだけで、シンボル検索機能を利用することができますが、必要に応じて細かくコントロールすることも可能です。

基本ルール

Peggyはシンボル検索のために、etagsコマンド、またはctagsコマンドを実行し、プロジェクトに登録されているファイル中のシンボル定義位置を調べ、<TAGS>ファイルを作成します。それをアウトプットウインドウの[TAGS]ページへ読み込みます。以後のシンボル検索は、[TAGS]ページからシンボルの定義位置を調べます。

etags/ctags の実行方法

etags/ctagsコマンドは、プロジェクトファイルのあるフォルダで実行し、そこへ<TAGS>ファイルを作成します。従って、書き込み可能なフォルダ内にあるプロジェクトを開いた状態でないと、シンボル検索を利用することができません。

また、新規に作成してまだ一度も保存していないプロジェクトの場合、ctags/etagsコマンドを実行するフォルダが決まらないため、シンボル検索は利用できません。まずどこかへプロジェクトを保存してください。

定義ファイル

etagsコマンドでシンボルの定義位置を抽出するためには、ファイルの言語が何かを指定しなければなりません。また、etagsが処理できる言語の制限もあります。

そこでPeggyの言語モードからetagsの言語へ対応させるために、定義ファイル<share¥etagsdef.txt>を使います。etagsが受け付ける言語名が第1カラムにあり、それに対応付けるPeggyの言語モード識別子が続いています。例えば、Cのいろいろな方言を、すべてまとめてCとして処理させています。

ctagsを使う場合は、代わりに<share¥ctagsdef.txt>を定義ファイルとして読み込みます。etagsとほぼ同じ書式により、Peggyの言語モードとctagsの言語名を結び付けます。詳しくは、<share¥ctagsdef.txt>内に記述されている説明を参照してください。

Peggyが<TAGS>ファイルを作成するときは、まずこの定義ファイルを参照します。次にプロジェクトの中を調べ、定義ファイルに記述されている順番に従ってetagsまたはctagsを起動し<TAGS>ファイルを生成します。定義ファイルにない言語モードのファイルは、シンボル定義位置を調べる対象となりません。例えば、テキストやメールのファイルは無視されます。

定義ファイルは、必要に応じてカスタマイズしてください。その際、RCSなどを使って編集履歴を残しておくとよいでしょう。特定の言語をTAGS作成から除外するには、そのエントリを削除するか、コメントアウトしてください。

【ヒント】etagsコマンドがサポートしていない言語であっても、正規表現を使うことである程度は対応することができます。例として、<share¥etagsdef.txt>ファイルにVisual BasicやRubyのタグを作成するためのサンプルがあります。参考にしてください。　ctagsも同様の方法で独自言語の解析ルールを追加できます。詳しくは、

<share¥ctagsdef.txt>ファイルと<share¥ctagsvb.txt>などのサンプルファイルを参照してください。

グループのプロパティ

プロジェクトに登録してあるファイルでも、シンボル検索の対象にしたくないものもあります。グループのプロパティに、[TAGSの対象にする]を設定することで、グループ単位に切り替えることができます。また、この設定によって、プロジェクトウインドウのグループアイコンに印が付くので、一目で確認できます。

1.	チェックなし (OFF状態)		このグループに属するファイルやサブグループをTAGS作成の対象としません。
2.	黒いチェック (ON状態)		このグループに属するファイルやサブグループをTAGS作成の対象とします。
3.	灰色チェック (ニュートラル状態)		親のグループの設定に従います。この状態がデフォルトです。

グループの階層構造の途中に、TAGSの対象としないグループ[ ]があると、それ以下のファイルをTAGS作成の対象とはしません。ただし、サブグループの中に[ ]があると、再びそのサブグループ以下はTAGS作成の対象となります。[ ]は、TAGS作成のON/OFFには影響せず、親グループの設定に従います。

すべてのグループのルートであるプロジェクトにも同じ[TAGSの対象にする]プロパティがあります。ただし、ニュートラル状態はありません。

1.	チェックなし (OFF状態)		プロジェクトに属するファイルやグループをTAGS作成の対象としません。
2.	チェックあり (ON状態)		プロジェクトに属するファイルやグループをTAGS作成の対象とします。この状態がデフォルトです。

デフォルトでは、プロジェクトがON状態、グループがニュートラル状態であるため、プロジェクトに登録されているすべてのファイルがTAGS作成の対象となります。

【ヒント】プロパティダイアログボックスを開かなくても、グループアイテムを選択してからTキー（大文字）をタイプすることで、3つの状態を順番に切り替えることができます。

【ヒント】どのグループをTAGS作成の対象にするかを設定する場合、以下の方針が参考になります。

1. プロジェクトをON状態とし、必要でないグループだけOFF状態にする。
   仕事単位に分かれた小さなプロジェクトでシンボル検索を利用するときに便利です。
2. プロジェクトをOFF状態とし、必要なグループだけON状態にする。
   複数の仕事に関連したファイルが登録してある大きなプロジェクトで便利です。必要なとき、必要な部分だけをONにしてTAGSを生成できます。

TAGS のインポート

ライブラリのヘッダファイルなど、ほとんど変化しないファイルに対して毎回ctags/etagsコマンドを実行しても、時間がかかるだけで意味がありません。そこで予め<TAGS>を作成しておき、出来た<TAGS>ファイルをプロジェクトに登録します。

TAGSを作成するとき、プロジェクト中のファイルを定義ファイルに指定されている順番で処理したあと、プロジェクトに登録してあるTAGSファイルをアウトプットウインドウの[TAGS]ページへ追加読み込みします。

【例】以下は、Microsoft Visual C++のMFCライブラリのTAGSを作成する例です。詳しくは、etagsコマンドを参照してください。

```
C:¥Program Files¥Microsoft Visual Studio¥VC98¥MFC>etags --language=c++
include¥*.h include¥*.inl src¥*.cpp
```

インポートするTAGSの中のファイルパスが相対である場合、そのTAGSがあるフォルダからと解釈します。また、インポートしたTAGSは、ライブラリなどほとんど変化しない情報を含んでいると考えられるため、これらのファイルの行数変化は追跡しません。

【ヒント】予め大きなライブラリのTAGSを作成しておくことで、TAGSを作成する時間の節約になります。また、ライブラリのソースファイルをプロジェクトに登録する必要もなくなります。

【メモ】TAGSファイルをインポートするかどうかも、そのTAGSが登録されているグループのプロパティ[TAGSの対象にする]に従います。[定義位置ブラウザ]ダイアログボックスの[プロジェクトのTAGSをインポートする]をOFFにすることで、すべてのインポートを禁止できます。

比較

高度な設定が可能なダイアログボックスを表示してから行なうファイル比較と、お手軽に使えるクイック比較があります。

ボタン	メニュー操作	キー操作
	[検索]-[比較]-[ファイル比較]	Ctrl+Shift+D
	[検索]-[比較]-[クイック比較]	Ctrl+Shift+Q

クイック比較

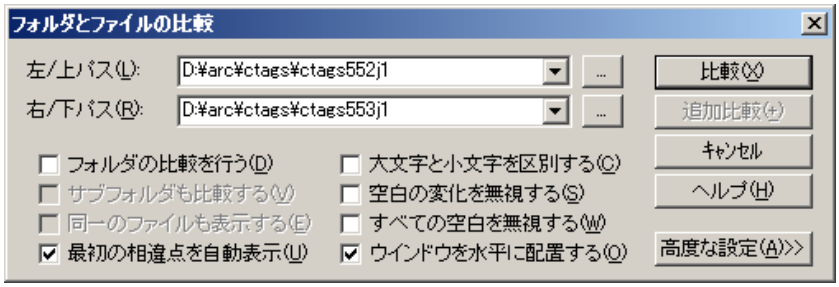
比較したいファイルを既に開いているときに便利な方法です。以下の手順で比較できます。

1. 2つ以上のファイルを開いているときは、比較するファイルのウインドウが上になるよう、まず両方のウインドウを順にクリックしてください。
2. メニューから[検索]-[クイック比較]を選択します。[ ]ボタンをクリックしても同じです。
3. 比較結果がアウトプットウインドウへ出力されます。
4. F4、Shift+F4キーのタグジャンプ機能で、相違箇所を順番に見て行けます。

【メモ】クイック比較時のその他の比較パラメタは、最後に[フォルダとファイルの比較]ダイアログボックスでの設定したものをそのまま使います。

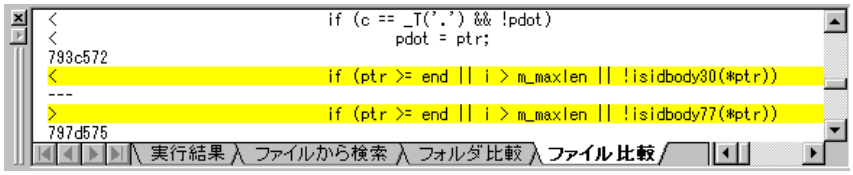
ファイル比較

ファイル比較とは、2つのテキストファイルの相違点を調べ、順番に見て行く機能です。

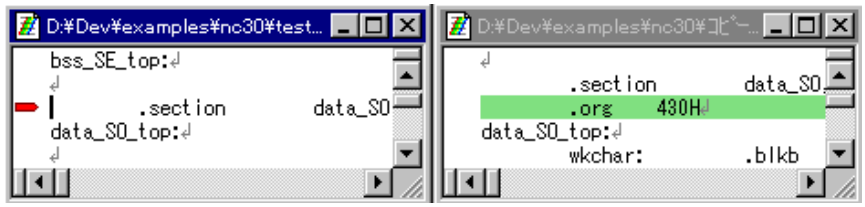


2 つのファイルを比較するには

1. まずメニューから [編集]-[ファイル比較] を選択し、[フォルダとファイルの比較] ダイアログボックスを呼び出します。
2. [フォルダの比較を行う] がチェックされていないことを確認してください。これが指定されていると、ファイルではなくフォルダの比較になります。
3. 比較したい 2 つのファイルのパスを、それぞれ [左 / 上パス] と、[右 / 下パス] にタイプします。タイプする代わりに、それぞれの [...] ボタンをクリックして、ファイルを選んでもかまいません。あるいは、エクスプローラからドラッグ＆ドロップすることもできます。
4. さらに細かい指定がある場合には、必要な比較オプションを指定します。
5. [比較] ボタンをクリックし、ファイル比較を実行してください。結果はアウトプットウインドウの [ファイル比較] ページへ出力されます。



比較の結果、２つのファイルに相違点が見つかった場合は、メニューの[編集]-[タグジャンプ]-[次のタグ]を選択するか、F4キーを押すことで、違いのある行を順番に見て行くことができます。[編集]ツールバーの[ ]ボタンでも同じです。変更されている行、新しく挿入されている行、削除された行、それぞれに異なった背景色を使って変更箇所を強調表示します。



前の相違点に戻るには、メニューの[編集]-[タグジャンプ]-[前のタグ]か、Shift+F4キーを押してください。

ボタン	メニュー操作	キー操作	機能
	[編集]-[タグジャンプ]-[次のエラー]	F4	次の相違行へ
	[編集]-[タグジャンプ]-[前のエラー]	Shift+F4	前の相違行へ

相違箇所のコピー

比較中の相違箇所の一方を他方へ合わせるコマンドがあります。このコマンドを使うと、クリップボードなどを使わなくても、ボタン一つで右から左(上から下)、左から右(下から上)のウインドウへテキストを移動できます。

ボタン	メニュー操作	キー操作
	[検索]-[比較]-[左から右へコピー]	
	[検索]-[比較]-[右から左へコピー]	

背景色のカスタマイズ

違いがあった行の背景色は、カスタマイズできます。違いのある部分を示す背景色は、メニューの[ツール]-[オプション]で表示される[オプション設定]ダイアログボックスの[表示色]ページで変更できます。変更行、追加行、削除行、それぞれの背景色が別々に設定できます。

アウトプットウインドウの[ファイル比較]ページのポップアップメニューから[クリア]を選択して比較結果をクリアするか、アウトプットウインドウの別のページを選択すると、背景色が通常の色に戻ります。

比較ウインドウの配置

比較ウインドウの配置は、次の2通りの方法で変えることができます。

1. [フォルダとファイルの比較] ダイアログボックスの [ウインドウを水平に配置する] をチェックする。またはチェックを外す。
2. アウトプットウインドウの [ファイル比較] ページのポップアップメニューから、[ウインドウを水平に配置] を選択する。

1. の方法は、ファイル比較を始める前に選ばなければなりませんが、2. は、いつでも使うことができます。

【ヒント】相違点を表示している左右、または上下に並んだウインドウは、編集ウインドウそのものです。必要ならいつでもエディットしてかまいません。その結果テキストの行数が多少変化しても、Peggyは行の増減を補正して表示します。しかしあまりに多くの挿入・削除を行うと、ずれが生じてしまうこともあります。その場合はいったんファイルを保存してから、比較しなおしてください。なお、Shift+F4を使って後ろから修正すると、行ずれの問題は起こりません。

【ヒント】Peggyのファイル/フォルダ比較機能は、子プロセスとしてGNU diffを起動し、その出力を取り込み解析することで実現しています。別のGNU diff互換検索コマンドと交換することも可能です。

【注意】異なる漢字コードを使ったファイル間の比較は意味がありません。また付属のGNU diffコマンドがUnicodeをバイナリファイルと判断してしまうため、Unicodeファイル同士の比較もできません。しかし、それ以外の同じ漢字コードファイル同士の比較は可能です。

左から右へコピー・右から左へコピー

ファイルの比較中に、相違行を両者の間でコピーし合います。これらのコマンドを使うと、クリップボードを経由することなく、相違箇所をもう一方に合わせることが可能です。

ボタン	メニュー操作/機能	キー操作
	[検索]-[比較]-[左から右へコピー]	
	[検索]-[比較]-[右から左へコピー]	

フォルダ比較

[フォルダとファイルの比較]ダイアログボックスの[フォルダの比較を行う]をチェックすると、2つのフォルダの中にある複数のファイルをまとめて比較できます。さらに[サブフォルダも比較する]をチェックすると、サブフォルダの中にあるファイルも再帰的に比較して行きます。

ボタン	メニュー操作	キー操作
	[検索]-[比較]-[ファイル比較]	Ctrl+Shift+D

2 つのフォルダを比較するには

1. まずメニューの [編集]-[ファイル比較] を選択して、[フォルダとファイルの比較] ダイアログボックスを呼び出してください。

2.[フォルダの比較を行う] をチェックします。

3. 比較したい 2 つのフォルダを、それぞれ [左 / 上パス]、[右 / 下パス] にタイプします。タイプする代わりに、[...] ボタンをクリックするか、エクスプローラからドラッグ＆ドロップしてください。

4. さらに細かい指定がある場合には、必要な比較オプションを指定します。ここでサブフォルダも比較したい場合は、[サブフォルダも比較する] をチェックしてください。

5.[比較] ボタンをクリックして比較を実行します。結果はアウトプットウインドウの [フォルダ比較] ページへ出力されます。

フォルダ比較を中止するには

[プロジェクト]-[実行中止]を実行します。ツールバーの[ ]でも同じです。

結果を見る

比較結果は、1対のファイルに付いて1行で出力されます。[同一のファイルも表示する]をチェックしていた場合、内容が同じであったファイルもその旨を表示します。以下は、出力行とその意味の説明です。

アウトプットウインドウ表示	意味
Only in ...	そのファイルがどちらかのフォルダにしかなかったことを表わしています。
Files xxx and yyy differ	ファイルが両方のフォルダに存在するが、その内容に違いがあったことを意味しています。
Files xxx and yyy are identical	ファイルが両方のフォルダにあり、その内容も同じであったことを示しています。このメッセージは、[同一のファイルも表示する] をチェックしているときだけ表示されます。
Binary files xxx and yyy differ	2つのバイナリファイルファイルxxxとyyyの内容が異なっていることを示しています。バイナリファイルであるため、この行をダブルクリックしてもファイル比較を行いません。ファイル比較で差異を見ることができるのは、テキストファイルだけです。
Common subdirectories: xxx and xxx	同じ名前のサブフォルダが両方にあることを表しています。

違いのあった行を順番に見て行くには、以下の操作で次、または前の相違ファイルを選択できます。

機能	メニュー操作	キー操作
次の相違ファイル	[編集]-[タグジャンプ]-[次のタグ]	F4
前の相違ファイル	[編集]-[タグジャンプ]-[前のタグ]	Shift+F4

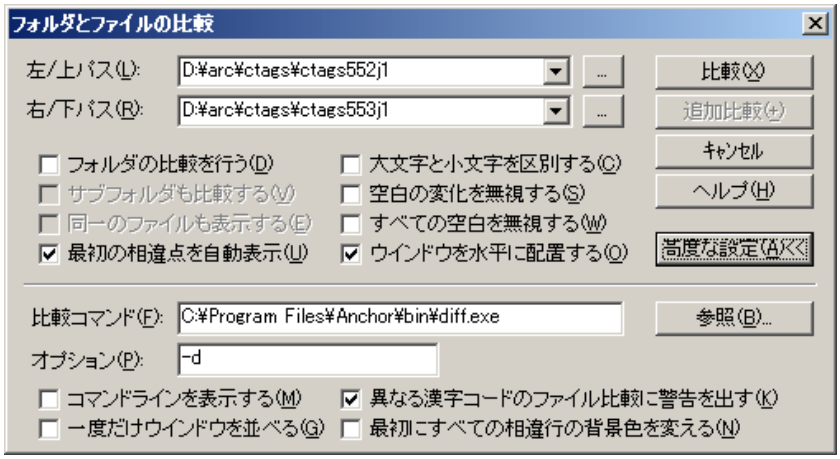
フォルダ比較の結果を元に個々のファイルの相違点を詳しく見たいときには、

```
Files xxx and yyy differ
```

という行の上をダブルクリックしてください。この行の上にカーソルを移動させてから、Enterキーをタイプしても同じです。メッセージ中に表示された2つのファイルパスを使って、ファイルを比較します。あとは『ファイル比較』と同じ操作で、相違行を順番に表示できます。

【ヒント】フォルダ比較を行う場合は、2つのエディタ、[フォルダ比較]ページ、[ファイル比較]ページと多くのウインドウ間でカーソルを移動させなければなりません。このカーソル移動は、Ctrl+Shift+3、Ctrl+Shift+4を使うと簡単に行えます。

キー操作	機能
Ctrl+Shift+3	カーソルをアウトプットウインドウの[フォルダ比較]ページへ移動する
Ctrl+Shift+4	カーソルをアウトプットウインドウの[ファイル比較]ページへ移動する



[左 / 上パス]、[右 / 下パス]

比較する２つのファイルまたはフォルダをそれぞれに入力します。[...]ボタンを押せば、ファイルやフォルダを選択入力することもできます。エクスプローラからドラッグ&ドロップで比較したいファイルを指定することもできます。

[フォルダの比較を行う]

ここをチェックするとフォルダ比較になります。チェックしないとファイル比較になります。

[サブフォルダも比較する]

フォルダ比較の場合だけ有効です。これをチェックすると指定したフォルダ内のファイルだけでなく、その下にあるサブフォルダも再帰的に比較対象になります。

[同一のファイルも表示する]

フォルダ比較の場合だけ有効です。これをチェックすると、比較した結果同じファイルであっても表示されます。

[最初の相違点を自動表示]

これをチェックすると比較終了後、アウトプットウインドウの最初に表示されている相違箇所を自動的に表示します。

[大文字と小文字を区別する]

これをチェックすると、半角アルファベットの大文字と小文字を区別して比較します。なお、全角文字の仮名、たとえば「あ」と「ぁ」は常に別の字として扱います。

[空白の変化を無視する]

これをチェックすると、空白文字が連続している箇所はその数が違っていても同一と見なします。ただし、空白が無かったところに空白が挿入されている場合は、常に違うと判断します。

[すべての空白を無視する]

空白文字を無視して比較をします。

[ウインドウを水平に配置する]

２つのファイルを並べて相違行を表示するとき、これをチェックしていると上下に並べて表示します。

[比較コマンド]

実際に比較を行うコマンドを指定します。

[オプション]

実際の比較コマンドへ与える追加のオプションを指定します。どのようなオプションが使えるかは比較コマンドによるため、詳しくはそのコマンドのマニュアルなどを参照してください。デフォルトは-dです。

[コマンドラインを表示する]

比較コマンドを実行するときのコマンドラインをアウトプットウインドウへ表示します。比較コマンドを他のものと入れ替えたような場合、これをチェックすることで、正しいオプションが渡されているか確認できます。

[異なる漢字コードのファイル比較に警告を出す]

漢字コードの異なるファイル間の比較は、画面上同じ文字でも相違していると判断されます。何が違うのかわかりにくい状態となります。そのようなとき警告を出してユーザに知らせます。

[一度だけウインドウを並べる]

ファイルを比較した直後の一回だけウインドウを上下、または左右に並べて配置します。このオプションをOFFにすると、毎回配置し直します。

[最初にすべての相違行の背景色を変える]

比較後にすべての相違行の背景色を変えて表示します。以後、各相違行へ順番にジャンプしても背景色を消しません。

置換

文字列を他の文字列で置換します。

ボタン	メニュー操作/機能	キー操作
	[検索]-[置換]	Ctrl+R

基本的な文字列置換操作

1. 何を（検索する文字列）、何に（置換後の文字列）置き換えるかを、それぞれ入力します。
2. ［次を検索］ボタンをクリックします。検索した文字列を選択表示します。
3. その文字列を［置換後の文字列］に置き換えたいなら、［置換］ボタンをクリックします。文字列の置き換えを行って次へ進みます。置き換えずに次の文字列へ進めるには、［次を検索］ボタンをクリックします。また、［すべて置換］をクリックすると、一度に置換できます。
4. ［閉じる］ボタンをクリックして、ダイアログボックスを閉じます。

【注意】編集禁止のテキストは置換できません。

［検索する文字列］、［置換後の文字列］

何を(検索する文字列)、何に(置換後の文字列)置き換えるかを入力します。[正規表現]をチェックしていれば、正規表現パターンも入力できます。以前に入力した文字列を覚えているので、リストから選ぶこともできます。

正規表現をチェックしているときは､マッチした文字列の一部分または全体を引用することができます。正規表現の引用したい部分を括弧(､)で囲んでおくと、[置換後の文字列]の中で、左から順に¥1～¥9として参照できます。また一致した文字列全体は、¥0で引用可能です。

[単語単位で探す]

長い単語の一部ではなく、全体が一致する単語を検索するにはこれをチェックします。

[大文字と小文字を区別する]

これをチェックすると、半角アルファベットの大文字と小文字を区別して検索します。なお、全角文字のかな、たとえば「あ」と「ぁ」は常に別の字として扱います。

[正規表現]

[検索する文字列]を正規表現パターンとみなして、それにマッチする文字列を探します。

【参照】『第4章 検索・比較・置換/正規表現』

[あいまい検索]

これをチェックすると、現在の「あいまい検索の設定」に従ったあいまい検索が有効になります。あいまい検索の設定を変更するには、右側にある[ ]ボタンをクリックしてください。[あいまい検索の設定]ダイアログボックスが表示されます。正規表現とあいまい検索は同時に指定できますが、一部制限があります。詳しくは、正規表現とあいまい検索を参照してください。

[置換後の文字列も初期化する]

[置換後の文字列]の初期値として、[検索する文字列]と同じ文字列をセットします。次に[置換]ダイアログボックスを開いたときから有効になります。

[置換の範囲]

文字列の置換を行う範囲を指定します。このダイアログボックスを呼び出す前にテキストのある部分を選択していると、デフォルトで[選択範囲]が選ばれ、その選択範囲に限定して文字列置換を行うことができます。[すべて置換]ボタンをクリックしても、選択範囲内の文字列しか置換しません。

[先頭へ]

置換開始位置をテキストの先頭に移動します。

[ ]

アンドゥです。置換した部分を元に戻します。複数ファイル置換の時でも、現在置換中のファイルしかアンドゥしません。別のファイルの置換をアンドゥする場合は、[ ]などでアンドゥするファイルに移動してください。

[ ]

リドゥです。アンドゥした部分を再度置換し直します。

[次を検索]

置換せず、[検索する文字列]と一致する部分へ進めます。

[置換]

置換してから、[検索する文字列]と一致する部分へ進めます。

[すべて置換]

対象ファイル中、または選択範囲中の[検索する文字列]と一致するすべての文字列を、[置換後の文字列]で置き換えます。

【注意】複数ファイル置換のとき、現在カーソルのあるファイルは、フィルタに一致する/しないに関わらず置換対象となるので注意してください。

[閉じる]

このダイアログボックスを閉じます。

[複数ファイル・単一ファイル]

複数のファイルを対象に置換を行うときに押します。複数ファイル置換のためのオプションが表示されます。すでに表示されている場合は隠します。

[TAGS をスキップする]

<TAGS>ファイルを置換の対象から除外します。

[走査中のアイテムを示す]

置換中のテキストが、プロジェクトウィンドウのどの位置にあるかを表示します。

[走査中の画面を更新する]

置換中のテキストを編集ウィンドウに順に表示します。検索文字列が含まれないファイルも１度表示るため、画面がちらつくことがあります。

[単一ファイル置換に戻す]

このダイアログボックスを閉じた後、再び置換を行うとき、「複数ファイル」置換のオプションが表示されず、単一ファイル置換に戻ります。

[走査の対象]

置換の対象となるファイルの範囲を指定します。[開いているファイル]、[グループ内のファイル]、[グループ以下のファイル]、[プロジェクト全体]の４つから選択します。

【注意】[すべて置換]で複数ファイルを連続して置換しているとき、プロジェクト中に複数回登録されているファイルに対しては1回しか置換を行いません。2回目以降は、置換を行わずスキップします。

【注意】検索の途中で[走査の対象]を変更すると、その時点でカーソルのあるファイルを基準に新しく走査の対象を再設定したと見なします。例えば、[グループ内のファイル]を選択すると、現在カーソルのあるファイルが登録されているプロジェクトグループ内のファイルが新しい走査対象となります。

[フィルタ]

ワイルドカードを使って置換対象のファイルを絞り込みます。複数のパターンを指定することができます。

【注意】フィルタを変更した場合でも、現在カーソルのあるファイルは検索/置換の対象となります。フィルタに一致するファイルまで移動するには、[⏮]、または[⏭]ボタンをクリックしてください。

[⏮]

走査対象のファイルの中で、走査の順序が最初のファイルを表示し、置換開始位置もファイルの先頭に戻します。

[⏪]

走査の順序が一つ前のファイルを表示し、置換開始位置をそのファイルの先頭に置きます。

[⏭]

走査の順序が次のファイルを表示し、置換開始位置をそのファイルの先頭に置きます。

[⏭!]

走査対象のファイルの中で、走査の順序が最後のファイルを表示し、置換開始位置もファイルの最後に置きます。

【ヒント】置換対象のファイルが多いと置換に時間がかかることがありますが、Escキーで中断できます。

【注意】[大文字小文字の区別]、[単語として検索]は、正規表現検索においても意味を持ちます。

条件保存

プロジェクトを使っている場合には、検索や比較の条件を保存することができます。

メニュー操作/機能	キー操作	ボタン
[検索]-[条件保存]-[文字列検索]		
[検索]-[条件保存]-[ファイルから検索]		
[検索]-[条件保存]-[ファイル比較]		

【参照】『第6章 プロジェクト機能/検索、比較条件を登録する』

正規表現

Peggyは正規表現によるパターン検索、置換機能をサポートしています。正規表現を使うと、柔軟な検索・置換が行えます。Peggyでは、標準のパターンに加えて、より使い易い「簡易表記」も用意しました。

基本正規表現一覧

正規表現	機能
.(半角ドット)	任意の1文字に一致します。全角でも半角でも1文字として扱います。

[　]	[ ]の中で指定した文字、または文字範囲に一致します。 [ABC]のように列挙しても、[A-Z]のように範囲で指定しても、両方を混在させてもかまいません。[ ]の中では、特殊文字 ^、$、*、+、?、(,)、. などは特別な意味を失い、普通の文字として扱います。ただし[の直後の ^ は、「範囲外」を意味します。 範囲を指定する代わりに以下のPOSIX名による文字クラス、および補文字クラスも使用できます。名前による日本語文字クラス以外は、1バイト文字の範囲で文字のクラスを判定します。2バイト文字は対照にならないので注意してください。名前による文字クラスは、システムのロケールによってヒットする文字集合が変わることがあります。文字クラス[]と補文字クラス[^]の中でだけ使用することができます。 [:alnum:]　アルファベットと数字 [:alpha:]　アルファベット [:ascii:]　USASII文字セット [:blank:]　ブランク文字(タブとスペース) [:cntrl:]　コントロール文字 [:digit:]　数字 [:graph:]　グラフ(形のある文字) [:lower:]　小文字のアルファベット [:print:]　印刷可能文字 [:punct:]　記号/区切り文字 [:space:]　空白文字 [:upper:]　大文字のアルファベット [:xdigit:]　16進文字 [:^alnum:]　アルファベットと数字以外 [:^alpha:]　アルファベット以外 [:^ascii:]　USASII文字セット以外 [:^blank:]　ブランク文字(タブとスペース)以外 [:^cntrl:]　コントロール文字以外 [:^digit:]　数字以外 [:^graph:]　グラフ(形のある文字)以外 [:^lower:]　小文字のアルファベット以外 [:^print:]　印刷不可能文字 [:^punct:]　記号/区切り文字以外 [:^space:]　空白文字以外 [:^upper:]　大文字のアルファベット以外 [:^xdigit:]　16進文字以外 さらにPOSIX風名前による名前による日本語文字クラスとその補文字クラスも使用できます。

	[:zenkaku:]　全角文字 [:kanji:]　全角漢字 [:hirakana:]　全角ひらかな [:katakana:]　全角カタカナ [:hankakukana:]　半角カタカナ [:^zenkaku:]　全角文字以外 [:^kanji:]　全角漢字以外 [:^hirakana:]　全角ひらかな以外 [:^katakana:]　全角カタカナ以外 [:^hankakukana:]　半角カタカナ以外
[^]	[ ]の中で指定した文字以外、または範囲外の文字に一致します。
( )	( )の中で指定した文字列を、置換のために記憶します。記憶された各文字列は、左から順番に番号が付けられ、¥1、¥2、¥3、... の形式で呼び出すことができます。
(?:)	番号の付かない部分パターンです。括弧( )と同じように部分パターンを指定しますが、マッチした部分は残らず、置換などで¥1、¥2のように参照することはできません。
|	パターンをOR条件で選択します。
^	行頭に一致します。
$	行末に一致します。
*	0回以上の繰り返しを指定します。(最長一致)
*?	0回以上の繰り返しを指定します。(最短一致)
+	1回以上の繰り返しを指定します。(最長一致)
+?	1回以上の繰り返しを指定します。(最短一致)
?	0回か1回パターンが現れることを指定します。(最長一致)
??	0回か1回パターンが現れることを指定します。(最短一致)
{m}	m回の繰り返しを指定します。
{m,n}	m回以上、n回以下の繰り返しを指定します。m、nを省略すると、それぞれ0、上限なしとして扱います。(最長一致)
{m,n}?	m回以上、n回以下の繰り返しを指定します。m、nを省略すると、それぞれ0、上限なしとして扱います。(最短一致)
¥b	単語境界(¥wと¥Wの境界)と一致します。
¥B	単語境界以外(¥wが続いている途中、または¥Wが続いている途中)と一致します。
¥<	単語の先頭とマッチします。
¥>	単語の直後にマッチします。
(?=subpattern)	サブパターンにマッチする境界を指定します。マッチした部分は残らず、置換などで¥1、¥2のように参照することはできません。

(?!subpattern)	サブパターンにマッチしない境界を指定します。マッチした部分は残らず、置換などで¥1、¥2のように参照することはできません。
¥	直後の特殊文字([、]、(、)、{、}、.、?、+、*、^、$、¥)の意味を取り消します。 ¥[　　1文字の[¥(　　1文字の(C言語の文字列定数のように、16進、または8進で文字コードを指定することも可能です。その他にもC言語相当のコントロール文字や簡易表記も指定します。特殊文字以外が後に続いている場合は、その文字の文字コードとなります。 ¥a　　0x07　　アラーム [¥b]　　0x08　　バックスペース【※2】 ¥t　　0x09　　タブ ¥n　　0x0A　　ラインフィード【※1】 ¥v　　0x0B　　垂直タブ ¥f　　0x0C　　フォームフィード ¥r　　0x0D　　キャリッジリターン【※1】 ¥e　　0x1B　　エスケープ ¥xHH　　0xHH　　2桁の16進数 ¥ooo　　0ooo　　3桁の8進数

簡易表記一覧

簡易表記	機能
¥s	全角を含む任意の空白文字1文字に一致します。[¥t-¥n　]
¥S	全角を含む半角の空白以外の1文字に一致します。[^¥t-¥n　]
¥w	英単語に一致します。[A-Za-z0-9_]+
¥W	英単語以外の1文字に一致します。[^A-Za-z0-9_]
¥d	半角数字1文字に一致します。[0-9]
¥D	半角数字以外の1文字に一致します。[^0-9]
¥z	全角文字1文字に一致します。
¥Z	全角以外の1文字に一致します。
¥k	全角漢字1文字に一致します。
¥K	全角漢字以外の1文字に一致します。
¥h	全角ひらがな1文字に一致します。
¥H	全角ひらがな以外の1文字に一致します。
¥j	全角カタカナ1文字に一致します。
¥J	全角カタカナ以外の1文字に一致します。
¥y	半角カタカナ1文字に一致します。
¥Y	半角カタカナ以外の1文字に一致します。
¥i	C/C++言語などの識別子に一致します。 ¥<[A-Za-z_][A-Za-z0-9_]*¥>
¥l	アセンブラなどのラベルに一致します。 ¥<[A-Za-z__][A-Za-z0-9__]*[　¥t]*:

¥q	二重引用符で囲まれた文字列に一致します。 "[^"¥¥]*(:?¥¥.[^"¥¥]*)*"
¥n	改行文字(CR-LF、LF、CR)に一致します。
¥r	改行文字(CR)に一致します。【※1】
¥x0a,¥012	改行文字(LF)に一致します。【※1】
¥r¥012	改行文字(CR-LF)に一致します。【※1】

【※1】ただしラインフィード(¥n)とキャリッジリターン(¥r)は行末として扱われるので、行中には一致する文字はありえません。改行を表す文字は、正規表現の最後に1つだけ指定することができ、1つの改行とマッチします。アスタリスク(*)などにより、繰り返し回数を指定することはできません。

【※2】¥bは単語境界を意味するので、バックスペースは[¥b]と記述します。

正規表現例

正規表現	機能
^$	何も文字のない行に一致します。
¥t	1文字のタブに一致します。
¥.	1文字のドット(.)に一致します。
¥¥	1文字の円記号(¥)に一致します。
¥x61	文字コード0x61、つまり小文字の(a)1文字に一致します。
[^¥t]+	空白でない文字の連続部分に一致します。
a+	1文字以上の連続した(a)に一致します。
aa+	2文字以上の連続した(a)に一致します。
[abc]+	1文字以上(a)、または(b)、または(c)が連続している部分に一致します。
[[:alnum:]]+	英数文字が1文字以上連続した部分に一致します。
//.*$	C++言語の行コメントに一致します。
[A-Za-z_]*_[A-Za-z0-9_]*	アンダースコア(_)を1文字以上含んでいる任意のシンボルに一致します。

簡易表記例

簡易表記	機能
[+-]?¥d+	符号付き、または符号無しの数字に一致します。
¥s+	1文字以上の連続した空白に一致します。
^¥s*$	何もないか空白文字だけからなる行に一致します。
¥s+$	行末の1文字以上の空白文字に一致します。

[¥k¥j]+¥h*	漢字またはカタカナと後に続くひらがなに一致します。
TRUE|FALSE	TRUEまたはFALSEと一致します。
^|$	行頭または行末と一致します。
¥d+(¥.¥d*)?	小数点と小数があるかもしれない数値と一致します。
#[¥dA-Fa-f]{1,6}	#の後に16進数が1～6桁並んだものと一致します。

【ヒント】正規表現の表記法を忘れたときは、[文字列の検索]ダイアログボックスの[▶]ボタンをクリックしましょう。利用できる正規表現のメニューがポップアップします。

【注意】正規表現による検索は純粋に文字列パターン一致によるものなので、プログラミング言語の文法を理解しているわけではありません。したがって、括弧のバランスとかどこまでが引用符の中なのかを文法に従って正確に判定させることはできません。

正規表現とあいまい検索

正規表現とあいまい検索は同時に使用できますが、以下の制限があります。

1. 文字クラス[　]と補文字クラス[^]の中はあいまい検索の対象外となります。
2. ¥wなど定義済みの文字クラスや簡易表記もあいまい検索が適用されません。
3. 全角と半角を区別しないオプションがONのときは、半角カタカナに濁点、半濁点が続いている「ｶﾞ」、「ﾎﾟ」のような2文字の半角文字が1文字として扱われます。【例】全角と半角を区別しないオプションがONのとき正規表現 ｶﾞ+ は、ｶﾞｶﾞｶﾞｶﾞｶﾞ にマッチしますが、ｶﾞﾞﾞﾞﾞﾞﾞﾞﾞﾞﾞﾞ にはマッチしません。これは、「ｶﾞ」が1文字として扱われるためです。このオプションがOFFになると、ｶﾞｶﾞｶﾞｶﾞｶﾞｶﾞ にはマッチせず、ｶﾞﾞﾞﾞﾞﾞﾞﾞﾞﾞﾞﾞ にマッチします。
4. 全角半角を区別しないあいまい検索で、¥<word¥>のような正規表現に全角のｗｏｒｄはマッチしません。単語境界を指定する¥<、¥>、¥bは、単語を構成する文字を半角に限定しているためです。
5. あいまい検索の大文字と小文字を区別しないオプションは、半角アルファベットの(A～Z/a～z)、全角アルファベットの(Ａ～Ｚ/ａ～ｚ)、全角のギリシャ文字(Α～Ω/α～ω)、キリル文字(А～Я/а～я)すべてに影響しますが、通常検索の大文字と小文字を区別しないオプションは、半角アルファベットだけに影響します。

正規表現その他のヒント

ここでは、検索、置換をよりスムーズに行うためのヒントを掲載します。

【ヒント】タブ文字の扱い

タブ文字を検索するには、正規表現を使わなければなりません。¥tでタブ文字と一致します。正規表現OFFのときに、¥tで検索すると一致する文字は、¥tという¥とtの2文字でできた文字列です。

置換も同様になりますから、タブ文字が3つ連続している部分をすべてタブ文字1つに置換したいときは、[正規表現]をチェックした上で、[検索する文字列]に ¥t¥t¥t [置換後の文字列]に ¥tと記述します。

【ヒント】( ) 文字の扱い

( ) は、正規表現で一致した文字列の指定部分を記憶するという意味を持っています。そのため、正規表現中で1文字の (や) を表すには ¥(、 ¥) と記述する必要があります。if(という文字列に一致する正規表現は、if¥(になります。

置換時には、検索する文字列中の ( ) の中で指定した文字列を記憶します。記憶された各文字列はその左からの順番に従って番号が付けられ、左から¥1、¥2、¥3、...、¥9の形式で呼び出すことができます。したがって、abcdefghijklmnoという文字列に対して[正規表現]をチェックした上で、

[検索する文字列]　　abc(def)ghi(jkl)mno

[置換後の文字列]　　abc¥2ghi¥1mno

と指定すれば、実行後の文字列は、abcjklghidefmnoになります。

¥0は、一致した文字列全体を参照します。正規表現を使った置換後の文字列には、制御コードなどを表わす以下の指定が可能です。

置換後文字列の表記法	置換する文字列
¥1 ～ ¥9	検索する文字列中の括弧で指定した部分の1番目～9番目
¥0	マッチした文字列全体
¥a	0x07(アラーム)
¥b	0x08(バックスペース)
¥t	0x09(水平タブ)
¥n	そのファイルの改行コードCR-LF、LF、CR)【※3】
¥x0A	0x0a(ラインフィード)
¥v	0x0b(垂直タブ)
¥f	0x0c(フォームフィード)
¥r	0x0d(キャリッジリターン)
¥e	0x1b(エスケープ)
¥xHH	2桁の16進数(HH)で文字コードを指定

【※3】¥nは、0x0a(ラインフィード文字)ではなく、そのファイルの改行コードに置き換えられます。Windowsのファイルであれば、CR-LFの2文字、UNIXのファイルならばLFの1文字となります。WindowsのファイルでLFだけを指定したい場合は、¥x0Aと16進数で指定してください。

第５章　言語モード

このリファレンスでは、言語モードに関する解説とより詳細な使い方を記載しています。

【参照】CRECユーザ定義キーワードファイルの具体的な作成方法は、

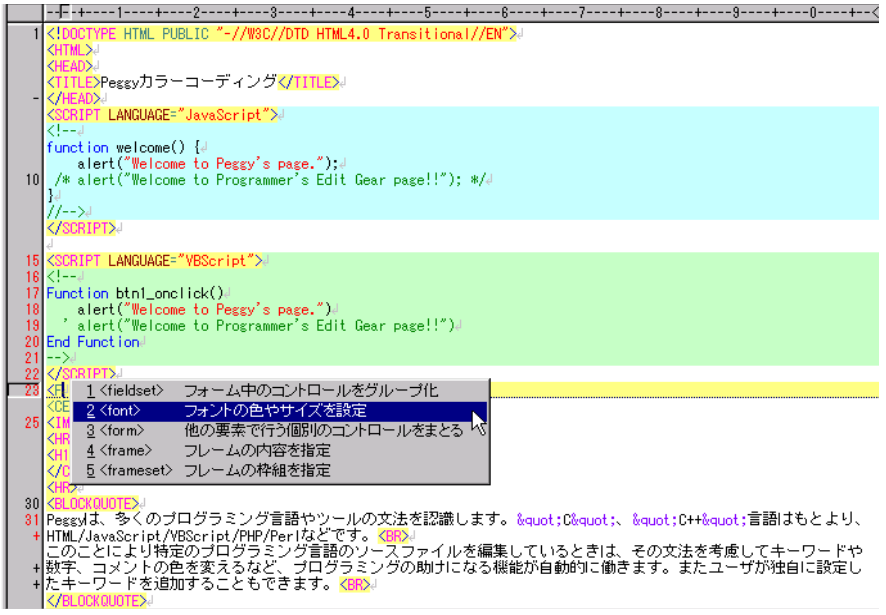
オンラインヘルプ『コンテキスト正規表現によるカラー表示の定義』を参照してください。

カラーコーディング

カラーコーディングとは

Peggyは、多くのプログラミング言語の文法やドキュメントの書式を認識します。C、C++言語はもとより、HTML、PHP、ASP、スクリプト言語、各種アセンブリ言語などです。

この機能により、ソースファイルの文法を考慮してキーワードや数字、コメントの色を変えるなど、プログラミングの助けになる機能が自動的に働きます。またユーザが独自に設定したキーワードを追加することもできます。



カラーコーディングを使う

カラーコーディングを使うには、言語を設定する必要があります。この言語の設定は、Peggy全体の言語設定と、プロジェクトの言語設定による動作の2段階になります。

デフォルト動作

カラーコーディングにおける言語のデフォルトは、[ツール]-[オプション]の[言語]ページの[拡張子マップ]に示される、ファイル拡張子と言語の組み合わせになります。

例えば、ファイル名<xxxx.c>のファイルをPeggyで開くと、デフォルトでC言語になります。エディタ上に表示されるテキストは、C言語の文法にしたがって、プリプロセッサ、数値、コメントなどが色分け表示されます。また、<xxxx.html>のファイルであればWWW HTMLとなり、HTML言語はもちろん、その中のサブ言語(JavaScriptやVBScriptなど)で書かれた部分もきっちりと認識し、色分けを行います。他にもASPなど複数言語の混在にも対応しています。

各部分の色は、[ツール]-[オプション]の[表示色]ページで変更できます。チェックが付いていない項目はデフォルト色です。

【ヒント】この言語認識は、新規対応言語が増えるごとにDLLファイルかCRECキーワード定義ファイルとして弊社より提供します。最新情報を弊社Webにてご確認ください。追加ファイルはすべてご自由にダウンロードしてお使いいただけます。

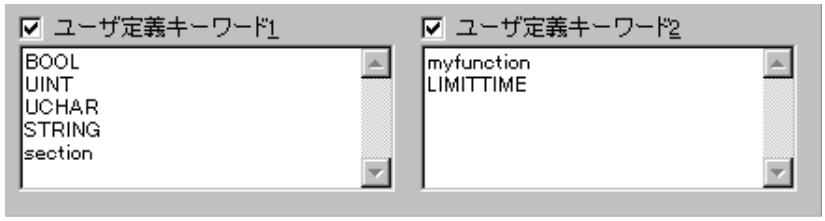
プロジェクト動作

プロジェクトにおける設定・動作は、この次の『言語モードと拡張子マップ』それに『第6章 プロジェクト機能』を参照してください。

ユーザ定義キーワード設定

カラーコーディングでユーザ定義キーワードを設定するには、[ツール]-[オプション]の[言語]ページにある[ユーザ定義キーワード1]、[ユーザ定義キーワード2]の2つの入力ボックスを使います。

ます定義したい言語モードを[言語]から選択し、キーワードとしたい単語を入力し、Enterキーを押します。キーワード数に制限はありません。また、[ユーザ定義キーワード1]、[ユーザ定義キーワード2]に設定された単語はそれぞれに、異なる色付けを行うことができます。



【ヒント】より柔軟にキーワード定義やカラーリングを行うには、CRECキーワード定義ファイルを作成してください。

言語モードと拡張子マップ

言語モードとは

いろいろなプログラミング言語に対応するためには、言語毎に文法の認識方法を変えなければなりません。これを「言語モード」または[言語]と呼びます。1つの言語モードは1つのプログラミング言語に対応していて、そのプログラミング言語のソースファイル編集をサポートします。またソースファイルを印刷するときも、言語の種類毎に多くの印刷オプションを設定できるようになっています。

Peggyには一般的なC/C++言語と、HTMLなどインターネットで使われる言語、Perl、Rubyのようなスクリプト言語、さまざまなアセンブリ言語などの文法が最初から組み込まれています。また、全くオリジナルの言語モードを作成することもできます。

これらを定義している言語サポートDLLとCRECキーワード定義ファイルは、CDに最初から添付されているものと、追加のものがあります。追加のファイルは、弊社Webより無償でダウンロードしてお使いいただけます。

各々の言語モードには、次の機能が含まれています。

●プログラミング言語の文法解析機能

●文法に従った文字列の色分け表示

●キーワードの説明のチップ表示

●キーワード補完入力

●ユーザ定義キーワード

●印刷の設定

●デフォルトのタブ間隔設定

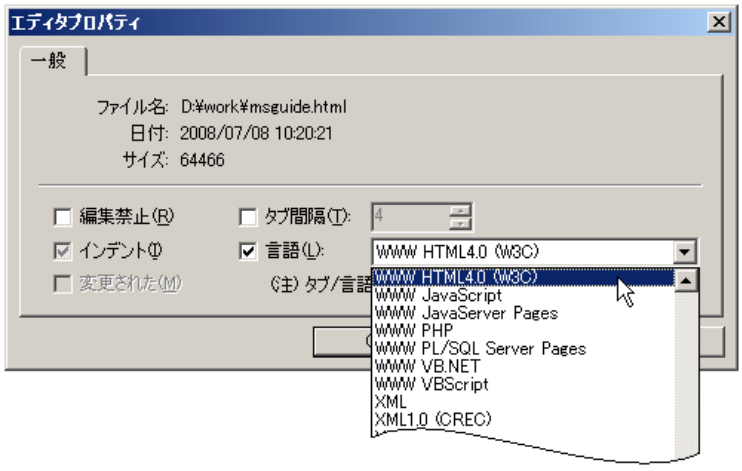
●デフォルトの拡張子設定

ファイルを開くとき

ファイルを開くときは、原則としてそのファイルの拡張子に従ってどの言語かを判定します。もし対応する言語モードがない場合は、自動的に言語[テキスト]を選択します。

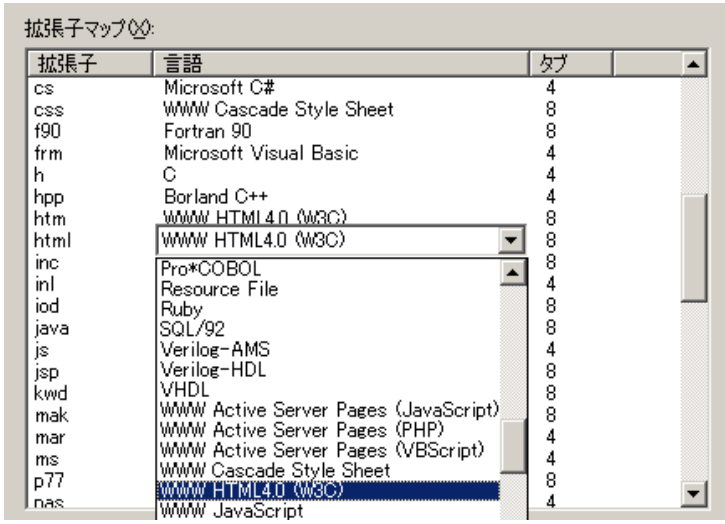
言語と拡張子の組み合わせは、ユーザが自由に追加、変更、削除できます。また、ユーザが独自の言語を定義することも可能です。

編集中のファイルの言語モードを確認したり別の言語モードへ変更するには、[エディタプロパティ]ダイアログボックスの[言語]で行ないます。



拡張子マップ

ソースファイルを開くときには、ソースファイルの拡張子から対応する言語を設定します。ファイルの内容を調べているのではありません。ファイル名の中でも拡張子はプログラミング言語によってほぼ決まっているので、言語を判定する上での重要なヒントとなります。Peggyは内部に「拡張子/言語」対応表を持っていて、これを「拡張子マップ」と呼びます。



もちろん拡張子マップはユーザがカスタマイズ可能です。拡張子が同じ<.c>でも、いろいろなCの方言があります。例えば、<.c>をある特定のCコンパイラの設定に変更したい場合、メニューの[ツール]-[オプション]を選択して[オプション設定]ダイアログボックスを開き、[言語]ページの下半分にある[拡張子マップ]で設定することができます。言語名が書かれている部分を左マウスボタンでクリックし、リストから言語を選んでください。

【注意】新規に作成したテキストでまだファイルに保存していないものには、ファイル名がありません。名前を付けて保存するまでは、文法のないテキストとして扱います。強制的に言語を設定するには、メニューの[編集]-[プロパティ]を選択するか、Alt+Enterキーをタイプし、[エディタプロパティ]ダイアログボックスの[言語]で設定できます。

新規言語モード作成・CREC キーワード定義ファイル

Peggyでは、非常に多くの言語をサポートしているため、新たに言語モードを追加する必要はほとんどありません。ただユーザが独自に規定した拡張子と新規モードを対応させることもできます。

まず、新規言語モードを作る簡易的な方法は、[オプション設定]ダイアログボックスの[言語]ページで[新規]ボタンを押し、ユーザ定義キーワードの指定、タブ間隔、印刷オプションの設定をします。ただし、この方法では独自の文法規則を定義することはできません。あくまで特定の文字列を強調表示するために表示色や印刷フォント、タブ間隔を変えるために使ってください。

もう一つの方法はCREC(コンテキスト正規表現カラー表示)を使って新規に言語モードを作ることです。コンテキスト正規表現とは、コンテキストの状態と正規表現を組み合わせることにより、テキストファイルのより高度で正確な色分け表示を可能にするまったく新しい技術です。何行にも続くコメントや文字列も、これだけで自由に定義可能になります。

もちろん、より単純な言語はコンテキストを使わず、キーワードを書き並べるだけなので、とても簡単に追加できます。

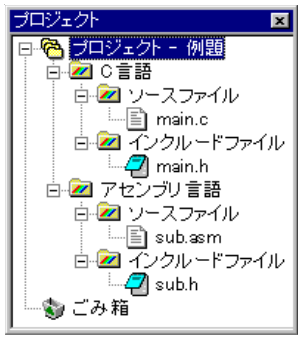
<share>フォルダ内の<ac_kwd.kwd>を基にCRECキーワード定義ファイルを作成してください。なお、詳しい説明はオンラインヘルプ『コンテキスト正規表現によるカラー表示の定義』を参照してください。

プロジェクトとの連携

Peggyの特徴として、プロジェクト機能があります。プロジェクトの下にはグループを作ることができます。ファイルは、プロジェクトの下の任意のグループに登録することができ、一度登録したファイルに対して以下の操作が可能になります。

グループ分け

意味のあるグループ単位にファイル群を登録することができます。グループの下にまたグループを階層的に作成することができます。



言語の設定

拡張子に対応する言語の設定を、グループごとに行うことができます。例えば拡張子<.h>は、デフォルトではC言語に対応していますが、上の図で<アセンブリ言語>グループにアンカーアセンブラを設定すると、<sub.h>はアンカーアセンブラであると認識されます。

第 6 章　プロジェクト機能

このリファレンスでは、プロジェクトとアイテムのプロパティに関するコマンドを分類し、そのコマンド一覧とより詳細な使い方を記載しています。

プロジェクト機能

コマンド一覧

ボタン	メニュー操作/機能	キー操作
	[プロジェクト]-[新規プロジェクト]	
	[プロジェクト]-[保存する]	
	プロジェクトを開く	
	[プロジェクト]-[閉じる]	
	[プロジェクト]-[新規グループ]	
	[プロジェクト]-[ファイル挿入]	
	複数のファイルやフォルダを一度に登録する	
	プロジェクトのファイルを開く	
	プロジェクトを移動する	
	[プロジェクト]-[フルパス表示]	
	[プロジェクト]-[位置の確認]	Ctrl+Shift+I
	アイテムの順番を入れ替える	
	[プロジェクト]-[浅くソート]	
	[プロジェクト]-[深くソート]	
	アイテムのコピー・切り取り・貼り付け	
	登録したアイテムを削除する	
	ごみ箱から削除する	
	[プロジェクト]-[存在しないファイルの削除]	
	[プロジェクト]-[ごみ箱を空にする]	
	プロジェクト名を変更する	
	アイテムの名前の変更	
	[プロジェクト]-[実行中止]	Shift+Esc
	キーボードマクロを登録する	
	キーボードマクロを再生する	
	キーボードマクロのコピー、移動、名前の変更	
	キーボードマクロを呼び出す	
	検索、比較条件を登録する	
	検索、比較を実行する	
	検索、比較条件のコピー、移動、名前の変更	
	検索、比較条件を設定する	
	マウスアクションを設定する	

	ツールアクションを設定する	
	スクリプトファイルの実行	

プロジェクトとは

「プロジェクト」こそ、Peggyの真髄と言えます。プロジェクトを使いこなすことでその力を発揮できます。

今1つの開発プロジェクトがあり、Peggyでソースファイルを編集することを想定してみます。まず、そのソースファイルをプロジェクトに登録することから始めます。ファイルを登録するとプロパティ設定、ウインドウの位置と大きさ、カーソルの位置など作業中の状態も「プロジェクト」が自動的に記録し、ユーザの編集作業を支援します。

プロジェクトにはいくつでもファイルを登録できるので、開発プロジェクトに関係するソースファイル、ヘッダファイルなどをすべて登録しましょう。また、ワープロソフトや表計算ソフトウェアで作成した仕様書・工程管理表など他のアプリケーションのファイルも登録しておくと便利です。それらのファイルをダブルクリックするだけで対応アプリケーションが立ち上がり、すぐ編集作業に取りかかれます。

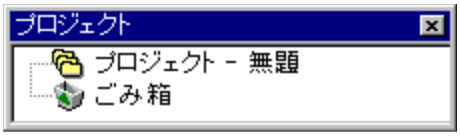
さらに、プロジェクトにはバッチファイルやMakefile、デバッガプログラムも登録できるのです。Peggyはそれらを簡単に実行させることができます。もちろん実行時のフォルダ、引数などの設定も可能です。

ここまでで「プロジェクト」って、関係するファイルをひとまとめに記録しておくための物だな、という感じはつかんでいただけたと思います。まずはプロジェクトを作り、ソースファイルを登録するところから始めてください。仕様書やデバッガを登録するのは、少し慣れてからでもかまいません。プロジェクトを使いこなせるようになると、Peggyを中心として「設計」-「プログラミング」-「コンパイル」-「デバッグ」という一連の開発作業すべてを行うことができます。その他にもキーボードマクロや検索、比較条件など、ファイル以外のツールも登録できます。

プロジェクトを作る

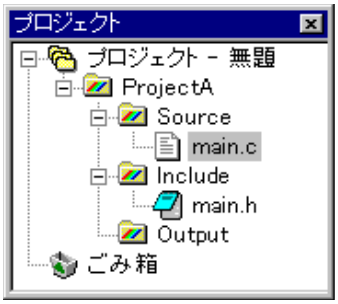
プロジェクトを新しく作ります。方法は２通りあります。

●基本的な方法です。メニューの[プロジェクト]-[新規プロジェクト]を選択します。プロジェクトウインドウに「無題」という空のプロジェクトができます。ツールバーの[]でも同じです。



●こちらは応用的な方法です。プロジェクトを作るとともにファイルの登録もしてしまいます。プロジェクトウインドウに何も表示されていない、つまりプロジェクトを開いていないときに、エクスプローラからファイルまたはフォルダをドラッグして、プロジェクトウインドウにドロップします。「無題」というプロジェクトが作成され、ファイルやフォルダが自動的に登録されます。
フォルダの場合は、フォルダ内のファイルを登録するかどうか選択できます。図

は「ProjectA」フォルダをドラッグ＆ドロップした例です。<ProjectA>、<Source>などの各フォルダは、プロジェクトの中ではグループとなります。また、<main.c>などのファイルはファイルアイテムとして登録されます。



なお、いずれの場合も最初のプロジェクト名は<無題>となります。この名前は、後でプロジェクトを閉じるときに適当な名前に換えてください。プロジェクトはファイルの形で保存されます。プロジェクトファイルの名前は、

<プロジェクト名 + .peg>

となります。

initial.peg

Peggyをインストールしたフォルダに<initial.peg>という名前のプロジェクトファイルがあると、その内容(ゴミ箱を除く)が自動的に新規作成プロジェクトへコピーされます。いつも使うキーボードマクロや設定を<initial.peg>に登録しておけば、新規プロジェクトと同時に挿入されるので便利です。

Shiftキーを押しながらプロジェクトを新規作成した場合、<initial.peg>の内容はコピーされません。<initial.peg>の内容によらず、空のプロジェクトが作成されます。

プロジェクトを保存する

プロジェクトを開いたままプロジェクトの状態を保存します。2通の方法があります。

- メニューの[プロジェクト]-[保存する]を選択します。
- プロジェクトウインドウの上をクリックした後、次のいずれかの操作をします。

  メニューから[ファイル]-[上書き保存]を選択します。
  メニューから[ファイル]-[すべて保存]を選択します。
  Ctrl+Sキーをタイプします。
  [ファイル]ツールバーの[ ]ボタンを押します。

他にも、プロジェクトを閉じるときや、Peggyを終了するときにプロジェクトの内容が変更されていれば、保存するかどうかを問い合わせてきます。

プロジェクトを開く

プロジェクトは、前回閉じたときの状態を記憶しています。プロジェクトを開くと、編集中だったファイルも開かれ、カーソルの位置など作業中の状態も再現されます。

方法は４通りあります。

●メニューの[ファイル]-[開く]を選択し、[開く]ダイアログボックスからプロジェクトファイルを指定します。

●エクスプローラからプロジェクトファイルをドラッグして、実行中のPeggy上にドロップします。

●エクスプローラからプロジェクトファイルをドラッグして、Peggyのアイコン上にドロップします。新しいPeggyが起動してプロジェクトが開きます。

●[オプション設定]ダイアログボックスの[全般]ページにある[以前に開いていたプロジェクトを自動的に開く]をチェックしていると、Peggyの起動と同時に、前回Peggyを終了したとき開いていたプロジェクトを自動的に開きます。このオプションは、デフォルトでONになっています。

【参照】『第11章 カスタマイズ/オプション設定/オプション[全般]』

【注意】Peggyは、一度に一つのプロジェクトしか開くことができません。複数のプロジェクトを開いて同時に作業をしたい場合は、その数だけPeggyを起動してください。

【注意】1つのプロジェクトファイルを複数のPeggyから同時に開くことはできません。どうしても開く必要がある場合は、前もってコピーを作成し、そのコピーを開くようにします。

プロジェクトを閉じる

３通りの方法があります。

●メニューから[プロジェクト]-[閉じる]を選択します。

●メニューから[ファイル]-[Peggyの終了]を選択すると、終了前にプロジェクトを閉じます。この方法でPeggyを終了すると、次に起動したとき、自動的に同じプロジェクトを開きます。

●別のプロジェクトを開きます。現在開いているプロジェクトは自動的に閉じられます。

プロジェクトを閉じると、必要に応じてPeggyがプロジェクトをファイルに保存します。

プロジェクトにファイルを登録する

５通りの方法があります。

●メニューの[プロジェクト]-[ファイル挿入]を選択します。ダイアログボックスが開くのでファイルを選択します。

●ツールバーの[ ]ボタンを押します。

●プロジェクトウインドウで右マウスボタンをクリックし、現れるポップアップメニューから[ファイル挿入]を選択します。ダイアログボックスが開くのでファイルを選択します。

●エクスプローラからファイルをドラッグして、プロジェクトウインドウにドロップします。このときファイルはドロップした位置に登録されます。

●作業中のファイルをプロジェクトに登録するには、[プロジェクト]-[位置の確認]コマンドを実行します。ツールバーの[ ]ボタンをクリックしても同じです。これは、作業中のファイルがプロジェクトのどこにあるかを確認するコマンドですが、プロジェクト中に見つからない場合は、登録するかどうかを尋ねてきます。そこで[はい]を選択すると、プロジェクトに登録できます。

【メモ】エクスプローラからのドラッグ&ドロップで登録する場合は、[オプション設定]ダイアログボックスの[プロジェクト]ページの[無視するパターン]に指定されているパターンにマッチするファイルやフォルダを登録しません。

プロジェクトにグループを作る

プロジェクトにアイテムが増えてくると整理する必要があります。そんなときはグループ分けするとよいでしょう。グループごとに拡張子と言語のモードの対応づけができるので、C言語とアセンブリ言語のファイルを別々のグループに分けて登録し、拡張子.hをそれぞれC言語とアセンブリ言語に設定しておけば、カラー表示やタブ間隔などを正しく行うことができます。

方法は４通りあります。

●メニューの[プロジェクト]-[新規グループ]を選択します。

●ツールバーの[ ]ボタンを押します。

●プロジェクトウインドウで右マウスボタンをクリックし、現れるポップアップメニューから[新規グループ]を選択します。

●エクスプローラからフォルダをドラッグして、プロジェクトウインドウにドロップします。このときフォルダはドロップした位置にグループとして登録されます。フォルダ内のファイルとサブフォルダは登録するかどうか問い合わせてくるので選択できます。

複数のファイルやフォルダを一度に登録する

３通りの方法があります。

●メニューの[プロジェクト]-[ファイル挿入]を選択します。ダイアログボックスが開くので、ファイル名を""(ダブルクォーテーション)で囲むと複数のファイルを指定できます。Ctrlキー、Shiftキーとマウスを使いながらファイルを複数選択すると簡単に指定できます。

●プロジェクトウインドウで右マウスボタンをクリックし、現れるポップアップメニューから[ファイル挿入]を選択します。ダイアログボックスが開くので、１番目の方法と同じようにファイル名を複数入力します。

●エクスプローラでCtrlキーやShiftキーを押しながら複数のファイルやフォルダをドラッグして、プロジェクトウインドウにドロップしてください。フォルダ内のファイルとサブフォルダは登録するかどうかを選択できます。

【ヒント】３番目の方法のとき、Ctrlキーを押しながらドロップすると、問い合わせ無しにフォルダ内の、すべてのファイルとサブフォルダを登録します。問い合わせに「すべてはい」と答えたのと同じです。

【ヒント】３番目の方法のとき、Shiftキーを押しながらドロップすると、ファイルは登録しますが、フォルダはその中のファイルとサブフォルダは登録しません。問い合わせに「すべていいえ」と答えたのと同じです。

【ヒント】３番目の方法のとき、Ctrl+Shift キーを押しながらドロップすると、[ファイル登録]ダイアログボックスが表示されます。様々なプションにより、ファイルやフォルダの登録方法を細かく指定できます。

【ヒント】3番目の方法のとき、登録したくないファイル名パターンを[オプション設定]ダイアログボックスの[プロジェクト]ページに予め設定しておくことができます。

ファイルを開く

プロジェクトに登録しているファイルを開く方法はいろいろあります。ファイルを開くと、前回閉じたときの作業状態が再現されます。もし、前回の状態を再現させたくないときは、Ctrlキーを押しながら開いてください。またShiftキーを押しながら開くと、ウインドウを大きく表示します。

●ファイルアイテムを右マウスボタンでクリックし、現れるポップアップメニューから[開く]を選択します。

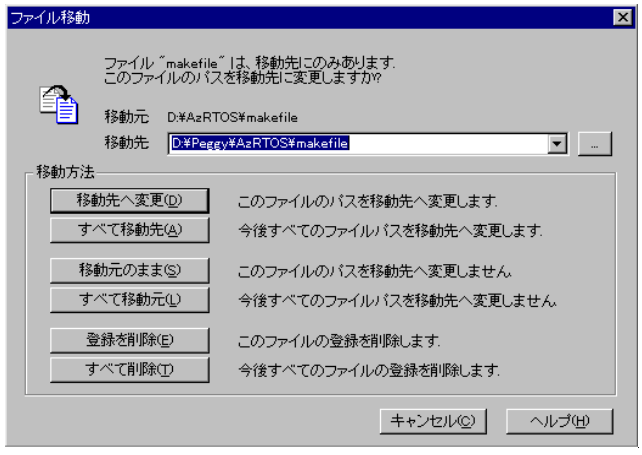
●アイテムを左マウスボタンでドラッグして、プロジェクトウインドウ以外のPeggy上でドロップします。

後で説明する「マウスアクション」が「エディタで開く」になっていると次の方法も使えます。

●アイテムを左マウスボタンでダブルクリックします。または、アイテムを選択してからEnterキーを押します。

プロジェクトを移動する

プロジェクトファイルは自分自身のパスを記憶しています。プロジェクトを開くとき、Peggyはそのパスと現在のパスに違いがないかどうかを調べます。もし違いがあると、プロジェクトに登録しているファイルアイテムのパスにも変化があるのではないかと推測し、次のような問い合わせをしてきます。



Peggyはプロジェクトファイルのパスの変化の割合から、最適と思われるファイルアイテムのパスを「移動先」に表示しています。[...]ボタンを押してダイアログ

ボックスから選択したり、直接入力し直して正しいパスを指定することもできます。そして、[移動方法]の中の６つのボタンから１つを選択します。

アイテムの順番を入れ替える

プロジェクトウインドウのアイテムの順番に意味はありません。作業しやすい順番に整理しましょう。方法は簡単です。

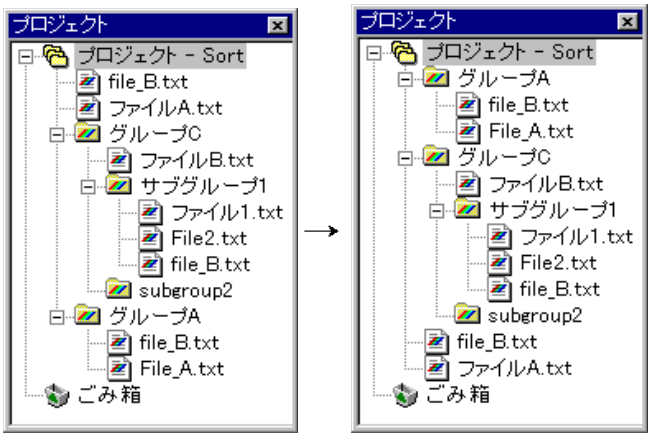
●アイテムをドラッグして適当な位置でドロップします。

それだけです。グループごと移動することもできます。

【ヒント】ドロップするときにCtrlキーを押していると、移動ではなくグループやアイテムのコピーとなります。コピーはプロジェクト内だけのもので、実際のファイルが移動したりコピーされるのではありません。

アイテムを浅くソートする

アイテムの順番を文字コード順に一度に並べ替えることができます。並べ替え対象のプロジェクトやグループのすぐ下のアイテムだけが並べ替えられます。図はプロジェクトを選択して「浅くソート」した例です。グループとそれ以外のアイテムは別々にソートされます。



方法は３通りあります。いずれの場合もプロジェクトウインドウのプロジェクトかグループアイテムを選択してから実行します。

●メニューの[プロジェクト]-[浅くソート]を選択します。

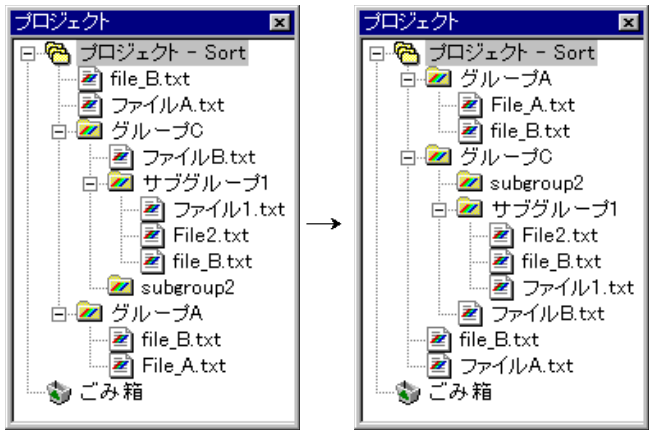
●右マウスボタンをクリックし、現れるポップアップメニューから[浅くソート]を選択します。

●ツールバーの[ ]ボタンを押します。

ソートを１回するごとに昇順と降順が切り替わります。

アイテムを深くソートする

アイテムの順番を文字コード順に一度に並べ替えることができます。並べ替え対象のプロジェクトやグループの下に、グループがあれば再帰的にその中も並べ替えられます。図はプロジェクトを選択して「深くソート」した例です。同じレベルのグループとそれ以外のアイテムは別々にソートされます。



方法は３通りあります。いずれの場合もプロジェクトウインドウのプロジェクトかグループアイテムを選択してから実行します。グループアイテムの場合は、その中のアイテムのみがソート対象です。

●メニューの[プロジェクト]-[深くソート]を選択します。

●右マウスボタンをクリックし、現れるポップアップメニューから[深くソート]を選択します。

●ツールバーの[ ]ボタンを押します。

ソートを１回するごとに昇順と降順が切り替わります。

アイテムのコピー・切り取り・貼り付け

プロジェクトウインドウのアイテムをクリップボードへコピーしたり、クリップボード経由の切り取り、貼り付け操作によってプロジェクト間を移動させることができます。プロパティの設定もすべてコピー・貼り付けされます。

プロジェクトアイテムのコピー、切り取りは、アイテムをクリックしてからメニューの[編集]-[コピー]、[編集]-[切り取り]を選択します。プロジェクトウインドウを右マウスボタンでクリックしたときに現れるポップアップメニューから選択しても同じです。グループをコピー、切り取りした場合は、そのグループに含まれるすべてのアイテムがコピー、または切り取られます。

貼り付けは、挿入したいアイテムを選択してから、[編集]-[貼り付け]を実行してください。そのアイテムの直前に挿入されます。また、グループを選択していた場合は、そのグループの最後の子アイテムとして挿入されます。

プロジェクトアイテムのコピー、切り取り、貼り付けはプロジェクト間でも行えます。プロジェクトAの一部分をプロジェクトBへコピーすることが可能です。

【注意】アイテムやグループを切り取ると、内容をクリップボードへ移したあとプロジェクトから削除されます。ゴミ箱へ移動するのではないので、注意してください。間違って切り取った場合は、貼り付けで戻すことができます。

登録したアイテムを削除する

４通りの方法があります。いずれの場合も、まずプロジェクトウインドウの削除したいアイテムをマウスで選択してから行います。

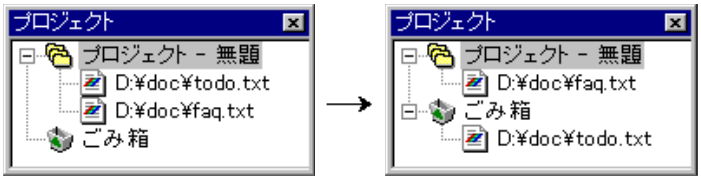
●Deleteキーを押します。

●メニューの[編集]-[削除]を選択します。

●右マウスボタンをクリックし、現れるポップアップメニューから[削除]を選択します。

●ツールバーの[✕]ボタンを押します。

ここで注意する点は、「削除」の意味です。削除されてもプロジェクトから完全に消えるのではなく、実は[ごみ箱]に移動しているだけです。



また、図の場合では<todo.txt>を削除していますが、ディスク上に存在するファイル<D:¥doc¥todo.txt>そのものを削除するわけではありません。ファイルの実体には全く影響を与えません。

「アイテムの削除」がアイテムを[ごみ箱]に移すだけだということが理解できれば、方法はもう一つあります。

●アイテムをドラッグして「ごみ箱」でドロップします。

【注意】グループを削除するとその下に登録されたアイテムやグループも[ごみ箱]に移動します。

ごみ箱から削除する

アイテムを削除すると[ごみ箱]に移ります。[ごみ箱]には削除されたアイテムがたまってきます。将来的にも不要だと思うアイテムは[ごみ箱]からも削除して完全にプロジェクトから消し去りましょう。なお、「登録したアイテムを削除する」と同様に、ディスク上に存在するファイルそのものを削除するわけではありません。

方法は４通りあります。いずれの場合も[ごみ箱]の中のアイテムを選択してから実行します。

●Deleteキーを押します。

●メニューの[編集]-[削除]を選択します。

●右マウスボタンをクリックし、現れるポップアップメニューから[ごみ箱から削除]を選択します。

●ツールバーの[✕]ボタンを押します。

【ヒント】[オプション設定]ダイアログボックスの[プロジェクト]ページにある[ごみ箱を空にする]をチェックしておくと、プロジェクトを開くとき、自動的にごみ箱が空になります。

ごみ箱を空にする

「ごみ箱から削除する」のように一つずつではなく、ごみ箱の中を一度に空にすることができます。なお、「登録したアイテムを削除する」と同様に、ディスク上に存在するファイルそのものを削除するわけではありません。

方法は２通りあります。

●メニューの[プロジェクト]-[ごみ箱を空にする]を選択します。

●ごみ箱で右マウスボタンをクリックし、現れるポップアップメニューから[ごみ箱を空にする]を選択します。

【ヒント】[オプション設定]ダイアログボックスの[プロジェクト]ページにある[ごみ箱を空にする]をチェックしておくと、プロジェクトを開くとき、自動的にごみ箱が空になります。

実体のないファイルアイテムを削除する

プロジェクトに登録しているファイルをエクスプローラなどで削除しても、プロジェクトのファイルアイテムは連動して削除されません。このようなプロジェクトに登録していても無意味と思われるファイルアイテムを削除します。

方法は２通りあります。

●メニューの[プロジェクト]-[存在しないファイルを削除]を選択します。

●ツールバーの[ ]ボタンを押します。

フルパスで表示する

プロジェクトウインドウのファイルアイテムの表示方法には２種類あります。ファイル名のみとフルパス表示です。これらはユーザの好みで切り替えることができます。

方法は３通りあります。いずれの場合もプロジェクトウインドウを左マウスボタンで選択してから実行します。

●メニューの[プロジェクト]-[フルパスで表示]を選択します。

●右マウスボタンをクリックし、現れるポップアップメニューから[フルパスで表示]を選択します。

●ツールバーの[ ]ボタンを押します。

プロジェクト名を変更する

プロジェクト名は、プロジェクトファイル<xxxx.peg>のファイル名で表されています。プロジェクト名を変更するには、このプロジェクトファイル<xxxx.peg>のファイル名を変更してください。拡張子は必ず<.peg>としてください。

アイテムの名前を変更する

方法は２通りです。

●アイテムを右クリックで選択するとポップアップメニューが現れるので、[名前の変更]を選択します。

●アイテムを左クリックで選択します。もう一度左クリックをします。その間隔が短いとダブルクリックになるので、少し間をあけるのがコツです。

名前の部分が変更可能になるので、適当な名前を入れ、最後にEnterキーを押します。

【ヒント】ファイルアイテムの名前の変更は、[オプション設定]ダイアログボックスの[プロジェクト]ページにある[ファイル名の変更を許す]をチェックしている場合だけ可能です

位置の確認

編集中のテキストがプロジェクトウインドウのどのアイテムかを表示します。

方法は２通りです。

●メニューの[プロジェクト]-[位置の確認]を選択します。

●ツールバーの[ ]ボタンを押します。

【ヒント】編集中のファイルがまだプロジェクトに登録されてない場合、このコマンドを実行することでプロジェクトへ登録するかどうかを問い合わせてきます。またプロジェクトを開いていない状態では、[無題]の新規プロジェクトを作成して登録できます。

実行中止

検索・比較やバッチファイルの実行などを途中で止めるときに使います。

方法は２通りです。

●メニューの[プロジェクト]-[実行中止]を選択します。

●ツールバーの[ ]ボタンを押します。

キーボードマクロを登録する

Peggyが記憶するキーボードマクロは１チャンネルしかないため、新しく記録するたびに上書きされてしまいます。複数のキーボードマクロを使い分けたい場合はプロジェクトに登録します。

[ツール]-[キーボードマクロ]-[プロジェクトに保存]を実行します。

【メモ】プロジェクトにキーボードマクロを登録すると、以下のような利点があります。
・数に制限がありません。
・名前を付けることができ、区別が容易になります。
・プロジェクトに登録したキーボードマクロの内容は記憶されています。プロジェクトを開くと、すぐにキーボードマクロが使えます。

キーボードマクロのコピー、移動、名前の変更

プロジェクトに登録したキーボードマクロは、左マウスボタンで選択した後、Ctrl+Cキーでコピーを作成することができます。

コピーしたキーボードマクロは、ドラッグ&ドロップで移動したり、名前の変更ができます。

キーボードマクロを再生する

キーボードマクロを再生する方法は、５通りあります。

●プロジェクトウインドウ中のアイテムを左マウスボタンでダブルクリックします。

●アイテムを選択してから、Enterキーを押します。

これら２つの方法は、プロパティ [マウスアクション]が[再生する]になっていなければなりません。

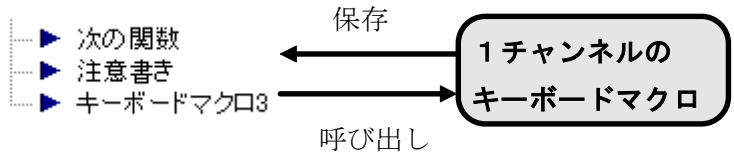
●アイテムのポップアップメニューから[キーボードマクロ再生]を選択します。

●プロパティの[ツールアクション]でツールボタンに割り当て、そのボタンをマウスでクリックします。このときは[ツールアクション]が[再生する]になっていなければなりません。

●メニューの[ツール]-[キーボードマクロ]-[再生]を選択します。

キーボードマクロを呼び出す

プロジェクトに登録しているキーボードマクロを、Peggyが再生に使う１チャンネルのキーボードマクロに呼び出すことができます。



方法は４通りあります。

●プロジェクトウインドウ中のアイテムを左マウスボタンでダブルクリックします。

●アイテムを選択してから、Enterキーを押します。

これら２つの方法はプロパティで[マウスアクション]が[リストアする]になっていなければなりません。

●アイテムのポップアップメニューから[キーボードマクロ呼び出し]を選択します。

●プロパティの[ツールアクション]でツールボタンに割り当て、そのボタンをマウスでクリックします。このときは[ツールアクション]が[リストアする]になっていなければなりません。

検索、比較条件を登録する

Peggyは文字列検索やファイルから検索、またファイル比較などの条件をプロジェクトに登録・保存することができます。検索や比較の種類により、[条件保存]は、[文字列検索]、[ファイルから検索]、[ファイル比較]に分類されます。キーボードマクロの登録と同じように、それぞれのコマンドが現在の設定条件の状態を一時記憶しており、その条件を登録することになります。

[検索]-[条件保存]-[文字列検索]、[ファイルから検索]、[ファイル比較]を実行します。実行した条件が登録されます。

【メモ】検索、比較条件を登録する利点
・数に制限がありません。
・名前を付けることができ、プロジェクトウインドウに表示され、その区別も容易になります。

●プロジェクトに登録した検索、比較条件の内容は記憶されています。プロジェクトを開くと、すぐに検索、比較が使えます。

検索、比較条件のコピー、移動、名前の変更

プロジェクトに登録した検索、比較条件は、左マウスボタンで選択した後、Ctrl+Cキーで複製をプロジェクト内に作成することができます。コピーは、ドラッグ&ドロップで移動したり、名前の変更ができます。

検索、比較を実行する

検索、比較を実行する方法は、４通りあります。

●プロジェクトウインドウ中のアイテムを左マウスボタンでダブルクリックします。

●アイテムを選択してから、Enterキーを押します。

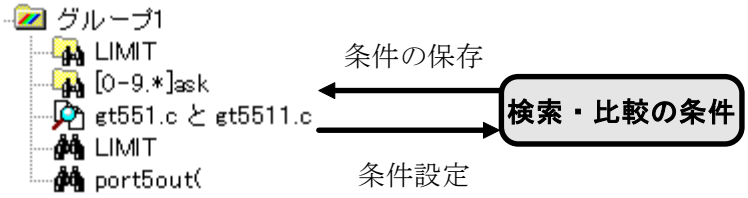
ただし、これらの方法はプロパティ[マウスアクション]が、[検索する]、[追加検索する]、[比較する]、[追加比較する]のいずれかになっていなければなりません。

●アイテムのポップアップメニューから実行します。

●プロパティの[ツールアクション]でツールボタンに割り当て、そのボタンをマウスでクリックします。このときは[ツールアクション]が[検索する]、[追加検索する]、[比較する]、[追加比較する]のいずれかになっていなければなりません。

検索、比較条件を設定する・リストアする

プロジェクトにアイテムとして登録している検索・比較の条件を、Peggyが記憶している検索・比較の条件に設定することができます。それを「設定する」あるいは「リストアする」と言います。ちょうど、「検索、比較条件を登録する」の逆の動作です。



方法は４通りあります。

●プロジェクトウインドウ中のアイテムを左マウスボタンでダブルクリックします。

●アイテムを選択してから、Enterキーを押します。

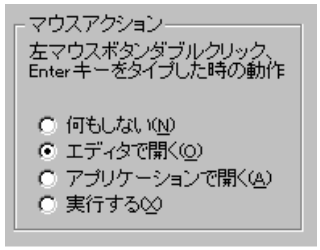
これら２つの方法はプロパティで[マウスアクション]が[リストアする]になっていなければなりません。

●アイテムのポップアップメニューから[検索条件の設定]または[比較条件の設定]を選択します。

●プロパティの[ツールアクション]でツールボタンに割り当て、そのボタンをマウスでクリックします。このときは[ツールアクション]が[リストアする]になっていなければなりません。

マウスアクションを設定する

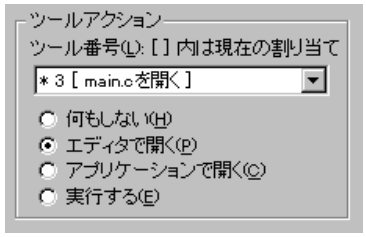
マウスアクションとは、プロジェクトに登録されたアイテムを左マウスボタンでダブルクリックしたり、Enterキーをタイプしたときの動作のことです。Peggyは、それぞれのアイテムごとにマウスアクションを決めることができます。マウスアクションは、[プロパティ]ダイアログボックスを開いて設定します。



ダイアログボックスの[ファイル]ページに、上図のようなマウスアクションを設定する部分があります。どれか一つを選択します。以後、そのアイテムをダブルクリックすると、図の場合はファイルをPeggyで開くことになります。動作の説明はこの章の『[ファイル]プロパティページ』を参照してください。

ツールアクションを設定する

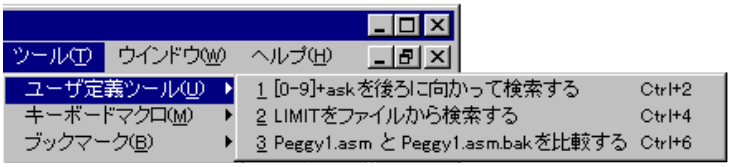
ツールアクションとは、ツールバーのボタンにアイテムに関連した動作を割り当てることを言います。Peggyは24個のツールボタンを持っています。各アイテムは1つだけツールボタンに動作を割り当てることができます。ツールアクションは、プロパティダイアログボックスを開いて設定します。



ダイアログボックスの[ファイル]ページ上に図のようなツールアクションを設定する部分があります。コンボボックスをドロップダウンして、1～24までのボタン番号を選択します。例えば、3番を選んだとします。

コンボボックスの下にある動作を選び、[OK]ボタンを押して、プロパティのダイアログボックスを閉じます。3番のツールボタンがアクティブになります。

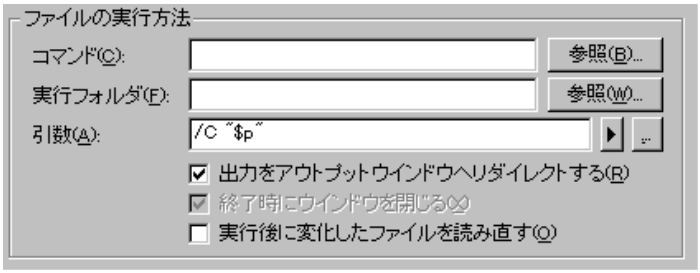
以後は3番のツールボタンを押すだけで、この場合は、そのファイルをPeggyで開くことになります。また、割り当てられているツールアクションをメニューからも実行することができます。



スクリプトファイルの実行

Makefileやバッチファイルなどタイプが[スクリプト]のアイテムに前述のマウスアクションとツールアクションを設定することで、それらのスクリプトを簡単に起動できます。

まず、プロパティダイアログボックスの[実行方法]ページで、[コマンド]、[引数]などを必要に応じて入力します。



次に[ファイル]ページのマウスアクション、ツールアクションを[実行する]に設定すれば、アイテムをマウスでダブルクリックしたり、ツールボタンをクリックするだけで、バッチファイルなどを起動するようになります。

【注意】このプロジェクト機能における「スクリプト」とは、一般的な外部プログラムのスクリプト言語を意味します。Peggyのビルトインスクリプト「MocaScript」の登録、実行とは関係ありません。

【ヒント】アイテムを右クリックして出てくるポップアップメニューで、[実行]を選択してもスクリプトファイルを実行することができます。

その他コマンド、引数などの詳細は、この後の『[実行方法]プロパティページ』を参照してください。

実行結果

スクリプトファイルなどの実行結果は、アウトプットウインドウに表示されます。

アウトプットウインドウ文字列の活用

アウトプットウインドウに出力された文字列を直接編集することはできません。マウスで文字列を選択してエディタ上にドラッグ＆ドロップすることでテキストとして編集することができます。また、アウトプットウインドウポップアップメニューから、[ウインドウへ保存]を選択すると、新しい編集ウインドウを作成し、アウトプットウインドウの内容をコピーする方法もあります。

バッチファイルを実行するときの注意

バッチファイルの中でオペレータからの入力を期待する記述は書かないでください。バッチファイルを実行させても、Peggyのアウトプットウインドウからはキー入力することができないからです。どうしても、キー入力が必要な場合は、実行ファイルのプロパティの[実行方法]のページにある[出力をアウトプットウインドウへリダイレクトする]のチェックをはずしてください。その場合、実行結果はコンソールウインドウを表示し、そこで実行します。キー入力も可能です。

タグジャンプ

プロジェクトにMakefileやバッチファイルなどを登録して実行すると、コンパイラやアセンブラからのエラーメッセージがアウトプットウインドウに表示されます。そこには多くの場合、そのエラー原因となったソースファイルのパスと行番号が含まれています。Peggyはこれらの情報を把握していて、ファイルを開いてエラーのあった行へジャンプできます。

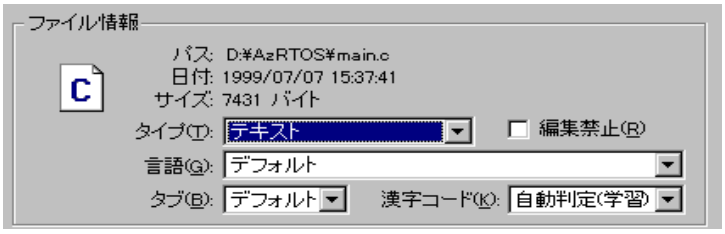
【注意】タグジャンプでは、エラーメッセージ中の行番号を頼りにジャンプするため、コンパイルの後にソースファイルを編集して行数が増減していると、正しいエラー個所へジャンプできないことがあります。

プロジェクトの言語モード

テキストファイルをプロジェクトに登録するとき、Peggyはそのファイルの拡張子から言語モードを決定しています。『第5章　言語モード/言語モードと拡張子マップ』で解説しているように、Peggyがプログラミング言語を知ることで、カラーコーディングが使えるなどの利点があります。では、具体的にその決定の手順を見てみましょう。

ファイルの言語が「デフォルト」とは

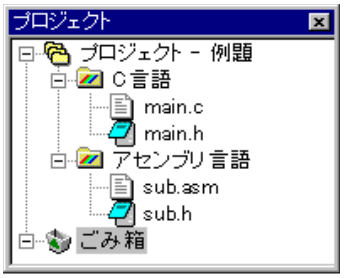
プロジェクトに登録されたテキストファイルのプロパティを見てみると、[言語]と[タブ]は「デフォルト」になっています。



この「デフォルト」の種明かしをする前に「プロジェクトデフォルト」と「グループデフォルト」の説明をします。

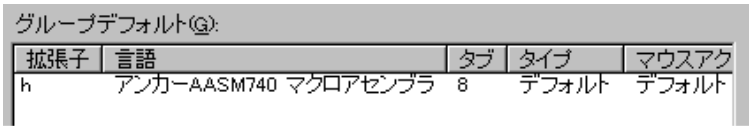
プロジェクトデフォルトとグループデフォルト

拡張子と言語の対応関係は[オプション設定]ダイアログボックスの[拡張子]ページにある[拡張子マップ]に定義されています。実はPeggyはこの[拡張子マップ]だけで言語モードを決定しているのではないのです。例えば、次のようなプロジェクトの場合で説明します。



「C言語」と「アセンブリ言語」という２つのグループがあります。両方に拡張子が<.h>のファイルが登録されています。[拡張子マップ]では、<.h>がC言語に対応しています。これでは、<sub.h>が実際はアセンブリ言語であっても、C言語と認識されてしまいます。

それを解決するため、プロジェクトとグループのプロパティの中に「プロジェクトデフォルト」と「グループデフォルト」があります。それぞれプロジェクト内、グループ内という適用範囲の違いはありますが、拡張子と言語・タブの対応を設定できます。この例の場合、「アセンブリ言語」グループの[グループデフォルト]を次のように設定すると、<sub.h>の言語とタブが正しく認識されます。



言語決定手順

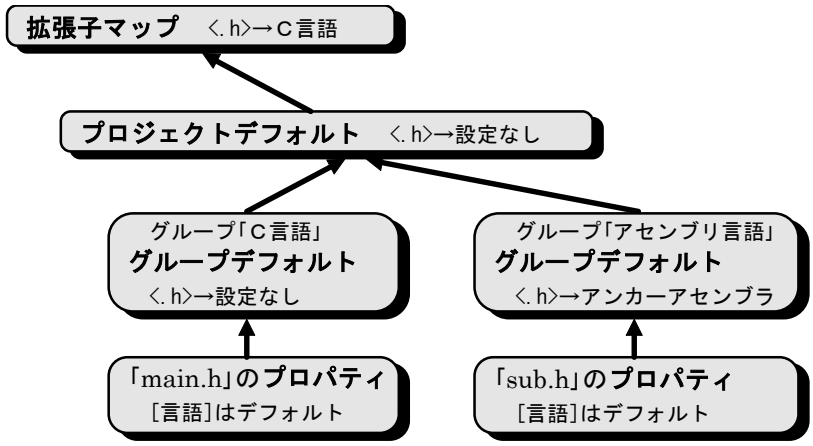
「デフォルト」の話に戻りましょう。プロパティの[言語]と[タブ]が「デフォルト」になっていると、Peggyはファイルの拡張子と

●拡張子マップ

●プロジェクトデフォルト

●グループデフォルト

の３つを参考に言語を決定します。ただし、グループデフォルトはグループごとにあるので、グループが階層構造になっていると複数のグループデフォルトが関係します。実際にPeggyが言語を判断する手順は図のようになっています。



テキストファイルの[言語]と[タブ]が「デフォルト」になっていると、そのファイルが属するグループの「グループデフォルト」に拡張子の設定を探し、グループが階層構造になっていたら下から順に「グループデフォルト」を見ていきます。さらに「プロジェクトデフォルト」、「拡張子マップ」の順に拡張子の設定を探します。

例の場合、<sub.h>は「グループデフォルト」で「アンカーアセンブラ」であると認識されます。<main.h>は途中の「グループデフォルト」と「プロジェクトデフォルト」に設定が無いので、「拡張子マップ」から「C 言語」と判断されるのです。

統合環境作り

前節で記述されているツールアクション、マウスアクションとプロジェクトの下に作成するグループ単位での言語設定を組み合わせれば、グループを一つのソフトウエア開発単位としてとらえ、統合マネージャやランチャとして使うことができます。具体的には、

1. 開発ターゲットに対してプロジェクトを新規に作成
2. 必要であればグループを作成
3. 必要なソースファイルを登録
4. ソースファイルをコンパイル、リンクするための Makefile を作成
5. その Makefile も登録
6. Makefile に対するアクションを設定

   ファイルの[タイプ]は[スクリプト]、

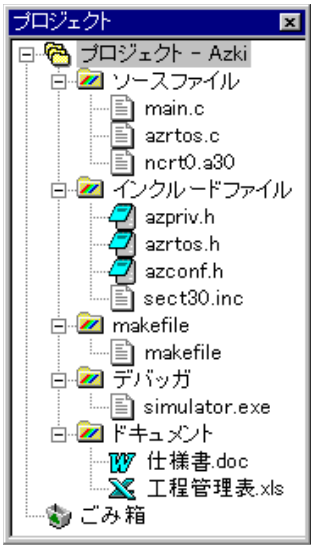
   実行方法の[コマンド]は[<make.exe>]

   実行方法の[引数]は[-f "$p"]
7. デバッガも登録

   ファイルの[タイプ]は[実行可能]

ここまで登録すれば、統合マネージャやランチャに相当する動作が行えます。また、ここに登録できるファイルの種類に制限はありませんから、必要に応じて Microsoft Wordで書かれた仕様書、Microsoft Excelで書かれたスケジュール表や状態遷移表などもグループに登録できます。これらファイルはデフォルトで[アプ

リケーションデータ]と認識されます。あとはユーザツールボタンやマウスのクリックで簡単に開くことができます。



プロパティ

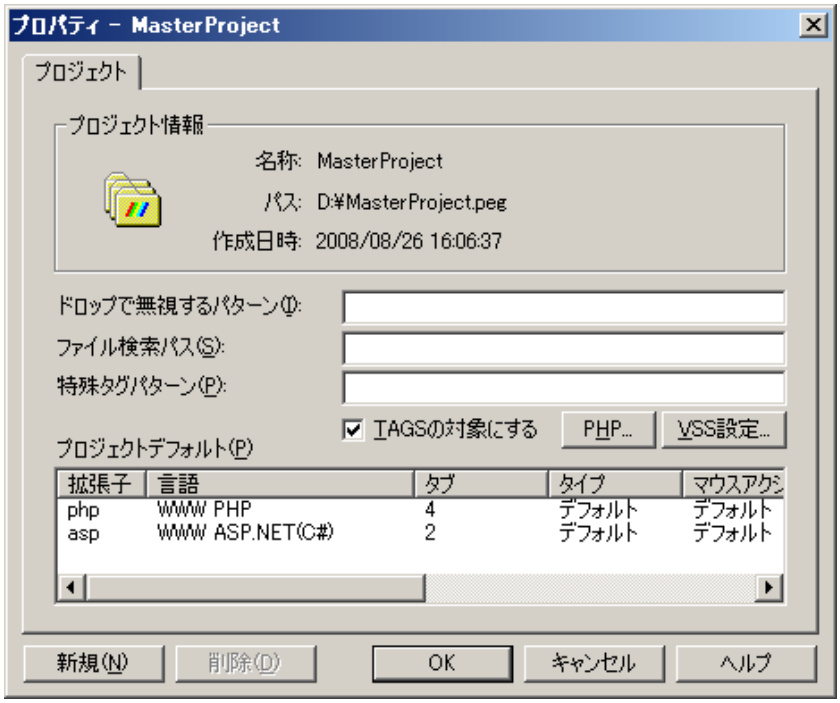
プロジェクトは個々のアイテム・グループに対して言語やマウスアクションなど種々の設定を記憶しています。それらをプロパティと呼びます。アイテムの種類によってプロパティの項目も変わってきます。

【メモ】プロジェクトに登録していないファイルのプロパティは、[エディタプロパティ]ダイアログボックスしか設定できません。

メニュー操作/機能	キー操作
[編集]-[プロパティ]	Alt+Enter

[プロジェクト] プロパティページ

[グループ] プロパティページ



[プロジェクト]プロパティページには、[プロジェクト情報]、[プロジェクトデフォルト]があり、グループやグループ直下のアイテムの場合、グループに関する情報として[グループ情報]、[グループデフォルト]になります。意味的には同じですので、以下の説明では「プロジェクト」で行います。

[名称]

プロジェクトの名称を表示しています。

[パス]

プロジェクトファイルのパスです。

[作成日時]

最初にプロジェクトが作成された日時です。

[ドロップで無視するパターン]([プロジェクト] プロパティページのみ)

エクスプローラなどからプロジェクトウインドウへドロップしたときに無視するプロジェクト固有のファイルパターンを指定します。Peggy全体の設定に加えて、このプロジェクトに限りドロップしたくないパターンを設定してください。

[ファイル検索パス]([プロジェクト] プロパティページのみ)

[ダイレクトオープン]コマンドで開くファイルを探すパスを指定します。複数のフォルダを指定するときは、セミコロン(;)で区切ってください。ここで指定された順にフォルダを探し、最初に見つかったファイルを開きます。

パスを直接指定する代わりに、${と}の間に環境変数名を記述することもできます。たとえば、${include}と指定すると、環境変数includeに指定してあるインクルー

ドファイル検索パスを取り込むことが可能です。パスと環境変を混在させてもかまいません。

【例】D:¥dev;${include}

[特殊タグパターン]([プロジェクト]プロパティページのみ)

このプロジェクトに特有な特殊タグジャンプパターンを正規表現形式で設定します。Peggy全体の設定は、[全般]ページの[特殊パターン]で行います。指定方法は[全般]ページの[特殊パターン]と同じです。

[TAGSの対象にする]

シンボル検索の対象にするかどうかを指定します。

【参照】『第４章 検索・比較/シンボル検索とプロジェクト』

[PHP] (Peggy Pro for PHPのみ)

このボタンを押すと[PHPデバッガの設定]ダイアログボックスが表示されます。プロジェクト固有のソースファイル位置とURLの関係を定義できます。

[VSS設定]

このボタンを押すと[VSSの設定]ダイアログボックスが表示されます。このグループ(またはプロジェクト)に対するVSSの設定を入力します。

[プロジェクトデフォルト]・[グループデフォルト]

拡張子と言語などの対応関係を表示しています。表示されている値を変更する場合は、値のところを左マウスボタンでクリックしてください。リストから選択するかキーボードから入力します。ファイルをプロジェクトに登録するとき、その拡張子が[グループデフォルト]にあれば、それに対応する言語などが設定されます。なければ、すぐ上のグループの設定、それもなければさらにその上というようにさかのぼり、最後はプロジェクトのプロパティ[プロジェクトデフォルト]の設定が採用されます。そこにもなければ、Peggyが最適と考える値を設定します。なお、[言語]と[タブ]については、さらに[オプション設定]ダイアログボックスの拡張子マップまでさかのぼります。

[新規]

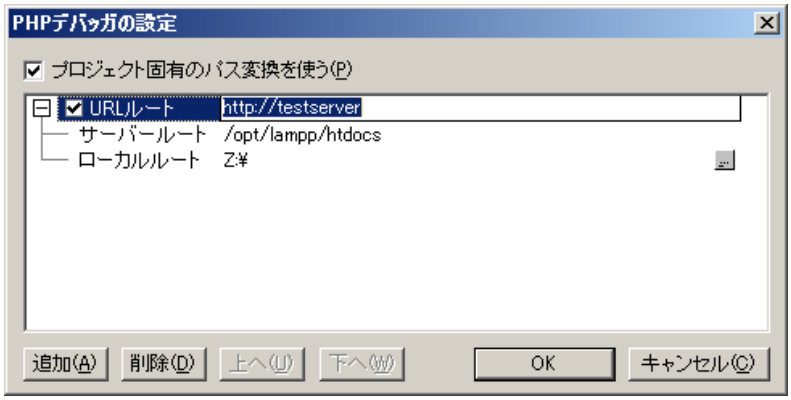
[プロジェクトデフォルト]のリストにない新たな拡張子と言語などの対応を追加します。このボタンを押すとダイアログボックスが表示されるので、拡張子を入力します。

[削除]

[プロジェクトデフォルト]のリストにある拡張子と言語などの対応の組み合わせを削除します。

[PHP デバッガの設定] ダイアログボックス

プロジェクト別にPHPデバッグの設定するためのダイアログボックスです。デバッグ対象のURLと、それに対応するWebサーバー、ローカルコンピュータそれぞれのファイルの場所を指定します。



【注意】このダイアログボックスが表示されるのは、Peggy Pro for PHPのみです。

[プロジェクト固有のパス変換を使う]

プロジェクト固有のパス変換を使うか、デフォルトのパス変換を使うかを選択します。デフォルトのパス変換は、[オプション設定]ダイアログボックスの[デバッグ]ページで指定します。

[URL ルート]

対象となるWebサーバーのトップレベルURLを指定します。チェックを外すと、一時的にOFFにできます。

[サーバールート]

Webサーバーからみたソースファイル群れが置いてあるトップレベルのフォルダ(ドキュメントルート等)を指定します。Peggy と Webサーバーが同じコンピュータ上にあるときは省略できます。省略すると、[ローカルルート] と同じと見なします。

[ローカルルート]

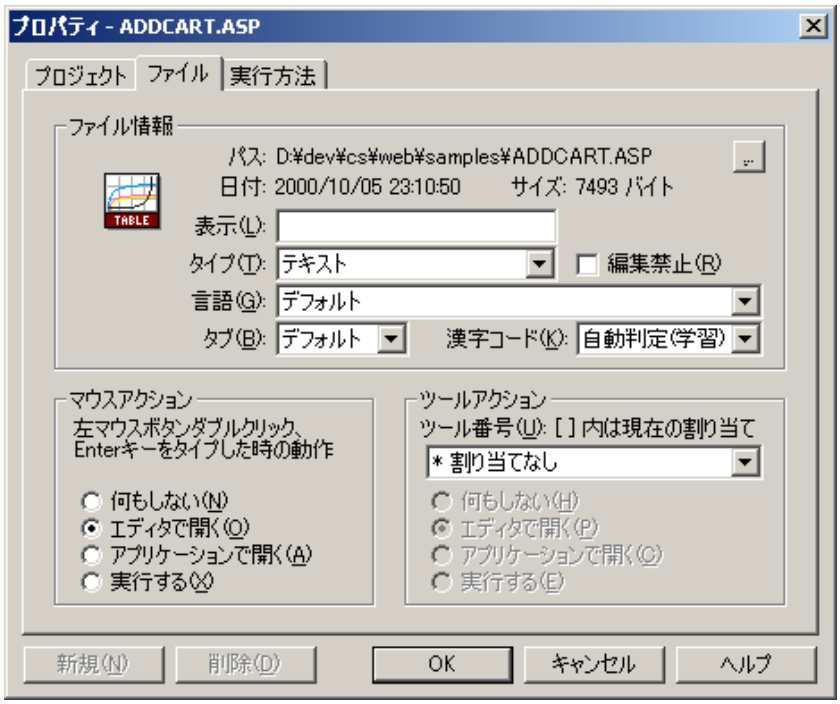
Peggy Pro for PHPからみたソースファイル群が置いてあるトップレベルフォルダ(ドキュメントルート等)のパスを指定します。ここで指定したフォルダ以下にあるファイルを「ソースファイル」と見なします。

[追加]、[削除]、[上へ]、[下へ]

複数登録されているときは、上から順番に一致するものを検索します。[追加]、[削除]、[上へ]、[下へ]ボタンにより、追加、削除、上へ移動、下へ移動できます。

【参照】『第10章 PHPプログラムのデバッグ／デバッグの準備』

[ファイル] プロパティページ



ファイルアイテムの場合に表示されます。

[パス]

ファイルのフルパスを表示しています。

[...]

参照しているファイルを変更します。プロジェクトの移動により参照先のファイルを修正するときに使います。

[日付]

ファイルの最終更新日時を表示しています。

[サイズ]

ファイルサイズを表示しています。

[表示]

ファイル名の代わりにプロジェクトウインドウへ表示する文字列を指定します。同じファイルが複数登録してある場合、[表示]文字列を指定することで区別しやすくなります。

[タイプ]

[テキスト]、[スクリプト]、[アプリケーションデータ]、[実行可能]の４タイプの中から分類されたファイルタイプを表示しています。

[言語]、[タブ]

言語の種類とタブ間隔を表示しています。[デフォルト]になっている場合でも、Peggyは言語の種類などを判断しています。その手順は、この章の『プロジェクトの言語モード』を参照してください。

[編集禁止]

ファイルが書き込み禁止属性の場合、[編集禁止]にチェックがつきます。書き込み禁止属性のファイルは編集できません。このチェックをはずすことで編集できるようになりますが、別のファイル名で保存する必要があります。

[漢字コード]

このファイルを開くときの漢字コードを指定します。

選択	動作
[自動判定(毎回)]	毎回漢字コードを調べてファイルを開きます。
[自動判定(学習)]	前回開いたときの自動判定結果、または保存したときの漢字コードと同じであると仮定してファイルを開きます。これがデフォルトです。いったん自動判定が成功すると、その結果をプロジェクト内に記憶します。次回からは判定の必要がなくなるので、ファイルを開く操作が少し早くなります。
[Shift JIS]	漢字コードがShift JISのファイルであるとして開きます。
[EUC]	漢字コードがEUCのファイルであるとして開きます。
[JIS]	漢字コードがJISのファイルであるとして開きます。
[UTF-7]	UTF-7ファイルであるとして開きます。
[UTF-8/8N]	UTF-8、UTF-8Nファイルであるとして開きます。
[Unicode]	Unicodeファイルであるとして開きます。

【注意】漢字コードの自動判定は、正しく判定できないこともあります。特に短いファイルでは、判定の材料が少ないため誤った漢字コードでファイルを開く可能性が高くなります。

正しくない漢字コードで開いた場合は、メニューの[ファイル]-[開き直す]コマンドで正しい漢字コードを指定し、ファイルを読み直してください。[自動判定(学習)]モードであれば、ユーザが指定して開き直した漢字コードを記憶し、次からは正しく開くことを学習します。

[マウスアクション]

アイテムを左マウスボタンでダブルクリックしたり、Enterキーをタイプしたときの動作を選択します。

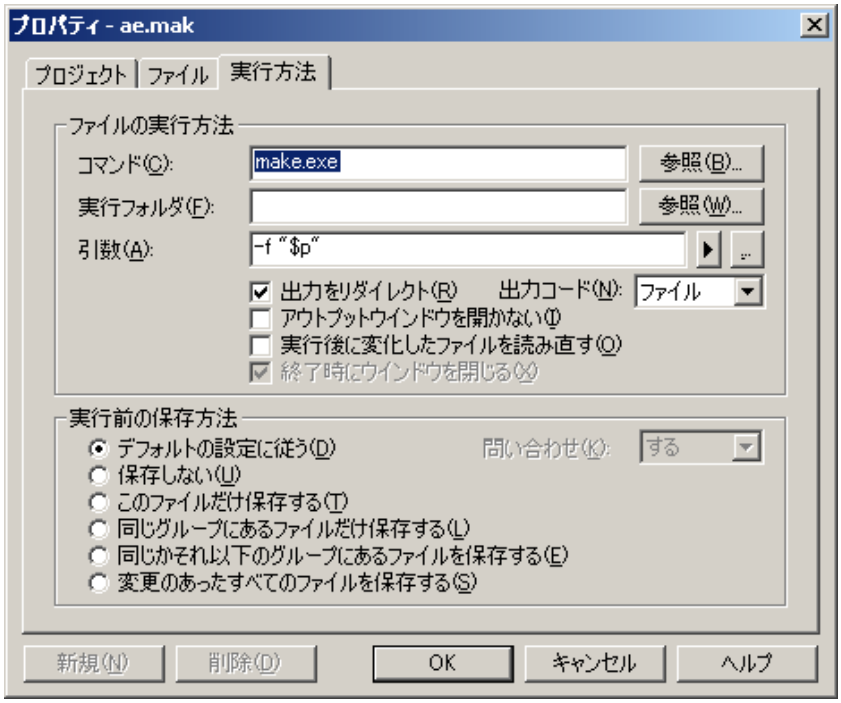
選択	動作
[何もしない]	Peggyは何もしません。
[エディタで開く]	Peggyでファイルを開きます。
[アプリケーションで開く]	Windowsに登録されているファイルタイプに従い、関連付けされたアプリケーションを起動し、ファイルを開きます。
[実行する]	[スクリプト]または[実行可能]タイプのファイルの場合だけ選択できます。Windows上で実行します。

[ツールアクション]

[マウスアクション]で選択できる４つの動作を、ツールボタン[ ]に割り当てることができます。そこには、そのアイテムに割り当てられているツール番号と、押されたときの動作を表示しています。

[ツール番号]が[* 割り当てなし]になっていたら、そのアイテムはツールボタンにまだ割り当てられていません。すでに割り当てられている番号の変更は、リストから別の番号を選んでください。４種類の動作は、[マウスアクション]と同じです。

[実行方法] プロパティページ



ファイルアイテムで[タイプ]が[スクリプト]、[実行可能]の場合のみ、設定可能です。

[コマンド]

[タイプ]が[スクリプト]の場合だけ入力可能です。スクリプトを動かすコマンドを表示しています。[参照]ボタンを押すと[コマンドの指定]ダイアログボックスが開くので、コマンドを選択することができます。ファイルによってはPeggyが初期値を入れています。

ファイルの種類	コマンドの初期値
Makefile、拡張子が　.mak	make.exe
拡張子が　.pl	perl.exe
拡張子が　.awk	awk.exe
拡張子が　.sed	sed.exe
拡張子が　.rb	ruby.exe
拡張子が　.ms	moca.exe

【ヒント】グループのプロパティの[グループデフォルト]の[コマンド]に値が設定されていれば、その値が初期値として入ります。

【ヒント】バッチファイルのときは、[コマンド] は必要ありません。OS により、<command.com>または<cmd.exe>が呼ばれます。

[実行フォルダ]

コマンドを実行するフォルダを表示しています。

●指定無し…そのファイルがあるフォルダで実行。

●絶対パス…その指定のフォルダで実行。

●相対パス…ファイルがあるフォルダからの相対パスで示されるフォルダで実行。

指定のフォルダが存在しない場合は、エラーになります。次の[引数]で説明する$hのような特殊記号(ただしアドホックパラメタ$i、$j、$o を除く)も使えます。

【注意】ドル記号($)を含むパスを指定するときは、$$を指定してください。これは、$が特殊記号の始まりとして解釈されてしまうためです。詳しくは、次の[引数]を参照してください。

[引数]

コマンドを実行するときの引数を表示しています。ファイルによってはPeggyが初期値を設定します。

ファイルの種類	引数の初期値
Makefile、拡張子が　.mak	-f ″$p″
バッチファイル、拡張子が　.bat	/C ″$p″
拡張子が　.cmd	/C ″$p″
拡張子が　.pl	″$p″
拡張子が　.rb	″$p″
拡張子が　.awk	-f ″$p″
拡張子が　.sed	-f ″$p″
拡張子が　.ms	″$p″

【ヒント】グループのプロパティの[グループデフォルト]の[引数]に値が設定されていれば、その値が初期値として入ります。

引数の中の特殊記号は、実行時に次のように展開されます。これらの特殊記号は、[▶]ボタンをクリックして現れるメニューから選んで入力することができます。

記号	意味
$f[:opt]	ファイル名と拡張子
$p[:opt]	フルパス
$s[:opt]	ショート形式(DOSの8.3形式)のフルパス
$n[:opt]	拡張子なしのファイル名
$e[:opt]	拡張子のみ
$d[:opt]	ファイルのあるフォルダ
$a[:opt]	現在アクティブなドキュメントのフルパス

$b[:opt]	現在アクティブなドキュメントのファイル名と拡張子
$c[:opt]	現在アクティブなドキュメントのフォルダ
$m[:opt]	現在アクティブなドキュメントの拡張子なしのファイル名
$A[:opt]	2番目にアクティブなドキュメントのフルパス
$B[:opt]	2番目にアクティブなドキュメントのファイル名と拡張子
$C[:opt]	2番目にアクティブなドキュメントのフォルダ
$M[:opt]	2番目にアクティブなドキュメントの拡張子なしのファイル名
$w	現在アクティブなドキュメントのカーソル位置の単語
$i[.ext][:opt]	現在アクティブなドキュメントの選択範囲を保存したテンポラリファイルのフルパス。.extを指定すると、テンポラリファイルの拡張子を変更できる。省略すると.tmpが拡張子となる。
$j[.ext][:opt]	$I[.ext]と同じであるが、選択範囲がない場合はテンポラリファイルを作らず、常にNULとなる。この場合は拡張子なしとなる。
$o[.ext][:opt]	コマンドからの出力を受け取るテンポラリファイルを作成し、そのフルパスと置き換える。このファイルに出力されたデータは、実行終了後、そのときアクティブであるドキュメントの選択範囲と置換される。.extを指定すると、テンポラリファイルの拡張子を変更できる。省略すると.tmpが拡張子となる。
$t[.ext][:opt]	アクティブなドキュメントの内容をShift JISファイルとしてテンポラリファイルへ出力し、そのフルパスと置き換える。$tを使うことで、ドキュメントを対応するファイルへ保存することなくコマンドへ渡すことが可能となる。 .extを指定すると、テンポラリファイルの拡張子を変更できる。省略すると.tmpが拡張子となる。
$l	現在アクティブなドキュメントの選択範囲の最初の行の行番号
$g	現在アクティブなドキュメントの言語モード識別名
$1～$9	アドホックパラメタ(実行フォルダの指定には使用不可)
${ENV}	環境変数ENVの値
$h[:opt]	プロジェクトファイルのあるフォルダ
$.[:opt]	カレントフォルダ
<	<の後にあるファイルを標準入力へリダイレクト
>	>の後にあるファイルへ標準出力をリダイレクト
>>	>>の後へあるファイルへ標準出力を追加モードでリダイレクト
$$	’$’ そのもの
$>	’>’ そのもの
$<	’<’ そのもの

[:opt]の部分には、コロン(:)に続けて英文字と数字を使ってパスを変形させるためのオプションを指定できます。オプションは複数指定できます。オプションは、以下の英文字か数字、ハイフン(-)以外の文字で終了したとみなされるため、途中に空白を入れることはできません。また、コロン(:)によりオプションの終わりを明示的に指定することも可能です。

u	パス中の英文字をすべて大文字に変換します。
l	パス中の英文字をすべて小文字に変換します。
f	パスの区切り文字である円記号(¥)をすべてフォワードスラッシュ (/)に変換します。
s	ショートパス形式に変換します。
数値の範囲	数字とハイフンでコマンドの終了コードの範囲を指定します。範囲が指定できるのは、$oの後に続けたオプションだけです。ハイフンを使うことで範囲を指定します。範囲の前後を省略すると、それぞれ0、255と見なします。 【例】0 終了ステータス0を指定 1-3 終了ステータス1～3を指定 2- 終了ステータス2～255を指定 -5 終了ステータス0～5を指定
:	オプションの終わりを示します。省略することもできます。

【例】特殊記号の使用例

″$p″	このアイテムのフルパスを引数として渡します。パスの中にスペースが含まれている場合もあるため、全体を二重引用符で囲んでいる方が安全です。
$s	このアイテムのフルパスをショート形式で渡します。ロング形式やスペースを含んだパスを上手く処理できないコマンドに対して使うとよいでしょう。
>>“$d¥output.log“	コマンド標準出力をこのアイテムと同じフォルダにあるファイルoutput.logへ追加モードでリダイレクトします。
″$i″	選択範囲を保存したテンポラリファイルのフルパスを渡します。単語や文字の種類をカウントするコマンドなどと組み合わせて使えば、選択範囲の文字の構成を調べることもできます。Windows 2000などでは、テンポラリファイルのパスに空白が含まれることがあるため、全体を二重引用符で囲んでください。または、:sオプションによりショート形式のパスに変換する方法もあります。
″$t:sf″	[実行前の保存方法]を[保存しない]とし、引数に$tを使うと、編集中のファイルを保存せずに最新の状態をコマンドに渡すことが可能です。 オプションによりテンポラリファイルのパスをショート形式、フォワードスラッシュ (/)区切りに変換します。
<$i >$o:0	アクティブドキュメントの選択範囲をコマンドの標準入力へリダイレクトし、コマンドの標準出力を選択範囲と置換します。これによりいわゆるフィルタコマンドを編集機能の一部として組み込むことが可能になります。 :0と指定しているので、コマンドの実行終了ステータス値が0のときだけ置換します。つまり、コマンドが失敗したときには置換しません。

>> $o	コマンドの標準出力をアクティブドキュメントの選択範囲の後へ追加挿入します。
"$i.pl"	アクティブドキュメントの選択範囲を.plの拡張子を持つテンポラリファイルへ保存してからコマンドを実行します。Perlのスクリプトを部分実行させることができます。
-E "$i.c"	Cコンパイラ自体をプロジェクトに登録している場合、プリプロセッサに選択部分だけを処理させ、結果をアウトプットウインドウへ表示させるようなこともできます。マクロの展開結果を簡単に調べることが可能です。
"$1" "$2"	実行時に2つの引数をユーザに問い合わせる[アドホックパラメタ入力]ダイアログボックスを表示し、入力された文字列を二重引用符で囲んでコマンドへ渡します。
"$a" "$A"	編集しているファイルの上から2つのフルパスを引数として渡します。外部の比較コマンドなどを起動するような場合に便利です。

【ヒント】>、>>により標準出力をリダイレクトしても、標準エラー出力は常にアウトプットウインドウへ送られます。

【注意】選択範囲との入出力を表す$i、$j、$oは、[出力をアウトプットウインドウへリダイレクトする]モードのときのみ利用できます。このチェックがOFFの場合は、コマンドの終了を待たないため、テンポラリファイルの後始末することができなくなるためです。

[▶]

このボタンをクリックすると、引数として指定できる特殊パラメタのメニューを表示します。特殊パラメタを忘れてしまっても、このメニューから選んで入力できます。

[…]

アドホックパラメタのラベルやデフォルト値を設定するための[アドホックパラメタ設定]ダイアログボックスを表示します。

[出力をリダイレクト]

これをチェックすると、コマンドが出力するメッセージをアウトプットウインドウに表示し、コマンドの実行終了を待ちます。したがって、コマンドが終了するまで次のコマンドを実行できません。引数に選択範囲を表す特殊記号$i、$j、$oを使う場合は、必ずチェックしてください。

チェックをはずすと、コンソールアプリケーションであればコンソールウインドウが表示されます。コマンドの実行終了を待ちません。

[出力コード]

ツールの出力がShift JIS以外の漢字コードで出力される場合、それをShift JISに変換しながらアウトプットウインドウへリダイレクトします。以下の漢字コードが選択できます。

[ファイル]	ファイルの漢字コードと同じコードで出力されると仮定します。例えば、EUCで記述されているPerlスクリプトからはEUCのコードで出力されるとみなし、Shift JISに変換しながらアウトプットウインドウへリダイレクトします。
[アクティブ]	編集中のファイルと同じコードで出力されると仮定します。引数に$a(アクティブドキュメントのフルパス)などを使って編集中のスクリプトを実行させたいときに便利です。
[Shift JIS]	Shift JISで出力されると仮定し、そのままアウトプットウインドウへリダイレクトします。
[EUC]	EUCで出力されると仮定し、Shift JISに変換しながらアウトプットウインドウへリダイレクトします。
[JIS]	JISで出力されると仮定し、Shift JISに変換しながらアウトプットウインドウへリダイレクトします。
[UTF-7]	UTF-7で出力されると仮定し、Shift JISに変換しながらアウトプットウインドウへリダイレクトします。
[UTF-8]	UTF-8で出力されると仮定し、Shift JISに変換しながらアウトプットウインドウへリダイレクトします。
[Unicode]	Unicodeで出力されると仮定し、Shift JISに変換しながらアウトプットウインドウへリダイレクトします。BOM(バイトオーダーマーク)がある場合は、それに従ってバイト順を決めます。BOMがない場合は、LITTLE ENDIANとみなして変換します。

[アウトプットウインドウを開かない]

コマンドを実行する直前に、アウトプットウインドウが閉じられていても開かず、そのまま実行します。

[実行後に変化したファイルを読み直す]

スクリプトやコマンドの実行後に、Peggyで開いていて外部で変更された可能性のあるファイルを読み直します。

[終了時にウインドウを閉じる]

これをチェックするとコマンド実行後、コンソールウインドウは自動的に閉じられます。チェックをはずすと、何かのキーを押すまでコンソールウインドウは開いたままです。

[実行前の保存方法]

コマンドを実行するときに、変更があったファイルを保存する方法を選択します。[デフォルトの設定に従う]は、[オプション設定]ダイアログボックスの[編集]ページ、[実行前保存]の設定が使われます。

アドホックパラメタ

スクリプトや実行ファイルを実行するときに、その引数をユーザから入力できるようにするための特殊パラメタです。

[実行方法]プロパティページの[引数]に$1～$9の記述があると、それは実行時にユーザが入力する引数であると解釈し、[アドホックパラメタ入力]ダイアログボックスを表示します。

【注意】アドホックパラメタは、$1～$9の9個しか使えません。$10は、$1と0と解釈されます。

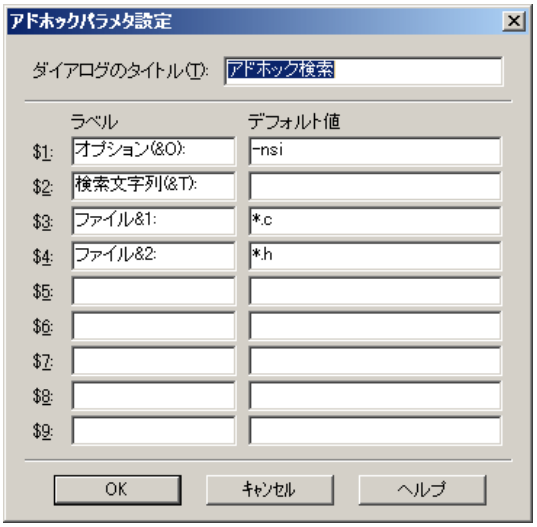
【注意】アドホックパラメタは、引数にのみ使用できます。アドホックパラメタで実行フォルダを指定することはできません。

【ヒント】[アドホックパラメタ設定]ダイアログボックスにより、予めパラメタのラベル名や初期値、入力ダイアログボックスのタイトルなどを設定しておくと、意味のある[アドホックパラメタ入力]ダイアログボックスが表示され、入力間違いを防止できます。

[アドホックパラメタ設定] ダイアログボックス

[実行方法]プロパティページから[...]ボタンをクリックすると、[アドホックパラメタ設定]ダイアログボックスが表示されます。

コマンドやスクリプト実行時、引数にアドホックパラメタが使用されていた場合に表示される[アドホックパラメタ入力]ダイアログボックスに初期値として表示されるデフォルト値や、ラベル名を設定します。意味のあるラベル名やデフォルト値を設定しておくことで、アドホックパラメタを使いやすくできます。



[ダイアログのタイトル]

[アドホックパラメタ入力]ダイアログボックスのタイトルバーに表示する文字列を指定します。意味のあるタイトルを表示すると、何のための入力かが分かりやすくなります。

[$1] ～ [$9]

それぞれアドホックパラメタの$1～$9に対応したラベルとデフォルト値であることを示しています。

[ラベル]

[アドホックパラメタ入力]ダイアログボックスの各入力欄のラベルとして表示する文字列を指定します。入力欄のキーシンボルを指定したい場合は、アンパサンド(&)に続けて英数字1文字を指定してください。

[デフォルト値]

[アドホックパラメタ入力]ダイアログボックスを表示した直後に各欄のデフォルト値となる文字列を指定します。

[アドホックパラメタ入力] ダイアログボックス

スクリプトや実行ファイルの引数としてアドホックパラメタ($1～$9)が指定されていた場合、実行開始直前にこの[アドホックパラメタ入力]ダイアログボックスが表示され、ユーザが実引数の入力をします。

表示されている実引数を入力し、[OK]ボタンをクリックすると、アドホックパラメタ($1～$9)を実パラメタに置換してからコマンドやスクリプトを実行します。

予め[アドホックパラメタ設定]ダイアログボックスにより、各パラメタのラベルやデフォルト値を設定しておくと、上図のように意味のあるアドホック入力ダイアログボックスにすることができます。

[アドホックパラメタ設定]ダイアログボックスで何も指定していない場合、ラベルには$1～$9と、アドホックパラメタそのものが表示されます。

【メモ】アドホックパラメタは、$1～$9のうち、実際に使われているものしか表示しません。

[キーボードマクロ] プロパティページ

キーボードマクロアイテムの場合に表示されます。

[マクロ名]

このアイテムに付けられた名前を表示しています。

[作成日付]

キーボードマクロをプロジェクトに登録した日時を表示しています。

[ストローク数]

キーボードマクロのキーストローク数を表示しています。

[マウスアクション]

アイテムを左マウスボタンでダブルクリックしたり、Enterキーをタイプしたときの動作を選択します。キーボードマクロについての詳しい説明は、キーボードマクロを参照してください。

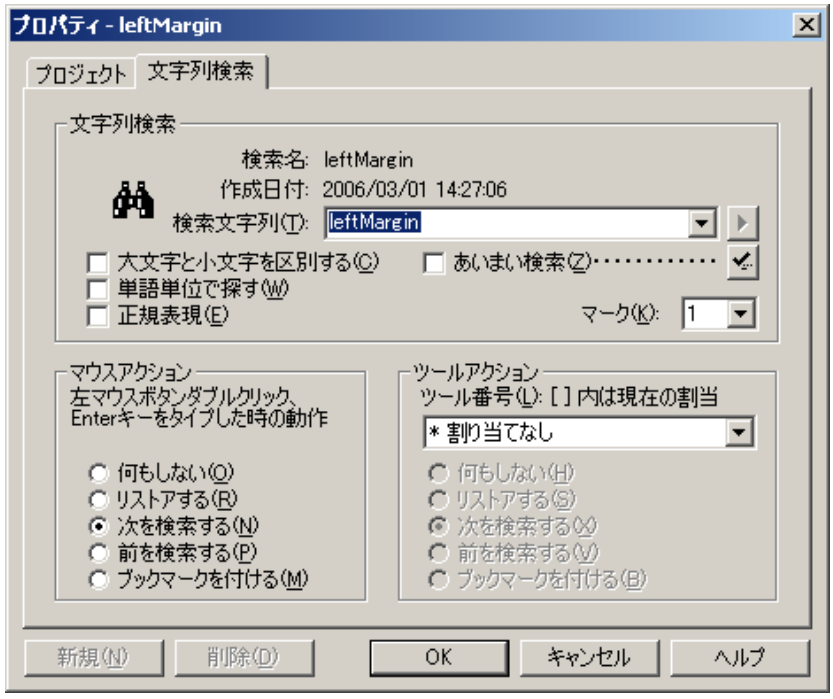
選択	動作
[何もしない]	Peggyは何もしません。
[リストアする]	アイテムのキーストロークの内容を、1チャンネルのキーボードマクロにコピーします。
[再生する]	そのアイテムに記憶されているキーストロークを再生します。

[ツールアクション]

[マウスアクション]で選択できる3つの動作を、ツールボタン[]に割り当てることができます。そこには、そのアイテムに割り当てられているツール番号と、押されたときの動作を表示しています。

[ツール番号]が[* 割り当てなし]になっていたら、そのアイテムはツールボタンにまだ割り当てられていません。すでに割り当てられている番号の変更は、リストから別の番号を選んでください。3種類の動作は、[マウスアクション]と同じです。

[文字列検索] プロパティページ



文字列検索アイテムの場合に表示されます。

[検索名]

このアイテムに付けられた名前を表示しています。

[作成日付]

このアイテムをプロジェクトに登録した日時を表示しています。

[検索文字列]、[大文字と小文字を区別する]、[単語単位で探す]、[正規表現]、[あいまい検索]、[マーク]

これらは文字列検索と同じです。詳しくは、『第4章 検索・比較・置換/検索』を参照してください。

[マウスアクション]

アイテムを左マウスボタンでダブルクリックしたり、Enterキーをタイプしたときの動作を表示しています。それぞれの動作の意味を説明します。

選択	動作
[何もしない]	Peggyは何もしません。

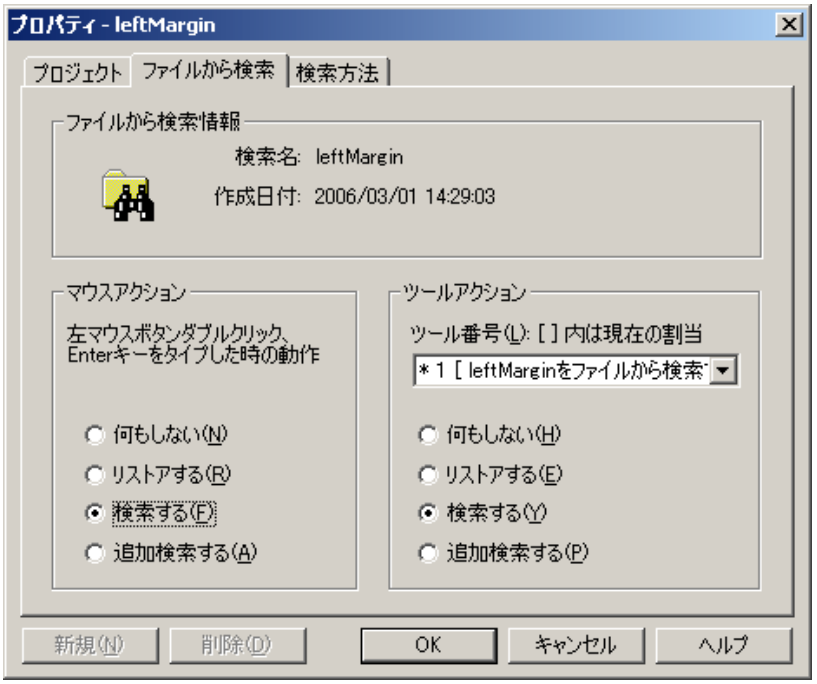
[リストアする]	アイテムの検索条件を、Peggyの現在の検索条件にコピーします。
[次を検索する]	アイテムの検索条件で、カーソル位置から下方向へ検索を行います。
[前を検索する]	アイテムの検索条件で、カーソル位置から上方向へ検索を行います。
[ブックマークをつける]	アクティブなドキュメントの全体を検索し、文字列の見つかった行にブックマークを付けます。

[ツールアクション]

[マウスアクション]で選択できる5つの動作を、ツールボタン[ ]に割り当てることができます。そこには、そのアイテムに割り当てられているツール番号と、押されたときの動作を選択します。

[ツール番号]が[* 割り当てなし]になっていたら、そのアイテムはツールボタンにまだ割り当てられていません。すでに割り当てられている番号の変更は、リストから別の番号を選んでください。5種類の動作は、[マウスアクション]と同じです。

[ファイルから検索] プロパティページ



ファイルから検索アイテムの場合に表示されます。

[検索名]

このアイテムに付けられた名前を表示しています。

[作成日付]

このアイテムをプロジェクトに登録した日時を表示しています。

[マウスアクション]

アイテムを左マウスボタンでダブルクリックしたり、Enterキーをタイプしたときの動作を表示しています。それぞれの動作の意味を説明します。

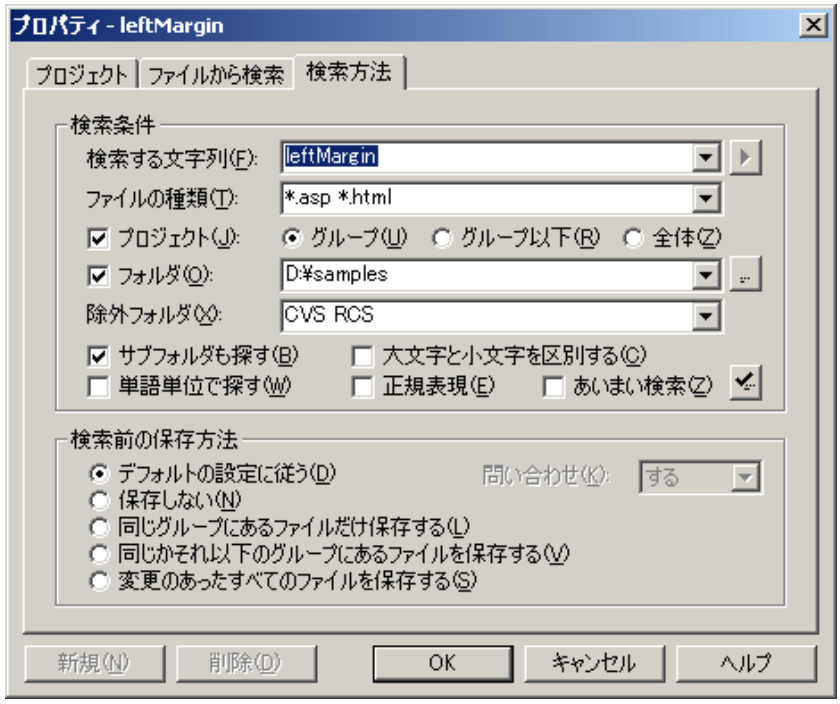
選択	動作
[何もしない]	Peggyは何もしません。
[リストアする]	アイテムの検索条件を、Peggyの現在の検索条件にコピーします。
[検索する]	アイテムの検索条件で、ファイルから検索を実行します。
[追加検索する]	アイテムの検索条件で検索を開始しますが、前回の検索結果を消さずに、今回の検索結果をその後に追加します。

[ツールアクション]

[マウスアクション]で選択できる4つの動作を、ツールボタン[ ]に割り当てることができます。そこには、そのアイテムに割り当てられているツール番号と、押されたときの動作を表示しています。

[ツール番号]が[* 割り当てなし]になっていたら、そのアイテムはツールボタンにまだ割り当てられていません。すでに割り当てられている番号の変更は、リストから別の番号を選んでください。4種類の動作は、[マウスアクション]と同じです。

[検索方法] プロパティページ



ファイルから検索アイテムの場合に表示されます。

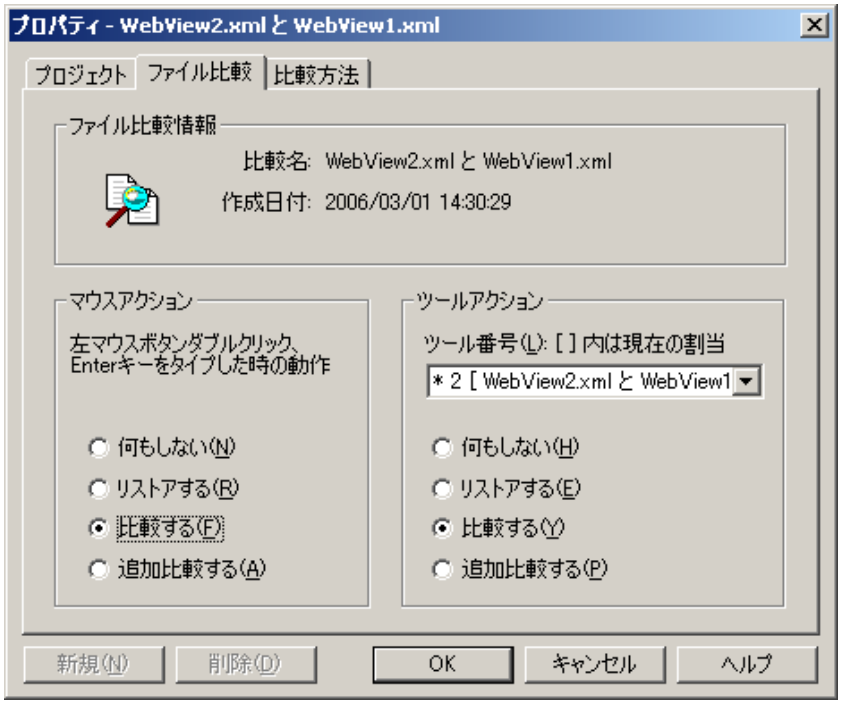
[検索する文字列]、[ファイルの種類]、[プロジェクト]、[フォルダ]、[除外フォルダ]、[サブフォルダも探す]、[正規表現]、[あいまい検索]、[大文字と小文字を区別する]、[単語単位で探す]

これらはファイルから検索と同じです。詳しくは、『第4章 検索・比較・置換/ファイルから検索』を参照してください。

[検索前の保存方法]

検索を実行する前に、開いているファイルの保存方法を選択します。[デフォルトの設定に従う]をチェックした場合は、[オプション設定]ダイアログボックスの[編集]ページでの[実行前保存]の設定に従います。ファイルを保存するときに確認したい場合は、[問い合わせ]を[する]にしてください。

[ファイル比較] プロパティページ



ファイル比較アイテムの場合に表示されます。

[比較名]

このアイテムに付けられた名前を表示しています。

[作成日付]

このアイテムをプロジェクトに登録した日時を表示しています。

[マウスアクション]

アイテムを左マウスボタンでダブルクリックしたり、Enterキーをタイプしたときの動作を選択します。それぞれの動作の意味を説明します。

選択	動作
[何もしない]	Peggyは何もしません。
[リストアする]	アイテムの比較条件を、Peggyの現在の比較条件にコピーします。

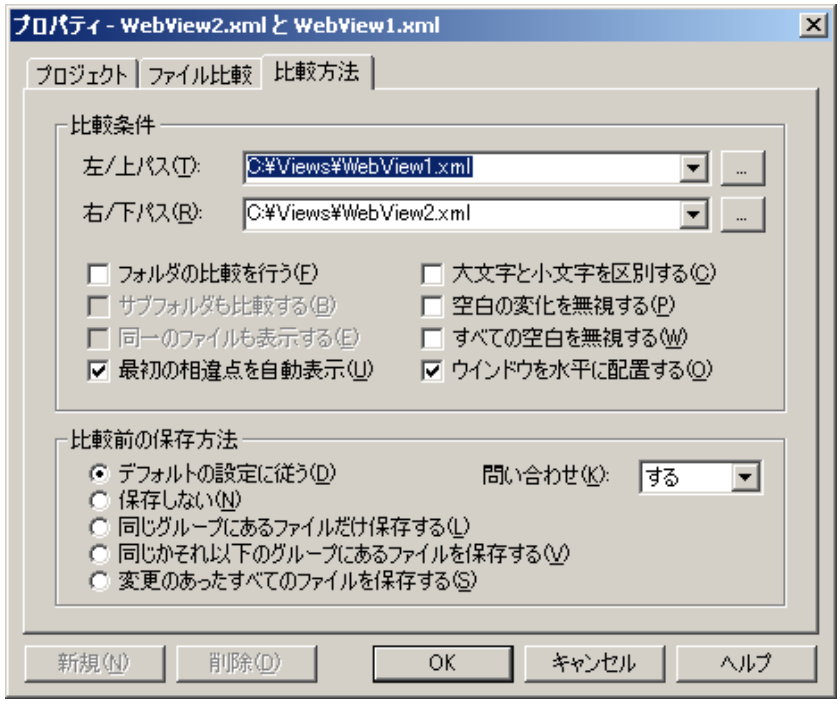
[比較する]	アイテムの比較条件で、ファイル比較を実行します。
[追加比較する]	アイテムの比較条件で比較を開始しますが、前回の比較結果を消さずに、今回の比較結果をその後に追加します。

[ツールアクション]

[マウスアクション]で選択できる4つの動作を、ツールボタン[ ]に割り当てることができます。そこには、そのアイテムに割り当てられているツール番号と、押されたときの動作を表示しています。

[ツール番号]が[* 割り当てなし]になっていたら、そのアイテムはツールボタンにまだ割り当てられていません。すでに割り当てられている番号の変更は、リストから別の番号を選んでください。4種類の動作は、[マウスアクション]と同じです。

[比較方法] プロパティページ



ファイル比較アイテムの場合に表示されます。

[左 / 上パス]、[右 / 下パス]、[フォルダの比較を行う]、[サブフォルダも比較する]、[同一のファイルも表示する]、[最初の相違点を自動表示]、[大文字と小文字を区別する]、[空白の変化を無視する]、[すべての空白を無視する]、[ウインドウを水平に配置する]

これらはファイル比較と同じです。詳しくは、『第4章 検索・比較・置換/ファイル比較』を参照してください。

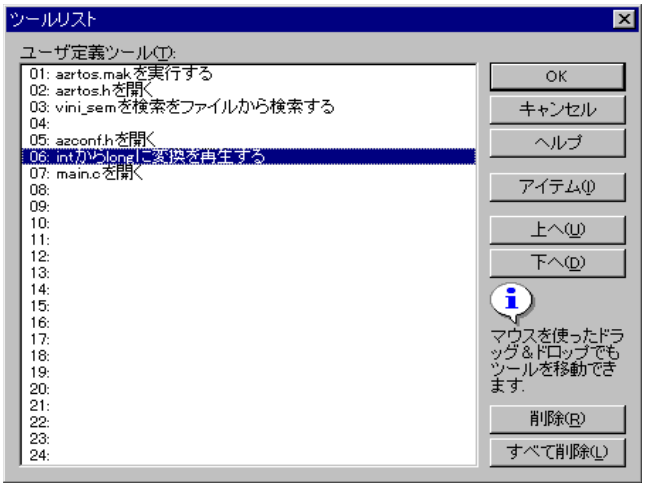
[比較前の保存方法]

ファイル比較を実行する前に、開いているファイルの保存方法を選択します。[デフォルトの設定に従う]をチェックした場合は、[オプション設定]ダイアログボックスの[編集]ページでの[実行前保存]の設定に従います。ファイルを保存するときに確認したい場合は、[問い合わせ]を[する]にしてください。

[ツールリスト] ダイアログボックス

[ツール]-[ツールリスト] コマンド

ユーザが定義できる24個のツールの割り当て一覧を表示し、番号の付け替えや削除が行えます。このダイアログボックスは、[ツール]-[ツールリスト]コマンドで表示されます。



[ユーザ定義ツール]

現在定義されているユーザツールの一覧を表示します。ツールを選択して以下の操作ができます。

キー	動作
U、u	ツールを上に移動します。
D、d	ツールを下に移動します。
Delete	ツールを削除します。

[アイテム]

ツールを設定しているプロジェクトのアイテムを特定します。

[上へ]

選択しているツールを1つ上へ移動します。

[下へ]

選択しているツールを1つ下へ移動します。

[削除]

選択しているツールを削除します。

[すべて削除]

定義されているツールをすべて削除します。

第７章　ビルトインスクリプト

ここでビルトインスクリプト「MocaScript」のリファレンスすべてを記述することができないため、基本的な使い方のみを紹介しています。

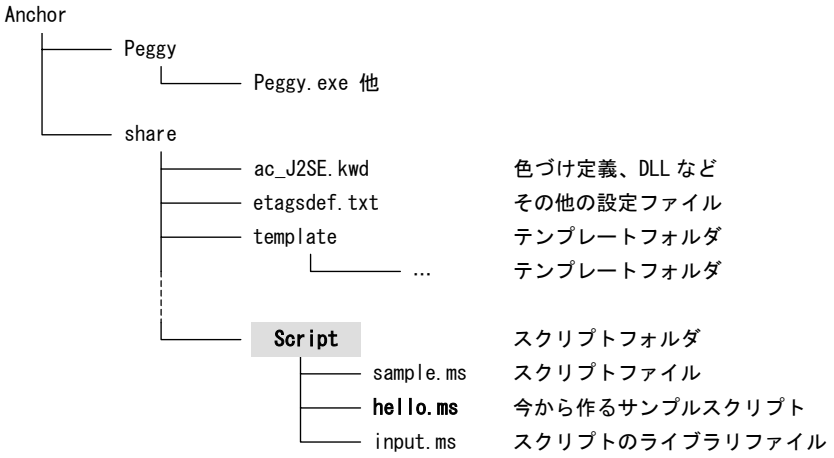
詳細リファレンスは、オンラインヘルプの『MocaScript』を参照してください。

Hello, world

定石どおりMocaScriptも、"Hello, world"から始めましょう。

スクリプトファイルの作成

スクリプトはファイルに記述し、Peggy　をインストールしたフォルダの<share¥script>に入れます。このとき拡張子は、必ず<.ms>としてください。Peggyの起動時には、このフォルダにある<*.ms>ファイルをスクリプトとして認識します。



このフォルダに、<hello.ms>という名前のファイルを作成し、その中にスクリプトとして実行したいMocaScriptの文を記述します。

```
// アクティブな編集ウインドウに Hello, world と挿入する。
insertText("Hello, world¥n");
```

スクリプトファイルの追加や削除を行ったときは、メニューの[ツール]-[スクリプト]-[再ロード]を実行してください。<share¥script>フォルダ内を調べ直して、プロジェクトウインドウのスクリプトページを更新します。これによって、先ほど作成した、<hello.ms>ファイルも表示されます。

スクリプトの実行

helloを開くとペンのアイコンの横に[実行]と表示されています。これを「スクリプトコマンド」と呼ぶことにします。スクリプトコマンドを実行するには、このペンのアイコンをダブルクリックしてください。編集ウインドウのカーソル位置に「Hello, world」という文字列が挿入されます。

【メモ】ここでエラーが発生するかもしれません。それは、文字列を挿入すべきウインドウが開かれていないためです。insertText()は、アクティブな編集ウインドウへ文字列を挿入する関数なので、ウインドウが見つからないとエラーとなります。この場合、[ファイル]-[新規]などで新しい編集ウインドウを開き、再度[実行]をダブルクリックしてください。

スクリプトは、Hello, worldサンプルのような単純なものから複雑なテキスト処理まで、さまざまな用途で使われます。そしてそれらを使う状況もいろいろです。状況に合わせて便利に使えるよう、ダブルクリック以外にも色々な方法で実行させることができます。

ドラッグ＆ドロップ実行

プロジェクトウインドウの[スクリプト]ページからスクリプトコマンドをドラッグし、編集ウインドウ上へドロップする方法です。ドロップした編集ウインドウのカーソル位置でスクリプトコマンドが実行されます。これをドラッグ＆ドロップ実行と呼びます。ドラッグ＆ドロップ実行では、アクティブでない編集ウインドウへドロップして実行させることもできます。

ピンポイント・ドロップ実行

スクリプトコマンドをドラッグ中にCtrlキーを押すと、編集ウインドウにドロップポイントを示すマーカーが表示されます。そのままドロップすると、カーソルをドロップポイントに設定してからスクリプトの実行が始まります。Hello, worldの例では、好みの挿入場所を決めてから実行できることになります。文字列を挿入するスクリプトには都合のよい実行方法と言えるでしょう。

他にも、ツールバーのボタンやショートカットから実行する方法がありますが、詳しい説明は後回しにして先へ進みましょう。

【ヒント】長くかかるスクリプトの実行を途中で止めるには、Escキーを押してください。

操作編

コマンド宣言

1つ1つの機能を別々のスクリプトファイルにしていたのでは、ファイルだらけになってしまって大変です。そこで、1つのスクリプトファイルに複数のスクリプトコマンドを登録する方法を用意しました。スクリプトファイル中にどのようなスクリプトコマンドがあるのかを、スクリプトファイルの先頭で#command文により宣言するのです。

```
// ファイル中にあるスクリプトコマンドの宣言
#command Hello    "Hello, worldを挿入"
#command Goodbye  "Goodbye, worldを挿入"
```

#command文は、Peggyに対して「このファイル中には、...のスクリプトコマンドがありますよ」と宣言しているだけで、実際に実行したときは何もしません。Hello、Goodbyeをスクリプトコマンド名(または単にコマンド名)、"Hello, worldを挿入"、"Goodbye, worldを挿入"をコマンドラベル(または単にラベル)と呼びます。コマンド名は、MocaScriptの変数名などと同じ識別子です。ラベルは、Peggyのメニューなどに表示される任意の文字列です。あまり長くならない程度で判り易いラベルを付けておきましょう。ラベルは省略できます。

Peggyのメニューから[ツール]-[スクリプト]-[再ロード]を選択すると、各スクリプトファイル中の#command宣言文を調べて、スクリプトページを更新します。実

際、<hello.ms>に上記のコマンド宣言を追加してから再ロードすると、スクリプトページに2つのラベルが表示されるはずです。

コマンドの識別

どのコマンドを選んで実行しても、スクリプトは常にファイルの先頭から実行されます。何かの目印により途中をスキップしてくれるような魔法は存在しません。そこで必要になるのは、ファイル中のどのコマンドが選ばれて実行されたのかを知る方法です。

Peggy/MocaScriptでは、最も単純な方法を採用しました。それは、「commandという名前の変数にコマンド名が文字列として代入された状態で実行が開始される」というものです。実行が開始されたら、**if**文や**switch**文でcommand変数を調べ、適切な処理に分岐するのです。まとめると、以下のようになります。

```
// ファイル中にあるスクリプトコマンドの宣言
#command Hello    "Hello, worldを挿入"
#command Goodbye  "Goodbye, worldを挿入"

// コマンドの処理
if (command == "Hello")
    insertText("Hello, world¥n");
else if (command == "Goodbye")
    insertText("Goodbye, world¥n");
else
    error("不明コマンド");
```

最後のerror(...)は、コマンド宣言だけを追加して処理部分を忘れていたとか、コマンド名をミスタイプしたなど、万が一の場合に備えての安全策です。ビープ音を鳴らしてメッセージをステータスバーに表示し、スクリプトの実行を終了します。

コマンド数が増えると、**switch**文を使った方が読みやすくなります。

```
// ファイル中にあるスクリプトコマンドの宣言
...

// コマンドで分岐（MocaScriptでは、case 文に文字列を指定できる）
switch (command) {
case "Hello":
    insertText("Hello, world¥n");
    break;

case "Goodbye":
    insertText("Goodbye, world¥n");
    break;

default:
    error("不明コマンド");
}
```

【メモ】command変数以外にもう1つARGVという名前の変数が予め定義されています。ARGVはコマンドライン版MocaScriptと同じように、ARGV[0]にスクリプトファイル自身のパス、ARGV[1]にコマンド名(command変数と同じ文字列)を要素として持つ配列です。

コマンドの分類

さらにコマンドが増えると、グループに分けて表示したくなります。それには、#command宣言のコマンド名をドット(.)で区切って続けます。

```
// ファイル中にあるスクリプトコマンドを階層分けして宣言
#command Insert           "挿入(&I)"
#command Insert.Hello     "&Hello, world"
#command Insert.Goodbye   "&Goodbye, world"

#command Delete           "削除(&D)"
#command Delete.Word      "単語(&W)"
#command Delete.Line      "行(&L)"
```

このように宣言することで、プロジェクトウインドウの[スクリプト]ページ、[ツール]-[スクリプト]メニューなどでコマンドが階層表示されます。上手に分類することで、多くのコマンドを効率よく探すことができるようになるでしょう。階層の深さは8段まで可能です。

コマンドラベルの中の"(&D)"とか、"(&W)"のような記述は、このラベルがメニューに表示されたときのキーシンボルを指定するものです。スクリプトコマンドをメニューから実行するときに、キーボードからの操作が楽になります。

この例のInsertやDeleteは、その下のスクリプトコマンドをまとめる「コマンドグループ」です。必ずしもコマンドグループを宣言する必要はありませんが、コマンドグループにより判りやすいラベルを付けることができます。

スクリプト実行開始時のcommand変数の値は、"Insert.Hello"のようになります。

```
// ファイル中にあるスクリプトコマンドの宣言
...

// 階層付きのコマンドで分岐
switch (command) {
case "Insert.Hello":
    insertText("Hello, world¥n");
    break;

case "Insert.Goodbye":
    insertText("Goodbye, world¥n");
    break;

case "Delete.Word":
    deleteWord();
    break;

case "Delete.Line":
    deleteLine();
    break;

default:
    error("不明コマンド");
}
```

【メモ】コマンドの追加や削除を行ったときは、[ツール]-[スクリプト]-[再ロード]を実行し、プロジェクトウインドウの[スクリプト]ページを更新してください。

【メモ】プロジェクトウインドウの[スクリプト]ページをマウスの右ボタンでクリックし、[コマンド名を表示]を選択すると、ラベルとコマンド名の両方を確認できます。

スクリプトファイルを開く

スクリプトファイルを開くには、プロジェクトウインドウの[スクリプト]ページの編集したいスクリプトファイル、またはコマンドを右マウスボタンでクリックし、ポップアップメニューから[開く]を選択してください。スクリプトコマンドから[開く]を選択した場合は、そのコマンドの#command宣言を自動的に選択します。

【メモ】スクリプトファイルのアイコンをドラッグ＆ドロップすることでもオープンできます。

コーディング

MocaScriptのプログラムを記述するときは、Peggyの言語モードをMocaScriptにしておきましょう。それには、[オプション設定]ダイアログボックスの[拡張子]ページの[拡張子マップ]で、拡張子"ms"や"msl"をMocaScriptに対応付けます。言語モードMocaScriptが見当たらないときは、Anchor SystemsのホームページからCRECの色づけ定義ファイル<ac_moca.kwd>をダウンロードし、<share>フォルダに入れてください。

【メモ】言語モードをMocaScriptにすることで、スクリプトファイルが色分け表示され、キーワードチップやキーワード補完機能も働くので効率よくプログラミングできるようになります。

コンパイル

MocaScriptは、プログラムをバイトコード形式にコンパイルしてから実行します。コンパイルはスクリプトを実行するとき自動的に行われるため、意識する必要はありません。またコンパイルも非常に短時間で終わるため、初回の実行でもあまり気になることはないでしょう。コンパイルに成功すると、プログラムが修正されない限り同じバイトコードを繰り返し使います。

実行せずコンパイルだけしたい場合は、プロジェクトウインドウの[スクリプト]ページのポップアップメニューから[コンパイル]を選択してください。

文法エラーなどがあるとコンパイルを中断し、アウトプットウインドウの[スクリプト]ページにエラーメッセージを出力します。このエラーメッセージはタグジャンプ可能な形式なので、F4キーを押すことでエラー原因となった行へジャンプできます。

【メモ】Peggyはコンパイルしたバイトコードとともに、その時のスクリプトファイルのタイムスタンプを持っています。スクリプトコマンドを実行する場合、このタイムスタンプを調べて変化がなければ、バイトコードをそのまま使い、スクリプトファイルが更新されていれば再コンパイルします。

【注意】自動再コンパイルのためのタイムスタンプの比較は、スクリプトファイルに対してのみです。そのファイルからインクルードしているファイルはチェックされません。そのためインクルートされているファイルを編集しても、自動的には再コンパイルされません。これらのファイルを修正した場合は、プロジェクトウインドウの[スクリプト]ページのポップアップメニューから[コンパイル]を実行してください。

デバッグ

Peggyに搭載されたMocaScriptには、デバッグのための簡単なトレース実行機能を用意しました。トレース実行を行うには、プロジェクトウインドウの[スクリプト]ページの実行したいコマンドを右マウスボタンでクリックし、ポップアップメニューから[トレース実行]を選択してください。

どの命令をどの順番で実行したかの情報がアウトプットウインドウの[スクリプト]ページへ出力されます。これもタグジャンプ可能な形式なので、F4キーを連続して押すことにより、順番にたどることができます。

しかしトレース実行機能は実行した命令の足跡をたどるだけで、変数の内容を見たり変更することはできません。残念ながら現バージョンでは、途中で止めたり変数の内容を確認する機能は備わっていません。

変数の中身を確認したいときは、writeln()関数などを使って変数値をアウトプットウインドウへ出力してください。これら関数の出力も[スクリプト]ページへ送られるので、トレース結果に挟まれる形で実行した行とその時の値を確認することができます。

スクリプトの実行

スクリプトの実行方法を紹介します。

場所	実行方法	最大数
[スクリプト]ページ	コマンドをダブルクリックする。 コマンドを選択してEnterキーを押す。 コマンドをドラッグして編集ウインドウ上へドロップする。 コマンドをドラッグして編集ウインドウ上へCtrlキーを押しながらドロップする。 ポップアップメニューから[実行]を選択する。 ポップアップメニューから[トレース実行]を選択する。	無制限
編集ウインドウ	ポップアップメニューから[スクリプト]-....を選択する。 MocaScriptの実行文を選択して、ポップアップメニューから[スクリプト]-[選択範囲を実行]を選択する。	8階層500コマンド
メニュー	[ツール]-[スクリプト]の下にあるスクリプトコマンドを選択する。	8階層500コマンド
ツールバーボタン	ツールバーに配置したボタンをクリックして実行する。	72個
ショートカット	ショートカットキーによりスクリプトコマンドを実行する。	72個
Peggyの起動	<share¥script>フォルダに<startup.ms>という名前のスクリプトファイルがあると、Peggyの起動時に自動実行される。	

メニューと編集ウインドウのポップアップメニューは同じものです。プロジェクトウインドウの[スクリプト]ページに表示されているコマンド階層がメニュー形式で選択できるようになっています。

[選択範囲を実行]コマンドは、編集中のウインドウの選択範囲をMocaScriptのプログラムと見なして実行する機能です。短いプログラムを試してみたい場合に便利です。

ツールバーのボタンとショートカットは、以下のようにしてPeggyをカスタマイズする必要があります。

ボタンの割り当て

スクリプトを実行させるためのボタンは、72個用意してあります。メニューから[ツール]-[カスタマイズ]で表示される[カスタマイズ]ダイアログボックスの[コマンド]ページにある[スクリプト]を選択してください。[ボタン]のところに72個のスクリプト実行用のボタンが表示されます。このボタンをドラッグして、好みのツールバー上へドロップしてください。これでスクリプトのボタンがツールバー上へ配置できました。

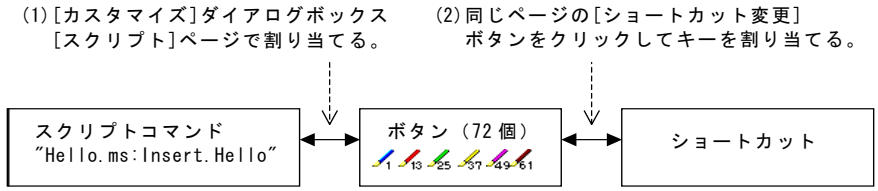
次に同じダイアログボックスの[スクリプト]ページで、スクリプトコマンドとボタンの結び付けを行います。[スクリプト]ページにはボタンの一覧があります。スクリプトコマンドを割り当てたいボタンの[コマンド]の欄をクリックしてください。現在利用可能なスクリプトコマンドのリストが現れます。リストから割り当てたいコマンドを選択してEnterキーを押すと、コマンドの割り当てが完了します。

【注意】ボタンとスクリプトコマンドの結びつきは、スクリプトのファイル名とコマンド名に頼っています。具体的には、「ボタン1に割り当てたコマンドは、"<ファイル名>:コマンド名"である」という形式で記憶しています。Hello, worldの場合であれば、"Hello.ms:Insert.Hello"となります。そのためスクリプトファイルを編集してコマンド名が変わった、またはスクリプトファイルの名前を変えたような場合、ボタンとスクリプトコマンドのリンクが切れてしまいます。ボタンに割り当てたコマンドの名前を変更した場合は、再度ボタンの割り当てを行ってください。

【メモ】ボタンの割り当てを保存したファイルは、テキストファイルです。保存したファイルを開くと、ボタンとコマンドが上記の形式でマップされている様子を見ることができます。

ショートカットの割り当て

スクリプトコマンドはショートカットからも実行できますが、ショートカットを直接スクリプトコマンドに割り当てるのではありません。スクリプトコマンドは、スクリプトファイルを編集したり、ファイルの追加/削除で知らない間に変わってしまうためです。そこで、スクリプトのボタンに対してショートカットを割り当てることにしました。



スクリプト用のツールバーボタンは、全部で72個あります。まず、メニューから[ツール]-[カスタマイズ]を選択して[カスタマイズ]ダイアログボックスを開き、[スクリプト]ページをクリックしてください。そこに72個のスクリプトボタンの一覧が表示されています。

次に、リストからボタンを選んでスクリプトコマンドを割り当てます。最後に[ショートカット変更]ボタンをクリックして、ショートカットキーを割り当ててください。以後、ショートカットキーを使ってスクリプトを実行できるようになります。ショートカットは、[ツール]-[ショートカット]ダイアログボックスを使っても変更できます。

さらに、ボタンのイメージを変更することも可能です。ボタンイメージを変更するには、[ボタン変更]ボタンをクリックし、好みのイメージを選択してください。

第 8 章　バージョン管理

この章では、CVS、RCS、Subversion、VSSを用いたバージョン管理について解説します。

バージョン管理

「バージョン管理システム」と言うと何やら難しいもののように聞こえますが、実は簡単に使えてとても便利なソフトウェアツールです。バージョン管理システムは「ファイルの変更履歴を記録するソフトウェア」です。メモ書きでも巨大なソフトウェアのソースファイルでも同じです。繰り返し編集を重ね変化して行くファイルの内容を記録し、後から特定のバージョンを取り出したり、いつ、誰が、何の目的で、どう修正を加えたかを調べることが可能になります。

Peggyには、バージョン管理ソフトウェアとして全世界で広く使われているRCSとCVS(※1・※2)、Subversion(※2)、それにVSS(※2)のユーザインターフェイスを組み込みました。それにより強力なバージョン管理機能を簡単に利用することが可能になりました。テキストを編集し、そのままボタン1つで編集記録を残すことができます。ソフトウェア開発に限らず、作りかけのホームページや完成後のメインテナンス、メモ書き、メールのログ、書きかけの原稿など何でも記録してください。

もう一つ精神衛生上の利点もあります。いざとなればいつでも過去の履歴から取り出せるので、テキストの不要な部分を安心して削除できるのです。ひょっとしたら...と思って消さずに我慢する必要はありません。バージョン管理は簡単に始められます。

【※1】バージョン管理システムRCSとCVSは、GNU GENERAL PUBLIC LICENSEに基づいて配布されているソフトウェアです。

【※2】RCS、CVS、Subversion、VSS(Microsoft Visual SourceSafe)のユーザインターフェイスをサポートしているのは、Peggy Proです。Peggyは、RCSのみサポートします。

RCS、CVS、Subversion

RCS(Revision Control System)とCVS(Concurrent Versions System)は、親子関係にあるバージョン管理システムで、共に長い歴史と実績があります。最初にRCSが開発され、それをベースとしてCVSが開発されました。

CVSとSubversionにも深い関連があります。CVS経験に基づき、CVSの欠点を克服すべく新しく設計されたバージョン管理システムがSubversionです。

RCS、CVS、Subversionは多くの共通点を持っていますが、用途や性格はかなり異なります。以下の表を参考にして使い分けるとよいでしょう。

RCS 特徴

長所	シンプルでどこでもすぐに使える。
短所	管理の単位が「1つのファイル」なので、たくさんのファイルを扱うのは面倒。
用途	メモ書き、メールのログ、ちょっとしたバッチやスクリプトファイルなど、個人で使っている単独のテキストファイルの履歴管理に適している。

CVS の特徴

長所	・複数のフォルダに分かれた数多くのファイルをまとめて効率よく扱うことができる。 ・グループでの作業を前提に設計されているので、ファイルの共有と共同作業をサポートしている。 ・複数のユーザが同時に同じファイルを編集することも可能。
短所	・高機能な分だけ使い方が多少複雑。 ・最初にある程度理解し、設定しなければならないことがある。
用途	・プログラムのソースファイル群、ホームページのHTMLファイル群をまとめて管理する場合。 ・一般的に数個以上の関連ファイルがある場合は、CVSを使ってまとめた方が便利に作業できる。 ・グループでの編集作業を支援するので、バージョン管理のデータベース(リポジトリ)をサーバーに置き、ネットワークからグループ全員でアクセスし、変更記録を残したい場合など。

Subversion の特徴

長所	CVSに加えて以下の利点がある。 ・フォルダの履歴管理が可能。 ・ファイルやフォルダのリネームや移動も履歴として追跡できる。 ・バイナリファイルの履歴も効率的に管理できる。 ・ネットワーク負荷を軽減する配慮がある。 ・HTTPサーバー経由でもアクセスすることもできる。 ・同時にコミットしたファイルやフォルダを特定できる。
短所	(CVSと同じ)
用途	(CVSと同じ)

以上をまとめると、個人の単独ファイルはRCSで、それ以上はSubversionまたはCVSをお勧めします。個人で利用する場合でも、例えばホームページのように数個以上のファイルから構成されているなら、SubversionやCVSの方が便利になります。Peggy ProからGUIで操作できるので、基本事項さえ理解すれば、SubversionやCVSも簡単に利用できます。

【ヒント】RCSとCVSは同じフォーマットのファイルを使って変更履歴を管理するので、最初はシンプルなRCSを使っておき、ファイルが増えてきたらCVSへ移すことも可能です。Subversionのファイルフォーマットは全く異なるので、RCS/CVSと履歴ファイルレベルでの互換性はありません。

RCS によるバージョン管理

ここでは、初めてRCSを使ってバージョン管理を体験するための手順を示しています。Peggyを使った最も簡単なバージョン管理の入門コーナーです。

(1) RCSへチェックイン

(2) 編集して再びRCSへチェックイン

(3) ログを見る

(4) 古いリビジョンを見る

(5) 変更箇所を確認する

【ヒント】Peggyに組み込まれているRCSのユーザインターフェイスは、個人ユーザが少数のファイルの履歴を管理する場合を想定しています。グループでの作業や、多数のファイルを同時に管理したい場合は、Peggy ProとCVSまたはSubversionの組み合わせを推奨します。

RCS へチェックイン

「習うより慣れろ」です。何はともあれ、一度使ってみましょう。

1.	まず、Peggyで適当なテキストファイルを開いてください。練習なので、内容を書き換えてもかまわないファイルを選ぶとよいでしょう【注意】参照
2.	次に、[ ]ボタンをクリックしてください。すると、[RCSチェックイン]ダイアログボックスが現れます。
3.	[ログメッセージ]のところへ、"チェックインの練習"などとタイプして、
4.	[チェックイン]ボタンをクリックします。

これだけで、現時点でのテキストファイルの状態が記録されました。この操作を「チェックイン」と呼びます。ログメッセージは、後で見たときに何のための変更かを思い出す「手がかりメモ」と思ってください。

テキストファイルの状態は、どこに記録されるのでしょうか?チェックインすると、そのテキストファイルがあるフォルダにRCSというサブフォルダを作成し、その中に「RCSファイル」と呼ばれる特別なフォーマットのファイルを作成して記録します。元のファイル1つに付き、1つのRCSファイルが作成されます。RCSファイルの名前は、元のファイル名の末尾に,vを付けたものとなります。

元のファイル	C:¥work¥sample.txt
RCSファイル	C:¥work¥RCS¥sample.txt,v

【注意】RCSは外国生まれのソフトウェアです。そのため日本語ファイル名で混乱することがあります。バージョン管理をしたいファイルに日本語のファイル名を付けるのは避けた方がよいでしょう。ファイルの内容自体は、どんな言語、どんな漢字コード(Unicodeを除く)でも問題ありません。Unicodeのテキストファイルは、UTF-8(N)として保存してからチェックインしてください。

【ヒント】RCSファイルもテキストファイルなので、Peggyで開いて中を見ることができます。しかし、RCSが読み書きするので、内容を変更してはいけません。安全のため、RCSファイルは書き込み禁止になっています。

【ヒント】過去の履歴もまとめてすべて消したいときは、RCSファイルを削除してください。

【注意】チェックイン完了後にファイルからテキストを読み直すため、アンドゥ情報がクリアされます。そのため、[編集]-[元に戻す]コマンドでチェックイン以前の状態へ戻すことができなくなります。

編集して再び RCS へチェックイン

ファイルを開いたまま数行付け加えるなどして内容を編集し、再びチェックインしてみましょう。ログメッセージは変えた方がいいかもしれませんが、その他の操作は前回とまったく同じです。

1.	テキストファイル編集して数行追加する。
2.	[ ]ボタンをクリックして、再び[RCSチェックイン]ダイアログボックスを呼び出す。
3.	適当なログインメッセージをタイプして、
4.	[チェックイン]ボタンをクリックする。

これでテキストファイルの新しい状態が記録されました。RCSは、前回から変化した行だけを記憶します。そのため何度もチェックインしても、消費するディスクスペースは僅かです。

RCS ログを見る

さて、ここまでのチェックインからテキストファイルの変更履歴を表示させてみましょう。チェックイン履歴のことをRCSでは「リビジョンログ」、または単に「ログ」と呼びます。操作はとても簡単です。

1.	リビジョンログを見たいファイルにカーソルがあることを確認します。
2.	[ ]ボタンをクリックします

これだけで[RCSリビジョンログ]ダイアログボックスが表示されます。[ ]がファイルを表していて、その下にある2つの[ ]がチェックイン記録です。

これをリビジョンと呼びます。リビジョンには、それを識別するためにリビジョン番号が自動的に付けられます。リビジョン番号は、ドットで区切られた偶数個の数字で、1.1から始まり、チェックインするたびに最後の番号が+1されます。例えば、1.1、1.2、1.3、... となります。その他に、誰が(ユーザ)、いつ(日時)、何のために(ログメッセージ)、何行変更したか(増/減)も知ることができます。

【メモ】リビジョンは、「改訂」などと訳します。「バージョン」という言葉は氾濫していていろいろな意味で使われているため、RCS、CVSでは1つ1つのチェックインによる改訂をあえて「リビジョン」と呼んでいます。

【ヒント】ユーザ名は、RCSインストールの際に設定した環境変数LOGNAMEが元になっています。

RCS 古いリビジョンを見る

[RCSリビジョンログ]ダイアログボックスを開いたついでに、古いリビジョン1.1を表示してみましょう。

1.	[1.1]をクリックしてリビジョン1.1を選択します。
2.	[表示]ボタンをクリックします。

これで、過去のテキストが表示されたはずです。ウインドウのタイトルバーには、[<filename> - Rev 1.1]と表示され、テキストが過去のリビジョンであることを示しています。

【ヒント】[表示]ボタンは、過去のリビジョンをテンポラリファイルに取り出すため、オリジナルのファイルを壊す心配はありません。

【ヒント】過去のリビジョンを表示している編集ウインドウは、編集禁止です(過去を変えることはできません)。過去のリビジョンを元に編集したいときは、[ファイル]-[名前を付けて保存]コマンドにより別ファイルとして保存してから行ってください。

RCS 変更箇所を確認する

いま編集しているファイル(これを作業コピーと呼びます)と昔のリビジョン1.1を比較してみましょう。

1.	[RCSリビジョンログ]ダイアログボックスの[1.1]をクリックします。
2.	ダイアログボックスの[比較]ボタンをクリックします。

ウインドウが2つ開いて、作業コピーとリビジョン1.1の相違箇所を表示します。他にも変更点があるときは、F4キー、Shift+F4キーでそれぞれ前後の変更行へタグジャンプできます。変更箇所を確認し終わったら、ウインドウを閉じてください。

次に、リビジョン1.1と1.2で何が変わったのかを調べてみましょう。リビジョン間の比較です。

1.	[RCSリビジョンログ]ダイアログボックスの[1.1]をクリックします。
2.	Ctrlキーを押しながら、[1.2]をクリックします。この時点で、1.1と1.2の両方が反転表示されているはずです。
3.	最後に[比較]ボタンをクリックしてください。

編集ウインドウが2つ開いて、リビジョン1.1と1.2の相違箇所を表示します。隣同士でなくても任意のリビジョン間で比較できます。

基本的なRCSの使い方はこれだけです。Peggyを使ってテキストファイルを編集しているとき、一区切りついたところでチェックインするだけです。あとは、過去の変更履歴が必要になったとき、[RCSリビジョンログ]ダイアログボックスを開けば、すべてのリビジョンにアクセスできます。

RCS コマンド

ボタン	メニュー操作	キー操作	機能
	[ツール]-[RCS]-[チェックイン]		[RCSチェックイン]ダイアログボックスを表示して、作業コピーの変更をRCSファイルへチェックインします。ci.exe -l、またはci.exe -uに相当します。
	[ツール]-[RCS]-[チェックアウト]		RCSファイルから編集可能な作業ファイルをチェックアウトします。co.exe -lに相当します。
	[ツール]-[RCS]-[取得]		RCSファイルから書き込み禁止状態でファイルを取得します。取り出したファイルを編集することはできません。co.exe -uに相当します。
	[ツール]-[RCS]-[チェックアウト取消]		チェックアウトを取り消します。rcs.exe -uに相当します。

※	[ツール]-[リビジョンログ]※		[RCSリビジョンログ]ダイアログボックスを表示します。任意リビジョンの表示や比較が行えます
※	[ツール]-[リビジョン比較]※		作業コピーとRCSファイル中の最新リビジョンとの比較します。
§	[ツール]-[RCS]-[ログ]§		[RCSリビジョンログ]ダイアログボックスを表示します。任意リビジョンの表示や比較が行えます。
§	[ツール]-[RCS]-[比較]§		作業コピーとRCSファイル中の最新リビジョンとの比較します。

※Peggy Proのみ・§Peggyのみ

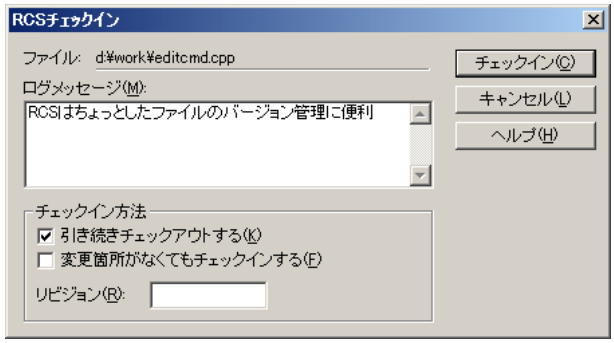
【注意】※印の付いたボタンとコマンドは、Peggy Proのみサポートしています。Peggyは代りに、§印の付いたコマンドをメニューから選択できます。

【注意】[ツール]-[リビジョンログ]と[ツール]-[リビジョン比較]は、RCS、CVS、Subversion共通コマンドです。編集しているテキストがどのバージョン管理システムに登録されているかを自動的に判断します。

【注意】RCSによるバージョン管理を行うには、パスの通ったフォルダに4つの実行ファイル(ci.exe、co.exe、rlog.exe、rcs.exe)が必要です。少なくともci.exeが見つからないと、RCS関連のコマンドが有効になりません。

[RCS チェックイン] ダイアログボックス

RCSを使ってファイルをチェックインします。このダイアログボックスは、[ツール]-[RCS]-[チェックイン]コマンドを実行すると表示されます。



[ファイル]

チェックインするファイルパスを表示しています。

[ログメッセージ]

チェックインに伴うメッセージを書きこみます。複数行でも、また省略することもできます。

[引き続きチェックアウトする]

チェックイン後もファイルを編集できる状態にします。チェックを外すと、チェックイン後はファイルが書き込み禁止となり、編集できなくなります。

【ヒント】チェックインして書き込み禁止になったファイルは、チェックアウトすると再び編集できる状態になります。

[変更箇所がなくてもチェックインする]

ファイルを変更していなくても強制的にチェックインし、リビジョン番号を進めます。このオプションを指定していない場合は、ファイルに変更があったときだけチェックインします。

[リビジョン]

1.30から2.00へなど、特別にバージョン番号を指定してチェックインします。無指定のときは、自動的に新しいリビジョン番号が付けられます。

[チェックイン]

チェックインを実行します。

CVS によるバージョン管理

まず始めに、Peggy ProがサポートするCVSのユーザインターフェイスは、基本部分のみであることを理解してください。CVSはもともとグループで共同作業を行うことを前提にデザインされているので、日頃ほとんど使わない「管理」に関係した機能が多くあります。これらはCVSデータベース(リポジトリと呼ぶ)の管理者の仕事なので、エディタから操作するような種類のものではありません。

管理者以外、つまり普通にCVSを使うユーザのほとんどは、ソースプログラムやテキストを編集しては新しいリビジョンとして登録することの繰り返しです。Peggy ProのCVSサポート機能は、この部分を簡単・便利にすることを目的としています。Peggy ProがサポートしているCVSのバージョンは、Windows版1.10以降です。

CVSを使ったバージョン管理の仕組みに関しては、<share¥document¥peggycvs.pdf>を参照してください。

【注意】CVS関連のコマンドが利用できるのは、Peggy Proだけです。

【注意】CVSは外国生まれのソフトウェアです。そのため日本語ファイル名・フォルダ名で混乱することがあります。特に2バイト目が円記号(¥、文字コード5C)の漢字を含んでいると、ファイルやフォルダを登録することができません。バージョン管理をしたいファイルに日本語のファイル名を付けるのは避けた方がよいでしょう。特に、UnixとWindowsが混在した環境での日本語ファイル名は、トラブルの原因になります。ファイルの内容自体は、どんな言語でも問題ありません。Peggyは、ほとんどすべての漢字コードを扱うことができます。

CVS コマンド

ボタン	メニュー操作	機能
	[ツール]-[CVS]-[アップデート]	[CVSアップデート]ダイアログボックスを開いて、現在編集しているファイルをアップデートします。 プロジェクトウインドウのファイルやグループアイテムを選択しているときは、それに対応したファイルやフォルダをアップデートします。

	[ツール]-[CVS]-[コミット]	[CVSコミット]ダイアログボックスを開いて、現在編集しているファイルをコミットします。 プロジェクトウインドウのファイルやグループアイテムを選択しているときは、それに対応したファイルやフォルダをコミットします。
	[ツール]-[CVS]-[ステータス]	[CVSステータス]ダイアログボックスを開いて、同じフォルダにある他のファイルとともに、現在編集しているファイルのステータスを表示します。
	[ツール]-[リビジョンログ]	[CVSリビジョンログ]ダイアログボックスを開いて、編集しているファイルのリビジョンログを表示します。**【メモ】参照**
	[ツール]-[リビジョン比較]	編集しているファイルと、その元になったリポジトリのBASEリビジョンを比較します。 **【メモ】参照**
	[ツール]-[CVS]-[ログ]	[CVSリビジョンログ]ダイアログボックスを開いて、編集しているファイルのリビジョンログを表示します。
	[ツール]-[CVS]-[比較]	編集しているファイルと、その元になったリポジトリのBASEリビジョンを比較します。

【メモ】[ツール]-[リビジョンログ]と[ツール]-[リビジョン比較]は、RCS、CVS、Subversionのどれでも使えるコマンドです。これらは、指定されたファイルがどのシステムでバージョン管理されているかを調べ、必要な処理を実行します。

厳密にCVSのログ、CVSのリビジョン比較を実行したい場合は、[ツール]-[CVS]-[CVSログ]や[ツール]-[CVS]-[比較]コマンドを使ってください。

【注意】コミット完了後にファイルからテキストを読み直すため、アンドゥ情報がクリアされます。そのため、[編集]-[元に戻す]コマンドでチェックイン以前の状態へ戻すことができなくなります。アップデートによりファイルが更新された場合も、同様にファイルからテキストを読み直します。

【注意】CVS関連のコマンドが利用できるのは、Peggy Proだけです。さらに、Peggy Proを起動したときcvs.exeコマンドがパスの通ったフォルダに見つからないと、これらのコマンドは有効になりません。cvs.exeをパスの通ったフォルダにコピーしてから、Peggy Proを起動しなおしてください。

RCS/CVS/Subversion 共通コマンド

これらのコマンドは、ファイルがRCS、CVS、Subversionのうちどれによってバージョン管理されているかを自動的に判断して動作します。

ボタン	メニュー操作	機能

	[ツール]-[リビジョンログ]	選択しているファイルにより、Subversion/CVS/RCSの[リビジョンログ]ダイアログボックスを表示します。
	[ツール]-[リビジョン比較]	選択しているファイルにより、Subversion/CVS/RCSの作業ファイルと、その元になっているリビジョン（**【メモ】参照**）を比較します。

【注意】これらのコマンドは、Peggy Proのみサポートしています。

【注意】1つのファイルを2つ以上のシステムに登録している場合は、[オプション設定]ダイアログボックス[バージョン管理]ページの[優先順位]により、Subversion、CVS、RCS の優先順位を指定できます。

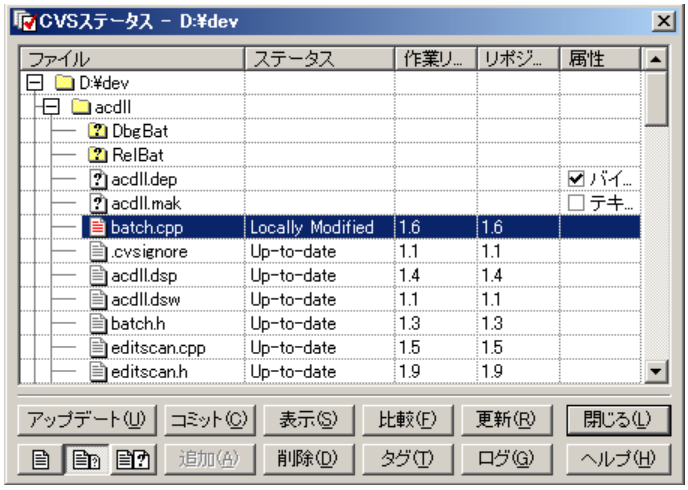
【メモ】元になっているリビジョンとは、Subversion/CVSであれば作業コピーのBASEリビジョン、RCSならRCSファイルの最新リビジョンを意味します。

[CVS ステータス] ダイアログボックス

[CVSステータス]ダイアログボックスは、CVS作業ファイルの状態を表示します。このダイアログボックスから、[アップデート]、[コミット]など、CVSの主要な操作をすべて行うことが可能です。

このダイアログボックスを表示するには、メニューから[ツール]-[CVS]-[ステータス]コマンドを実行します。そのとき編集しているファイルの状態を、同じフォルダにある他のファイルと共に表示します。

プロジェクトウインドウのファイルアイテムを選択してから[ツール]-[CVS]-[ステータス]コマンドを実行することで、そのファイルの状態を表示させることも可能です。このダイアログボックスは、リサイズできます。



[ファイル] 欄

CVS作業フォルダの階層と、その中に登録されているファイルの名前を表示します。ヘッダのこの部分をクリックすると、ファイル名順に並べ替えて表示できます。

[ステータス] 欄

[ファイル]欄のアイコンと共に作業ファイルの状態を表示します。ヘッダのこの部分をクリックすると、ステータスに従って並べ替えることができます。

【ヒント】ステータスでソートすると、同じ状態のファイルが近くに並ぶので、作業コピーの状態を把握しやすくなります。

アイコン	ステータス	意味
	Up-to-date	作業ファイルが最新の状態であることを示しています。
	Locally Modified	作業ファイルに変更を加えていることを示しています。
	Needs Checkout Needs Patch	リポジトリに作業コピーより新しいリビジョンがコミットされていることを示しています。通常は、[アップデート]により、作業コピーをアップデートすることが必要です。
	Needs Merge	リポジトリには新しいバージョンがコミットされ、同時に作業コピーも編集している状態を表しています。[アップデート]により、両方の変更箇所をマージ(併合)する必要があります。
	File had conflicts on merge	マージの際に変更部分が衝突してしまったことを示しています。衝突箇所を手作業で編集してからコミットします。
	Locally Added	追加したけれど、まだコミットしていないファイルであることを示しています。
	Locally Removed	削除したけれど、まだコミットしていないファイルであることを示しています。
		CVSの管理下にあるフォルダを表しています。
		ファイルがCVSの管理下のものではないことを示しています。
		CVSの管理下ではないフォルダを表しています。[CVSステータス]ダイアログボックスでは、管理外のフォルダを開いてその下のファイルを表示することはできません

[作業リビジョン] 欄

作業ファイルの元になっているリビジョンを表示しています。

[リポジトリリビジョン] 欄

現在のリポジトリのリビジョンを表示しています。[作業リビジョン]よりも進んでいれば、誰かが新しいリビジョンをコミットしたことを意味します。

[属性] 欄

リビジョン、日時、キーワード展開に関するスティッキーフラグを表示します。-kbは、バイナリファイルであることを表しています。

CVSに未登録のファイル[ ]を表示しているときは、バイナリかテキストファイルを選択するチェックボックス[☑バイナリ]、[☐テキスト]が表示されます。

[アップデート]

[CVSアップデート]ダイアログボックスを開いて、選択している作業ファイルやフォルダを、リポジトリの状態に従ってアップデートします。

[コミット]

[CVSコミット]ダイアログボックスを開いて、選択している作業ファイルやフォルダに行った変更をリポジトリへコミットします。

【注意】コミット完了後にファイルからテキストを読み直すため、コミットしたファイルのアンドゥ情報がクリアされます。そのため、[編集]-[元に戻す]コマンドでコミット以前の状態へ戻すことができなくなります。

[表示]

選択している作業ファイルを開きます。

[比較]

選択しているのがファイルかフォルダかで動作が異なります。ファイルを選択している場合は、作業コピーと対応するリポジトリのHEADリビジョンを比較します。違いがあれば2つのエディタウインドウを開き、最初の相違行へジャンプします。以後、F4、Shift+F4キーで前後の相違箇所へジャンプ可能です。

フォルダを選択しているときは、そのフォルダとサブフォルダ内の作業コピーをリポジトリのHEADリビジョンに対しまとめて比較します。結果は、アウトプットウインドウの[バージョン管理]ページに出力されます。

【ヒント】フォルダの比較は、cvs.exe -q -n update -R -d -r HEADを実行します。

【ヒント】[リビジョン比較]、[CVSリビジョン比較]コマンドは、作業コピーの元になっているBASEリビジョンと比較ですが、[CVSステータス]ダイアログボックスの[比較]ボタンは、HEADリビジョンと比較します。以下に、BASEとHEADの違いをまとめました。

BASEリビジョンとの比較	作業コピーに加えた変更だけを表示します。
HEADリビジョンとの比較	リポジトリに新しくコミットされた変更も含んで、作業ファイルと比較します。

[更新]

現在選択しているフォルダ内のステータスを読み直して、最新の状態を表示します。

【注意】ステータスを更新するのは、選択しているファイルのあるフォルダ内だけです。他の部分は更新されないので注意してください。

[ ]、[ ]、[ ]

CVSでバージョン管理を行っていないファイルやフォルダの表示方法を切り替えます。[追加]ボタンにより新しいファイルをCVSへ追加する場合は、[ ]や[ ]をクリックし、CVSに登録されていないファイルやフォルダを表示させてください。

ボタン	表示内容
	CVSに登録されているファイルとフォルダのみを表示します。

	CVSに登録してあるファイルとフォルダに加え、.cvsignoreに指定されている「CVSが無視するファイル名パターン」と、「CVSがデフォルトで無視するパターン（**【メモ】参照**)」のどちらとも一致しなかったファイルやフォルダも表示します。
	RCS、CVS、.svnを除くすべてのファイルとフォルダを表示します。

【注意】.cvsignoreは、そのフォルダにあるものだけを調べます。リポジトリにあるcvsignoreファイルに書かれているファイル名パターンは反映されません。

【メモ】[ ]が選択されているときに表示しないデフォルトのファイルパターンは、SCCS、RCS、CVS、CVS.adm、RCSLOG、cvslog.*、TAGS、*~、#*、.#*、,*、_$*、*$、*.old、*.bak、*.orig、*.rej、.del-*、*.a、*.olb、*.o、*.obj、*.so、*.exe、*.Z、*.elc、*.ln、core、.make.state、.nse_depinfo、Thumbs.db、mssccprj.scc、vssver.scc、*.mine、*.left、*.rightです。

[追加]

[CVSファイル追加]ダイアログボックスを表示して､CVSに新しくファイルやフォルダを追加します。追加対象となるファイルやフォルダは、[ ]、[ ]ボタンをクリックすると表示されます。まだCVSに登録されていない[ ]や[ ]を選択してから、[追加]ボタンを押してください。

ファイルを追加するときは、Peggyが予めファイルの内容を調べてバイナリかテキストかを推測し、[属性]欄にチェック[☑バイナリ]、[☐テキスト]を付けています。もしも推測が誤っていたら、[追加]ボタンを押す前にチェックをクリックし、修正してください。

【注意】Unicodeのテキストを登録する場合は、UTF-8またはUTF-8N形式で保存してください。UTF-16ファイルでは改行コードCR-LFもそれぞれ16ビットとなるため、リポジトリと作業ファイル間での行末変換に問題が生じます。また、リビジョン間の比較を行うときも、diffコマンドが正しく動作しません。

【ヒント】ファイルを追加しても、コミットするまでは正式に登録が完了したことになりません。コミット前であれば、[削除]で追加を取り消すことができます。

【ヒント】CVSは、テキストファイルの行末をCR-LFからLFに変換してリポジトリへ格納します。これは、Unixを含んだすべてのプラットフォームで共通のファイル形式を保つためです。しかしバイナリファイルまで勝手に変換して欲しくありません。これを避けるには、ファイルの登録時にバイナリファイルであることを指定する必要があります。バイナリとして登録すると、[属性]の欄に-kbと表示されます。

[削除]

[CVSファイル削除]ダイアログボックスを表示して、CVSからファイルを削除します。削除したい1つまたは複数のファイルを選択してから、[削除]ボタンをクリックしてください。

[タグ]

[CVSタグ設定]ダイアログボックスを表示して、選択しているファイルやフォルダ内のファイルにタグを設定します。ファイルを選択した場合はそのファイルに、フォルダを選択している場合はフォルダ内のファイルすべてが対象となります。

【メモ】[オプション設定]ダイアログボックス[バージョン管理]ページの[[CVSステータス]ダイアログボックスの[タグ]ボタンを有効にする]をOFFにすると、[タグ]ボタンが無効になります。これは誤って[タグ]ボタンを押してしまわないための安全策です。

[ログ]

選択しているファイルの変更履歴を表示する、[CVSリビジョンログ]ダイアログボックスを表示します。このボタンをクリックすると、[CVSステータス]ダイアログボックスは閉じられます。

[CVS リビジョンログ]・[RCS リビジョンログ] ダイアログボックス

CVS、RCSで管理しているファイルの履歴を表示するダイアログボックスです。このダイアログボックスから過去の任意リビジョンの表示や、作業ファイルとの比較、過去のリビジョン同士の比較が行えます。

このダイアログボックスを表示するには、メニューから[ツール]-[リビジョンログ]コマンドを実行してください。そのとき編集しているファイルのログを表示します(下図)。

プロジェクトウインドウのファイルアイテムを選択してから[ツール]-[リビジョンログ]コマンドを実行することで、そのファイルのログを表示させることも可能です。

[リビジョン]欄

ファイル名と登録されているリビジョンをツリー形式で表示します。リビジョンにタグが設定されているときは、リビジョン番号に続けてタグも表示します。

リビジョンの表示順は、デフォルトで新しいものが上になっていますが、[CVSログフィルタ]ダイアログボックスのオプションにより、古いものから表示させることも可能です。

アイコン	意味
	作業ファイルを表しています。
	1つのリビジョンを表しています。リビジョン番号と、(あれば)タグが続けて表示されます。
	作業ファイルのベースとなったリビジョンであることを表しています(CVSのみ)。
	リビジョンがここで枝分かれ(ブランチ)していることを表しています。枝分かれした先のリビジョンは、このアイテムの子として表示します。

[状態] 欄

各リビジョンの状態を表示します。

[ユーザ] 欄

そのリビジョンをコミットした(CVS)、またはチェックインした(RCS)ユーザのログイン名を表示します。

[日付] 欄

そのリビジョンをコミットした日時を表示します。

【注意】ここに表示される日時は、UTC(Universal Coordinated Time)が基準となっているため、日本時間より9時間遅れています。[オプション設定]ダイアログボックス[バージョン管理]ページの[ログの日時を現地時間で表示する]をチェックすることで、ローカルタイムで表示するように変更できます。

[増] 欄

直前のリビジョンから挿入された行数を表示します。

[減] 欄

直前のリビジョンから削除された行数を表示します。

[ログメッセージ] 欄

コミット、またはチェックインしたときに入力したログメッセージを表示します。

【ヒント】リポジトリをUnixマシン上に置いているサイトでは、ログメッセージがEUCやUTF-8で記録されていることがあります。その場合は、[オプション設定]ダイアログボックスの[バージョン管理]ページにある[ログメッセージをEUCに変換する]、または[ログメッセージをUTF-8に変換する]をONにしてください。

[表示]

選択しているリビジョンをテンポラリファイルへ取り出して表示します。表示ウインドウは編集禁止となっているため、保存するには、[ファイル]-[名前を付けて保存]を選択し、別ファイルに保存してください。リビジョンではなくファイル[ ]を選択しているときは、作業コピーそのものを開きます。

[取得]

特定のリビジョンを取り出して作業ファイルと置き換えます。現在編集中の作業ファイルを捨て、古いリビジョンへ戻すことができます。

CVSの場合は、[取得]によりスティッキーフラグをクリアしたり、逆に作業ファイルのBASEリビジョンを特定のものに設定する(スティッキーにする)かを選択できます。スティッキーフラグを変更せずに任意のリビジョンを取り出すことも可能です。

【注意】この操作により、コミットまたはチェックインしていない作業ファイルへの変更は失われてしまいます。十分注意してから実行してください。現在のファイルをバックアップ(<オリジナルファイル名>.bak)として残すこともできます。

【ヒント】[アップデート]によってリポジトリから特定のリビジョンを取り出すと、現在ある作業ファイルのと間でマージ(併合)されてしまうことがあります。[取得]は[アップデート]と異なり、作業ファイルが存在しない状態で取り出すため、マージが発生することはありません。

[比較]

選択している数によって、リビジョンと作業ファイル、またはリビジョン間の比較を行います。比較を実行すると2つのエディタウインドウが開きます。ファイル比較と同じ操作で相違点をチェックできます。

1つだけ選択している	現在ある作業コピーと、選択しているリビジョン間を比較します。
2つ選択している	2つのリビジョンを比較します。

[詳細]

cvs.exeが出力するステータス情報をそのまま[アウトプットウインドウ]の[バージョン管理]ページへ出力します。非常に長いログメッセージの全文を読むことができます。

リビジョンを選択しているときは、そのリビジョンのログを表示します。ファイルを選択しているときは、[CVSログフィルタ]ダイアログボックスの設定に合ったリビジョン範囲のログを表示します。ただし、メッセージパターンによるフィルタは効きません。

[更新]

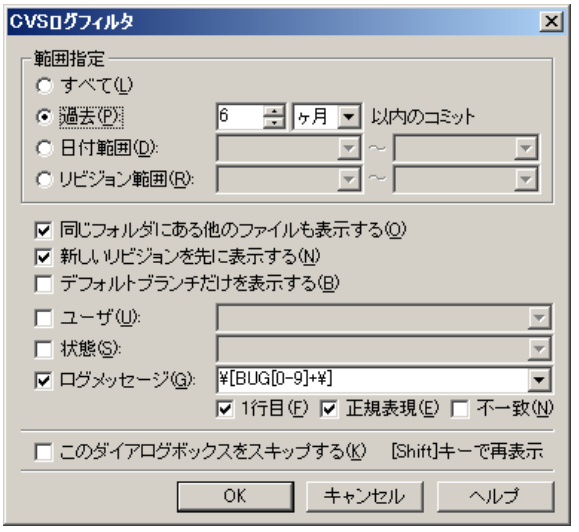
最新のログ情報を取り出して、表示内容をリフレッシュします。[CVSログフィルタ]ダイアログボックスが表示されるので、無駄な部分を表示しないようにできます。

【注意】[CVSログフィルタ]ダイアログボックスによりログの取得範囲を制限できるのは、CVSだけです。RCSでは、常にすべてのリビジョンログを取得します。

[CVS ログフィルタ] ダイアログボックス

大規模なリポジトリの場合、そこに記録されているリビジョンログは膨大な量になります。範囲を指定することで、不要なログをカットして表示します。

[ツール]-[リビジョンログ]コマンドを実行して[CVSリビジョンログ]ダイアログボックスを表示・更新する場合、または、[CVSリビジョンログ]ダイアログボックスの[更新]ボタンをクリックすると表示されます。



[範囲指定]

リビジョンログを取り出す範囲を指定します。

選択	範囲
すべて	すべてのリビジョンログを取得します。
過去	指定した時点から現在までのリビジョンログを取得します。1～100分､1～100時間､1～100日、1～100週間、1～100ヶ月の範囲が指定可能です。
日付範囲	開始日以降、終了日より前のリビジョンログを取得します。開始または終了日を省略すると、終了日以前、開始日以降のログを取得します。 【例】2000/1/1 ～ 2000/2/1 【例】　　～ 1999/12/31
リビジョン範囲	リビジョン範囲を指定してログを取得します。リビジョン番号による指定以外に、タグ名も使えます。 【例】Release1 ～ Release1_1

[同じフォルダにある他のファイルも表示する]

特定のファイルを指定してリビジョンログを表示する場合でも､同じフォルダ中にあるファイルのリビジョンログもまとめて取得します｡一度に多くのファイルのログを調べることが可能です。

[新しいリビジョンを先に表示する]

最近チェックインしたリビジョンが上になるよう表示します。OFFにすると古い順に表示します。

また､同じリビジョンに複数のタグが設定されているとき、このオプションをONにしておくと、後から設定したタグを前に、古いものを後ろに表示します。

[デフォルトブランチだけを表示する]

チェックすると､デフォルトブランチに沿ったリビジョンだけを表示し､枝分かれした部分から先を表示しません。

[ユーザ]

特定のユーザが行ったコミットに限定してリビジョンログを取得します｡リビジョンログを取得したいユーザのログイン名を入力してください｡複数ユーザは、コンマで区切って指定できます。

【例】paul,scott,peggy

[状態]

指定した状態のログだけを取得できます｡コンマで区切ることにより複数の状態も指定可能です。

[ログメッセージ]

ログメッセージに、ここで指定した文字列が含まれている場合だけ表示します。

[1 行目]

ログメッセージの第1行だけを調べて、指定文字列が含まれているかどうかを判定します。2行目以降は調べません。ある運用ルールによって、例えばバグの追跡番号などを必ず1行目に入れているような場合に有効です。

[正規表現]

指定文字列を正規表現としてログメッセージを検索します。正規表現を使うことにより、柔軟なログメッセージの指定が行えます。1つのリビジョンのログメッセージ各行に対して正規表現と一致するかを調べ、1行でも一致すればそのリビジョンを表示します。ただし、[1行目]が指定されているときは、先頭行しか調べません。

[不一致]

指定したログメッセージに一致しなかったリビジョンを表示します。

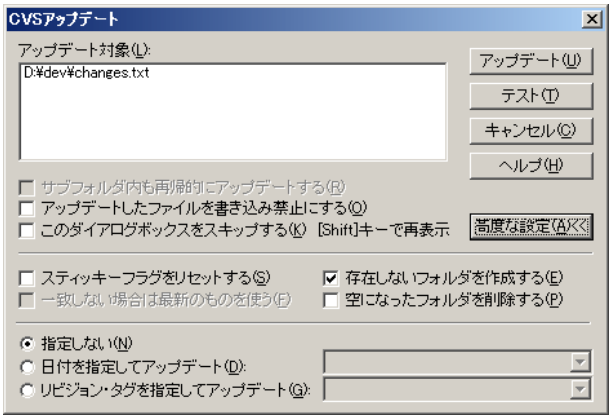
[このダイアログボックスをスキップする]

このオプションをチェックすると、以降このダイアログボックスを表示せず、即座にログを表示します。

再びログフィルタの設定を変えたくなったときは、Shiftキーを押しながら[ツール]-[リビジョンログ]コマンドを実行するか、[CVSリビジョンログ]ダイアログボックスの[更新]ボタンをクリックしてください。

[CVS アップデート] ダイアログボックス

CVSアップデートを実行するためのダイアログボックスです。このダイアログボックスは、[ツール]-[CVS]-[アップデート]コマンドを選択するか、[CVSステータス]ダイアログボックスから[アップデート]ボタンをクリックしたときに表示されます。



[アップデート対象]

アップデートするファイルやフォルダに間違いがないことを確認するためのリストです。

[サブフォルダもアップデートする]

アップデート対象にフォルダが含まれているときここをチェックすると、そのサブフォルダの中まで再帰的にファイルをアップデートします。

[アップデートしたファイルを書き込み禁止にする]

アップデートにより変更したファイルを書き込み禁止にします。

[このダイアログボックスをスキップする]

このダイアログボックスをスキップして、即座にアップデートを実行します。再び表示させてオプション等を変えるには、Shiftキーを押しながら[ツール]-[CVS]-[アップデート]コマンドを選択してください。

[高度な設定]

通常の操作ではあまり必要のないアップデートオプションを表示したり隠したりします。

[スティッキーフラグをリセットする]

cvs update -Aオプションにより、アップデート対象の作業ファイルに設定されているスティッキーフラグをクリアします。

[一致しない場合は最新のものを使う]

日付やリビジョンを指定してアップデートを実行するとき、該当リビジョンがなかった場合に最新版を使うようにします。

[存在しないフォルダを作成する]

リポジトリにあって作業コピーにない(おそらく前回のアップデート以降に誰かが追加した)フォルダを作成します。このオプションを指定しないと、フォルダは作成されません。注意してください。

[空になったフォルダを削除する]

アップデートの結果、空になったフォルダを削除します。

[指定しない]

アップデートするリビジョンを指定しません。スティッキーフラグに日付やリビジョンが設定されていない場合は、リポジトリの最新リビジョンへアップデートします。

[日付を指定してアップデート]

指定した日付に最も近い(が越えない)リビジョンに従って作業コピーをアップデートします。この指定は、スティッキーフラグをセットするので、以後のアップデートは、自動的にこの日付が基準となります。

[リビジョン・タグを指定してアップデート]

特定のリビジョンやタグを基準にして作業コピーをアップデートします。この指定は、スティッキーフラグを設定するので、以降のアップデートは、自動的のこのリビジョンやタグが基準となります。

[アップデート]

cvs updateコマンドを実行し、作業ファイルをアップデートします。

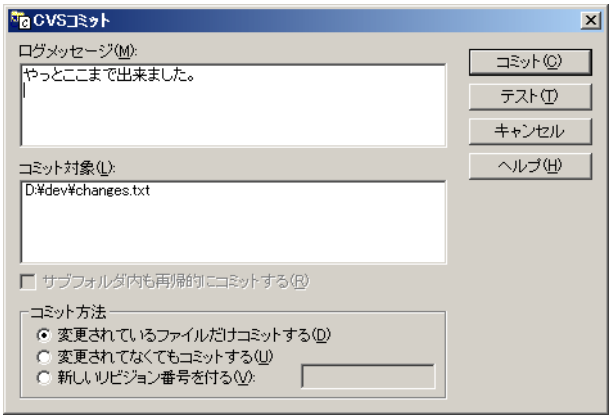
[テスト]

cvs -n updateコマンドを実行し、実際にアップデートすると何が起こるのかをテストします。作業コピーは一切書き換えられません。大量のファイルをアップデートする前にテストすることで、その後安心してアップデートを実行できます。

[CVS コミット] ダイアログボックス

作業コピーの変更をリポジトリへコミットする確認ダイアログボックスです。このダイアログボックスは、[ツール]-[CVS]-[コミット]コマンドを選択するか、[CVS

ステータス]ダイアログボックスの[コミット]ボタンをクリックしたときに表示されます。



[ログメッセージ]

コミットの理由などを簡素に表したログメッセージをタイプします。ログメッセージは、[CVSリビジョンログ]ダイアログボックスの[ログメッセージ]欄に表示され、あとから何のための変更かを思い出すヒントになります。

[コミット対象]

間違いがないかを確認するため、コミットしようとしているファイルやフォルダを表示しています。

[サブフォルダ内も再帰的にコミットする]

コミット対象にフォルダが含まれている場合、ここをチェックするとサブフォルダの中のファイルも再帰的にすべてコミットします。

[コミット方法]

以下の3つからコミットする方法を選択します。

[変更されているファイルだけコミットする]	変更されているファイルだけコミットし、リビジョン番号を進めます。変更されていないファイルには、何もしません。最も一般的なコミット方法です。
[変更されてなくてもコミットする]	変更の有無にかかわらずコミットし、リビジョン番号を進めます。
[新しいリビジョン番号を付る]	新しいリビジョン番号を指定してコミットします。変更のないファイルもコミットされます。新しいリビジョン番号は、そのフォルダに含まれるどのファイルのリビジョン番号よりも大きいものでなければなりません。

[コミット]

コミットを実行します。

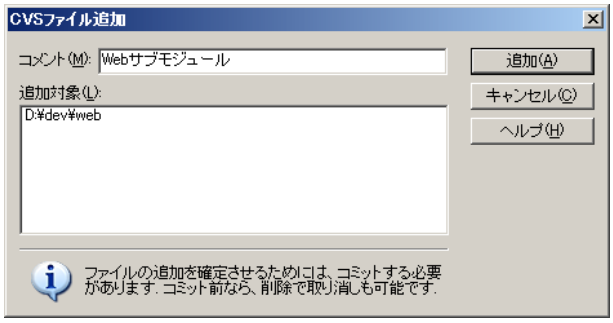
[テスト]

コミットすると何が起こるかを調べるために、コミット実行をシミュレートします。実際のリポジトリは変化しません。

【ヒント】cvs.exe -n commit ... と、-nオプションを付けてcvs.exeを実行します。

[CVS ファイル追加] ダイアログボックス

CVSに追加するファイルを確認し、コメントを入力してから、実際に追加を実行するダイアログボックスです。[CVSステータス]ダイアログボックスから、[追加]ボタンをクリックすることで表示されます。



[コメント]

一般にファイルの内容を説明する簡単なコメントを書きこみます。コメントは、省略することもできます。

[追加対象]

追加対象となっているファイルとフォルダのパスを表示します。追加しようとしているファイルやフォルダに間違いがないかを確認します。

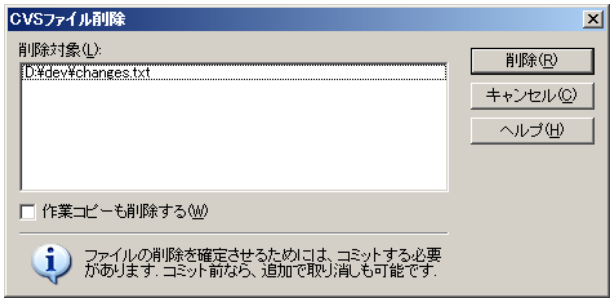
[追加]

すべて正しければ、このボタンをクリックすることで、CVSにファイルやフォルダを追加します。

【ヒント】ファイルを追加した場合は、コミットして初めて有効になります。コミット前であれば、削除することで追加の取消が可能です。

[CVS ファイル削除] ダイアログボックス

CVSに登録してあるファイルを削除します。このダイアログボックスは、[CVSステータス]ダイアログボックスの[削除]ボタンをクリックすると表示されます。



[削除対象]

CVSバージョン管理システムから削除するファイルのリストです。間違いないかを確認できます。

【メモ】CVSの場合、「フォルダを削除する」という操作はありません。リポジトリからフォルダを削除してしまうと、過去のリビジョンが取り出せなくなるからです。

その代りに、フォルダ内のすべてのファイルを削除します。その後のアップデートでは、[CVSアップデート]ダイアログボックスの[空になったフォルダを削除する]を

チェックして実行してください。アップデート処理の一部として空のフォルダを削除するため、実質的にフォルダを削除したのと同じになります。

[作業コピーも削除する]

チェックすると作業ファイルも削除します。

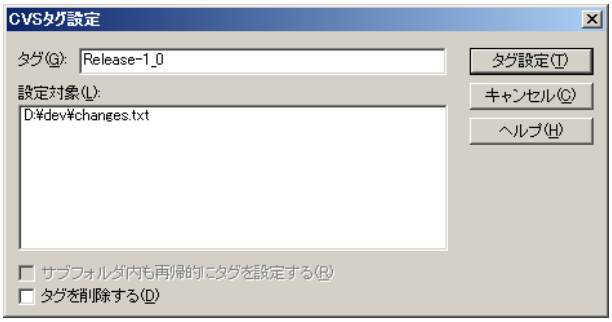
【注意】チェックを外して「作業コピーを削除」しないようにしても、削除を確定するコミットで削除されてしまいます。どうしても作業コピーを残しておきたい場合は、予め他の名前に変更するなどしてください。すべての変更をコミットしているのであれば、リビジョンを指定することで、いつでもリポジトリから取り出せます。

[削除]

削除を実行します。削除を確定させるには、コミットしなければなりません。コミット前であれば、再追加することで削除を取り消せます。

[CVS タグ設定] ダイアログボックス

現在のリビジョンにタグを設定します。このダイアログボックスは、[CVSステータス]ダイアログボックスの[タグ]ボタンをクリックすると表示されます。



[タグ]

設定するタグ名を指定します。タグ名に使える文字は、半角の英数字(A-Z、a-z、0-9)とハイフン(-)、それにアンダースコア(_)のみなので注意してください。

[設定対象]

[CVSステータス]ダイアログボックスで選択したタグを設定する対象を表示します。タグの設定対象に間違いがないことを確認します。

[サブフォルダ内も再帰的にタグを設定する]

設定対象にフォルダが含まれている場合、ここをチェックすることで、サブフォルダ中のファイルまで再帰的にタグを設定します。

[タグを削除する]

指定されたタグを削除します。

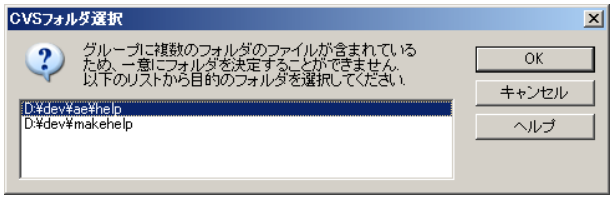
[タグ設定]

タグ設定を実行します。タグ設定はその場で実行されるため、コミットする必要はありません。

[CVS フォルダ選択]・[Subversion フォルダ選択] ダイアログボックス

プロジェクトウインドウのグループを選択してCVS/Subversionバージョン管理システムのコミット、アップデート、リビジョンログの表示、ステータスの表示をしようとしたとき、グループに対応するフォルダが一意に決定できない場合に表示さ

れるダイアログボックスです。目的のフォルダを選択してから、[OK]ボタンをクリックしてください。



本来プロジェクトのグループは、ファイルシステムのフォルダとは独立した概念であり、どこのフォルダにあるファイルでも自由に入れることができます。しかし、CVS/Subversionの作業コピーをフォルダごとまとめてプロジェクトへドロップしたときなどは、グループは作業コピーのフォルダと非常に近い関係になり、フォルダとしても扱えると便利です。

そこで、CVS/Subversionのコミット、アップデート、ステータス、リビジョンログコマンドでは、グループに登録してあるファイルからグループに対応するフォルダを推測します。その結果、唯一フォルダが特定できたときは、そのままコマンドを実行しますが、複数フォルダのファイルが混在している場合は、フォルダのリストを作成して、このダイアログボックスを表示します。

Subversion によるバージョン管理

まず始めに、Peggy ProがサポートするSubversionユーザインターフェイスは、CVSと同じく基本部分のみであることを理解してください。

【注意】Peggy ProからSubversionを使うには、システムにコマンドライン版のSubversionであるsvn.exeがインストールされていることが前提です。コマンドライン版のSubversionは、Subversionプロジェクトのホームページからダウンロードできます。インストーラ形式になったものが便利です。

Subversion コマンド

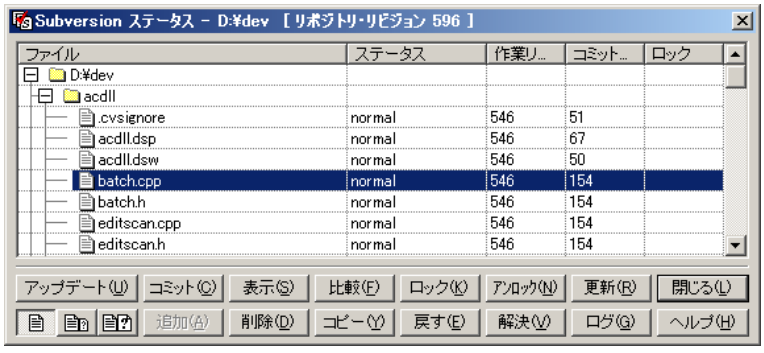
ボタン	メニュー操作	機能
	[ツール]-[Subversion]-[アップデート]	[Subversionアップデート]ダイアログボックスを開いて、現在編集しているファイルをアップデートします。 プロジェクトウインドウのファイルやグループアイテムを選択しているときは、それに対応したファイルやフォルダをアップデートします。

	[ツール]-[Subversion]-[コミット]	[Subversionコミット]ダイアログボックスを開いて、現在編集しているファイルをコミットします。 プロジェクトウインドウのファイルやグループアイテムを選択しているときは、それに対応したファイルやフォルダをコミットします。
	[ツール]-[Subversion]-[ステータス]	[Subversionステータス]ダイアログボックスを開いて、同じフォルダにある他のファイルとともに、現在編集しているファイルのステータスを表示します。
	[ツール]-[リビジョンログ]	[Subversionリビジョンログ]ダイアログボックスを開いて、編集しているファイルのリビジョンログを表示します。
	[ツール]-[リビジョン比較]	編集している作業ファイルと、作業ファイルの元になったBASEリビジョンを比較します。
	[ツール]-[Subversion]-[ログ]	[Subversion リビジョンログ]ダイアログボックスを開いて、編集しているファイルのリビジョンログを表示します。
	[ツール]-[Subversion]-[比較]	編集している作業ファイルと、作業ファイルの元になったBASEリビジョンを比較します。
	[ツール]-[Subversion]-[チェックアウト]	リポジトリから作業ファイルをチェックアウトします。
	[ツール]-[Subversion]-[リポジトリブラウザ]	リポジトリの内容を表示するリポジトリブラウザを表示します。

[Subversion ステータス] ダイアログボックス

[Subversion ステータス]ダイアログボックスは、Subversion作業コピーの状態を表示します。このダイアログボックスからアップデート、コミットなど、Subversionの主要な操作を行うことが可能です。

このダイアログボックスを表示するには、メニューから[ツール]-[Subversion]-[ステータス]コマンドを実行します。そのとき編集しているファイルの状態を、同じフォルダにある他のファイルと共に表示します。プロジェクトウインドウのファイルアイテムを選択してから[ツール]-[Subversion]-[ステータス]コマンドを実行することでも表示されます。



[ファイル] 欄

Subversion作業フォルダの階層と、その中に登録されているファイルの名前を表示します。ヘッダのこの部分をクリックすると、ファイル名順に並べ替えて表示できます。

[ステータス] 欄

[ファイル]欄のアイコンと共に作業ファイルやフォルダの状態を表示します。ヘッダのこの部分をクリックすると、ステータスに従って並べ替えることができます。

【ヒント】ステータスでソートすると、同じ状態のファイルが近くに並ぶので、作業コピーの状態を把握しやすくなります。

アイコン	ステータス	意味
	normal	作業ファイル編集されておらず、BASEリビジョンと同じあることを示しています。
	modified	作業ファイルに変更を加えていることを示しています。
	to be updated	リポジトリに作業コピーより新しいリビジョンがコミットされていることを示しています。[アップデート]により最新にできます。【注意】参照
	missing	作業ファイルがないことを示しています。[アップデート]により最新のファイルを取得できます。
	incomplete	作業ファイルが壊れているなど、不完全な状態にあることを示しています。[アップデート]により最新のファイルを取得できます。
	obstructed	作業ファイルが別のオブジェクトで置き換えられてることを示しています。[アップデート]により最新のオブジェクトを取得できます。
	to be merged	リポジトリには新しいバージョンがコミットされ、同時に作業コピーも編集している状態を表しています。[アップデート]により、両方の変更箇所をマージ(併合)する必要があります。【注意】参照
	conflicted	マージの際に変更部分が衝突してしまったことを示しています。衝突箇所を手作業で編集してから[解決]、[コミット]します。
	added	追加したけれど、まだコミットされていないファイルであることを示しています。
	deleted	削除したけれど、まだコミットされていないファイルであることを示しています。
		svn:ignore 属性などにより無視するように指定されているファイルです。
		ファイルがSubversionの管理下のものではないことを示しています。
		まだ状態を取得していないファイルです。
	normal	Subversionの管理下にあるフォルダを表しています。

	added	追加したけれど、まだコミットされていないフォルダであることを示しています。
	deleted	削除したけれど、まだコミットされていないフォルダであることを示しています。
	ignored	svn:ignore 属性などにより無視するように指定されているフォルダです。
		Subversionの管理下ではないフォルダを表しています。[Subversion ステータス]ダイアログボックスでは、管理外のフォルダを開いてその下のファイルを表示することはできません。
		これらはそれぞれ、[normal]、[modified]、[to be updated など]、[to be merged]、[conflicted]、[added]、[delete] の状態に svn:needs-lock プロパティが付いていることを表しています。 svn:needs-lock プロパティが付いているファイルは、チェックアウトやアップデートで書き込み禁止となり、ロックすることで書き込み可能になります。

【注意】リポジトリの参照
これらの状態はリポジトリへアクセスしなければ判断できないため、[オプション設定]ダイアログボックス[バージョン管理]ページの[Subversionを使う]-[ステータス表示時にリポジトリの最新状態を参照する]オプションが選択されているときだけ表示されます。

[作業リビジョン]欄

作業ファイルの元になっているBASEリビジョンのリビジョン番号を表示します。

[コミットリビジョン]欄

BASEリビジョンがコミットされたリビジョン番号を表示します。

[ロック]欄

ファイルがロックされているかどうかを表示します。この作業ファイルからロックしている場合は My Lock ....、他の作業ファイルやユーザーからロックされている場合はユーザ名を表示します。続けてロックした日時、あればコメントも表示します。

【ヒント】他のユーザーからロックされている間は、コミットできません。

[アップデート]

[Subversion アップデート]ダイアログボックスを開いて、選択している作業ファイルやフォルダをアップデートします。

[コミット]

[Subversion コミット]ダイアログボックスを開いて、選択している作業ファイルやフォルダに行った変更をリポジトリへコミットします。

[表示]

選択している作業ファイルを開きます。

【メモ】[オプション設定]ダイアログボックス[バージョン管理]ページの[外部ビューアで開くファイル]パターンに一致するときは、外部ビューアを起動します。それ以外は、編集ウインドウを開いて表示します。

[比較]

選択しているのがファイルかフォルダかで動作が異なります。ファイルを選択している場合は、作業コピーの元になっているBASEリビジョンと比較します。違いがあれば2つの編集ウインドウを開き、最初の相違行へジャンプします。以後、F4、Shift+F4キーで前後の相違箇所へジャンプ可能です。

フォルダを選択しているときは、そのフォルダの作業コピーを作業コピーの元になったBASEリビジョンに対しまとめて比較します。結果はアウトプットウインドウの[バージョン管理]ページに出力されます。

[ロック]

選択しているファイルをロックし、他のユーザがコミットできないようにします。

[アンロック]

選択しているファイルのロックを解除し、他のユーザがコミットできるようにします。

[更新]

現在選択しているフォルダ内のステータスを読み直して、最新の状態にします。

【注意】ステータスを更新するのは、選択しているファイルのあるフォルダ内だけです。他の部分は更新されないので注意してください。

[ ]、[]、[]

Subversionでバージョン管理を行っていないファイルやフォルダの表示方法を切り替えます。[追加]ボタンにより新しいファイルをSubversionへ追加する場合は、[]や[]をクリックし、Subversionに登録されていないファイルやフォルダを表示させてください。

ボタン	表示内容
	Subversionに登録されているファイルとフォルダのみを表示します。
	Subversionに登録してあるファイルとフォルダに加え、メモに記載されているパターン以外のファイル/フォルダも表示します。
	RCS、CVS、.svn を除くすべてのファイルとフォルダを表示します。

【メモ】[]が選択されているときに表示しないデフォルトのファイルパターンは、SCCS、RCS、CVS、CVS.adm、RCSLOG、cvslog.*、TAGS、*~、#*、.#*、,*、_$*、*$、*.old、*.bak、*.orig、*.rej、.del-*、*.a、*.olb、*.o、*.obj、*.so、*.exe、*.Z、*.elc、*.ln、core、.make.state、.nse_depinfo、Thumbs.db、mssccprj.scc、vssver.scc、*.mine、*.left、*.rightです。

[追加]

[Subversion 追加]ダイアログボックスを表示して、Subversionに新しくファイルやフォルダを追加します。追加対象となるファイルやフォルダは、[]や[]ボタンをクリックすると表示されます。

まだSubversionに登録されていない[ファイル]や [フォルダ]を選択してから、[追加]ボタンをクリックしてください。複数のファイルやフォルダを一度に追加できますが、それらはすべて同じフォルダ内のものでなければなりません。

【ヒント】ファイルやフォルダを追加・削除・コピーしても、コミットするまでは正式に完了したことになりません。コミット前であれば、[戻す]ボタンで取り消すことができます。

[削除]

[Subversion 削除]ダイアログボックスを表示して、Subversionからファイルを削除します。削除したい1つまたは複数のファイルやフォルダを選択してから、[削除]ボタンをクリックしてください。

[コピー]

コピーしたい1つまたは複数のファイルやフォルダを選択してから、[コピー]ボタンをクリックしてください。[Subversion コピー]ダイアログボックスを表示して、Subversionの管理下にあるファイルやフォルダをコピーします。

[戻す]

選択しているファイルやフォルダの変更を取り消しBASE リビジョンに戻す[Subversion 戻す]ダイアログボックスを表示します。

【重要】ファイルやフォルダ内のファイルに対行った変更がすべて失われてしまいます。慎重に実行してください。

[解決]

アップデート時に発生した衝突が解決したことをSubversionに知らせます。[conflicted]が表示されているファイルを選択し、このボタンをクリックしてください。

[ログ]

選択しているファイルの変更履歴を表示する[Subversion リビジョンログ]ダイアログボックスを表示します。このボタンをクリックすると、[Subversion ステータス]ダイアログボックスは閉じられます。

[Subversion リビジョンログ]ダイアログボックス

Subversion で管理しているファイルの履歴を表示するダイアログボックスです。このダイアログボックスから過去の任意リビジョンの表示や、作業ファイルとの比較、過去のリビジョン同士の比較が行えます。

このダイアログボックスを表示するには、メニューから[ツール]-[Subversion]-[ログ]コマンドを実行してください。そのとき編集しているファイルのログを表示します。

プロジェクトウインドウのファイルアイテムを選択してから［ ツール ］-[Subversion]-[ログ]コマンドを実行することで、ログを表示させることも可能です。

［リビジョン］欄

ファイル名と登録されているリビジョンを表示します。

アイコン	意味
	作業ファイルを表しています。
	1つのリビジョンを表しています。
	作業ファイルのベースとなったリビジョン(BASEリビジョン)であることを表しています。

【ヒント】最新リビジョンの表示
デフォルトではBASEリビジョンまでしか表示されません。BASEより新しいリビジョン、つまりリポジトリの最新リビジョンまで表示させたいときは、[Subversion ログフィルタ]ダイアログボックスの[範囲指定]で[リビジョン範囲]を選択し、範囲を（空欄）～ HEAD と指定します。[Subversion ログフィルタ]ダイアログボックスを表示するには、Shiftキーを押しながら[更新]ボタンをクリックしてください。

［ユーザ］欄

そのリビジョンをコミットしたユーザ名を表示します。

［日付］欄

そのリビジョンをコミットした日時を表示します。

［メッセージ］欄

コミットしたときに入力したログメッセージを表示します。

［表示］

選択しているリビジョンをテンポラリファイルへ取り出して表示します。編集ウインドウは編集禁止となっているため、保存するには[ファイル]-[名前を付けて保存]を選択し、別ファイルに保存してください。

【メモ】ファイルを表示するとき、[オプション設定]ダイアログボックス[バージョン管理]ページの[外部ビューアで開くファイル]パターンに一致する場合は、外部ビューアを起動します。それ以外は、編集ウインドウを開いて表示します。

［取得］

特定のリビジョンを取り出して作業ファイルと置き換えます。現在編集中の作業ファイルを捨て、古いリビジョンへ戻すことができます。

【注意】この操作により、まだコミットしていない作業ファイルへの変更は失われてしまいます。十分注意してから実行してください。現在のファイルをバックアップ(<オリジナルファイル名>.bak)として残すこともできます。

【ヒント】[アップデート]によってリポジトリから特定のリビジョンを取り出すと、現在ある作業ファイルのと間でマージ(併合)されてしまうことがあります。[取得]は[アップデート]と異なり、作業ファイルが存在しない状態で取り出すため、マージが発生することはありません。

[比較]

選択している数によって、リビジョンと作業ファイル、またはリビジョン間の比較を行います。比較を実行すると2つの編集ウインドウが開きます。ファイル比較と同じ操作で相違点をチェックできます。

1つだけ選択している場合は、現在ある作業コピーと選択しているリビジョンを比較します。2つ選択している場合は、それら2つのリビジョン間の比較を実行します。

[詳細]

svn.exe が出力するステータス情報をそのままアウトプットウインドウの[バージョン管理]ページへ出力します。非常に長いログメッセージの全文を読むことができます。

リビジョンを選択しているときは、そのリビジョンのログを表示します。ファイルを選択しているときは、[Subversion ログフィルタ]ダイアログボックスの設定に合ったリビジョン範囲のログを表示します。ただし、ユーザ名、メッセージパターンによるフィルタは効きません。

[更新]

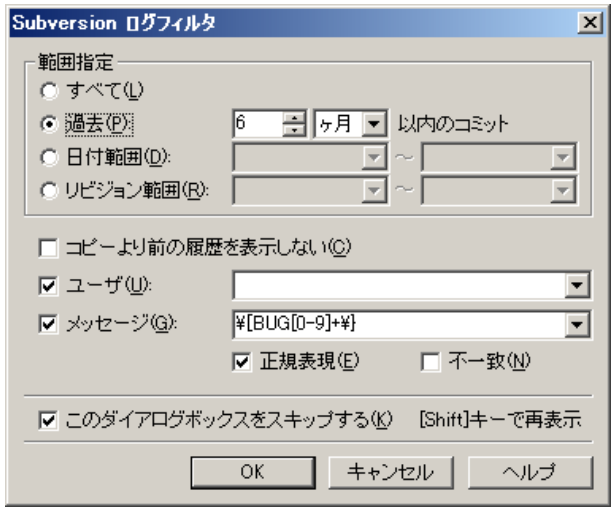
最新のログ情報を取り出して、表示内容をリフレッシュします。[Subversion ログフィルタ]ダイアログボックスが表示されるので、無駄な部分を表示しないようにできます。

[Subversion ログフィルタ]ダイアログボックス

大規模なリポジトリの場合、そこに記録されているリビジョンログは膨大な量になります。範囲を指定することで、不要なログをカットして表示します。表示に要する時間も短縮されます。

[ツール]-[Subversion]-[ログ]や[ツール]-[リビジョンログ]コマンドを実行して[Subversion リビジョンログ]ダイアログボックスを表示・更新するとき、または

[Subversion ステータス]ダイアログボックスの[更新]ボタンをクリックするときに表示されます。



[範囲指定]

リビジョンログを取り出す範囲を指定します。

選択	範囲
すべて	すべてのリビジョンログを取得します。
過去	指定した時点から現在までのリビジョンログを取得します。1～100分、1～100時間、1～100日、1～100週間、1～100ヶ月の範囲が指定可能です。
日付範囲	開始日以降、終了日より前のリビジョンログを取得します。開始または終了日を省略すると、終了日以前、開始日以降のログを取得します。 【例】2000/1/1 ～ 2000/2/1 【例】　　～ 2007/12/31 日時による指定以外に、HEAD、BASE、COMMITTED、PREV も使えます。
リビジョン範囲	リビジョン範囲を指定してログを取得します。リビジョン番号による指定以外に、HEAD、BASE、COMMITTED、PREV も使えます。

【ヒント】リビジョンの表示範囲を狭くすると、サーバーの処理が軽く早くなります。

【ヒント】最新リビジョンの表示
デフォルトではBASEリビジョンまでしか表示されません。BASEより新しいリビジョン、つまりリポジトリの最新リビジョンまで表示させたいときは、[範囲指定]で[リビジョン範囲]を選択し、範囲を（空欄）～ HEAD と指定します。

[コピーより前の履歴を表示しない]

対象ファイルがコピーされた枝分かれの中にある作業ファイルである場合、コピー時点より古いリビジョンを表示しません。

[ユーザ]

特定のユーザが行ったコミットに限定してリビジョンログを取得します。リビジョンログを取得したいユーザ名を入力してください。複数ユーザは、コンマで区切って指定できます。

【例】paul,scott,peggy

[正規表現]

指定文字列を正規表現としてログメッセージを検索します。正規表現を使うことにより、柔軟なログメッセージの指定が行えます。1つのリビジョンのログメッセージに対して正規表現と一致するかを調べ、一致すればそのリビジョンを表示します。

[不一致]

指定したログメッセージに一致しなかったリビジョンを表示します。

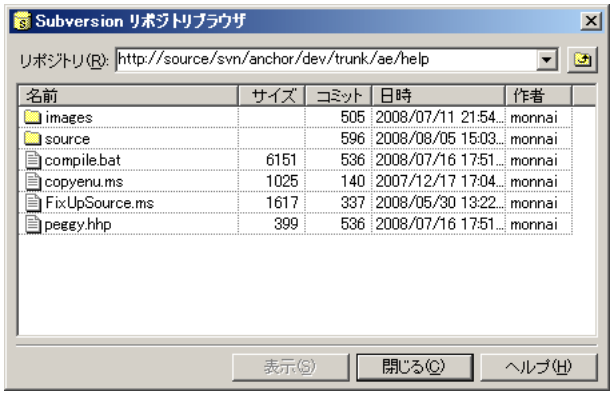
[このダイアログボックスをスキップする]

このオプションをチェックすると、以降このダイアログボックスを表示せず、即座にログを表示します。

再びログフィルタの設定を変えたくなったときは、Shiftキーを押しながら[ツール]-[リビジョンログ]コマンドを実行するか、[Subversion リビジョンログ]ダイアログボックスの[更新]ボタンをクリックしてください。

[Subversion リポジトリブラウザ] ダイアログボックス

Subversion リポジトリの内容をブラウズするためのダイアログボックスです。作業コピーとして取り出すことなく、Subversioin リポジトリの構造やファイルを表示できます。



[リポジトリ]

リポジトリの場所を指定するURLを記述します。http://...、https://...、file:///...、svn://...、svn+ssh://... のようなスキーマを持ったURLを指定します。詳しくは、Subversion のドキュメントを参照してください。

【メモ】[Subversion チェックアウト]ダイアログボックスから呼び出されているときは、フォルダしか表示されません。

[ ]

リポジトリ内の1つ上のフォルダへ移動します。

[ファイル・フォルダリスト]

[リポジトリ]で指定されているフォルダ内にあるファイルとフォルダの一覧を表示します。

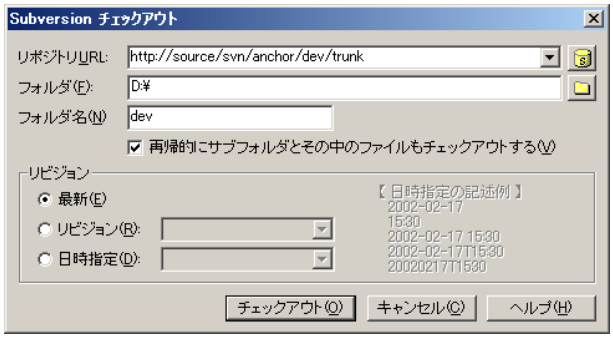
[表示]

リストでフォルダを選択しているときは、そのフォルダを開いて内部を表示します。リストでファイルを選択しているときは、そのファイルを表示します。

【メモ】ファイル表示するとき、[オプション設定]ダイアログボックス[バージョン管理]ページの[外部ビューアで開くファイル]パターンに一致する場合は外部ビューアを起動します。それ以外は、編集ウインドウを開いて表示します。

[Subversion チェックアウト] ダイアログボックス

Subversionによりバージョン管理されているファイルやフォルダを作業コピーとしてチェックアウトするためのダイアログボックスです。このダイアログボックスは、[ツール]-[Subversion]-[チェックアウト]コマンドを選択すると表示されます。



[リポジトリ URL]

リポジトリの場所を指定するURLを記述します。http://...、https://...、file:///...、svn://...、svn+ssh://... のようなスキーマを持ったURLを指定できます。

詳しくは、Subversion のドキュメントを参照してください。[ ]ボタンをクリックし、[Subversion チェックアウトリポジトリの選択]ダイアログボックスを使ってチェックアウトするリポジトリを選択することも可能です。

[フォルダ]

作業コピーを置くフォルダを指定します。[ ]ボタンをクリックしてフォルダを選択することもできます。

[フォルダ名]

[フォルダ]で指定したフォルダ内に作成する作業コピーのトップレベルフォルダ名を指定します。[フォルダ名]を省略すると、[リポジトリURL]の最後の部分がトップレベルフォルダ名となります。

[再帰的にサブフォルダとその中のファイルもチェックアウトする]

サブフォルダ内のファイルも再帰的にチェックアウトします。

[リビジョン]

特定のリビジョンや日時を指定すると、その時点の状態を取り出すことができます。

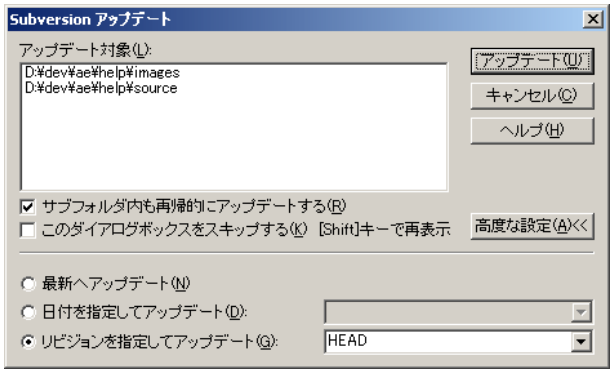
[最新]	最新版をチェックアウトします。
[リビジョン]	リビジョンを指定してチェックアウトします。
[日時指定]	日時を指定してチェックアウトします。2008/4/1、2008-04-01 23:59:59 のような指定が可能です。

[チェックアウト]

チェックアウトを実行し、作業コピーを取得します。

[Subversion アップデート] ダイアログボックス

Subversionアップデートを実行するためのダイアログボックスです。このダイアログボックスは、[ツール]-[Subversion]-[アップデート]コマンドを選択するか、[Subversion ステータス]ダイアログボックスから[アップデート]ボタンをクリックしたときに表示されます。



[アップデート対象]

アップデートするファイルやフォルダに間違いがないことを確認するためのリストです。

[サブフォルダ内も再帰的にアップデートする]

アップデート対象にフォルダが含まれているときここをチェックすると、そのサブフォルダの中まで再帰的にファイルやフォルダをアップデートします。

[このダイアログボックスをスキップする]

このダイアログボックスをスキップして、即座にアップデートを実行します。再び表示させてオプション等を変えるには、Shiftキーを押しながら[ツール]-[Subversion]-[アップデート]コマンドを選択してください。

[最新へアップデート]

アップデートするリビジョンを指定せず、最新リビジョンへアップデートします。

[日付を指定してアップデート]

指定した日付に最も近い(が越えない)リビジョンに従って作業コピーをアップデートします。2008/4/1、2008-04-01 12:00:00 のような日付や時間の他に、HEAD、BASE、COMMITTED、PREV などの定義済みリビジョンも指定可能です。

[リビジョンを指定してアップデート]

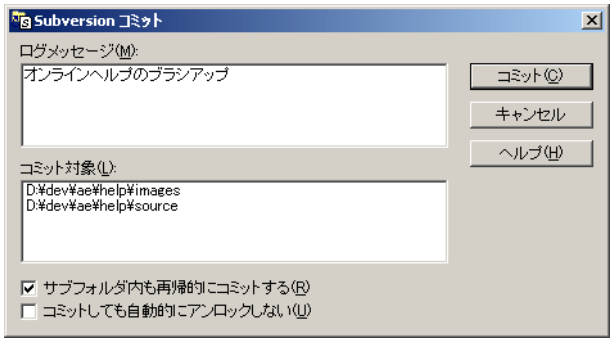
特定のリビジョンやタグを基準にして作業コピーをアップデートします。120、121 のようなリビジョン番号の他に、HEAD、BASE、COMMITTED、PREV などの定義済みリビジョンも指定可能です。

[アップデート]

作業コピーをアップデートします。

[Subversion コミット] ダイアログボックス

作業コピーの変更をリポジトリへコミットする確認ダイアログボックスです。このダイアログボックスは、[ツール]-[Subversion]-[コミット]コマンドを選択するか、[Subversion ステータス]ダイアログボックスの[コミット]ボタンをクリックしたときに表示されます。



[ログメッセージ]

コミットの理由などを簡素に表したログメッセージをタイプします。ログメッセージは、[Subversion リビジョンログ]ダイアログボックスの[ログメッセージ]欄に表示され、あとから何のための変更かを思い出すヒントになります。

[コミット対象]

確認のためコミットしようとしているファイルやフォルダの一覧を表示します。

[サブフォルダ内も再帰的にコミットする]

コミット対象にフォルダが含まれている場合、ここをチェックするとサブフォルダの中のファイルやフォルダも再帰的にコミットします。

[コミットしても自動的にアンロックしない]

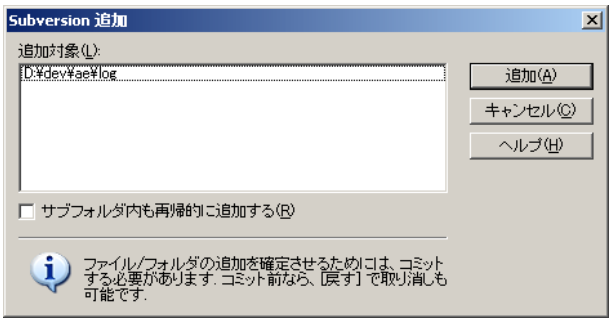
コミットによりロックを自動解除しないようにします。

[コミット]

コミットを実行します。

[Subversion 追加] ダイアログボックス

Subversionに追加するファイルやフォルダを確認し、追加するダイアログボックスです。[Subversion ステータス]ダイアログボックスから、[追加]ボタンをクリックすることで表示されます。



[追加対象]

追加対象となっているファイルとフォルダのパスを一覧表示します。追加しようとしているファイルやフォルダに間違いがないかを確認します。

[サブフォルダ内も再帰的に追加する]

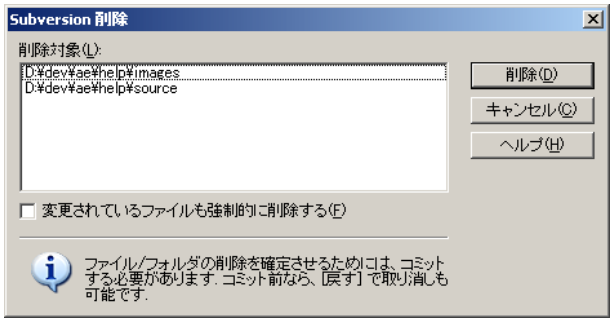
サブフォルダ内のファイルやフォルダも再帰的に追加します。

[追加]

Subversionにファイルやフォルダを追加します。追加を確定させるには、コミットしなければなりません。コミット前であれば、[Subversion ステータス]ダイアログボックスの[戻す]ボタンにより追加を取り消すことが可能です。

[Subversion 削除] ダイアログボックス

Subversionに登録してあるファイルやフォルダを削除します。このダイアログボックスは、[Subversion ステータス]ダイアログボックスの[削除]ボタンをクリックすると表示されます。



[削除対象]

Subversionバージョン管理システムから削除するファイルやフォルダのリストです。間違いないかを確認できます。

[変更されているファイルも強制的に削除する]

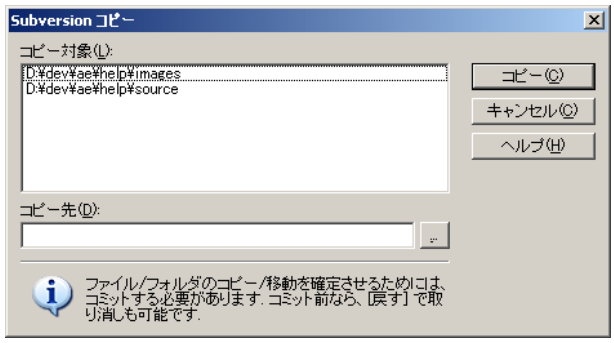
作業ファイルが変更されていると削除できませんが、このオプションを指定することで強制的に削除できます。

［削除］

削除を実行します。削除を確定させるには、コミットしなければなりません。コミット前であれば、[Subversion ステータス]ダイアログボックスの[戻す]ボタンにより削除を取り消すことが可能です。

[Subversion コピー］ダイアログボックス

Subversion内でコピーするファイルやフォルダを確認し、コピーを実行するダイアログボックスです。[Subversion ステータス]ダイアログボックスから、[コピー]ボタンをクリックすることで表示されます。



［コピー対象］

コピーの対象となるファイルやフォルダの一覧です。

［コピー先］

コピー先のフォルダを指定します。

【注意】コピー先フォルダは、同じリポジトリの作業コピーを収めたフォルダでなければなりません。リポジトリURLを直接指定してのコピーはサポートしていません。

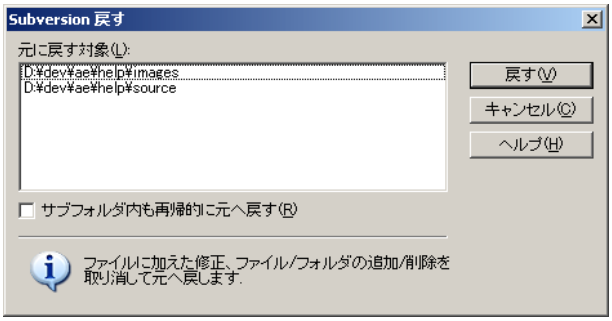
［コピー］

コピーを開始します。コピーを確定させるには、コミットしなければなりません。コミット前であれば、[Subversion ステータス]ダイアログボックスの[戻す]ボタンによりコピーを取り消すことが可能です。

【メモ】Windows ExplorerやコンソールウインドウのcopyコマンドでコピーしたファイルをSubversionに追加した場合は、新しいファイルが Subversion に追加されたと見なされ、過去の履歴が参照できません。しかしこの方法でコピーすると、ファイルやフォルダの履歴を引き継いだ状態でコピーできます。

[Subversion 戻す] ダイアログボックス

[Subversion ステータス]ダイアログボックスから、[戻す]ボタンをクリックすることで表示され、変更されたファイルやフォルダをBASEリビジョンの状態へ戻します。



【重要】ファイルやフォルダ内のファイルに対行った変更がすべて失われてしまいます。慎重に確認してから[戻す]ボタンをクリックしてください。

[元に戻す対象]

元に戻すファイルやフォルダの一覧を表示します。

[サブフォルダ内も再帰的に元へ戻す]

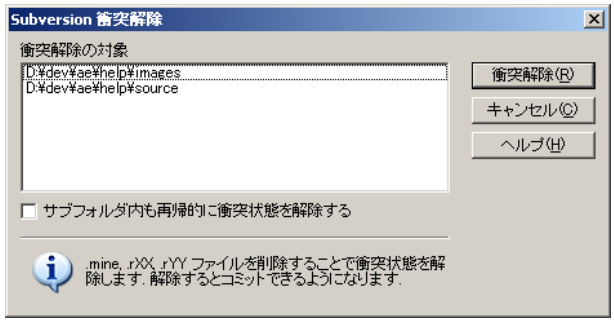
サブフォルダ内も再帰的にBASEリビジョンへ戻します。

[戻す]

BASEリビジョンへ戻す処理を実行します。

[Subversion 衝突解除] ダイアログボックス

[Subversion ステータス]ダイアログボックスから[解決]ボタンをクリックすることで表示されます。変更箇所の衝突状態が解決したことをSubversionに知らせます。



【メモ】実際には、衝突に関係した中間ファイル〈filename.mine〉、〈filename.r*OLDREV*〉、〈filename.r*NEWREV*〉（*OLDREV*、*NEWREV* は前後のリビジョン番号）を削除するだけです。

[衝突解除の対象]

衝突状態を解決するファイルやフォルダの一覧を表示します。

[サブフォルダ内も再帰的に衝突状態を解除する]

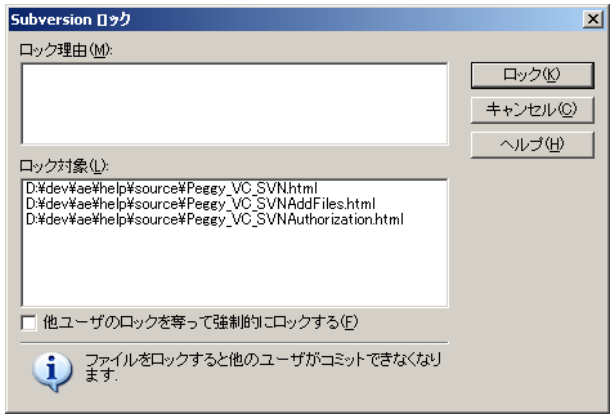
サブフォルダ内も再帰的に衝突状態が解決したことにします。

[衝突解除]

衝突解除を実行します。

[Subversion ロック] ダイアログボックス

[Subversion ステータス]ダイアログボックスから[ロック]ボタンをクリックすることで表示され、選択しているファイルをロックします。



[ロック理由]

ロックの理由を記述すると[Subversion ステータス]ダイアログボックスに表示されるので、他のユーザに理由を知らせることができます。

[ロック対象]

ロックの対象となるファイルの一覧を表示します。

[他ユーザのロックを奪って強制的にロックする]

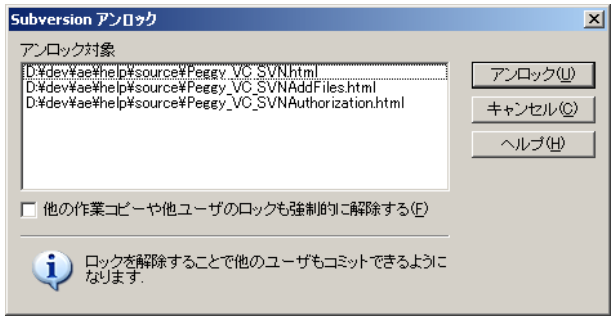
他のユーザがロックしているファイルでも、そのロックを奪って強制的にロックします。

[ロック]

ロックを実行します。ロックは直接リポジトリへ反映されるため、コミットする必要はありません。

[Subversion アンロック] ダイアログボックス

[Subversion ステータス]ダイアログボックスから[アンロック]ボタンをクリックすることで表示され、選択しているファイルをアンロックします。



[アンロック対象]

アンロックの対象となるファイルの一覧を表示します。

[他の作業コピーや他ユーザのロックも強制的に解除する]

他の作業コピーからのロックや、他のユーザが設定しているロックを強制的に解除します。

[アンロック]

アンロックを実行します。アンロックは直接リポジトリへ反映されるため、コミットする必要はありません。

VSS（Visual SourceSafe）によるバージョン管理

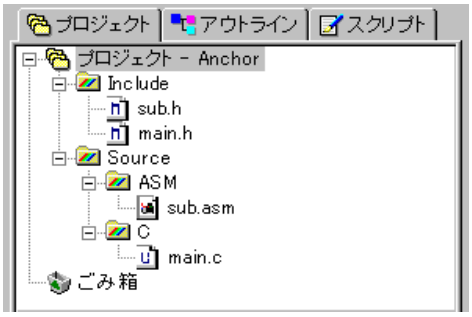
Peggy Proは普段よく使うチェックイン･チェックアウトなどVSSの基本機能のユーザインターフェイスを備えています。ただし、VSSのすべてのコマンドを使えるわけではありません。Peggy Proは、普段よく使うコマンドに機能をしぼっています。

VSS を使うための初期設定

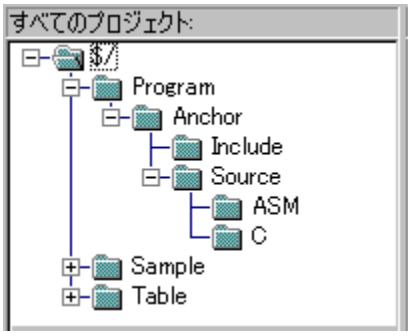
【注意】以下ではPeggy Proのプロジェクトと、VSSのプロジェクト、という２つの「プロジェクト」が出てきます。混同しないようにお読みください。

1. VSSのコマンドは、Peggy Proのプロジェクトに登録しているファイルが対象です。まず、ファイルをプロジェクトに登録してください。ここでは、Peggy Proのプロジェクトを新しく作ります。
2. 次のようなファイルで説明します。
   D:¥Develop¥Anchor¥Includeに
   main.hとsub.hの２つのファイル
   D:¥Develop¥Anchor¥Source¥ASMに
   sub.asm
   D:¥Develop¥Anchor¥Source¥Cに
   main.c
   のファイルがあります。これら４つのファイルをPeggy Proに登録します。
   エクスプローラーでディレクトリAnchorをクリックし、Ctrl+A(すべて選択)でディレクトリIncludeとSourceを選択して、Peggy Proのプロジェクトウィンドウにドロップします。
3. Peggy Proで下図のようにプロジェクトができるはずです。プロジェクト名は最初"無題"となっていますがプロジェクトを保存するときに名前を付けてください。もし、下のようにグループができない場合は一つ一つグループとファイルを登録します。４つのファイルを同じグループに登録してもかまわないのですが、実際のディレクトリ構成と同じように登録する方が、利点が多いのでこのようにします。

【例】エクスプローラの画面



4. VSSにも同じような構成でプロジェクトを作成します。



５． Peggy Proの設定をします。プロジェクトAnchorのプロパティを開いて、[VSS設定]ボタンを押します。[データベースのパス]、[ユーザ名]はコンボボックスから選択します。[プロジェクトに対応するVSSプロジェクト]に"$/Program/Anchor"と入力します。
なお、ルートプロジェクトを表す‘$/’は省略してもかまいません。[VSS設定]ボタンがグレーアウトしている場合は、後述の【注意】を参照してください。

６． ファイルをVSSに追加します。すでにVSSに登録されている場合は手順７に進んでください。プロジェクトウィンドウでプロジェクト-Anchorにフォーカスがある状態で、[ツール]-[VSS]-[ソース管理へ追加]を実行します。

７． 以上でVSSに４つのファイルが追加され、以後、チェックイン、チェックアウトなどのコマンドを使うことができます。

【ヒント】ファイルのディレクトリ構成、Peggy Proのグループ構成、VSSのプロジェクト構成の３つが同じですと上記の例のように、Peggy Proのプロジェクトやグループに対するコマンドがそのサブグループにも再帰的に適用され、一度に操作が完了し便利です。

【ヒント】手順４で、プロジェクト<Anchor>は必ず作成してください。手順６で[ソース管理へ追加]コマンドを実行する前に、対応するVSSのプロジェクトを調べています。しかし、その下の<Include>,<Source>以下のプロジェクトはコマンド実行時に作られるので無くてもかまいません。

【ヒント】手順５で、グループ<Include>や<C>の[グループに対するVSSプロジェクト]の設定は必要ありません。たとえばグループ<ASM>にはプロジェクト-Anchorの設定"$/Program/Anchor"にグループ構成が加味された"$/Program/Anchor/Source/ASM"が入力されているのと同じです。また、ファイル<main.c>に対してVSSのコマンドを実行する場合でも、Peggy Proはまずファイルが属しているグループの設定を調べます。設定されていないと、その上位のグループを、設定が見つかるまで次々に調べます。この場合、最後にプロジェクト<Anchor>に"$/Program/Anchor"という設定（もし、ここにも設定がない場合、"$/"が設定されているものとします）があり、その設定に、たどってきたグループの名前を足して、"$/Program/Anchor/Source/C"を対応するVSSのプロジェクトとします。

【注意】VSS関連のコマンドが利用できるのは、Peggy Proです。

【注意】Peggy ProにはVSSのファイルは含まれていません。また、VSSがインストールされていなかったり、Peggy ProがVSSのコマンドを呼び出すときに使う<SSSCC.DLL>が見

つからないと、これらのコマンドは有効になりません。チェックポイントは以下の通りです。

最初に、Peggy Proはレジストリの値を基に<SSSCC.DLL>を探していますので、下記のレジストリの値を確認してください。

HKEY_LOCAL_MACHINE¥Software¥Microsoft¥SourceSafeキーのSCCServerPathエントリーの値が<SSSCC.DLL>の場所を示しています。

次に上記のSCCServerPathエントリー自体がない場合、VSSのインストール時のオプション設定が原因と思われます。インストールタイプを「カスタム」にし、インストールするオプションを選択する画面で、"Visual SourceSafe統合オプション"のチェックをはずしてインストールされたと考えられますので、VSSの再インストールを行ってください。

【注意】Peggy ProにはVSSのファイルは含まれていません。また、VSSがインストールされていなかったり、Peggy ProがVSSのコマンドを呼び出すときに使う<SSSCC.DLL>が見つからないと、これらのコマンドは有効になりません。チェックポイントは以下の通りです。

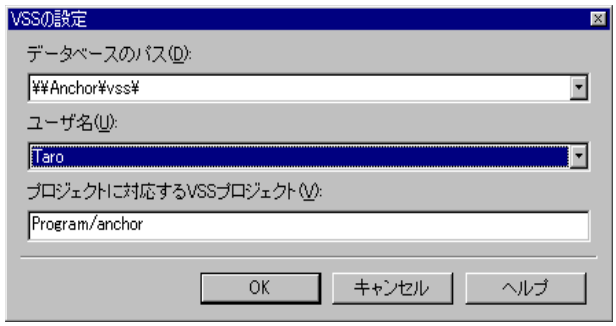
最初にPeggy Proはレジストリの値を基に<SSSCC.DLL>を探していますので、下記のレジストリの値を確認してください。

HKEY_LOCAL_MACHINE¥Software¥Microsoft¥SourceSafeキーのSCCServerPathエントリーの値が<SSSCC.DLL>の場所を示しています。

次に上記のSCCServerPathエントリー自体がない場合、VSSのインストール時のオプション設定が原因と思われます。インストールタイプを「カスタム」にし、インストールするオプションを選択する画面で、"Visual SourceSafe統合オプション"のチェックをはずしてインストールされたと考えられますので、VSSの再インストールを行ってください。

VSS の設定

VSSのコマンドを実行するためには次の３つの設定を行ってください。[VSSの設定]ダイアログボックスで値を指定します。



[データベースのパス]

VSSのデータを保存しているデータベースの場所を選択します。通常、そのサブディレクトリ<data>にデータが保存されています。

[ユーザ名]

VSSにアクセスする際のユーザ名を選択します。

[プロジェクトに対応する VSS プロジェクト]、[グループに対応する VSS プロジェクト]

VSSのコマンドを実行する時にそのプロジェクト・グループ内のファイルが、VSSのどのプロジェクトと対応しているのか（ファイルがVSSのどの場所で登録されているのか）を指定します。なお、ルートプロジェクトを表す‘$/’は省略してもかまいません。

【例】 $/anchor/source/asm

【注意】[データベースのパス]、[ユーザ名]はVSSのレジストリ設定やファイルからPeggy Proが選択肢を拾い集めています。もし、その中から選択するものがない場合は、VSSの管理者に相談してください。

【注意】[データベースのパス]、[ユーザ名]はプロジェクトの設定がプロジェクト内のすべてのグループとファイルに適用されます。そのためプロジェクトのプロパティを開いたときしか選択できず、グループのプロパティでは変更することはできません。

パスワードの入力

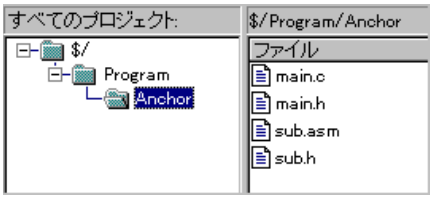
VSSのパスワードが設定されているときは、Peggy Proを起動して最初にVSSのコマンドを実行するときにパスワードを尋ねられます。Peggy Proはパスワードを記憶しないので、Peggy Proを起動するたびに入力してください。

複数ファイルをコマンドの対象とするときの注意点

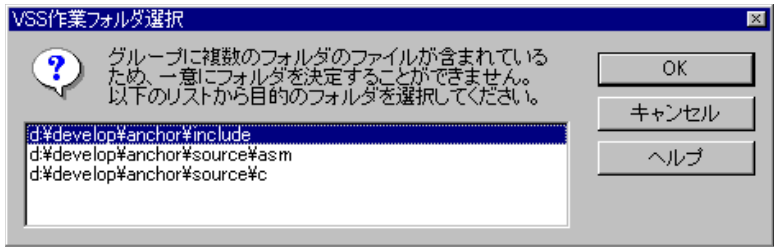
プロジェクトウィンドウのプロジェクトやグループが選択された状態でVSSのコマンドを実行すると、そのサブグループも含めたグループ内のすべてのファイルアイテムが対象になります。ただし、Peggy Proのプロジェクトは、同じグループ内に違うディレクトリのファイルでも登録できますが、このときは同時にVSSのコマンドの対象とすることはできません。具体例として、前項の「VSS(Visual SourceSafe)によるバージョン管理」で使った４つのファイルで説明いたします。

これらのファイルを次のようにグループ分けせずにプロジェクトに登録します。VSSの設定で、[プロジェクトに対応するVSSプロジェクト]は”$/Program/Anchor”と入力します。

VSSにも４つのファイルを同じプロジェクト”$/Program/Anchor”に登録します。

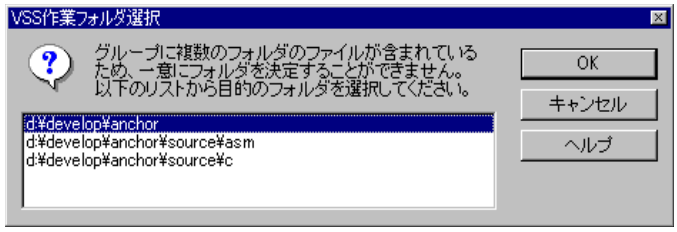


ここで、Peggyのプロジェクトウィンドウでプロジェクト-Anchor選択して、VSSのコマンドを実行するとダイアログボックスが表示されます。



Peggyのグループ構成とVSSのプロジェクト構成が同じなので、そのまま４つのファイルに対してVSSのコマンドを実行できそうですが、VSSでは、各プロジェクトに必ず作業フォルダを設定することになっていて、コマンドもその作業フォルダにあるファイルしか対象になりません。そのためPeggyがどのフォルダのファイルを対象にするかを尋ねています。選択するとそのフォルダ内のファイルだけを対象としてコマンドを実行します。なお、フォルダのリスト表示順序は対象となるファイルの数が多い順になっています。また、次のようにPeggy Proに登録した場合に注目してください。

ここでも作業フォルダを尋ねられるのですが、グループ<Include>内のファイルの作業フォルダは”d:¥develop¥anchor”と表示されています。これは、表示されるフォルダがコマンド実行対象となったグループ（この例の場合はプロジェクト-Anchor）から見たフォルダ表記になるので、グループ名<Include>を取った表示になります。ご注意ください。



このように複数のファイルを同時に（特にサブグループも含めて）VSSのコマンドを実行したいときは、ファイルのディレクトリ構成、Peggy Proのグループ構成、VSSのプロジェクト構成の３つが同じになるようにしておくことをお勧めします。例に挙げた４つのファイルの場合、

ファイルのディレクトリ……d:¥develop¥anchor

Peggy Proのグループ　……プロジェクト-Anchor

VSSのプロジェクト　……$/Program/Anchor

これらより下の構成が同じになるようにします。前項の「VSS(Visual SourceSafe)によるバージョン管理」を参考にしてください。

VSS コマンド

ボタン	メニュー操作	キー操作	機能
	[ツール]-[VSS]-[チェックイン]		[VSSチェックイン]ダイアログボックスを開いて、現在編集しているファイルをチェックインします。
	[ツール]-[VSS]-[チェックアウト]		[VSSチェックアウト]ダイアログボックスを開いて、現在編集しているファイルをチェックアウトします。
	[ツール]-[VSS]-[チェックアウト取消]		[VSSチェックアウトの取り消し]ダイアログボックスを開いて、現在編集しているファイルのチェックアウトを取り消します。
	[ツール]-[VSS]-[取得]		[VSS最新のバージョンを取得]ダイアログボックスを開いて、現在編集しているファイルの最新バージョンを取得します。
	[ツール]-[VSS]-[ソース管理へ追加]		[VSSソース管理へ追加]ダイアログボックスを開いて、現在編集しているファイルをVSSの管理下に入れます。
	[ツール]-[VSS]-[ソース管理から削除]		[VSSソースコード管理から削除]ダイアログボックスを開いて、現在編集しているファイルをVSSの管理から削除します。
	[ツール]-[VSS]-[履歴の表示]		[VSSファイルの履歴表示オプション]ダイアログボックスを開いて、編集しているファイルの履歴を表示します。
	[ツール]-[VSS]-[相違点の表示]		[VSS相違点表示オプション]ダイアログボックスを開いて、編集しているファイルとVSS内の最新バージョンの相違点を表示します。
	[ツール]-[VSS]-[プロパティ]		編集しているファイルの[VSSプロパティ]ダイアログボックスを表示します。

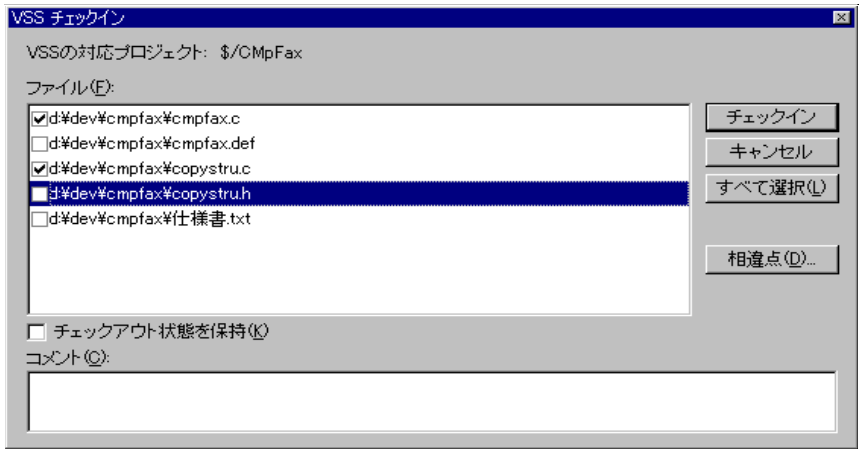
【メモ】これらのコマンドは、プロジェクトウインドウのファイルアイテムを選択しているときは、そのファイルアイテムが対象になります。グループアイテムを選択しているときは、そのグループ内のファイルアイテムとサブグループ内のファイルアイテムすべてが対象になります。

【注意】「履歴の表示」、「相違点の表示」、「プロパティ」の各コマンドはプロジェクトやグループに対しては使用できません。ファイルに対してのみ使用できます。

【注意】VSS関連のコマンドが利用できるのは、Peggy Proですが、Peggy ProにはVSSのファイルは含まれていません。VSSがインストールされていなかったり、Peggy ProがVSSのコマンドを呼び出すときに使う<SSSCC.DLL>が見つからないとこれらのコマンドは有効になりません。Peggy Proはレジストリの値を基に<SSSCC.DLL>を探していますので、下記のレジストリの値を確認してください。

HKEY_LOCAL_MACHINE¥Software¥Microsoft¥SourceSafeキーのSCCServerPathエントリーの値がSSSCC.DLLの場所を示しています。

[VSS チェックイン] ダイアログボックス



[データベースのパス]

VSSのデータを格納しているデータベースのパス。

[ユーザー名]

VSSにアクセスするときのユーザー名。

[対応する VSS プロジェクト]

Peggyが接続しようとしているVSSのプロジェクト名が表示されます。

[ファイル]

対象候補のファイルが表示されています。チェックインさせたいファイルのチェックボックスをオンにしてください。複数のファイルを選択することが出来ます。最初、チェックアウトしているファイルはオンになっています。

[チェックアウト状態を保持]

チェックインした後もファイルをチェックアウトされた状態に維持します。

[コメント]

チェックインするファイルの説明などを入力できます。ファイルの履歴を表示したとき、[詳細情報]を押すとそのバージョンをチェックインしたときのコメントが表示されます。

[チェックイン]

[ファイル]欄でチェックされているファイルをチェックインします。

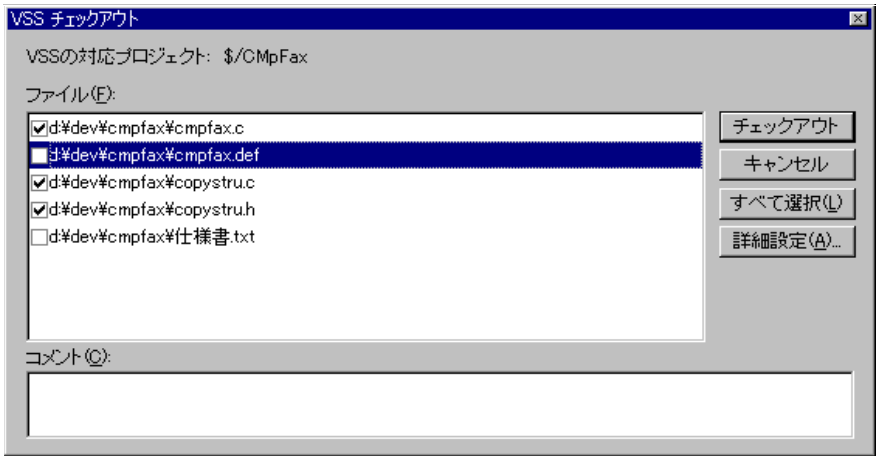
[すべて選択]

すべてのファイルを選択します。

[相違点]

[ファイル]欄でチェックされているファイルの相違点を表示するダイアログボックスを表示します。

[VSS チェックアウト] ダイアログボックス



[データベースのパス]

VSSのデータを格納しているデータベースのパス。

[ユーザー名]

VSSにアクセスするときのユーザー名。

[対応する VSS プロジェクト]

Peggyが接続しようとしているVSSのプロジェクト名が表示されます。

[ファイル]

対象候補のファイルが表示されています。チェックアウトさせたいファイルのチェックボックスをオンにしてください。複数のファイルを選択することが出来ます。最初、チェックアウトしていないファイルはオンになっています。

[コメント]

チェックアウトの理由などを説明する簡単なコメントを入力できます。チェックアウトされているファイルのプロパティを表示すると、このコメントが[チェックアウト状況]タブに表示されます。

[チェックアウト]

[ファイル]欄でチェックされているファイルをチェックアウトします。

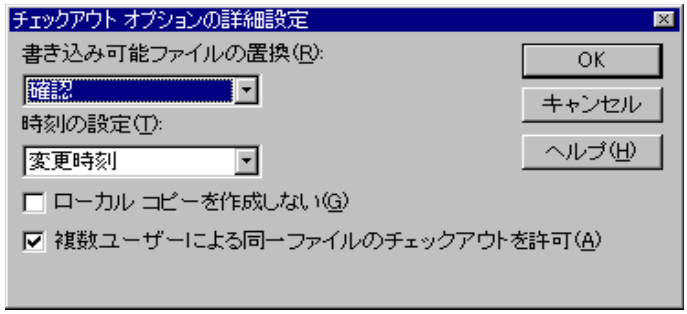
[すべて選択]

すべてのファイルを選択します。

[詳細設定]

オプションの詳細設定ダイアログボックスが開きます。

チェックアウトオプションの詳細設定ダイアログボックス



[書き込み可能ファイルの置換]

ファイルをチェックアウトするとき、既に作業フォルダに書き込み可能なローカルコピーがある場合の対処方法を設定します。

[時刻の設定]

チェックアウトしたファイルに設定する日付と時刻を指定します。[既定値]、[現在時刻]、[変更時刻]、[チェックイン時刻]から選択できます。既定値はVSSのオプションの設定が使われます。

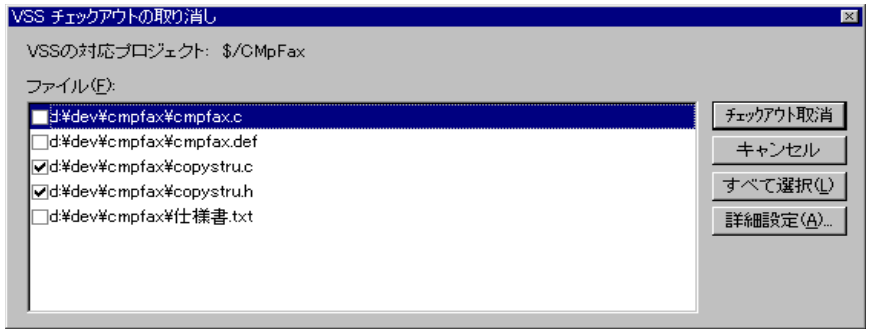
[ローカルコピーを作成しない]

ファイルが作業フォルダにコピーされません。ファイル履歴上ではチェックアウトとして記録されますが、ローカルコピーは作成されません。

[複数ユーザーによる同一ファイルのチェックアウトを許可]

複数のユーザーが同じファイルまたはプロジェクトを同時にチェックアウトできるようにします。このプロジェクトに対して多重チェックアウトが禁止されていると、オフの状態で固定になります。

[VSS チェックアウトの取り消し] ダイアログボックス



[データベースのパス]

VSSのデータを格納しているデータベースのパス。

[ユーザー名]

VSSにアクセスするときのユーザー名。

[対応する VSS プロジェクト]

Peggyが接続しようとしているVSSのプロジェクト名が表示されます。

[ファイル]

対象候補のファイルが表示されています。チェックアウトを取り消したいファイルのチェックボックスをオンにしてください。複数のファイルを選択することが出来ます。最初、チェックアウトしているファイルはオンになっています。

[チェックアウト取消]

[ファイル]欄でチェックされているファイルのチェックアウトを取り消します。

[すべて選択]

すべてのファイルを選択します。

[詳細設定]

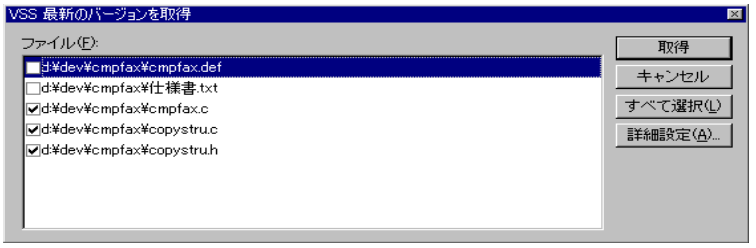
オプションの詳細設定ダイアログボックスが開きます。

チェックアウトの取り消しオプションの詳細設定ダイアログボックス

[ローカルコピー]

チェックアウトを取り消したとき、作業フォルダにあるローカルコピーに行う処理を指定します。

[VSS 最新のバージョンを取得] ダイアログボックス



[データベースのパス]

VSSのデータを格納しているデータベースのパス。

[ユーザー名]

VSSにアクセスするときのユーザー名。

[対応する VSS プロジェクト]

Peggyが接続しようとしているVSSのプロジェクト名が表示されます。

[ファイル]

対象候補のファイルが表示されています。取得したいファイルのチェックボックスをオンにしてください。複数のファイルを選択することが出来ます。最初、チェックアウトしていないファイルはオンになっています。

[取得]

[ファイル]欄でチェックされているファイルの最新バージョンを取得します。

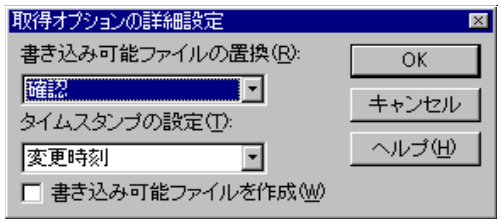
[すべて選択]

すべてのファイルを選択します。

[詳細設定]

オプションの詳細設定ダイアログボックスが開きます。

取得オプションの詳細設定ダイアログボックス



[書き込み可能ファイルの置換]

ファイルを取得するとき、既に作業フォルダに書き込み可能なローカルコピーがある場合の対処方法を設定します。

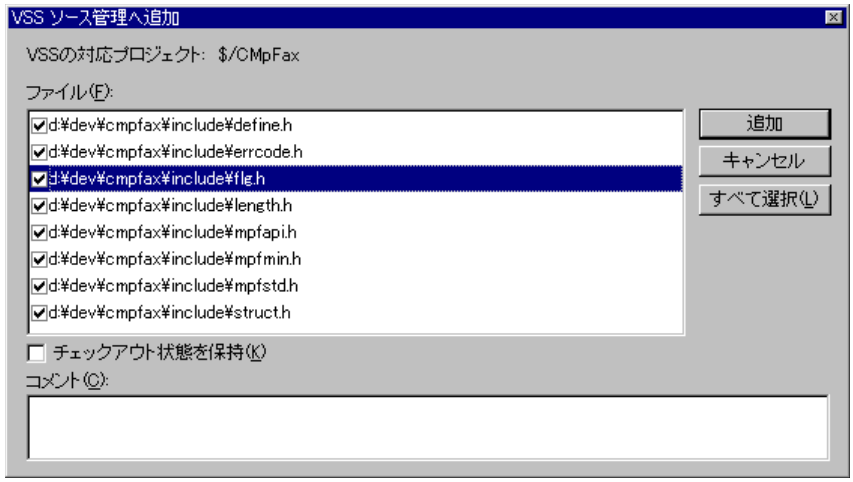
[タイムスタンプの設定]

取得したファイルに設定する日付と時刻を指定します。[既定値]、[現在時刻]、[変更時刻]、[チェックイン時刻] から選択できます。既定値はVSSのオプションの設定が使われます。

[書き込み可能ファイルを作成]

取り出したファイルを書き込み可能にします。

[VSS ソース管理へ追加] ダイアログボックス



[データベースのパス]

VSSのデータを格納しているデータベースのパス。

[ユーザー名]

VSSにアクセスするときのユーザー名。

[対応する VSS プロジェクト]

Peggyが接続しようとしているVSSのプロジェクト名が表示されます。このプロジェクトはすでに存在していなければなりません。このコマンドを実行したときエラーが表示されたら、VSSのプロジェクトが実在するか調べてください。

[ファイル]

対象候補のファイルが表示されています。VSSで管理したいファイルのチェックボックスをオンにしてください。複数のファイルを選択することが出来ます。最初、VSSで管理されていないファイルはオンになっています。

[チェックアウト状態を保持]

ファイルがVSSに追加された後、チェックアウトされます。

[コメント]

ソース管理へ追加した理由などコメントを入力できます。ファイルのプロパティを表示すると、このコメントが[全般]タブに表示されます。

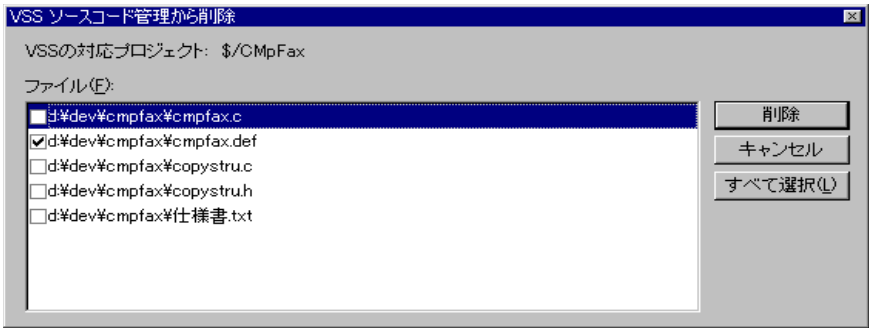
[追加]

[ファイル]欄でチェックされているファイルをソース管理に追加します。

[すべて選択]

すべてのファイルを選択します。

[VSS ソースコード管理から削除] ダイアログボックス



[データベースのパス]

VSSのデータを格納しているデータベースのパス。

[ユーザー名]

VSSにアクセスするときのユーザー名。

[対応する VSS プロジェクト]

Peggyが接続しようとしているVSSのプロジェクト名が表示されます。

[ファイル]

対象候補のファイルが表示されています。VSSの管理からはずしたいファイルのチェックボックスをオンにしてください。複数のファイルを選択することが出来ます。最初、VSSで管理されているファイルはオンになっています。

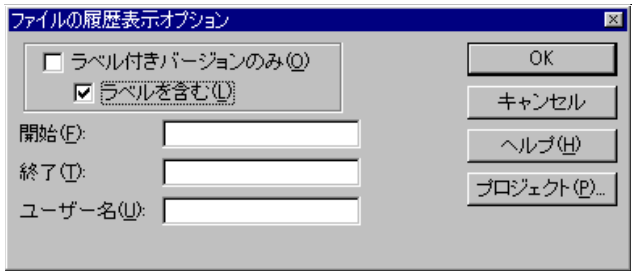
[削除]

[ファイル]欄でチェックされているファイルをソース管理から削除します。

[すべて選択]

すべてのファイルを選択します。

[VSS ファイルの履歴表示オプション] ダイアログボックス



[ラベル付きバージョンのみ]

ラベルが設定されているバージョンだけが[ファイルの履歴]ダイアログボックスに表示されます。

[開始]、[終了]

時刻、日付、バージョン、またはラベルを指定します。日時を指定するには先頭に"D"を付け、ラベルを指定するには先頭に"L"を付けます。たとえば、「D98/02/29;12:15」または「LBeta1」のように指定します。

[ユーザー名]

ユーザー名を指定すると、そのユーザーが操作したファイルの履歴だけが表示されます。

[プロジェクト]

[プロジェクトの履歴表示オプション]ダイアログボックスを表示します。なお、ここで言うプロジェクトとは、VSSのプロジェクトを指しています。このダイアログボックスのオプションは[履歴表示オプション]ダイアログボックスと基本的に同じですが、[ファイルの履歴も含む]と[サブプロジェクトも対象]の2つのチェックボックスが追加されています。[ファイルの履歴も含む]をオンにすると、[プロジェクトの履歴]ダイアログボックスに、プロジェクトの履歴情報のほかに、プロジェクト内のファイルの履歴情報も表示されます。[サブプロジェクトも対象]をオンにすると、指定したプロジェクトの下にあるすべてのサブプロジェクトとファイルの履歴が表示されます。オプションを設定して[OK]をクリックすると、プロジェクトの履歴が表示されます。

[VSS 相違点表示オプション] ダイアログボックス

[比較元のファイル]

比較するファイルの名前を入力します。[参照]をクリックしてファイルを探すこともできます。VSSプロジェクトまたは任意のWindowsフォルダでファイルを参照でき

ます。ファイルの特定のバージョンと比較するには、次のようにセミコロンとバージョン番号をファイル名に追加します。

HELP.C; 5

[比較先のファイル]

比較先のファイルの名前を入力します。[参照]をクリックしてファイルを探すこともできます。VSSプロジェクトまたは任意のWindowsフォルダでファイルを参照できます。

[形式]

[ビジュアル表示]を選ぶと、変更前と変更後の内容が左右に分けられ相違点が視覚的に表示されます。[ソースコード管理]を選ぶと、追加、削除、変更された行だけが表示されます。UNIX形式を使用するには[UNIX形式]を選択します。この形式では、変更した行だけが表示され、バージョンの新旧が示されます。

[スペース文字を無視]

オンにすると、スペース文字の相違点が無視されます。つまり、スペース、タブ、そのほかのスペース文字だけが異なる行は無視されます。

[大文字と小文字を区別しない]

大文字と小文字の相違点は無視されます。

[OK]

設定内容を確定します。

[レポート]

比較結果のレポートを作成し、クリップボード、プリンタ、またはファイルに出力します。

[プロジェクト]

[プロジェクトの相違点]ダイアログボックスを表示します。なお、ここで言うプロジェクトとは、VSSのプロジェクトを指しています。

[詳細設定]

オプションの詳細設定ダイアログボックスが開きます。

[OS の違いを無視]

オンにすると、改行文字などのオペレーティング システムによって異なるファイルの相違点が無視されます。

[設定を保存]

オンにすると、ここで指定した設定内容が保存され、次回の操作で既定値として使用されます。

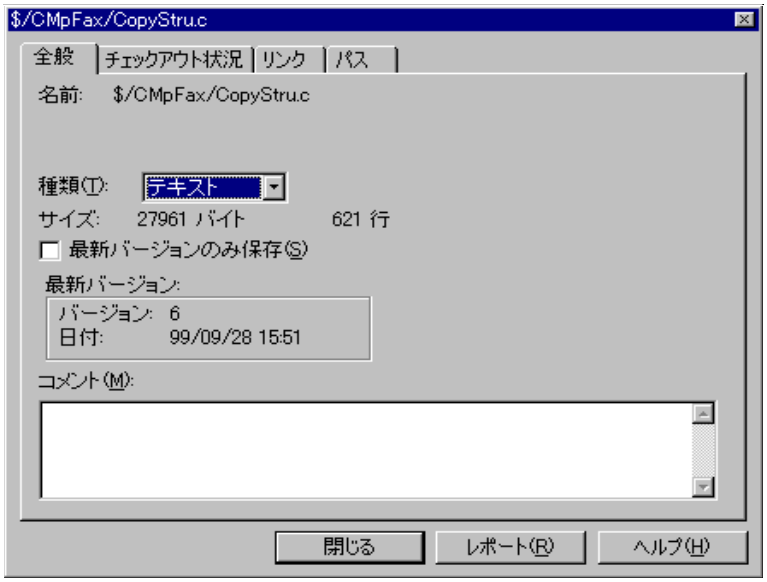
[コンテキストを表示]

オンにすると、相違点前後の情報が表示されます。つまり、変更のある行とその前後の行が表示されます。このオプションは、表示形式が[ビジュアル表示]の場合には無効で、[ソースコード管理]と[UNIX形式]の場合だけに有効です。

[行数]

[コンテキストを表示]をオンにしたときに、相違点のある行の前後に表示する行数を指定します。

[VSS プロパティ] ダイアログボックス



[全般] タブ

ファイルの名前、種類、サイズ、最新のバージョン番号、最終更新日時、およびコメントなどが表示され、設定し直すこともできます。

[チェックアウト状況] タブ

ファイルがチェックアウトされている場合は、ファイルの名前、チェックアウトしているユーザー名、排他チェックアウトの有無、チェックアウトの日時、カレントバージョン、チェックアウトフォルダ(およびコンピュータ名)、ファイルが属しているプロジェクト名、およびコメントが表示されます。コメントはここで変更できます。

[リンク] タブ

ファイルの共有リンクの一覧（そのファイルを共有しているプロジェクトの一覧）が表示されます。

[パス] タブ

分岐ファイルを持つプロジェクト（そのファイルを共有したことのあるプロジェクト）の一覧が表示されます。

第 9 章　DB ツール

このリファレンスでは、DBツールに関するコマンドを分類し、そのコマンド一覧とより詳細な使い方を記載しています。

DB ツールとは

OracleあるいはODBCを介して各種データベースと接続し、SQLデータベース言語や、PL/SQL言語（プロシジャ・ファンクション・パッケージ・トリガー）のプログラムを容易に作成・実行することができます。ファイル全体の実行はもとより、範囲を指定しての実行も可能です。クエリー結果はアウトプットウインドウに表示します。リダイレクト指定とセパレータ機能で、データのファイル化も可能です。実行文にエラーがあった場合は、エラー行へのタグジャンプを行うことができます。

Oracle クライアントのインストール

OracleのSQL*NET V2以上がインストール済みでOracleデータベースサーバーと接続可能な状態であることが条件です。Oracleクライアントのインストレーションに関してはOracleのマニュアルを参照してください。

ODBC ドライバとデータソース

データベースにODBCを介して接続するには、そのデータベースのODBCドライバがインストールされている必要があります。さらに、データソースの設定をしなければなりません。データソースの設定は「ODBCアドミニストレータ」を使い、行ってください。

DB ツールの設定

[ ]ボタン、またはメニューの[ツール]-[DBツール]-[設定]を選択して[DBツール設定]ダイアログボックスを表示させます。登録するDBツール番号を選択します。1から8を選択できます。DBツールのボタンイメージは、[カスタマイズ]ダイアログボックス[スクリプト]ページから一括してカスタマイズすることもできます。

[ボタン]

[変更]ボタンをクリックすると[ボタンギャラリー]ダイアログボックスが表示され、DBツールのボタンイメージを変更します。[戻す]ボタンをクリックすると、元のイメージへ戻ります。

[ツールタイトル]

ツールタイトル名を入力します。接続先やユーザ等の個々の設定を、最大8個まで設定が可能です。ツールタイトルはボタン[]等のツールチップと[ツール]-[DBツール]で登録されているDBツールが表示されます。設定を識別し、解りやすい名前を付けてください。

[DB タイプ]

データベースに接続する方法を指定します。

タイプ	意味
[Oracle]	Oracleクライアントを通じてOracleに接続します。
[ODBC]	ODBCを介してデータベースに接続します。

[TNS 名]

Oracleのサービス名を入力します。ODBC接続の場合はデータソース名を入力します。

[ユーザ名]

データベースに接続するためのユーザ名を入力します。ODBC接続の場合、データベースの種類によっては必ずしも必要ではありません。

[パスワード]

データベースに接続するためのパスワードを入力します。ODBC接続の場合、データベースの種類によっては必ずしも必要ではありません。

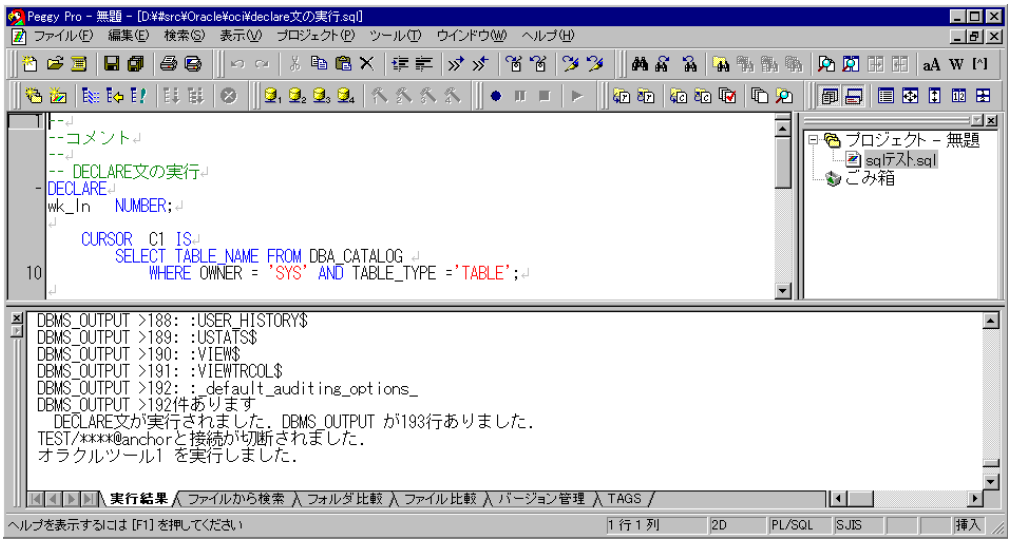
[DBMS_OUTPUT を表示する]

チェックすると、プロシジャー等のDBMS_OUTPUT文を表示します。

デフォルトではチェックです。

【例】DBMS_OUTPUT.PUT_LINE　文が実行されるとアウトプットウインドウに、"DBMS_OUTPUT>"の後、値を表示します。

```
FOR CURSOR  IN C1 LOOP
    DBMS_OUTPUT.PUT_LINE(TABLE_rec.TABLE_NAME );
END LOOP;
```



[実行終了時にコミットする]

チェックすると、UPDATE文、EDIT文、DELETE文実行後にSQL文の終了または、エラーにより中断した場合、直前の値までをコミットし終了します。デフォルトではロールバックします。

[セパレータ]

チェックをして、セパレータ文字を入力すると列名およびデータ間に指定されたセパレータを挿入します。セパレータを指定された場合、列名はデータ長に影響されずにフルスペルで書き出します。タブは(¥t)です。

[リダイレクト]

チェックをして、ファイル名を指定すると、列名とデータをファイルにリダイレクトします。メッセージ等はリダイレクトされません。

【例】リダイレクトファイルに<c:¥test.csv>を入力します。セパレータにコンマ(,)をいれます。実行後にPeggyのプロジェクトに<c:¥test.csv>を追加し、関連アプリケーション起動させると、CSVファイルがMicrosoft Excelに関連づけられている場合は、Excelが起動します。

［オプション］

チェックをしてオプション追加することで下記機能が有効となります。

オプション	機能
-m	メッセージを表示しません。デフォルトでは表示します。
-b　xxxxx	DBMS_OUTPUTのバッファー値を設定します。デフォルト値は20000です。
-n	列名かデータ長か長い方が優先されます。デフォルトではデータ長が優先されますので列名が表示されなかったり、途中で切れたりすることがあります。
-row　xxx	select文の先頭XXX行を表示します。
-t	経過時間を表示します。
-kt　xxx	カラムの型が下記の場合の、表示文字数を指定します。指定がないと24桁になります。 Microsoft　SQL　Serverのtext型・ntext型、Microsoft　Accessのメモ型、 Oracleのlong型、Microsoft　Excelの文字列(その列に256文字を超えるセルがある時)
-2	ODBC接続で列の型が下記の場合、表示する長さを２倍にします。Access,Excelなどのデータを読むとき、全角文字のデータがすべて表示されない問題を回避します。 Microsoft　SQL　Serverのvarchar型、Microsoft　Accessのテキスト型、 OracleのVARCHAR2型、Microsoft　Excelの文字列
-trn	データをすべて表示できずに切りつめたときのメッセージを表示しません。
-tro	同じくデータを切りつめたときのメッセージを、１つのSQL文の処理後に何カ所あったかだけを表示します。

DB ツールボタンをツールバーに追加する。

【参照】『第11章 カスタマイズ/ツールバー /ツールボタンの追加と削除』

マークがDBツールボタンに当ります。

拡張子・言語に付いて

[ファイル]プロパティページの言語設定をSQL/92またはPL/SQLに指定してください。デフォルトの場合は、[ツール]-[オプション]-[拡張子]の拡張子SQL言語により判断されます。SQL/92、PL/SQL以外が設定されていた場合は、ファイル全体がSQLとして判断され実行します。領域指定での実行は、実行時にSQL・PL/SQLの判断を行いますので拡張子には影響されません。

DB ツールの実行

DBツールボタン[ ]ボタンをクリックする。

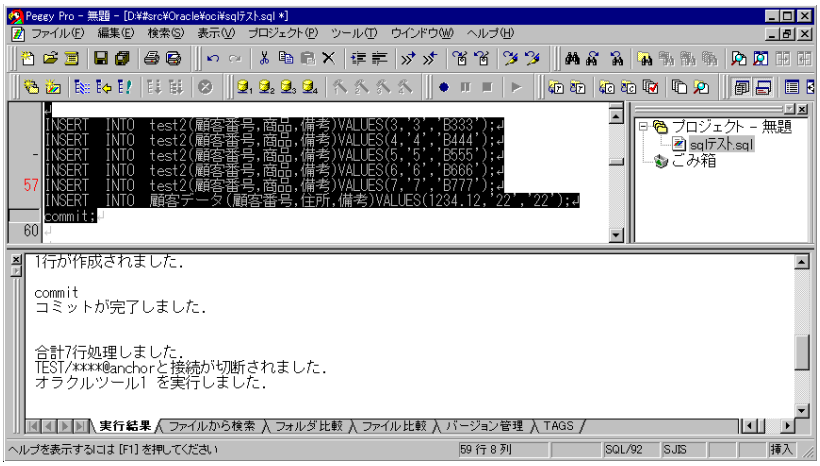
または[ツール]-[DBツール]-[１]から[８]を選択すると実行します。

【注意】SQL文は、１つの文は16Kバイト、また１Kステップに制限されています。

【注意】PL/SQLはサイズ制限はありませんが、Oracle 7は64Kバイトの制限があります。

範囲指定での実行

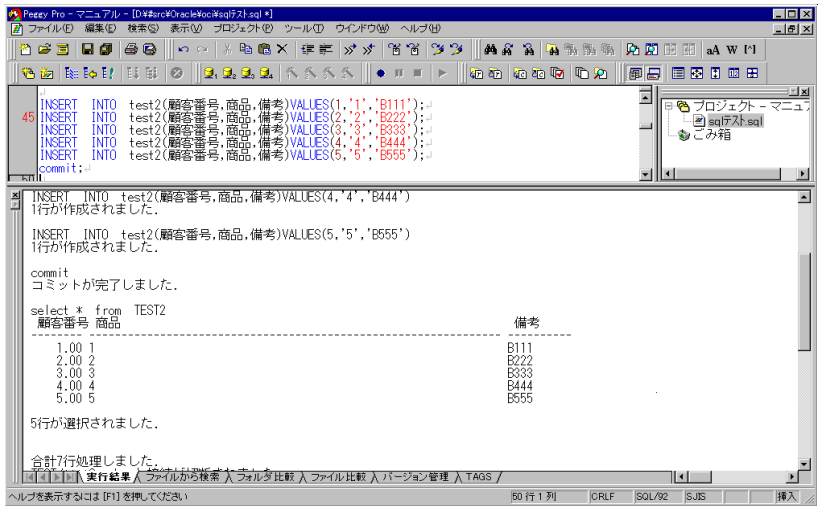
実行させたいSQL文を範囲指定し、DBツール[ ]ボタンをクリックする。または[ツール]-[DBツール]-[1]から[8]を選択すると実行します。



ファイル全体の実行

範囲指定をしないでDBツール[ ]ボタンをクリックする。または、[ツール]-[DBツール]-[1]から[8]を選択すると実行します。

【例】INSERT文・COMMIT文・SELECT文をまとめて実行したものです。



実行結果

実行結果は500列・5000行まで表示可能です。

これを超えるデータは、表示しません（デフォルト）。

500列を超えたデータを表示するには、実行結果ウィンドウで右クリックしポップアップメニューから[ウインドウへ保存]を選んでください。内容が編集ウインドウへ保存されます。

編集ウインドウでは、表示されなかったデータも表示されます。

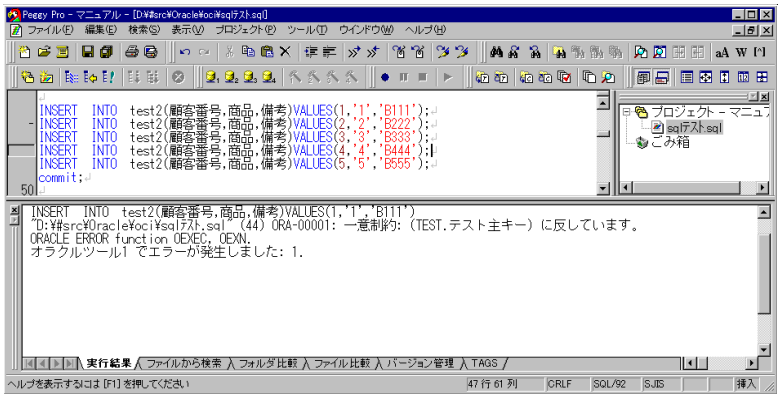
デフォルト値は5000行です。

変更するには[ツール]-[オプション]-[プロジェクト] -[実行結果ページの最大行数を制限する]を変更してください。

タグジャンプ

実行結果にエラーが表示された場合は、実行結果のエラーメッセージをクリックするとエラー行へジャンプします。

【例】INSERT文で一意制約エラーが起こった例です。



実行結果のエラーメッセージ行をクリックすると44行(エラーが起こったINSERT文)へジャンプします。保存されていないファイルを範囲指定して、実行させることも可能ですが、エラー行へのタグジャンプはできません。編集中のファイルを保存せずに実行させると以前に保存された文を実行し、正確な値(行)が表示されません。編集中のファイルが壊れることはありません。[ツール]-[オプション]-[編集]-[実行前保存]を[保存する]に設定することをお勧めします。

文法の注意点

●/*　*/　の間はコメント文とします。

●-- で始まる行は、改行までコメント文とします。

●REMで始まる行は、改行までコメント文とします。

●SQL文に関しては、文最後に；をつけてください。

●SQL文とPL/SQLの混在はできません。

●SQL文の最後は；です。

●PL/SQLの最後の/は不要です。あっても問題ありません。

●EXIT　ここでSQL文の実行を終了します。

PL/SQL に関して

PL/SQL の実行

例	SQL*Plus	Peggy
バインド変数の設定	VAR R1 NUMBER;	VAR R1 NUMBER;

実行	EXEC PL_TEST1(5,:R1)	EXEC PL_TEST1(5,:R1) または begin PL_TEST1(5,:R1);end;
バインド変数の表示	PRINT R1	なし (実行結果に自動表示)

SQL*PLUSとほぼ同じですが、print文は必要ありません。実行結果に表示されます。

DBMS_OUTPUTもデフォルトで実行結果に表示されますが、オプション指定する事で表示をしないようにできます。

Function の実行

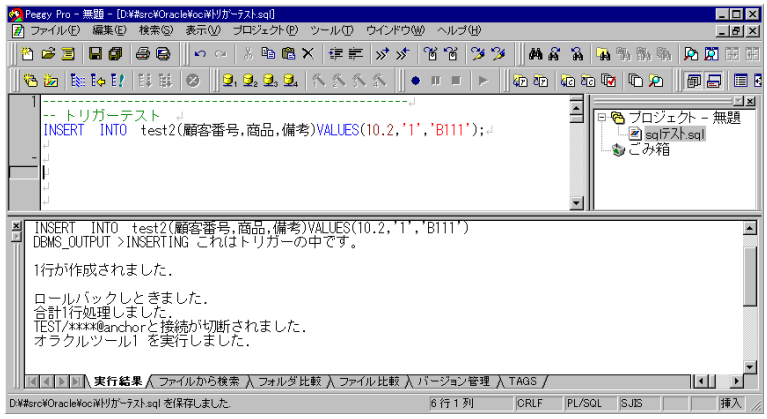
select xxxx from DUAL;　で実行します。

ファンクションの中にDBMS_OUTPUTが存在した場合は表示します。

TRIGGER について

TRIGGER中のDBMS_OUTPUT文があった場合はデフォルトで実行結果に表示されます。

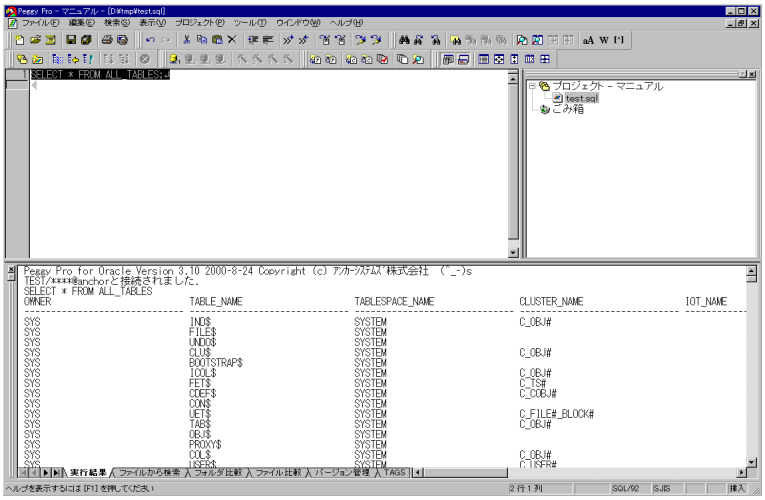
【例】INSERT文によりTRIGGERが動き、TRIGGER中DBMS_OUTPUTが、表示された例です。



応用例

その他サンプル接続および実行まで

[ツール]-[DBツール]-[設定]を選択して[DBツール設定]を表示させます。DBツールを「１」、ツールタイトルに「テスト」、TNS名・ユーザー名・パスワードを入力してOKボタンを押します。これで[ ]が有効になります。編集ウインドウに　select * from ALL_TABLES ;　と入力します。編集ウインドウの入力した部分を選択して[ ]を押します。実行結果に、クエリー結果が表示されます。



SQL文をファイルに保存する場合は、拡張子をSQLとして保存します。プロジェクトに追加します。プロジェクトウインドウの追加したファイルを右クリックし、プロパティの言語をSQL/92に変更します。これでファイル全体がSQL言語モードとなります。編集ウインドウを選択しないで[ ]を押します。ファイル全体を実行した場合で、エラーが発生する場合は、プロパティの言語を調べてください。

第 10 章　PHP プログラムのデバッグ

【注意】PHPプログラムのデバッグ機能が利用できるのは、Peggy Pro for PHPだけです。

PHP プログラムのデバッグ

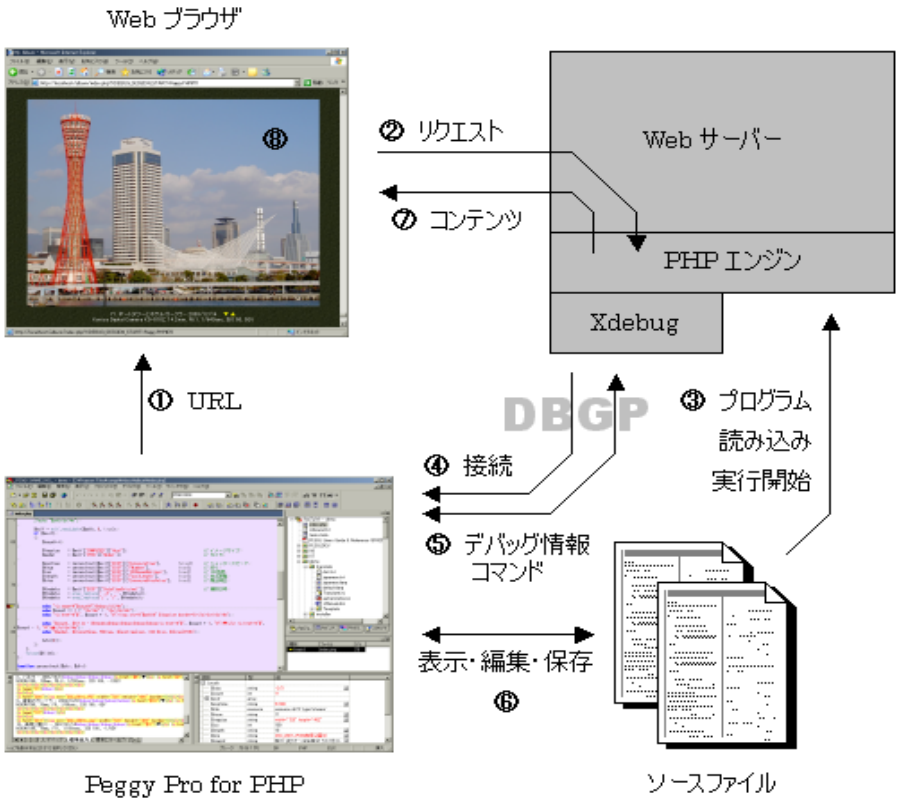
Peggy Pro for PHPでは、PHPプログラムのリモートデバッグ機能を搭載しました。以下のデバッグ機能により、Webサーバー上で実行されるPHPプログラムの不具合点を簡単に見つけることができます。

- ステップ実行(ステップイン、ステップオーバー、ステップアウト)
- カーソル位置まで実行
- ブレークポイントの設定と一時無効化
- コールスタック表示と切り替え
- 変数(ローカル変数・グローバル変数)表示と変数値の変更
- 変数値のチップ表示
- 軽快な動作
- 各種漢字コードを考慮した日本語対応

デバッグの概要

Peggy Pro for PHPは、以下のような構成と順序でPHPプログラムのデバッグを行います。

1. Peggy Pro for PHP で開いている PHP ソースファイルに対応する URL を作成し Web ブラウザを起動します。
2. Web ブラウザから Web サーバーへリクエストが送られます。
3. Web サーバーが PHP プログラム読み込み起動します。
4. PHP に組み込まれた拡張モジュール Xdebug が Peggy Pro for PHP へ接続します。
5. Xdebug と Peggy Pro for PHP でソースファイル、行番号、変数、ブレークポイントなどデバッグに必要な情報のやり取りが始まります。
6. Peggy Pro for PHP の画面にデバッグ情報が表示されます。
7. PHP プログラムの実行によりコンテンツが作成されて Web サーバーへ送り返されます。
8. Web ブラウザにページが表示されます。



【メモ】複雑そうに見えますが、設定後はボタン1つでデバッグを開始できます。

【メモ】DBGPは、PHPなどの言語エンジンとデバッガUI間の通信を定義した共通プロトコルです。詳細は、『DBGP - A common debugger protocol for languages and debugger UI communication (英文)』http://www.xdebug.org/docs-dbgp.phpを参照してください。

【注意】PHPソースファイルは、Webサーバー上に置かれます。Peggy Pro for PHPのデバッグでソースファイルを表示しながらステップ実行やブレークポイントを設定するには、Peggy Pro for PHPを実行しているコンピュータからWebサーバー上のPHPソースファイルを参照できる必要があります。
両方が同じコンピュータ上で実行されている場合は何の問題もありませんが、異なるコンピュータであれば、ファイル共有などでアクセスできるようにしてください。

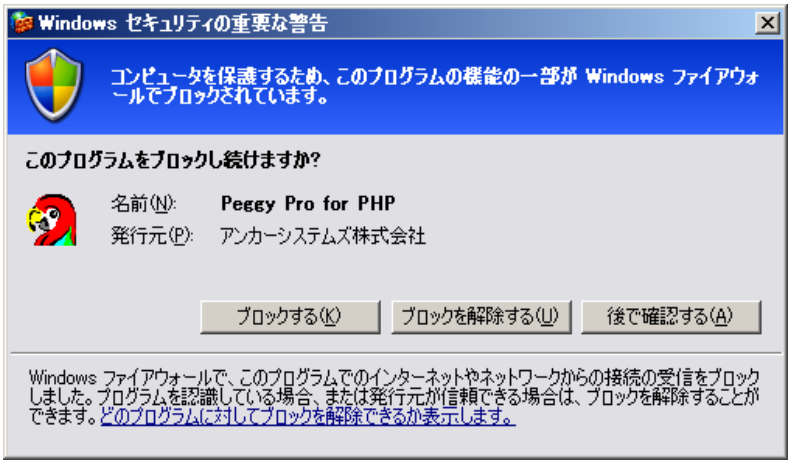
デバッグの準備

PHPプログラムのデバッグは、Peggy Pro for PHP、Webブラウザ、Webサーバーが連携してはじめて可能になります。お使いの環境に合わせて、それぞれ必要な設定をしてください。

【ヒント】Apache + PHP + MySQL = XAMPP
最初は、Peggy Pro for PHP、Webブラウザ、Webサーバーをすべて同じコンピュータ上で試すのが簡単・確実です。Apache Friendsによって開発・提供されているXAMPPを使

うと、とても簡単にApache + PHP + MySQL環境をインストールできます。
【参照】http://www.apachefriends.org/jp/

【重要】Windowsファイヤウォールからの警告
Peggy Pro for PHPはポートを開き、Webサーバー上のPHPからのデバッグのための接続を待ち受けます。そのためWindows XP SP2、Windows Vistaでは、最初にデバッグしようとしたときWindowsファイヤーウォールから以下のような警告が出ます。[ブロックを解除する]をクリックし、Windowsファイヤーウォールを通過できるようにしてください。



Web サーバーの設定

PHP 拡張モジュール Xdebug の組み込み

PHPプログラムのデバッグを行うには、サーバーに専用のPHP拡張モジュールXdebugを追加する必要があります。Xdebugは、Derick Rethansさんによって開発、無償で配布されているソフトウェアです。詳しくは、ホームページ XDEBUG EXTENSION FOR PHP を参照してください。

【参照】http://xdebug.org/

```
The Xdebug License, version 1.01
(Based on "The PHP License", version 3.0)
Copyright (c) 2002, 2003, 2004 Derick Rethans. All rights reserved.
```

以下の手順に従って拡張モジュールを追加してください。

1. Xdebug 拡張モジュールのダウンロード

使用しているPHPのバージョンに合った<php_xdebug.dll>をダウンロードし、PHPの拡張モジュールフォルダに保存します。

Xdebugの情報・ソースファイルやコンパイル済みのモジュールは、以下のサイトからダウンロードできます。

Xdebugの情報・ソースファイル	http://www.xdebug.org/
Windows版	http://www.xdebug.org/ 右側にあるメニューの Windows modules: から使用しているPHPバージョンに合ったものをダウンロード
Linux版	http://pecl.php.net/package/Xdebug

【注意】xdebug 2.1には対応していません。xdebug 2.0.xを使ってください。

2. <php.ini> ファイルの編集

WebサーバーのPHP初期化ファイル<php.ini>を編集し、[Zend]セクション(ファイルの終わり近く)に以下の太字で示した2行を追加します。Xdebug拡張モジュール<php_xdebug.dll>へのパスは、保存したフォルダに合わせて修正してください。

```
[Zend]
;zend_extension_manager.optimizer_ts =
                "C:¥Program Files¥xampp¥php¥zendOptimizer¥lib¥Optimizer"
;zend_extension_ts =
 "C:¥Program Files¥xampp¥php¥zendOptimizer¥lib¥ZendExtensionManager.dll"
;zend_optimizer.enable_loader = 1
;zend_optimizer.optimization_level=15
;zend_optimizer.license_path =
zend_extension_ts = "C:¥Program Files¥xampp¥php¥ext¥php_xdebug.dll"
xdebug.remote_enable = On
xdebug.remote_host = workstationname
xdebug.remote_port = 9000
```

【ヒント】システムによっては複数の<php.ini>ファイルがあり、どれを修正すべきか迷うことがあります。そのようなときは、phpinfo()を表示させると、初期設定に使った<php.ini>のフルパス表示されるので、簡単に確認できます。

System	Windows NT BALI 5.1 build 2600
Build Date	Nov 2 2006 11:50:55
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Program Files\xampp\apache\bin\php.ini

斜体で示した部分は、必要に応じて追加します。WebサーバーとPeggy Pro for PHPが異なるコンピュータ上で実行されているときは、xdebug.remote_hostにPeggy Pro for PHPを実行するコンピュータ名を指定します。

また、拡張モジュールXdebugとPeggy Pro for PHP間を9000以外のポートを使って接続したいときは、xdebug.remote_portを指定してください。同じコンピュータ上でポート9000を使って接続する場合はこれらの行を追加する必要はありません。

【ヒント】Peggy Pro for PHP側のポート番号は、[オプション設定]ダイアログボックスの[デバッグ]ページで変更できます。

【注意】オペレーティングシステムやPHPのバージョンによっては、zend_extension_ts = ... ではなく、zend_extension = ... を使うものもあるようです。
【例】zend_extension = /opt/lampp/lib/php/extensions/xdebug.so

【注意】Zend Optimizerとの競合(<php.ini>ファイルのアンダーライン行)
Xdebugは、Zend Optimizerと同時に使えないことがあります。Xdebugが正しく組み込めない場合、以下のようにZend Optimizerの読み込み行をコメント化してください。

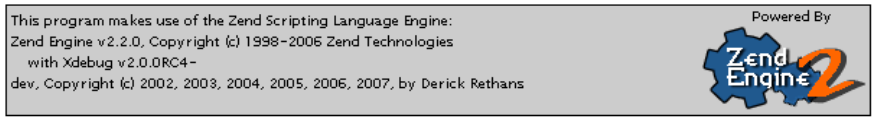
```
zend_extension_manager.optimizer_ts = "C:¥xampp¥php¥zendOptimizer¥lib¥Optimizer"
                    ↓
;zend_extension_manager.optimizer_ts = "C:¥xampp¥php¥zendOptimizer¥lib¥Optimizer"
```

3. Web サーバーの再起動

Webサーバーを再起動します。以上で、拡張モジュールXdebugがPHPに組み込まれました。

4. PHP 拡張モジュール Xdebug 組み込みの確認

phpinfo()を表示させ、拡張モジュールが正しく組み込まれていることを確認してください。下図のように、Zend Technologies社の著作権表示に続いてXdebugモジュールの著作権が表示されていれば、正常に組み込まれています。



5. リモートデバッグ設定の確認

同じ phpinfo() 画面を下にスクロールさせ、Xdebug セクションのxdebug.remote_enableがOnになっていることを確認します。<php.ini>のxdebug.remote_hostとxdebug.remote_portを設定しているときは、枠の部分も確認してください。

xdebug

xdebug support	enabled
Version	2.0.0RC4-dev

Supported protocols	Revision
DBGp - Common DeBuGger Protocol	$Revision: 1.116 $
GDB - GNU Debugger protocol	$Revision: 1.85 $
PHP3 - PHP 3 Debugger protocol	$Revision: 1.22 $

Directive	Local Value	Master Value
xdebug.auto_trace	Off	Off
.....		
xdebug.remote_autostart	Off	Off
xdebug.remote_enable	On	On
xdebug.remote_handler	dbgp	dbgp
xdebug.remote_host	localhost	localhost
xdebug.remote_log	*no value*	*no value*
xdebug.remote_mode	req	req
xdebug.remote_port	9000	9000
xdebug.show_exception_trace	Off	Off
xdebug.show_local_vars	Off	Off
xdebug.show_mem_delta	Off	Off
xdebug.trace_format	0	0
xdebug.trace_options	0	0
xdebug.trace_output_dir	/tmp	/tmp
xdebug.trace_output_name	crc32	crc32
xdebug.var_display_max_children	128	128
xdebug.var_display_max_data	512	512
xdebug.var_display_max_depth	3	3

以上でサーバーの設定と確認は完了です。

Peggy Pro for PHP の設定

ソースファイルへのアクセス

Peggy Pro for PHPからWebサーバーに置かれたPHPプログラムのソースファイルへアクセスする方法が必要です。Peggy Pro for PHPとWebサーバーが同じコンピュータ上で実行されているときは、特別な設定は必要ありません。別々のコンピュータ上にあるときは、Webサーバー側でファイル共有を設定し、Peggy Pro for PHPを実行しているコンピュータからアクセスできるようにしてください。

パス・URL の設定

Peggy Pro for PHPが、デバッグしようとしているPHPプログラムのソースファイルパスとURLの関係を理解し、相互に変換できるように設定します。具体的には、以

下の3つの項目を設定します。

1. Web ブラウザからみた URL
2. Web サーバーの PHP エンジンからみたソースファイルへのパス
3. Peggy Pro for PHP からソースファイルへアクセスするためのパス

これらは、オプション設定ダイアログボックスの[URL・パス変換]で設定します。

【ヒント】Peggy Pro for PHPとWebサーバーが同じコンピュータ上で実行されているときは、2.と3.は同じになります。

設定例

メニューから[ツール]-[オプション]を選択し、[オプション設定]ダイアログボックスの[デバッグ]ページをクリックします。[URL・パス変換]へ3つの項目を設定します。

PeggyとWebサーバーが同じコンピュータ上にあるときは、以下のように指定します。

【例1】

URLルート:	http://localhost
サーバールート:	(空欄)
ローカルルート:	C:¥Program Files¥xampp¥htdocs

異なるWindowsコンピュータ上にあるときは、以下のように指定します。

【例2】

URLルート:	http://testserver
サーバールート:	C:¥Program Files¥xampp¥htdocs
ローカルルート:	¥¥testserver¥C$¥Program Files¥xampp¥htdocs

異なるLinuxコンピュータ上にあるときは、以下のように指定します。(Linux版XAMPPを使い、Sambaによりサーバーのドキュメントルートを WindowsのZドライブとしてファイル共有をしている例です)

【例3】

URLルート:	http://testserver
サーバールート:	/opt/lampp/htdocs
ローカルルート:	Z:¥

【ヒント】URL・パス変換は、複数設定できます。

【ヒント】URL・パス変換は、プロジェクトのプロパティにも設定できます。

【参照】『第6章 プロジェクト機能／プロパティ／[PHPデバッガの設定]ダイアログボックス

【参照】『第11章 カスタマイズ／[デバッグ]ページ』

デバッグ開始

デバッグの開始方法

以下の方法によりデバッグを開始できます。

1. **ソースファイルを開いてからデバッグ開始**
   [実行]、[ステップイン]、[ステップオーバー]、[カーソル位置まで実行] などのボタンをクリックするだけで、簡単にデバッグを開始できます。
2. **カーソルをデバッグしたい PHP プログラムの URL 上に置いてデバッグ開始**
   デバッグしたいページの URL 上にカーソルを置いて、[実行]、[ステップイン]、[ステップオーバー] ボタンをクリックします。
3. **Web ブラウザにデバッグしたいページの URL を直接入力**
   URL の後に、?XDEBUG_SESSION_START=PeggyPHP を付け加え、このページに対してデバッグしたいことを指定します。この方法は、Web ブラウザが別のコンピュータ上にあってもかまいません。
   【例】http://localhost/sample/album.php?XDEBUG_SESSION_START=PeggyPHP

【注意】[実行]、[ステップイン]、[ステップオーバー]、[カーソル位置まで実行]ボタンでデバッグを開始するには、簡単スタートオプションがON(デフォルトでON)になっている必要があります。簡単スタートオプションは、[オプション設定]ダイアログボックスの[デバッグ]ページにあります。

【注意】[実行]ボタンによりデバッグを開始した場合、ブレークポイントを設定していないとそのまま最後まで実行してしまいます。止めたい行にブレークポイントを設定してから[実行]ボタンをクリックしてください。

【注意】イコール(=)の後のPeggyPHPはデバッグセッションの識別文字列です。PeggyPHP**以外の文字列でもかまいません。**

【注意】Webブラウザに直接URLを入力してデバッグを開始するには、[オプション設定]ダイアログボックス[デバッグページページにある[簡単スタートまでポートを開かない]オプションをOFFにし、Peggy Pro for PHPの起動と同時にポートを開くようにする必要があります。

Web ブラウザのキャッシュ

PHPプログラムのデバッグは、リクエストがWebブラウザからWebサーバーへ送られることで開始されます。

しかしWebブラウザには、過去に表示したページをキャッシュする機能があり、このキャッシュにヒットするとWebサーバーへリクエストが送られません。

デバッグ中にキャッシュからヒットしないようにするには、Webブラウザのオプションを変更します。インターネットエクスプローラの場合、[インターネットオプション]の[全般]ページ、[インターネット一時ファイル]の[設定]ボタンをクリックし、[保存しているページの新しいバージョンの確認]から[ページを表示するごとに確認する]を選択してください。

キャッシュから表示されてしまったら Ctrl+[更新]

キャッシュにヒットしWebブラウザへリクエストが送られなかったときは、Ctrlキーを押しながらWebブラウザの[更新]ボタンをクリックしてください。必ずリクエストが送られます。

【ヒント】<Peggyインストールフォルダ¥sample>にあるサンプルPHPプログラムをWebサーバーのドキュメントルートコピーすることで、簡単に試すことができます。

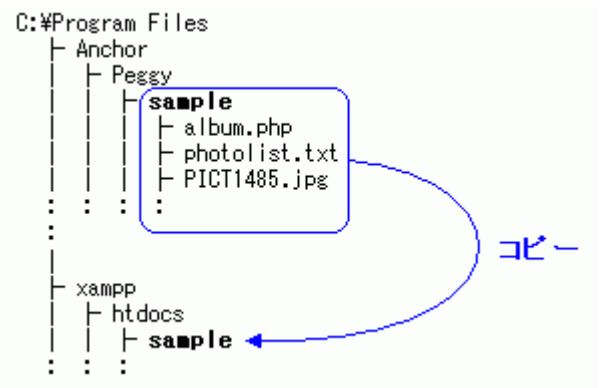
【メモ】デバッグ開始時には、MocaScriptのEvent.onDebugBegin()が呼び出されます。

サンプルプログラム

Peggy Pro for PHPをインストールしたフォルダの<sample>にPHPプログラムのサンプルがあります。

サンプルプログラムのコピー

サンプルプログラムを試すには、これらのファイルを<sample>フォルダごとWebサーバーのドキュメントルートにコピーしてください。下図の例では、Peggy Pro for PHPのサンプルプログラムを、同じコンピュータにインストールされているXAMPPのドキュメントルートにコピーしています。



【注意】サンプルプログラムの実行には、PHPの拡張モジュール<php_exif.dll>が必要です。<php.ini>ファイルを検索し、extension=php_exif.dllの行がコメントになっている場合は、先頭のセミコロンを削除し、モジュールをロードするように修正してください。修正後はWebサーバーの再起動が必要です。

```
;extension=php_exif.dll
        ↓
extension=php_exif.dll
```

デバッグ設定の確認

PHPプログラムをデバッグするには、[オプション設定]ダイアログボックスの[デバッグ]ページで、ソースファイルとURLの関係を指定してやる必要があります。

以下は、Peggy Pro for PHPとXAMPP(Apache+PHP+MySQL)が同じコンピュータにインストールされている場合の設定例です。

【例】

URLルート:	http://localhost
ローカルルート:	C:¥Program Files¥xampp¥htdocs
サーバールート:	(空欄)

ソースファイルを開く

デバッグしたいPHPプログラムのソースファイルを開きます。上記の例では、<C:¥Program Files¥xampp¥htdocs¥sample¥album.php>を開きます。

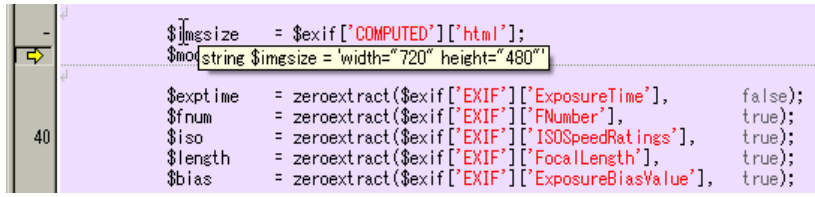
デバッグ開始

[ステップイン]ボタンをクリックします。デバッグが開始され、ステータスバーに「開始」と表示されます。また、カーソルがファイルの先頭に移動します。

続いて[ステップイン]ボタンをクリックすると、次に実行する文を指す黄色い矢印[]が順次移動します。

チップ表示

PHPプログラムの実行がブレークしているとき、変数の上にカーソルを移動させると、変数の型と値を表示します。変数値が簡単に確認できます。



標準出力のリダイレクト

PHPプログラムからの出力は、Webブラウザへ送られると同時にPeggy Pro for PHPのアウトプットウインドウの[標準出力]ページへリダイレクトさせることができます。これによりPHPプログラムが生成しているHTMLを簡単に確認できます。

【ヒント】リダイレクトは、[オプション設定]ダイアログボックスの[デバッグ]ページにある[リダイレクト]でON/OFFできます。アウトプットウインドウの[標準出力]ページのポップアップメニューでも切り替えられます。同様に漢字コードもポップアップメニューから選択可能です。

コマンドライン版の PHP プログラムのデバッグ

Webサーバーで実行されるPHPプログラムだけでなく、コマンドラインから実行したPHPプログラムもデバッグできます。Webサーバーなどを用意することなく、手軽にデバッグできます。

まず、以下の手順に従ってコマンドライン版デバッグの環境を設定してください。

1. Xdebug 拡張モジュールのダウンロード

Xdebug拡張モジュールをまだダウンロードしていないときは、Webサーバーの設定と同じ方法でダウンロードし、PHPの拡張モジュールフォルダに保存してください。

2. <php.ini> ファイルの編集

PHPの初期設定ファイル<php.ini>を編集し、拡張モジュールXdebugを組み込みます。

このファイルを開き、[Zend]セクション(ファイルの終わり近く)に以下のゴシック体太字で示した1行を追加します。<php_xdebug.dll>へのパスは、保存したフォルダに合わせて修正してください。

```
[Zend]
;zend_extension_manager.optimizer_ts =
               "C:¥Program Files¥xampp¥php¥zendOptimizer¥lib¥Optimizer"
;zend_extension_ts =
 "C:¥Program Files¥xampp¥php¥zendOptimizer¥lib¥ZendExtensionManager.dll"
;zend_optimizer.enable_loader = 1
;zend_optimizer.optimization_level=15
;zend_optimizer.license_path =
zend_extension_ts = "C:¥Program Files¥xampp¥php¥ext¥php_xdebug.dll"
```

【ヒント】コマンドライン版用の<php.ini>は、Webサーバー上で動作するPHP用とは別の場所にあります。通常は、コマンドライン版PHPの実行ファイル<php.exe>と同じフォルダにあります。

3. XDEBUG_CONFIG 環境変数の設定

コマンドラインから以下のように環境変数を設定します。この環境変数により、PHPとXdebug拡張モジュールはデバッグ用の実行であると判断し、デバッグセッションを開始しようと試みます。

```
D:¥test>set XDEBUG_CONFIG="idekey=PeggyPHP remote_enable=1"
```

【メモ】remote_host=myworkstationを設定すれば、<php.exe>とPeggy Pro for PHPが別のコンピュータ上で実行・デバッグできます。また、remote_port=XXXXにより使用するポートを変更できます。デフォルトホストはlocalhost、ポート番号は9000です。

【メモ】remote_host、remote_enable、remote_portは、コマンドライン版<php.exe>が読み込む初期化ファイル<php.ini>でも設定できます。

【メモ】idekey= の後のPeggyPHPは、デバッグセッションの識別文字列です。PeggyPHP以外の文字列でもかまいません。

4. Peggy Pro for PHP のオプション変更

オプションを変更して、Peggy Pro for PHPを起動すると同時にPHP/Xdebugからの接続を待ち受けるポートを開くようにします。[オプション設定]ダイアログボックス[デバッグ]ページの[簡単スタートまでポートを開かない]をOFFにしてください。

5. 接続待ちの確認

このオプションを有効にするため、Peggy Pro for PHPを再起動します。ステータスバーに[接続待ち]と表示されていることを確認してください。

以上で準備完了です。

6. 実行

あとはコマンドライン版<php.exe>を実行するだけです。例えば、<? phpinfo(); ?> 1行だけのファイル<info.php>を実行します。

```
D:¥test>type php.info
<? phpinfo(); ?>

D:¥test>php php.info
```

起動すると、Peggy Pro for PHPと接続しデバッグが開始されます。

デバッグコマンド

PHPプログラムのデバッグを行うコマンドです。[デバッグ]メニューには、以下のコマンドがあります。

ボタン	メニュー操作	キー操作	機能
	[デバッグ]-[実行]	F5	デバッグ中のプログラムを実行します。
	[デバッグ]-[中止]	Shift+F5	デバッグを中止します。
	[デバッグ]-[ステップイン]		ステップイン実行します。
	[デバッグ]-[ステップオーバー]		ステップオーバー実行します。
	[デバッグ]-[ステップアウト]		関数からリターンするまで実行します。
	[デバッグ]-[カーソル位置まで実行]		カーソル位置までデバッグ中のプログラムを実行します。
	[デバッグ]-[ブレークポイント]	F9	カーソル行にブレークポイントを設定、または解除します。
	[デバッグ]-[全ブレークポイント削除]		すべてのブレークポイントを削除します。
	[デバッグ]-[ブレークポイント有効切替]		ブレークポイントの有効・無効を切り替えます。
	[デバッグ]-[全ブレークポイントの無効化]		すべてのブレークポイントを無効にします。
	[デバッグ]-[ブレークポイントリスト]	Alt+F9	ブレークポイントリストを表示します。
	[デバッグ]-[変数]		変数ウインドウの表示・非表示を切り替えます。
	[デバッグ]-[スタック]		スタックウインドウの表示・非表示を切り替えます。
	[デバッグ]-[デバッグレイアウト]		ボタンや各種バーのレギュラーレイアウトとデバッグレイアウトを切り替えます。

実行と中止

［実行］コマンド

PHPプログラムを実行します。すでにデバッグを開始しているときは、現在停止しているところから実行します。まだデバッグを開始していないときは、指定プログラムのデバッグ実行を開始します。

ブレークポイントに達すると、プログラムの実行が止まります。

【メモ】[実行]コマンドでデバッグを開始するには、[オプション設定]ダイアログボックス[デバッグ]ページの[簡単スタートを有効にする]を選択してください。

【メモ】デバッグ開始時には、MocaScriptのEvent.onDebugBegin()が呼び出されます。

【注意】実行すると、ブレークポイントに達するか、プログラムの最後まで実行します。その他の方法でプログラムの実行を止めることはできないので注意してください。

【注意】[実行]コマンドによってデバッグを開始した場合でも、Webブラウザ、Webサーバー経由でPeggy Pro for PHPに接続されるまで2.5秒以上かかると、PHPプログラムの先頭で停止した状態になることがあります。

［中止］

PHPプログラムの実行をここまでで中止し、デバッグを終了します。以降のプログラムは実行されません。

【メモ】デバッグ終了時には、MocaScriptのEvent.onEndBegin()が呼び出されます。

［ステップイン］

PHPプログラムを1ステップ実行します。関数呼び出し文であった場合は、関数の内部へステップインします。

［ステップオーバー］

PHPプログラムを1ステップ実行します。関数呼び出し文であった場合は、それ全体を1ステップとして実行し、内部へは立ち入りません。

［ステップアウト］

実行中の関数からリターンするまで実行します。

［カーソル位置まで実行］

ソースファイルのカーソルがある行まで実行します。

【注意】PHP実行文のない行にカーソルがあるときに[カーソル位置まで実行]を選択すると、プログラムは止まらず、ブレークポイントかプログラムの最後まで実行してしまいます。必ず実行文のある行にカーソルを置いてから、[カーソル位置まで実行]コマンドを選択してください。

【注意】[カーソル位置まで実行]コマンドによってデバッグを開始した場合でも、Webブラウザ、Webサーバー経由でPeggy Pro for PHPに接続されるまで2.5秒以上かかると、PHPプログラムの先頭で停止した状態になることがあります。

【メモ】簡単スタートが有効になっているとき、これらのコマンドを使ってデバッグを開始することも可能です。

ブレークポイント

PHPソースファイルの行にブレークポイントの設定と削除、有効/無効の切替をします。

【ヒント】ブレークポイントは、プロジェクトとともに保存されます。プロジェクトを開くと、それまで設定されていたブレークポイントは開いたプロジェクトのものに置き換えられます。

[ブレークポイント] コマンド

編集ウインドウのカーソル行にブレークポイントを設定します。すでに設定されている場合は、削除します。

【重要】ブレークポイントの設定できるファイル・できないファイル
ブレークポイントは、デバッグ対象のPHPプログラムと思われるソースファイルの行のみに設定できます。デバッグ対象のPHPプログラムファイルとは、[オプション設定]ダイアログボックスプロジェクトプロパティの[ローカルルート]に設定されているフォルダ以下にあるファイルです。[ローカルルート]を設定していないときは、すべてのファイルにブレークポイントを設定できます。

[全ブレークポイント削除] コマンド

すべてのブレークポイントを削除します。

[ブレークポイント有効切替] コマンド

編集ウインドウのカーソル行にあるブレークポイントの有効・無効を切り替えます。

[全ブレークポイントの無効化] コマンド

すべてのブレークポイントを無効化します。

[ブレークポイントリスト] ダイアログボックス

[ブレークポイントリスト]ダイアログボックスを表示します。

[ブレークポイントリスト] コマンド

[ブレークポイントリスト]ダイアログボックスを表示します。

[ブレークポイントリスト] ダイアログボックス

現在設定されているブレークポイントの一覧を表示します。[ ]上をクリックすると、そのブレークポイントの有効・無効を切り替えられます。

パスカウント

パスカウントは、ブレークポイントを何回通過したときにブレークするかを指定します。[パスカウント]の欄を2回クリックすると、パスカウントを変更できます。[ ]ボタンで増減させることも可能です。

[ソース表示]

選択しているブレークポイントが仕掛けられているソース行を表示します。ブレークポイントリストのブレークポイントをダブルクリックしても表示します。

[すべて有効]

すべてのブレークポイントを有効にします。

[すべて無効]

すべてのブレークポイントを無効にします。

[削除]

選択しているブレークポイントを削除します。

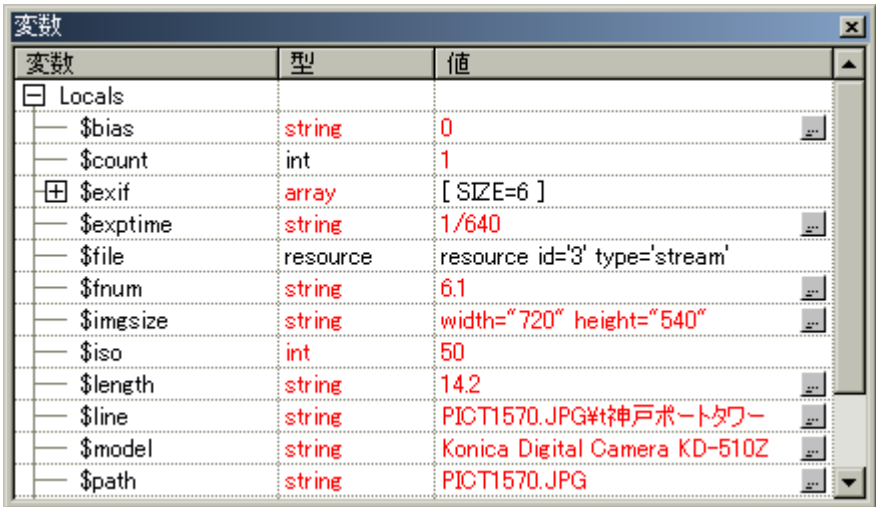
[すべて削除]

すべてのブレークポイントを削除します。

【ヒント】編集ウインドウのブレークポイントマーク[●]をダブルクリックしても、[ブレークポイントリスト]ダイアログボックスを表示できます。

[変数] ウインドウ

[デバッグ]-[変数]を選択すると、変数ウインドウが表示されます。PHPプログラムの実行がブレーク状態のとき、変数ウインドウを使って各種変数の値を表示・変更できます。データ型や値が変化した変数は、赤色で表示されます。



【注意】変数ウインドウは、デバッグレイアウトのときだけ表示・非表示を切り替えられます。

【注意】漢字コードにUTF-8を選択している場合でも、Shift JISにない文字は?と表示されます。

【ヒント】変数ウインドウの各部分の色は、[オプション設定]ダイアログボックスの[表示色]ページでカスタマイズできます。

【ヒント】漢字コード、整数フォーマットは、[オプション設定]ダイアログボックス[デバッグ]ページでも設定できます。

ワンキーコマンドとポップアップメニュー

変数ウインドウに以下のキーを入力することで、表示フォーマットや値の変更が簡単に行えます。また、これらはポップアップメニューからも選択できます。

キー操作	ポップアップメニュー	機能
A	[文字列漢字コード]-[自動]	ソースファイルの漢字コード
S	[文字列漢字コード]-[Shift JIS]	文字列をShift JISとして表示
E	[文字列漢字コード]-[EUC]	文字列をEUCとして表示
U、8	[文字列漢字コード]-[UTF-8]	文字列をUTF-8として表示

I	[整数表示フォーマット]-[そのまま]	そのまま表示
D	[整数表示フォーマット]-[10進]	符号付10進数として表示
H、X	[整数表示フォーマット]-[16進]	符号なし16進数として表示
O	[整数表示フォーマット]-[8進]	符号なし8進数として表示
C	[変化優先ソート]	変化した変数を先に表示
F2、V	[変数値の編集]	変数の値を変更する
.（ドット）	(なし)	[文字列の編集]ダイアログボックスを表示
M	[続きの要素]	配列やオブジェクトの続きの要素・プロパティを表示
(なし)	[すべての要素]	配列やオブジェクトのすべての要素・プロパティを表示
?	(なし)	ワンキーコマンドの簡単な説明文をステータスバーに表示

【注意】要素やプロパティが非常に多くある配列、オブジェクトに対して[すべての要素]を実行すると、すべてを表示するために長い時間が必要となる場合があります。[ステップイン]などが遅く感じられるときは、[+]ボタンをクリックして配列やオブジェクト内の表示を閉じてください。

変数値の変更

変数の値を変更するには、変数を選択しF2キーまたはポップアップメニューから[変数値の編集]を選択してください。最後にEnterキーを押すと、新しい値が設定されます。中止するには、Escキーを押してください。

【注意】値の変更や[...]ボタンによるテキスト編集は、PHPプログラムがブレーク状態のときだけ可能です。

[変化優先ソート]

値が変化した変数を先に表示します。ポップアップメニューから[変化優先ソート]を選択するか、ワンキーコマンドCで切り替えられます。

[続き]

配列など多くの要素を持ったオブジェクトをすべて表示しきれなかった場合、最後に表示されている[続き...]をダブルクリックすることで、続きの要素を追加表示します。Shiftキーを押しながらダブルクリックすると、残りすべてを表示します。

ポップアップメニューから[続きの要素]、[すべての要素]を選択しても同じです。

【ヒント】一度に追加する要素数は、[オプション設定]ダイアログボックス[最大要素数]でカスタマイズできます。

変数値のコピー

変数を選択しているときに、Ctrl+Cをタイプするか、メニューから[編集]-[コピー]を選択することで、変数の値をクリップボードへコピーできます。

[文字列の編集] ダイアログボックス

長い文字列や改行を含んだ文字列を変数ウインドウで確認するのは簡単ではありません。そのような場合、変数値の右端にある[...]ボタンをクリックしてください。[文字列の編集]ダイアログボックスが表示され、長い文字列を確認できます。

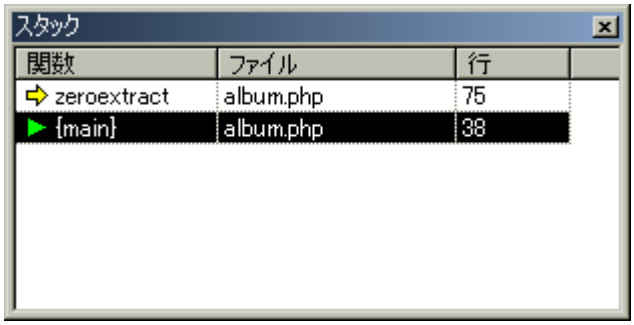
[文字列の編集]ダイアログボックスで表示されているテキストを編集し[OK]ボタンをクリックすると、編集したテキストが変数にセットされます。

[モード]

言語モードを選択して、テキストを色分け表示できます。

[スタック] ウインドウ

[デバッグ]-[スタック]を選択すると、スタックウインドウが表示されます。スタックウインドウでは、関数の呼び出し関係が表示されます。



ウインドの最も上に表示されるスタックが現在ブレークしている関数のスタックで、2行目が現在の関数を呼び出した関数、3行目が...となります。

スタックの切り替え

各行をダブルクリックすると、対応するソースファイル行が表示され、変数ウインドウがその関数のものに切り替わります。これにより関数呼び出しの任意のレベルでの変数が確認・変更できます。

【注意】スタックの切り替えは、PHPプログラムがブレーク状態のときだけ可能です。

【メモ】最上位のスタックが指している(次に実行されるステートメント)ソースファイル行には、[➪]が、最上位以外が選択されているときはスタックウインドウとソースファイル行に[▶]が表示されます。

【注意】スタックウインドウは、デバッグレイアウトのときだけ表示・非表示を切り替えられます。

【ヒント】スタックウインドウの各部分の色は、[オプション設定]ダイアログボックスの[表示色]ページでカスタマイズできます。

[デバッグレイアウト]

デバッグ時に無くてはならない変数やスタックウインドウも、デバッグ終わると邪魔者になってしまいます。そこで、通常の編集作業に便利な「レギュラーレイアウト」と、デバッグに便利な配置の「デバッグレイアウト」切り替えて使えるようにしました。

[デバッグ]-[デバッグレイアウト]コマンドにより、2つのレイアウトを交互に切り替えます。

【ヒント】オプションにより、デバッグ開始時とデバッグ終了時、自動的にレイアウトを切り替えることができます。詳しくは、『第11章 カスタマイズ／[デバッグ]ページ』を参照してください。

【ヒント】これらレイアウトには、すべてのボタンとドッキングウインドウの配置が含まれています。それぞれのレイアウトのボタンはカスタマイズできます。詳しくは、『第11章 カスタマイズ／ツールバー』を参照してください。

第 11 章　カスタマイズ

このリファレンスでは、カスタマイズに関するより詳細な使い方を記載しています。

ツールバー

Peggyでは、ツールバーの位置はもちろん、ボタンを追加、削除したり、ボタンの並びを入れ替えて好みのツールバーにカスタマイズできます。また、スクリプトのボタンへの割り当てもここで行います。

表示と消去

ツールバーは、表示したり消したりできます。ほとんど使わないツールバーは、消しておいたほうがスクリーンが広く使えます。フローティング状態のツールバーは、右上の[×]ボタンをクリックすれば簡単に消せます。

ドッキング状態のツールバー

フローティング状態のツールバー

ツールバーを表示したり消したりする

1. メニューの[ツール]-[カスタマイズ]を選択して、[カスタマイズ]ダイアログボックスを表示させます。
2. [ツールバー]ページの[ツールバー]のチェックをON/OFFしてください。ONでツールバーが現れます。
3. 必要なツールバーをON/OFFしたら、[OK]ボタンをクリックしてダイアログボックスを閉じます。

それ以外にも、[表示]-[ツールバー]で、[ツールバー]ダイアログボックスが開きます。各ツールバーのチェックをON/OFFしてください。ONでツールバーが現れます。

フローティングにする

1. 左マウスボタンでツールバーの左端にあるグリップハンドルをドラッグします。
2. 編集ウインドウの上下左右端ではないところまで持って行き、ドロップします。

フローティングツールバーの形状を、縦長、横長などに変えることもできます。フローティング状態のツールバーの境界を左マウスボタンでドラッグして移動させてください。ボタンの数に応じて適当な縦横サイズに変化します。

【ヒント】普通ならドッキングしてしまう位置にフローティング状態のまま配置したい場合は、Ctrlキーを押しながらドロップします。

【ヒント】ツールバーの左端にあるグリップハンドルをダブルクリックしてもフローティングにできます。

ドッキングする

1. 左マウスボタンでツールバーのタイトル部分をドラッグし、ドッキングさせたい所まで持っていきます。ドッキングできる個所は、編集ウインドウの上下左右の端です。
2. ドッキングできる所までくると、ドラッグ中のイメージが変化します。上下端にドッキングさせるときはイメージが少し細くなり、左右端にドッキングさせるときはイメージが縦長になります。ドッキングさせたい場所をねらってドロップしてください。

【ヒント】ツールバーのタイトルの部分をダブルクリックすると、元あった場所へドッキングします。

【ヒント】ツールバーを思いどおりの場所にドロップできるようになるには、多少のコツが必要です。最初は少しずれたりしますが、すぐに習得できます。

簡単カスタマイズ

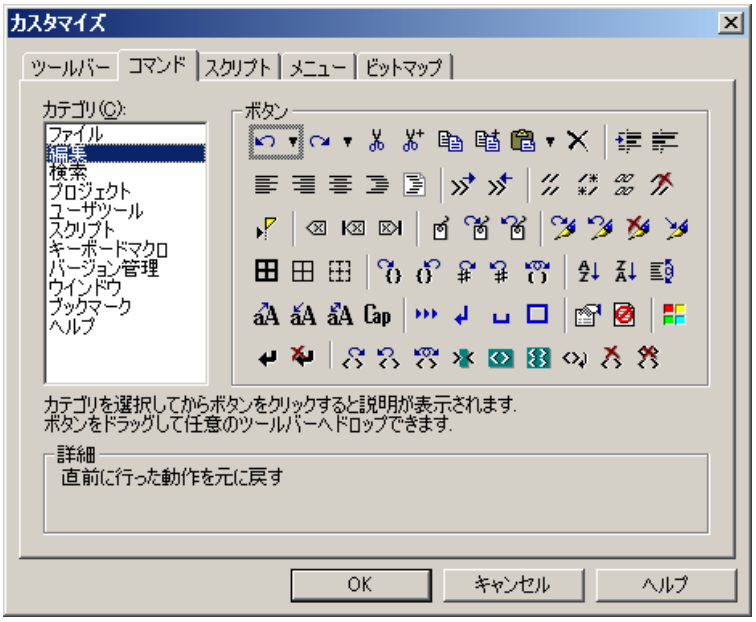
ちょっとボタンの順番を変えたり、ボタンを削除するといったカスタマイズは、[カスタマイズ]ダイアログボックスを表示させなくても簡単に行う方法があります。

●Altキーを押しながらツールバー上のボタンを左マウスボタンでドラッグしてください。ツールバーのないところでドロップすると、ツールバーからボタンを削除できます。また、他のツールバーや同じツールバーの違った場所でドロップすると、ボタンをそこへ移動できます。

●同じ場所でも、Altキーを押しながらボタンをドラッグし、少しだけ右へ移動させると、左にあるボタンとの間に間隔を空ける効果があります。視覚的にボタンをグループ化できます。

ツールボタンの追加と削除

もっと本格的にツールバーをカスタマイズするには、メニューの[ツール]-[カスタマイズ]を選んで[カスタマイズ]ダイアログボックスを呼び出し、[コマンド]ページを選択します。ボタンの追加もできます。



[カテゴリ]を選ぶと、そのカテゴリに属しているすべてのボタンが右の[ボタン]部分へ表示されます。そしてボタンをクリックすると、ボタンの説明が[詳細]部分へ表示されるので、何をするためのボタンかを確認できます。

ボタンをツールバーに追加する

[カスタマイズ]ダイアログボックスの[コマンド]ページにある[ボタン]からボタンをドラッグし、追加したいツールバー上にドロップしてください。ドラッグ中に中止するには、[コマンド]ページ上で左マウスボタンを離すか、Escキーを押します。

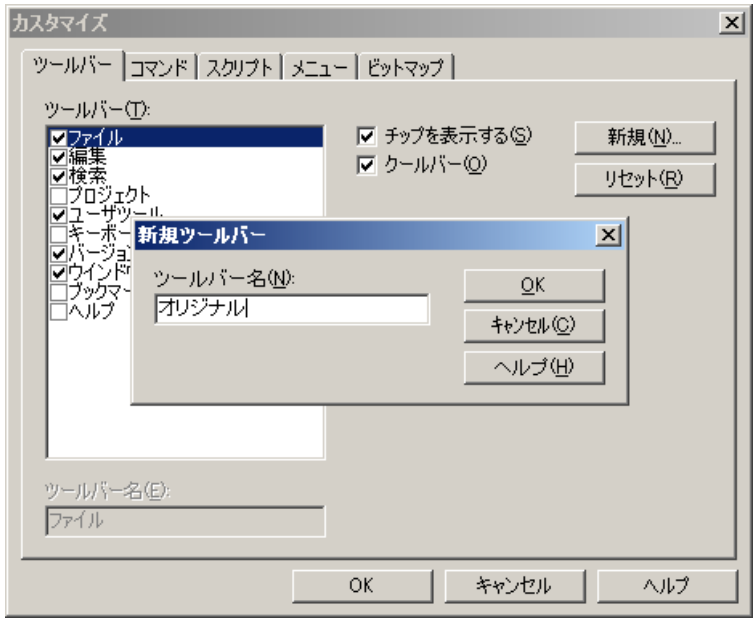
【注意】なにもない所(すでにあるツールバー上ではない所)へドロップすると、新しいツールバーを作成します。

新規ツールバーの作成と削除

既存のツールバー以外に、ユーザ独自のものを作成できます。

新しいツールバーを作成する

1. メニューの[ツール]-[カスタマイズ]を選択して、[カスタマイズ]ダイアログボックスを表示させます。
2. [ツールバー]ページにある[新規]ボタンをクリックしてください。
3. [新規ツールバー]ダイアログボックスが表示されるので、新しいツールバーの名前を入力してから、[OK]ボタンをクリックします。これで新しいツールバーが作成され、画面に表示されているはずです
4. [コマンド]ページを選び、新しいツールバーに入れたいボタンをドラッグ&ドロップで乗せてください。
5. [OK]ボタンをクリックして、[カスタマイズ]ダイアログボックスを閉じます。



ユーザが作成したツールバーを削除する

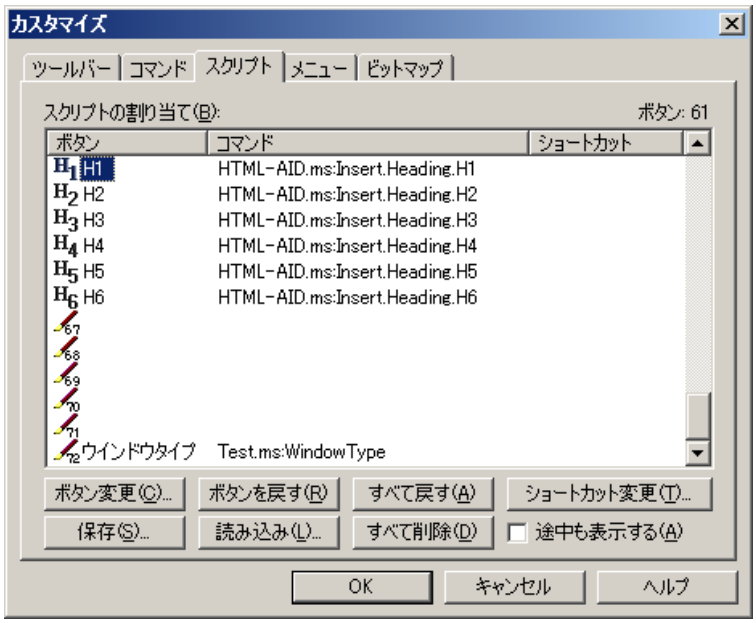
ユーザが作成したツールバーが不要になったら以下の手順で削除してください。

1. メニューの[ツール]-[カスタマイズ]を選択して、[カスタマイズ]ダイアログボックスを表示させます。
2. [ツールバー]から削除したいツールバーを選択します。
3. [削除]ボタンをクリックしてください。
4. [OK]ボタンをクリックして、[カスタマイズ]ダイアログボックスを閉じます。

デフォルトに戻す

最初からあるツールバーは削除できませんが、デフォルトの状態に戻すことができます。ユーザ作成ツールバーの削除と同じようにリストからバーを選択し、[リセット]ボタンをクリックしてください。カスタマイズする前の状態に戻ります。

スクリプトボタンのカスタマイズ



スクリプトをボタン[ ]に割り当てることができます。[スクリプト]ページの[コマンド]欄をクリックすると、全コマンドを表示したリストボックスが現れるので、割り当てたいスクリプトコマンドを選択してください。このとき[途中も表示する]をONにしているとリストにコマンドグループも表示されます。

以上でボタンにコマンドが割り当てられます。そのボタンをツールバーに表示させておけば、ボタンをクリックすることでコマンドを実行できます。[ショートカット]欄にはショートカットキーの割り当てが表示されます。

[ボタン変更]

リストで選択しているボタンのイメージを変更します。クリックすると[ボタンギャラリー]ダイアログボックスが表示されるので、変更したいボタンイメージを選択してから[変更]ボタンをクリックしてください。リスト中のボタンイメージ上をダブルクリックしても、[ボタンギャラリー]ダイアログボックスを開くことができます。

【メモ】<script>フォルダ内にボタンに割り当ててあるスクリプトファイルと同じ名前で拡張子だけが<.bmp>のビットマップファイルがあると、優先的にそのファイル内のボタンイメージを一覧表示します。スクリプトファイルとセットのビットマップファイルを用意しておくと、簡単にボタンイメージを設定できます。

[ショートカット変更]

リストで選択しているスクリプトにショートカットを設定、または設定されているショートカットを削除します。

[ボタンを戻す]

変更してあるボタンのイメージをオリジナルのものに戻します。

[すべて戻す]

すべてのボタンイメージをオリジナルのものに戻します。

[保存]、[読み込み]

ボタンへの割り当てをテキストファイルに保存したり、ファイルから読み込んだりします。状況により割り当てを切り替えたり、他のマシンやユーザーと割り当てを共通化することができます。

[すべて削除]

ボタンへの割り当てを全て削除します。ボタンのイメージもすべて元へ戻ります。

[ボタンギャラリー] ダイアログボックス

ボタンのイメージを選択するダイアログボックスです。<share¥bitmap>と<share¥script>に保存されているビットマップファイルからボタンイメージを表示・選択することができます。[設定]ボタンをクリックすると、ボタンイメージが選択されているものに変更されます。

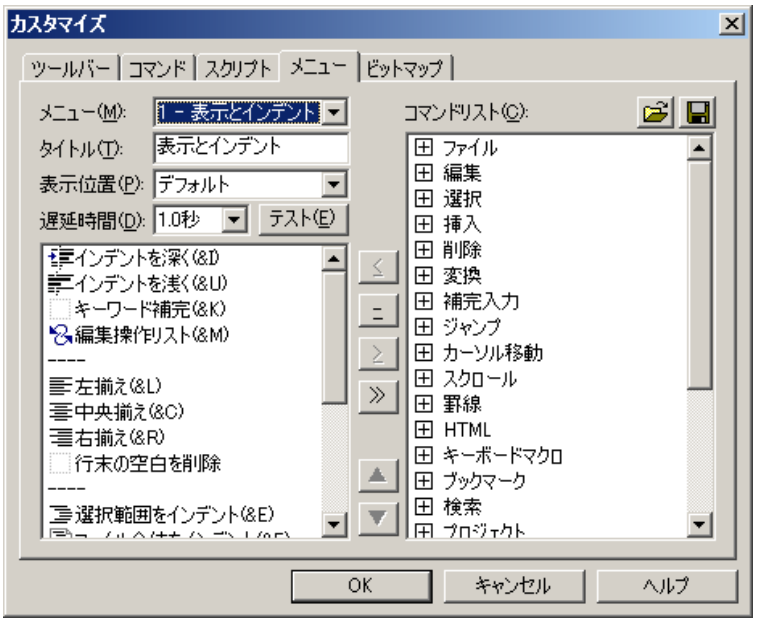
ビットマップファイル

1つのボタンの大きさは、横16ドット×縦15ドットです。1つのビットマップファイルには、複数のボタンイメージを収めることができます。そのためビットマップファイル全体の大きさは、縦15ドット、横は16ドットの倍数としてください。横は最大2048ドットまでです。縦横ドット数が多少多くても読み込むことは可能ですが、それらのドットは使用されません。

色は16色(4-bit)、256色(8-bit)、65535/32768色(16-bit)、フルカラー(24-bit)のどれでもかまいません。ただし、明るい灰色(ライトグレー:RGB=192,192,192)は背景色として扱われるので、実際に描画されるときはシステムで選択されているボタン表面の色に変換されます。

【注意】Peggyは、ボタンを表示するのに必要なビットマップファイルを1回だけ読み込みます。その後、そのファイルが更新されても再読み込みしません。編集したビットマップファイルを再読み込みさせるには、Peggyを再起動してください。

ユーザ定義メニューのカスタマイズ



独自のユーザ定義メニューを作成・変更します。12種類のメニューを定義できます。定義したメニューは、[ユーザ定義メニュー]コマンドで呼び出せます。

また、編集ウインドウの右マウスボタンで表示されるポップアップメニューを、ここで定義したものに変更することも可能です。右マウスボタンメニューの変更は、[オプション設定]ダイアログボックスの[言語]ページの[ポップアップ]から言語毎に指定してください。

[メニュー]

表示・修正したいユーザ定義メニューを選択します。12種類のメニューを定義できます。

[テスト]

定義したメニューをテストします。[テスト]ボタンにより表示されたメニューからコマンドを選択しても何も実行されません。

[タイトル]

定義したメニューのタイトルを指定します。

[表示位置]

ユーザ定義メニューがポップアップする場所を選択します。[カーソル]を選択すると、できるだけアクティブな編集ウインドウのキャレット位置にメニューを表示します。アクティブな編集ウインドウがないか、[マウス]を選択すると、マウスカーソルの近くに表示します。[デフォルト]を選択すると、[編集オプション]の設定に従います。

[遅延時間]

[ユーザ定義メニュー]コマンドを実行してから実際にメニューが表示されるまでの遅延時間を指定します。キーシンボルと遅延時間を組み合わせることで、メニューが表示される前にコマンドを選択・実行することが可能になります。

メニュー表示遅延中は、ステータスバーにキーシンボルの一覧が表示されます。また、メニュー表示遅延中に即メニューを表示させるには、スペースキーをタイプしてください。

[メニュー項目リスト]

ユーザ定義メニューに表示される項目リストです。項目を選択しているときに再度左マウスボタンでクリックすると、表示されるテキストを編集できます。(&K)のようなキーシンボルを付けておくと、ポップアップメニューが表示されたとき、キーボードから簡単に選択できるようになります。

キーシンボルと遅延時間を組み合わせることで、メニューが表示される前にコマンドを選択実行することも可能です。

メニュー項目の順番は、ドラッグ＆ドロップで変更できます。

[<]

[コマンドリスト]で選択しているコマンドをメニューに追加します。

[-]

セパレータをメニューに追加します。

[>]

メニューの選択しているコマンドを削除します。

[>>]

メニューのすべての項目を削除します。

[▲]

選択しているメニュー項目を上に移動します。メニュー項目をドラッグして移動させることも可能です。

[▼]

選択しているメニュー項目を下に移動します。メニュー項目をドラッグして移動させることも可能です。

[ ]

保存してあるユーザ定義メニューを読み込みます。現在設定しているメニューに追加読み込みする方法と、すべて入れ替える方法を選択できます。

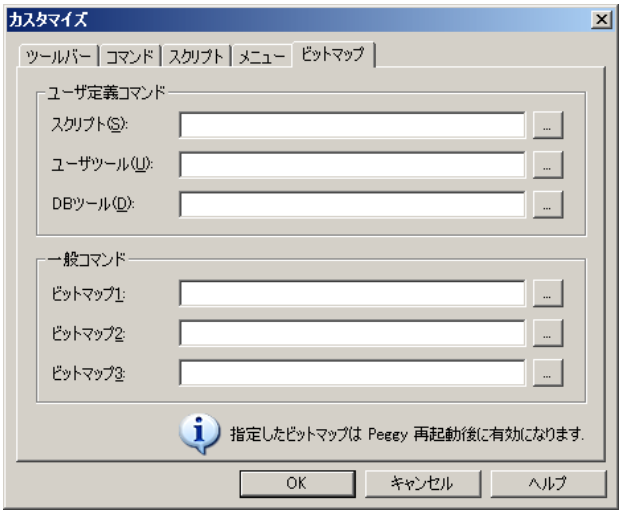
[ ]

現在設定されているユーザ定義メニューをファイルに保存します。

[コマンドリスト]

メニューに追加できるコマンド一覧です。コマンドは、ドラッグ＆ドロップでもメニューに挿入できます。

[カスタマイズ] ダイアログボックス [ビットマップ] ページ



ツールバーボタンのイメージをまとめて変更します。ここで指定したボタンイメージは、Peggyを再起動して有効になります。

[スクリプト]

スクリプトボタン1～72のイメージが収められているビットマップファイルを指定します。スクリプトボタンは、[カスタマイズ]ダイアログボックス[スクリプト]ページから個々のボタン単位で変更することも可能です。

[ユーザツール]

ユーザツールボタン1～24のイメージが収められているビットマップファイルを指定します。

[DB ツール]

DBツールボタン1～8のイメージが収められているビットマップファイルを指定します。DBツールボタンは、[DBツール設定]ダイアログボックスから個々のボタン単位で変更することも可能です。

[ビットマップ 1]

ファイル、編集、検索のカテゴリに属するボタンイメージが収められているビットマップファイルを指定します。

[ビットマップ 2]

プロジェクト、ユーザツール(ユーザツールボタン1～24、DBツールボタン1～8以外)、スクリプト(スクリプトボタン1～72以外)、キーボードマクロ、バージョン管理、ウインドウ、ブックマーク、ヘルプのカテゴリに属するボタンのイメージが収められているビットマップファイルを指定します。

[ビットマップ 3]

Subversion関連のボタンイメージが収められているビットマップファイルを指定します。

ビットマップファイル

<share¥bitmap>フォルダに、ボタンのイメージを収めたビットマップファイルがあります。これらのファイルのボタンイメージは、予めPeggyに組み込まれているものと同じです。オリジナルのボタンを作成する際の参考としてご活用ください。

1つのボタンの大きさは、横16ドット×縦15ドットです。1つのビットマップファイルには、複数のボタンイメージを収めることができます。そのためビットマップファイル全体の大きさは、縦15ドット、横は16ドットの倍数としてください。横は最大2048ドットまでです。縦横ドット数が多少多くても読み込むことは可能ですが、それらのドットは使用されません。

色は16色(4-bit)、256色(8-bit)、65535/32768色(16-bit)、フルカラー(24-bit)のどれでもかまいません。ただし、明るい灰色(ライトグレー:RGB=192,192,192)は背景色として扱われるので、実際に描画されるときはシステムで選択されているボタン表面の色に変換されます。

【参照】[ボタンギャラリー]ダイアログボックス

[カスタマイズ]ダイアログボックス

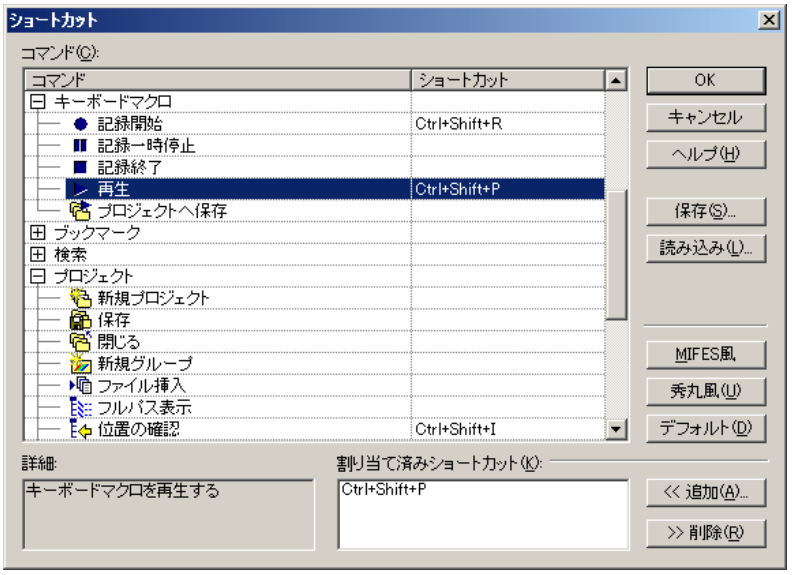
ショートカットキー

Peggyのデフォルトのキー割り当ては、標準的なWindowsアプリケーションとほぼ共通です。Windowsアプリケーションに親しんでいるユーザであれば、そのまま違和感なく使用できます。好みに応じてほとんどのキーをカスタマイズできます。

また、使用頻度の低いコマンドは、メニューにもショートカットキーにも割り当てられていない場合があります。そのようなコマンドを使うには、新しいショートカットキーを割り当てるしかありません。

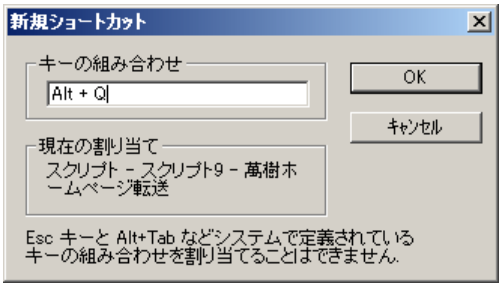
[ショートカット]ダイアログボックス

ショートカットキーをカスタマイズするには、メニューから[ツール]-[ショートカット]を選択してください。



【注意】Peggyにはバージョン管理システムVSS/CVS/Subversionのフロントエンド機能が含まれていないのにもかかわらず、VSS/CVS/Subversion関連コマンドが[ショートカット]ダイアログボックスの[コマンド選択]に表示され、キーに割り当てることができます。これは、PeggyとPeggy Proのキー割り当ての互換性を維持し、両者を切り替えて使った場合でもキー割り当てを失わないようにするための処置です。PeggyでCVS関連コマンドを実行しても、何の動作もしません。PeggyとPeggy Proは、レジストリのキー割り当てエントリーを共有しています。

新しいショートカットキーを割り当てる



1. メニューから [ツール]-[ショートカット] を選択して、[ショートカット] ダイアログボックスを表示します。
2. 割り当てたいコマンドを、[コマンド] リストから選んでください。
3. [追加] ボタンをクリックして、[新規ショートカット] ダイアログボックスを開きます。割り当てたいコマンドの上をダブルクリックしても開くことができます。
4. 割り当てたいキーの組み合わせをタイプします。タイプしたキーの組み合わせがすでに他のコマンドへ割り当てられていた場合、そのコマンドが [現在の割り当て] に表示されますので、確認してください。

5.[OK] ボタンをクリックして、[新規ショートカット] ダイアログボックスを閉じます。[割り当て済みショートカット] に、新しいショートカットキーの割り当てが表示されていることを確認してください。必要なら同じコマンドに複数のショートカットキーを割り当てることもできます。

6. さらにショートカットキーを追加する場合は、ステップ 2 ～ 5 を必要なだけ繰り返します。

7. 最後に、[OK] ボタンをクリックすると、新しいショートカットが有効になります。

【注意】Escを含んだキーの組み合わせ、修飾キーのないEnterやSpaceキー、システムコマンドのショートカットとして定義されているAlt+Tabなどは割り当てられません。また文字として入力できるキーも単独ではショートカットキーとしては使えません。必ずCtrl、Alt、Ctrl+Altなどの修飾キーと組み合わせる必要があります。

【注意】NEC PC-9800シリーズのWindows NT、Windows 95では、AltキーがGRAPHと表示されます。

【注意】[新規ショートカット]ダイアログボックスで矢印キー（←、→、↑、↓）は、それぞれLeft、Right、Up、Downと表示されます。

ショートカットキーを削除

1. メニューから [ツール]-[ショートカット] を選択して、[ショートカット] ダイアログボックスを表示します。

2. 削除したいコマンドを、[コマンド] リストから選んでください。

3. [割り当て済みショートカット] から削除したいキーの組み合わせを選択します。

4. [削除] ボタンをクリックして、キーの組み合わせを削除します。

5. さらにショートカットキーを削除するには、ステップ 2 ～ 4 を必要なだけ繰り返します。

6.[OK] ボタンをクリックすると、実際にショートカットキーが削除されます。

ショートカットキーを変更するには、いったん削除してから、新しいショートカットキーを割り当ててください。

【ヒント】必ず削除してからでないと新しい割り当てができないわけではありません。すでに割り当てのあるショートカットキーを別のコマンドに再割り当てすると、古いコマンドはそのショートカットキーを失います。

ショートカットキー定義の保存、読み込み

設定したオリジナルのショートカットキー定義をファイルに[保存]したり、保存した定義ファイルを[読み込み]することもできます。こうすることで複数のショートカットキー定義を使い分けることができます。

1. [ツール]-[ショートカット] の [保存] ボタンをクリックします。

2. ファイル名を入力して [保存] をクリックします。

これで、現在のショートカットキーの設定状態が保存できます。

【注意】ユーザ定義ファイルの拡張子は、<.sck>になります。逆に、保存したショートカットキーの設定を使いたいときは、以下のステップで読み込めます。

3. [ツール]-[ショートカット] の [読み込み] ボタンをクリックします。

4. ファイル名を入力して [開く] をクリックします。

5.[OK] をクリックしてダイアログボックスを閉じます。

【注意】Peggyを起動したときのショートカットキーの状態は、前回終了したときの状態が引き継がれています。

ユーザ定義ファイル

保存したユーザ定義ファイルはテキストファイルですから、Peggyで見ることも可能です。また、印刷すればショートカットキー定義一覧表になります。ユーザ定義ファイルのフォーマットは、

コマンド名,割り当てキー,... #コメント

です。下の例では、NewFileにCtrl+Nキーを割り当てることを意味します。#直後のコメントは、ショートカットキーダイアログボックスで表示されるコマンドリスト名称です。

【例】NewFile, Ctrl+N　　#ファイル - 新規作成 ,新規にファイルを作成する

【ヒント】この保存ファイルの内容を外部で編集することも可能ですが、A-Z,0-9以外のキーについては、表示がキーコードになることがあります。その場合、[190]のように10進数で表現されますが、ショートカットキーダイアログボックス内では正しく表示されます。

[MIFES風]、[秀丸風]ボタン

Peggyでは、一般的なWindowsのショートカットキー定義をデフォルトとしていますが、[MIFES風]、[秀丸風]ボタンをクリックし、ダイアログボックスを閉じると、それぞれのアプリケーションのデフォルトショートカットキー状態が再現されます。だたし、両者とPeggyのコマンドがまったく同じというわけではありませんので、すべての機能が同じショートカットキーでできるわけではありません。MIFES、秀丸に慣れている方はお試しください。またこれを読み込んだ状態からさらに変更を加え、オリジナルのユーザ定義ファイルとして保存することもできます。

[オプション設定]ダイアログボックス

[オプション設定]ダイアログボックスは、メニューの[ツール]-[オプション]で表示されます。[オプション設定]ダイアログボックスには、以下の13ページがあり、それぞれ好みに合わせてPeggyをカスタマイズすることができます。

●[全般]ページ

●[プロジェクト]ページ

●[移動]ページ

●[編集]ページ

●[補完]ページ

●[レイアウト]ページ

●[言語]ページ

●[拡張子]ページ

●[アウトライン]ページ

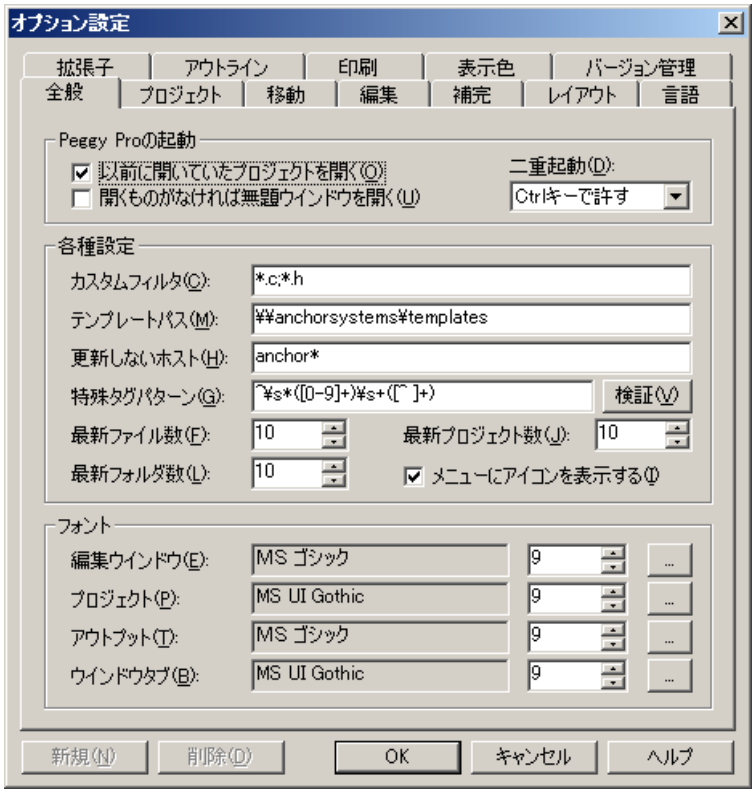
●[印刷]ページ

●[表示色]ページ

●[バージョン管理]ページ

●[デバッグ]ページ（Peggy Pro for PHPのみ）

[全般] ページ



Peggy の起動

[以前に開いていたプロジェクトを自動的に開く]

前回終了時に開いていたプロジェクトを自動的に開きます。そのプロジェクト内の状態はほとんど再現されます。

[何も開くものがなければ無題のウインドウを開く]

[以前に開いていたプロジェクトを自動的に開く]にチェックがないか、以前に開いていたプロジェクトがないときに、無題のウインドウを開いて編集入力待ち状態にします。

[二重起動]

Peggyを複数起動するかどうかを選択します。[許す]を選択すると複数起動できます。[Ctrlキーで許す]を選択すると、Ctrlキーを押しているときだけ2つ目のPeggyを起動できます。[許さない]を選択すると、1つしか起動できなくなります。

各種設定

[カスタムフィルタ]

[ファイルを開く]、[ファイル挿入]のファイル選択ダイアログボックスで表示されるファイルをフィルタする文字列を指定します。フィルタ記述に 2 種類のワイルドカード(クエスチョンマーク(?)とアスタリスク(*))を使うことができます。

フィルタ記述は、この入力枠に収まる文字数しか設定できません。

【ヒント】複数のフィルタパターンを記述するときはセミコロン(;)で区切ります。

m*.c*;*.h;List??.dat

[テンプレートパス]

[テンプレートから新規作成]コマンドでテンプレートファイルを探すフォルダを指定します。なお、色付けDLLファイルなどが入っている<share>フォルダの下の<template>フォルダはここで指定しなくても探索対象になっています。

[更新しないホスト]

Peggyがアクティブになったとき、開いているファイルが外部の他のアプリケーションプログラムによって変更されているかどうかチェックします。しかしネットワーク上のファイルなどによってはチェックに時間のかかる場合もあります。そのような場合、ここへ遅いマシン名を指定することで、それらのマシン上のファイルの更新チェックをスキップさせることができます。マシン名は、スペースで区切って複数指定できます。また、ワイルドカード(*と?)を使って一度に複数のマシンを指定することもできます。

【例】anchor*

[特殊タグパターン]

タグジャンプできるデフォルトの行形式は、ファイル名、区切り文字、行番号が続くものですが、ここに正規表現で特殊なタグパターンを記述することで、異なる行形式であってもタグジャンプが使えるようになります。

【例】　3　d:¥dev¥peggy1.asm SyntaxError No211

上記のように行番号が最初にあり、次にファイル名があるような場合、正規表現で以下の例のように記述することで、タグジャンプできるようになります。

```
^¥s*([0-9]+)¥s+([^ ]+)
```

行番号とファイル名にマッチする部分を括弧( )で囲む必要があります。2つの( )でくくられた文字列のどちらが行番号とファイル名でもかまいません。

[検証]

タグパターンに入力された文字列が正規表現として問題ないか検証します。検証しているのは、正しい正規表現であることと、ファイル名、行番号部分を示す括弧() が2組あることのみです。従って、検証で問題なくても必ずタグジャンプできるとは限りません。

[最新ファイル数]、[最新プロジェクト数]、[最新フォルダ数]

メニューの[ファイル]-[最近使ったファイル]、[最近使ったフォルダ]、[最近使ったプロジェクト]に表示する数を指定します。4から16の範囲で指定します。

[メニューにアイコンを表示する]

メニューにボタンアイコンを表示します。

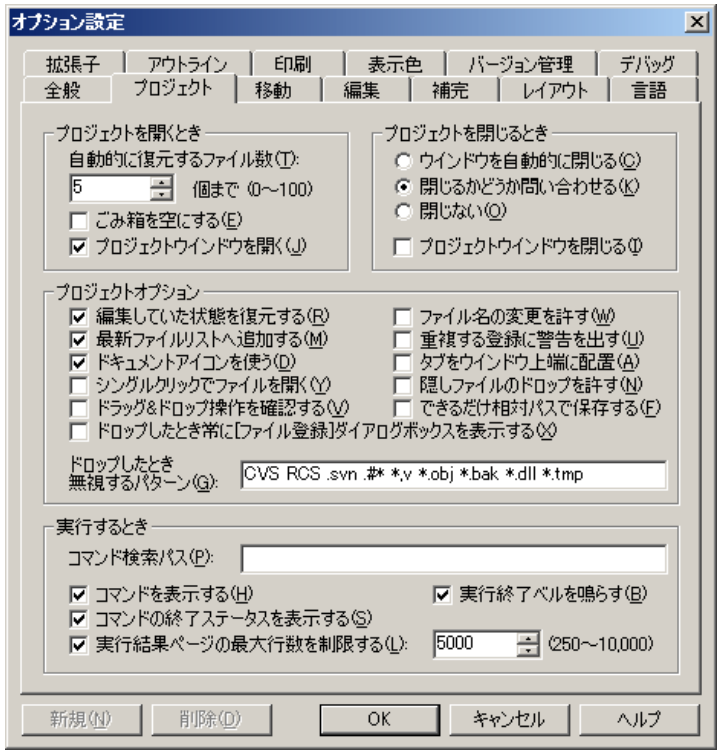
フォント

[編集ウインドウ]、[プロジェクト]、[アウトプット]、[ウインドウタブ]

[編集ウインドウ]、[プロジェクト]、[アウトプット]、[ウインドウタブ]のそれぞれのウインドウ内で使われるフォント、大きさ、太さなどを変更できます。

【注意】イタリック体を指定すると、文字の一部が欠けることがあります。また、ビットマップフォントでボールド体を指定した場合、文字幅に生じる誤差のため、正しく表示できないものがあります。

[プロジェクト] ページ



プロジェクトを開くとき

[自動的に復元するファイル数]

前回終了時に開いていたファイルの内､状態を再現して開くファイルの最大数を指定します。

[ごみ箱を空にする]

プロジェクトを開くたびに[ごみ箱]を空にします。

[プロジェクトウインドウを開く]

プロジェクトを開くとき、プロジェクトウインドウが非表示であれば表示します。

プロジェクトを閉じるとき

[ウインドウを自動的に閉じる]

プロジェクトを閉じるときに、すべての編集ウインドウも閉じます。

[閉じるかどうか問い合わせる]

プロジェクトを閉じるときに、「すべてのウインドウを閉じますか」と問い合わせます。

[閉じない]

プロジェクトを閉じるときに、編集ウインドウはすべて開いたままにします。別途、[保存]や[閉じる]処理を行うことになります

[プロジェクトウインドウを閉じる]

プロジェクトを閉じるときに、自動的にプロジェクトウインドウも閉じます。

プロジェクトオプション

[編集していた状態を復元する]

プロジェクトに登録しているファイルについては、閉じるときにブックマークや選択範囲、カーソル位置などが記録されます。次にファイルを開くときにそれらが復元されます。

[最新ファイルリストへ追加する]

プロジェクトに登録してあるファイルも最新のファイルリストに登録します。プロジェクトに登録されていないファイルを開いたときは、常に最新のファイルリストに登録されます。

[ドキュメントアイコンを使う]

プロジェクトウィンドウに表示されるファイルのアイコンを切り替えます。チェックするとドキュメントアイコン（エクスプローラで表示されるものと同じ）になります。有効にするにはPeggyを再起動してください。

[シングルクリックでファイルを開く]

ONにするとプロジェクトウインドウからシングルクリックでファイルを開く、または実行するようになります。同じウインドウにある[ウインドウリスト]、[アウトライン]、[スクリプト]の各ページも、シングルクリックでウインドウの切り替えやジャンプ、スクリプトの実行ができます。

[ドラッグ & ドロップ操作を確認する]

プロジェクト内部のドラッグ&ドロップ操作による移動、コピー操作に確認メッセージを表示します。これをONにすることにより、誤ったマウス操作でプロジェクトが変更されてしまうのを防ぐことができます。

[ドロップしたとき常に [ファイル登録] ダイアログボックスを表示する]

ONにすると、エクスプローラなどからプロジェクトウインドウへファイルやフォルダをドラッグ＆ドロップしたとき、常に[ファイル登録]ダイアログボックスが表示されます。OFFにすると、Ctrl+Shiftキーを押しながらドロップしたときだけ[ファイル登録]ダイアログボックスが表示されます。

[ファイル名の変更を許す]

チェックすると、プロジェクトウインドウからファイル名の変更ができるようになります。

[重複する登録に警告を出す]

プロジェクトにすでに登録しているファイルを、さらに登録しようとすると警告ダイアログボックスを表示し、注意を促します。

[タブをウィンドウ上端に配置]

プロジェクトウィンドウの「アウトライン」などのタブをウィンドウの上端に表示します。有効にするにはPeggyを再起動してください。

[隠しファイルのドロップを許す]

ファイルをドロップしたとき、隠しファイル、システムファイルの属性をもったファイルも受け付けるようにします。OFFにすると、これらのファイルを無視します。

[できるだけ相対パスで保存する]

ONにすると、登録されているファイルを相対パスとしてプロジェクトファイルに保存します。ただし、相対パスに変換するのは、プロジェクトファイルが保存されるフォルダか、そのサブフォルダ内のファイルだけで、それ以外のファイルは絶対パスで保存されます。ツールの実行フォルダなどのパスも変換されます。

[ドロップしたとき無視するパターン]

ドラッグ&ドロップ操作によりプロジェクトへファイルやフォルダを登録する場合、ここで指定したワイルドカードを含んだパターンに一致するものを無視します。パターンが複数あるときは、スペースで区切っていくつでも指定できます。

実行するとき

[コマンド検索パス]

このコマンド検索パスに入力されたパスは、ファイルアクションで設定されたコマンドを検索するときに使われます。

[コマンドを表示する]

プロジェクトに登録してある実行ファイルやスクリプトを起動するときのコマンドラインをアウトプットウインドウに表示します。

[コマンドの終了ステータスを表示する]

アウトプットウインドウに表示されたコマンドの実行結果の後に、コマンドの終了ステータスを表示します。

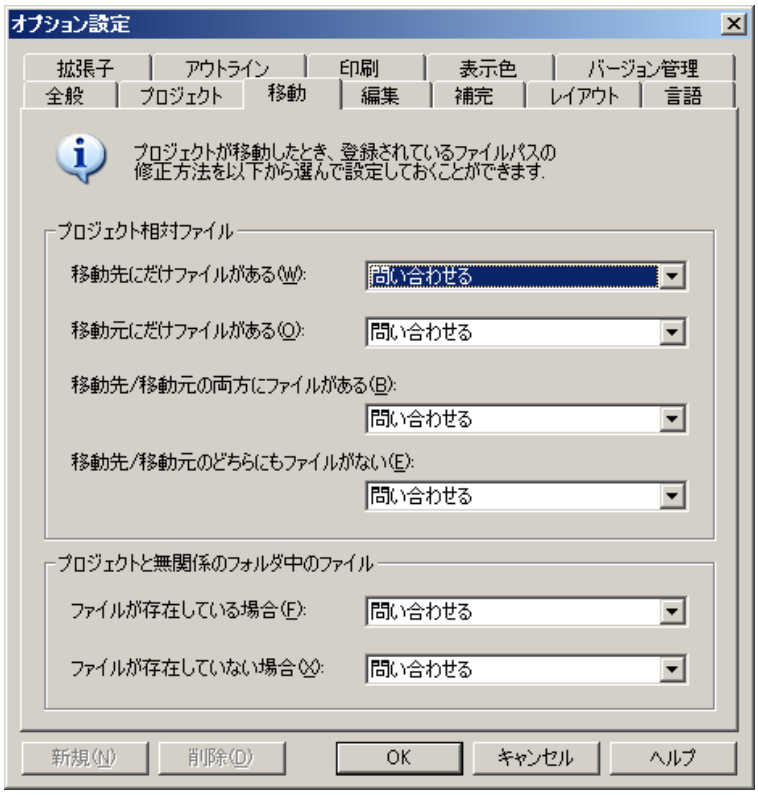
[実行結果ページの最大行数を制限する]

アウトプットウインドウの実行結果ページに表示される実行結果の最大行数を250～10,000行の範囲で設定します。特に膨大な文字列を出力するようなコマンドを実行したときに、制限行数を越えた古い結果から順に削除されます。

[実行終了ベルを鳴らす]

コマンドが終了した時点でベルを鳴らします。静かに実行させたい場合は、OFFにしてください。

[移動] ページ



プロジェクトが移動したとき、予想される6つのケースについて登録されているファイルパスの修正方法を予め設定しておくことで、簡単にプロジェクトを移動できるようになります。

[問い合わせる]を選択しておくと、移動したプロジェクトを開くときに[ファイル移動]ダイアログボックスが表示され、個別に処理方法を選択することができます。それ以外の処理を選択すると、自動的にそれに従って変更します。

[プロジェクト相対ファイル]

[移動先にだけファイルがある]

移動先にファイルがあり、移動元にはファイルがない場合の処理方法を設定します。

[移動元にだけファイルがある]

移動先にはファイルがないが、移動元にはファイルがある場合の処理方法を設定します。

[移動先 / 移動元の両方にファイルがある]

移動先にも移動元にもファイルがある場合の処理方法を設定します。

[移動先 / 移動元のどちらにもファイルがない]

移動先にも移動元にもファイルが存在しない場合の処理方法を設定します。

[プロジェクトと無関係のフォルダ中のファイル]

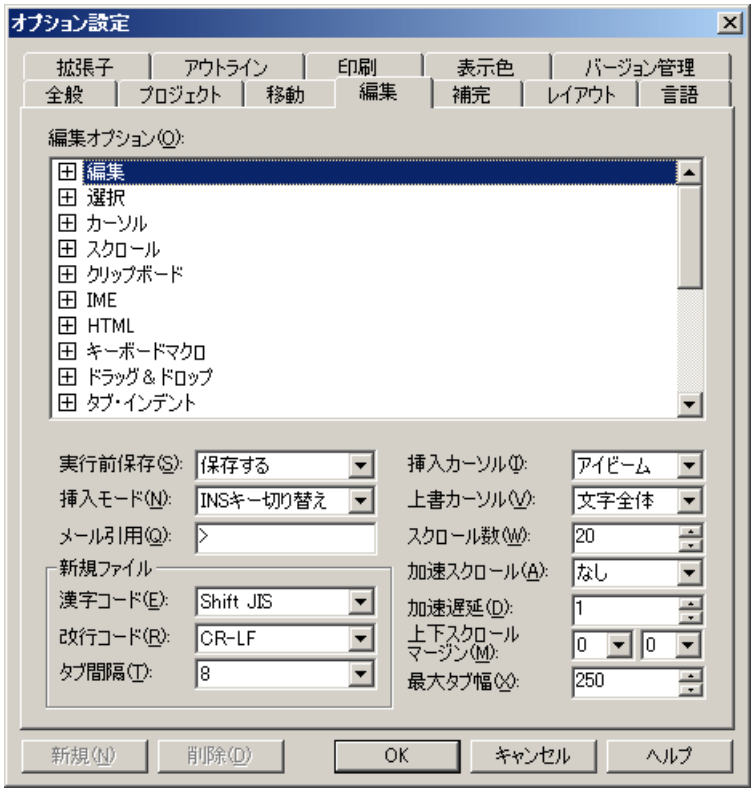
[ファイルが存在している場合]

プロジェクトと無関係のフォルダにあったファイルで、今も存在している場合の処理方法を設定します。

[ファイルが存在していない場合]

プロジェクトと無関係のフォルダにあったファイルで、今はもう存在していない場合の処理方法を設定します。

[編集] ページ



編集オプション

[編集]-[起動直後は上書きモードにする]

Peggy起動時は上書きモードになります。

[編集]-[桁位置をずらさないように上書きする]

上書きモードのとき、上書きしている部分より後の桁位置をずらさないように上書きします。チェックを外すと、文字単位で上書きします。

[編集]-[上書きモードの Enter キーで改行を挿入する]

上書きモードのとき、Enterキーで改行を挿入します。チェックを外すと改行は挿入されず、次の行の先頭へカーソル移動します。

[編集]-[RGB 値挿入のときの RGB 順序を逆転させる (BGR の順)]

[編集]-[挿入と削除]-[RGB値挿入]で挿入する値の上位と下位を逆転させます。

選択	順序	用途
OFF	R・G・Bの順	HTML色指定の順序
ON	B・G・Rの順	WindowsのCOLORREF値の順序

[編集]-[1 文字単位で Undo/Redo する]

Undo/Redoの記録単位を1文字にします。1文字単位でUndo/Redoできるようになります。

[編集]-[保存で行変更状態をクリアする]

変更した行には、変更行表示が付きます。この変更行表示をファイル保存したときにクリアします。

[編集]-[Undo/Redo でファイルと一致したら行変更状態をクリアする]

変更行表示が付いた状態でUndo/Redo操作を行い、元のファイルと同じ状態になったら、変更行表示をすべてクリアします。

[編集]-[行全体を削除したときは直後の行に変更マークを付ける]

行全体を削除したときに、直後の行へ変更マークを付けます。削除した行自体は消えてしまいますが、何か変更があったことを後から確認できます。

[選択]-[ダブルクリックで日本語文字列も選択する]

ダブルクリックで漢字･カタカナの後にひらがなが続いた部分を選択できるようになります。OFFにすると、ダブルクリックした箇所の1文字だけを選択します。

[選択]-[ダブルクリックでスペースと共に全角スペースも選択する]

連続している空白部分をダブルクリックしたとき､全角スペースも含めて選択します。このオプションをOFFにすると、半角スペースとタブの連続部分を選択します。

[選択]-[ダブルクリックで文法を認識しながら括弧を選択する]

括弧や括弧の内側をダブルクリックすると、括弧の内部を選択しますが、その範囲を判定するとき、言語の文法に従って選択します。括弧の内側にコメント行、文字列、正規表現がある場合に、本来は対とならない括弧を選択することがありますが、その問題を解決します。また、複数の言語が混在しているファイル、例えばJavaScriptが埋め込まれたHTMLファイルなどにも有効です。

[選択]-[トリプルクリックでファイル行を選択する]

トリプルクリックでファイル行全体を選択します。3回目のクリックでマウスボタンを押し下げたままドラッグすると、ファイル行単位で近くの行も選択できます。

[選択]-[マーカー領域で始めた行選択でもテキスト領域では文字単位で選択する]

マーカー領域でマウスの左ボタンを押しながら上下に移動して行選択を始めた場合でも、マウスがテキスト領域にあるときは、文字単位の選択になります。

[選択]-[非アクティブ編集ウインドウの選択範囲を表示する]

非アクティブ状態の編集ウインドウでも選択範囲を表示します。色は、[表示色] ページでカスタマイズできます。

[選択]-[選択中もキャレットを点滅させる]

範囲を選択しているときもキャレットを点滅させます。

[選択]-[文字選択と矩形選択モードをロックする]

文字選択モード、行選択、矩形選択モードコマンドで始めた選択モードをロックします。以後は、Shiftキーを押していなくても、カーソル移動で選択範囲を拡大・縮小します。選択モードは、Escキーや任意の編集コマンドで自動的に解除されます。

[選択]-[行指定ジャンプで選択モードを解除しない]

文字選択モード、行選択、矩形選択モードコマンドで選択モードに入ったとき、[指定行ジャンプ]コマンドでジャンプしても選択範囲が拡大・縮小します。このオプションは[文字選択と矩形選択モードをロックする]がチェックされていているとき有効に動作します。

[選択]-[Alt+Drag の代わりに Ctrl+Drag で矩形選択を開始する]

ONにすると、Alt+左マウスボタンの代わりにCtrl+左マウスボタンで矩形選択を開始します。

[選択]-[ブラウザを開くコマンドで URL 部分を選択する]

カーソル位置のURLを使ってブラウザを開くコマンドで、使ったURL部分を選択する。

[選択]-[ダイレクトオープンコマンドで参照したパス部分を選択する]

カーソル位置のファイルパスを使ってファイルを開くコマンドで、使ったパスの部分を選択する。

[カーソル]-[Home キーで常にカラム 1 へカーソル移動する]

ONにすると、Homeキーで常に行の先頭へカーソル移動させます。OFFにすると、最初のHomeキーで行の最初の空白ではない文字へ移動し、2回目のHomeキーで行の先頭へカーソルを移動させます。

[カーソル]-[Page Up,Page Down キーを半ページ単位にする]

Page Up/Page Downキーの動作を半ページ単位にします。このオプションをOFFにすると、1ページ単位で上下ページへ移動します。

[カーソル]-[行頭・行末をこえてカーソルを移動させない]

←、→キーでのカーソル移動で、行頭、行末を越えて前後の移動をしないようにします。

[カーソル]-[右マウスクリックでキャレットを移動する]

ONにすると、右クリックでもキャレットを移動します。OFFにすると移動しません。

[カーソル]-[キー入力中はマウスカーソルを隠す]

編集のためにキー入力を始めるとマウスカーソルを隠します。

[カーソル]-[カーソル移動でマウスカーソルを隠す]

矢印キーでカーソル移動を始めるとマウスカーソルを隠します。

[カーソル]-[表示リフレッシュでカーソル行を画面の中央に移動する]

メニューの[表示]-[リフレッシュ]コマンドで、画面を描きなおすと同時にカーソルのある行を画面の中央となるようにスクロールします。

[カーソル]-[アクティブでない編集ウインドウでも行縦カーソルを表示する]

アクティブではない編集ウインドウにも行カーソルを表示します。

[カーソル]-[アウトプットウインドウにも行カーソルを表示する]

アウトプットウインドウにも行カーソルを表示します。このオプションは、Peggyを再起動した後に有効になります

[カーソル]-[下線/ボックス行カーソルを実線で描く]

ONにすると下線/ボックス行カーソルを実線で描画します。OFFにすると点線で描画します。

[カーソル]-[縦線/ボックス桁カーソルを実線で描く]

ONにすると縦線/ボックス桁カーソルを実線で描画します。OFFにすると点線で描画します。

[スクロール]-[垂直スクロールバーを表示する]

ウインドウの右端に垂直スクロールバーを表示します。

[スクロール]-[水平スクロールバーを表示する]

編集ウインドウの下端に水平スクロールバーを表示します。

[スクロール]-[スクロール中はカーソルをウインドウ内に留める]

スクロールバーにより編集ウインドウをスクロールさせるとき、現在のカーソル位置を保持します。これをOFFにすると、カーソルもいっしょにスクロールします。

[スクロール]-[スクロールアップ/ダウンでカーソルを同じ表示位置に留める]

Ctrl+↑、Ctrl+↓などのスクロールコマンドによりウインドウを上下スクロールさせるとき、現在のカーソル位置を保持します。これをOFFにすると、編集ウインドウの上下端までカーソルもいっしょにスクロールします。

[スクロール]-[インテリマウススムーススクロール]

インテリマウスドライバが有効になっている場合、チェックするとマウスホイールの回転でスムーススクロール、チェックを外すとジャンプスクロールします。

[スクロール]-[ホイール押し下げによる連続スクロールをしない]

マウスホイールの回転でスクロールしているとき、ホイールを押し下げると回転させなくてもスクロールは継続されますが、その連続スクロール動作をやめます。

[スクロール]-[編集禁止のテキストをSpaceキーでスクロールする]

編集禁止テキストを表示しているとき、Spaceキーで1/3ページスクロールアップします。また、Shift+Spaceでは、1/3ページスクロールダウンします。

[スクロール]-[編集禁止のテキストをBackspaceキーでバックスクロールする]

編集禁止テキストを表示しているとき、Backspaceキーで1/3ページスクロールダウンします。

[クリップボード]-[クリップボードから貼り付けた部分を選択する]

クリップボードからのテキスト貼り付け後、貼り付けた部分を選択します。

[クリップボード]-[クリップボードへコピーした後に選択範囲をキャンセルする]

選択領域をクリップボードへコピーした後、選択領域をキャンセルします。

[クリップボード]-[選択範囲なしのコピーを無視する]

選択範囲がないときに[編集]-[コピー]を実行してもコピー動作を無視します。クリップボードの内容は変化しません。

[クリップボード]-[選択範囲なしのコピーはカーソルのある行を対象にする]

選択範囲がないときに[編集]-[コピー]を実行した場合、カーソルのある行全体をクリップボードへコピーします。ファイル行番号表示モードのときはファイル行が、表示行番号モードのときは表示行が対象となります。

[クリップボード]-[選択範囲なしの切り取りはカーソルのある行を対象にする]

選択範囲がないときにクリップボードへの切り取りコマンドを実行すると、カーソルのある行全体を切り取り対象とします。ファイル行番号表示モードのときはファイル行を、表示行番号モードのときは表示行を切り取ります。

[クリップボード]-[クリップボードとのデータ交換では常に CR-LF 改行を使う]

Peggyはさまざまな改行コードのファイルを編集することができます。しかし、そのままのテキストデータをクリップボード経由で他のWindowsアプリケーションと交換すると、問題が生じかねません。このオプションをONにすると、クリップボードとのデータ交換の前後で改行コードをCR-LFに変換するので、スムースにカット&コピーが行なえます。ONにしておくことをお勧めします。

[クリップボード]-[コピー履歴からの貼り付けサブメニューを表示しない]

ポップアップメニューの[貼り付け]に、コピー履歴からの貼り付けサブメニューを表示しないようにします。

[IME]-[再変換終了で IME をオフにする]

再変換開始でオンにしたIMEを再変換終了で自動的にオフにします。

[IME]-[ウインドウを閉じるときに IME をオフにする]

編集中のウインドウを閉じると、IMEをオフにします。

[IME]-[変換キーによる再変換を禁止する]

変換キーで再変換を開始しないようにします。禁止してもポップアップメニューやショートカットからの再変換は可能です。

[HTML]-[HTML タグ挿入は大文字で行う]

HTMLタグ挿入コマンドを実行するとき、大文字で挿入します。

[HTML]-[対応括弧ジャンプで HTML 対応タグへのジャンプも行う]

対応括弧ジャンプコマンドで、HTMLの対応するタグでもジャンプをします。

[HTML]-[前後や対となる HTML タグへジャンプ後にタグを選択する]

HTMLタグへ移動するコマンドを実行したとき、そのタグを選択します。

[HTML]-[ペア HTML タグ削除で片方しかない場合でも削除する]

１対のHTMLタグを削除するコマンドで、相手となるタグがない場合でも、削除します。

[キーボードマクロ]-[キーボードマクロの操作をまとめて Undo する]

キーボードマクロを再生することにより実行した編集操作を1つのUndoコマンドで戻せるようにします。

[キーボードマクロ]-[エラーが発生したらキーボードマクロの再生を中止する]

キーボードマクロの再生中にエラーが発生したら、途中であってもキーボードマクロの再生を中止します。

[ドラッグ & ドロップ]-[ドラッグ & ドロップによる編集を有効にする]

選択範囲をドラッグ&ドロップで移動やコピーができるようになります。

[ドラッグ & ドロップ]-[ファイルパスのドロップ挿入を有効にする]

エクスプローラなどから編集ウインドウ上へCtrlキーを押しながらドロップすると、そのファイルのパスを挿入できるようにします。

[ドラッグ＆ドロップ]-[ファイルパスのドロップ挿入で相対パスに変換する(Shiftキーで取り消し)]

ドラッグ&ドロップで編集中のテキストへパスをドロップするとき、パスを編集中のテキストからの相対パスに変換します。Shiftキーでもフルパス・相対パスドロップを切り替えられます。

[ドラッグ＆ドロップ]-[ファイルパスのドロップ挿入で ￥ を / に変換する]

ドラッグ&ドロップで編集中のテキストへパスをドロップするとき、パス中の円記号(¥)をスラッシュ (/)の置き換えます。

[ドラッグ＆ドロップ]-[ファイルパスのドロップ挿入でパスを二重引用符で囲む]

ドラッグ&ドロップで編集中のテキストへパスをドロップするとき、パスを二重引用符で囲みます。

[ドラッグ＆ドロップ]-[プロジェクトウインドウからのドラッグ＆ドロップを許可する]

ドラッグ&ドロップによる編集中テキストへのパス挿入をプロジェクトウインドウからできるようにします。

[タブ・インデント]-[スマートインデント]

行の終わりでEnterキーを押したときや、言語毎に設定するトリガー文字を入力したとき、インデントを言語の規則に従って調整します。スマートインデントに対応していない言語では、オートインデントとして扱います。

[タブ・インデント]-[スマートインデントに空白もそのまま含める]

スマートインデントがONのときは、空白を残したままインデントを行ないます。OFFにすると、行の最初の空白ではない文字の位置が揃うようにタブやスペースを調整します。

[タブ・インデント]-[オートインデント]

行の終わりでEnterキーを押したとき、直前の行のインデントと同じインデントが付きます。

[タブ・インデント]-[オートインデントに空白もそのまま含める]

オートインデントがONのときは、空白を残したままインデントを行ないます。OFFにすると、行の最初の空白ではない文字の位置が揃うようにタブやスペースを調整します。

[タブ・インデント]-[オートインデントのとき前行と同じ空白文字を使う]

オートインデントのとき前行と同じ空白文字(スペース、タブ、全角スペース)を使ってインデントします。このオプションをOFFにすると、[スペースでタブ位置まで進める]の設定に従って、タブまたはスペースを使ってオートインデントを行います。

[タブ・インデント]-[全角スペースもインデントの一部として扱う]

全角スペースもインデントの一部としてカウントし、自動インデントの位置を決定します。

[タブ・インデント]-[空白だけの行もインデントする]

[まとめてインデント]のとき、空行や空白だけの行もインデントするようにします。

[タブ・インデント]-[スペースでタブ位置まで進める]

タブ文字を使わずスペースで次のタブ位置まで進めます。オートインデントもすべてスペース文字だけを使って行うようにします。

[タブ・インデント]-[２カラム以上空いている空白のみをタブへ変換する]

[スペースをタブに変換]コマンドで１カラムだけの空白はタブに変換しません。

[コメント]-[行コメントとブロックコメントをフレキシブルに挿入する]

言語によっては、行単位のコメントとブロック単位のコメントのどちらかしかない場合があります。そのようなときに、[行コメント挿入]コマンドや[ブロックコメント挿入]コマンドを実行すると、その言語にある方のコメントにコマンドを読み替えて挿入します。

[コメント]-[コメント挿入時にスペースを追加する]

[行コメント挿入]コマンドなどのコメント記号を挿入するコマンドで、コメント記号を挿入するときコメント記号とテキストとの間に空白を挿入します。

[コメント]-[コメント削除時に前後のスペースも削除する]

[コメント削除]コマンドで、コメントを削除するとき、コメントと他のテキストとの間にある空白も一緒に削除します。

[罫線]-[罫線引きでタブを使う]

罫線を縦方向に引くとき、その左に発生する空間をスペースだけで埋めるか、タブで埋めるかを選択できます。タブで埋めても端数はスペースになります。

[罫線]-[罫線引きの Undo/Redo する単位を小さくする]

長い罫線引きを取り消すとき、何回かに分けて行います。

[検索・比較・タグジャンプ]-[検索やタグジャンプでファイルの終わりを越えたら先頭へ戻る]

検索やタグジャンプでファイルの終わりまで検索したら、また先頭へ戻ります。

[検索・比較・タグジャンプ]-[検索ツール / 検索 / 置換でカーソル位置の単語をコピーする]

Ctrl+Fキーなどで検索ツールへカーソルを移動させるときや、[検索]ダイアログボックス、[置換]ダイアログボックスを表示させるとき、編集ウインドウのカーソル位置の単語をコピーします。

[検索・比較・タグジャンプ]-[検索ツールで編集ウインドウと同じ固定幅フォントを使う]

検索ツールのフォントを編集ウインドウと同じものを使うようにします。このオプションは、Peggy再起動後に有効になります。

【注意】編集ウインドウに大きなフォントを使っている場合は、検索ツールに入りきらないことがあります。そのような場合は、このオプションをOFFにしてください。

[検索・比較・タグジャンプ]-[クイック検索で正規表現モードをクリアする]

クイック検索で自動的に正規表現をOFFにします。

[検索・比較・タグジャンプ]-[タグジャンプでウインドウをリサイクルする]

タグジャンプで複数のファイルにジャンプするとき、一つのウインドウを共用してファイルを表示します。チェックをはずすとファイルごとに新しいウインドウを開きます。

[検索・比較・タグジャンプ]-[タグジャンプもバックジャンプ可能にする]

定義位置ジャンプだけでなくタグジャンプでもバックジャンプできるようにします。

[検索・比較・タグジャンプ]-[ファイル比較中に片方を閉じたら他方の背景色をクリアする]

比較している2つのウインドウの片方を閉じた場合、もう一方のウインドウの比較相違部分を示している背景色を自動的にクリアします。

[検索・比較・タグジャンプ]-[検索強調表示]-[ファイルから検索のタグジャンプで検索強調表示する]

ファイルから検索でタグジャンプするときも見つかった個所を検索強調表示します。

[検索・比較・タグジャンプ]-[検索強調表示]-[ファイルから検索で古い検索強調表示をクリアする]

ファイルから検索のタグジャンプで検索強調表示をしないときでも、古い検索結果の強調表示があればクリアします。

[検索・比較・タグジャンプ]-[検索強調表示]-[ファイル比較で検索強調表示をクリアする]

ファイル比較で、比較対象ファイルの検索強調表示をクリアします。

[検索・比較・タグジャンプ]-[検索強調表示]-[Esc キーで検索強調表示をクリアする]

Escキーで編集ウインドウの検索強調表示をクリアします。

[検索・比較・タグジャンプ]-[検索強調表示]-[画面のリフレッシュで検索強調表示をクリアする]

[表示]-[リフレッシュ]コマンドで、検索強調表示をクリアします。

[検索・比較・タグジャンプ]-[検索強調表示]-[空文字列の検索で検索強調表示をクリアする]

検索文字列を指定せずに検索を実行したとき、そのとき設定されている検索強調表示をクリアします。

[補完]-[候補絞込みが一意に決まる場合は大文字 / 小文字を自動変換する]

補完ポップアップリストによる入力中に、入力した英文字に一致する候補がない場合、大文字と小文字を反転させて候補が見つかれば大文字と小文字を反転させて入力します。

[補完]-[補完入力中は補完の範囲を超える Undo/Redo を禁止する]

補完ポップアップリストによる入力中に、補完を開始した時点より前に戻るUndoや、補完開始以前の操作をやり直すRedoコマンドを禁止します。これにより、Undoしすぎて候補ポップアップリストが消えてしまうのを防ぐことが可能です。

[補完]-[補完入力中は Undo/Redo を 1 文字単位で行う]

補完ポップアップリストによる入力中に行った文字の入力・削除のUndo/Redo動作を1文字単位で行うようにします。ただし、Tabキーなどにより、共通する部分をまとめて入力した場合を除きます。

[補完]-[共通部分を自動的に入力する]

1文字入力したあと、残った候補の共通する部分を自動的に挿入します。

[補完]-[共通部分の挿入では常に大文字小文字を区別する]

補完ポップアップリストによる入力中に、自動、またはTabキーによって共通部分を入力するとき、常に大文字/小文字を区別しながら完全に一致する部分だけを挿入するようにします。

[補完]-[一致しない文字入力で補完を終了する]

補完ポップアップリストによる入力中に、補完候補にない文字を入力すると、自動的に補完ポップアップリストを閉じます。

[補完]-[補完以外のコマンド実行で補完を終了する]

補完ポップアップリストによる入力中に、補完には関係しない編集コマンドを実行すると、自動的に補完ポップアップリストを閉じます。

[表示]-[改行・空白・タブ]-[改行 /EOF 記号を表示する]

改行記号[↲]と、EOF[◀]の表示・非表示を切り替えます。[表示]メニューの[改行記号]の左横のチェックの有無が現在の状態を表しています。以下は、改行コードと表示される改行記号の関係です。これらの記号の色は、[表示色]ページで設定できます。

改行コード	改行記号
CR-LF	↲
LF	↓
CR	←

[表示]-[改行・空白・タブ]-[タブを表示する]

タブ[▸]の表示・非表示を切り替えます。[表示]メニューの[タブ]の左横のチェックの有無が現在の状態を表しています。

[表示]-[改行・空白・タブ]-[タブによりできたスペースを･･･と表示する]

タブによりできたスペース部分を･･･と表示します。

[表示]-[改行・空白・タブ]-[半角スペースを表示する]

半角スペース文字のある場所に半角スペース記号[␣]を表示します。半角スペースの表示は、[表示]メニューからも変更できます。

[表示]-[改行・空白・タブ]-[全角スペースを表示する]

全角スペース文字のある場所に全角スペース記号[□]を表示します。全角スペースの表示は、[表示]メニューからも変更できます。

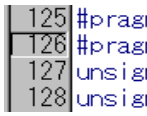
[表示]-[行番号・ルーラー]-[ルーラーを表示する]

ルーラーの表示・非表示を切り替えます。メニュー [表示]-[ルーラー]の左横のチェックの有無が現在の状態を表しています。

【ヒント】ルーラーのテキスト外領域との境界をマウスでドラッグし、折り返し位置を調整できます。境界が画面上にない場合でも、ルーラーの右端(マウスカーソルが変わります)からドラッグすることができます。

[表示]-[行番号・ルーラー]-[現在の行番号とルーラーのカラムをくぼめて表示する]

行番号表示部の現在行をくぼめて表示します。



[表示]-[行番号・ルーラー]-[行番号 / カラム番号をチップ表示する]

マウスカーソルをエディタ画面左端に近づけると、行番号をチップ表示するようになります。

[表示]-[テキスト外の領域を表示する]

折り返しカラムより右の部分(テキスト外の領域)を別の色で表示します。表示色は、[表示色]ページで設定できます。

[表示]-[自動的に対応する括弧を強調表示する]

カーソルを移動したとき、括弧を入力したときに対応する括弧を自動的に強調表示します。強調表示する括弧は、半角文字の(、)、{、}、[、]です。

[表示]-[対応括弧の強調表示を検索する範囲を 2000 行に広げる]

ONにすると、自動的に強調表示する括弧を探す範囲を前後2000行まで広げます。OFFにすると、前後250行の範囲で対応する括弧を探します。

[表示]-[キーワードチップを有効にする]

Peggyは、拡張子から言語を認識します。このオプションを指定すると、言語の命令、キーワードなどの文字列上にマウスカーソルを移動させると、キーワードチップが表示されます。命令実行によるフラグ変化などがマニュアルを見なくても理解できます。

[表示]-[ガイドラインを実線で描く]

ONにするとガイドラインを実線で描画します。OFFにすると点線で描画します。

[表示]-[ルーラー上にガイドライン位置を示すマークを表示する]

ONにするとルーラー上にもガイドライン位置を示すマークを表示します。

[表示]-[ルーラー上にガイドライン位置を示すマークを常に表示する]

ONにすると、ガイドライン表示のON/OFFにかかわらず、ルーラー上に設定されているガイドライン位置を示すマークを常に表示します。

[表示]-[文字コードインジケータに常に Shift JIS コードを表示する]

ONにすると、カーソル位置の文字コードを常にShift JISコードで表示します。OFFにすると、そのファイルの漢字コードで表示します。

[ファイル]-[外部で更新されたファイルを自動的に読み込む]

Peggyがファイルを開いているとき、そのファイルがPeggy以外のエディタで編集され保存されたときに、そのファイルを自動的に読み込みます。ただし、Peggyでも変更されている場合は、問い合わせのダイアログボックスが表示されます。

[ファイル]-[開いている間はファイルを書き込み禁止にする]

開いているファイルの属性を書き込み禁止にします。このことにより、他のエディタでファイルを開こうとすると書き込み禁止状態となり、ファイルの二重編集を防ぐことができます。

[ファイル]-[ファイルダイアログボックスにプレースバーを表示する]

ファイルを開く、保存ダイアログボックスに、プレースバーを表示します。Windows 2000以上でのみ有効です。

[ファイル]-[プロジェクトアイテムのホイールクリックでファイルを閉じる]

開いているファイルのプロジェクトアイテムをマウスのホイールボタンクリックで閉じます。

[ファイル]-[ウインドウリストのホイールクリックでファイルを閉じる]

ウインドウリストの項目をマウスのホイールボタンでクリックすることにより、対応するウインドウを閉じます。

[ファイル]-[開く]-[書き込み禁止ファイルは編集禁止にする]

書き込み禁止属性のファイルを開いたとき編集できないようにします。文字入力はすべて無視されます。

一時的に編集を許可するには、ファイルのプロパティの[ファイル]ページにある[編集禁止]のチェックをはずしてください。

[ファイル]-[開く]-[openFile() で開いたファイルをヒストリに追加する]

[ファイル]-[開く]-[openFile() で開いたファイルを MRU リストに追加する]

MocaScriptのopenFile()関数でファイルを開いたとき、それぞれファイルヒストリと、ファイルメニューの[最近使ったファイル]、[最近使ったフォルダ]リストへ登録します。

[ファイル]-[開く]-[アクティブドキュメントのフォルダを基準に [開く] を表示する]

[ファイル]-[開く]で表示する[開く]ダイアログボックスの初期フォルダを現在のアクティブドキュメントのフォルダにする。

[ファイル]-[開く]-[プロジェクトウインドウからファイルを開いた後プロジェクトウインドウを閉じる]

プロジェクトウインドウからファイルを開くと、プロジェクトウインドウを閉じます。

[ファイル]-[開く]-[コマンドラインで指定されたファイルが存在しないときは作成する]

コマンドラインで指定されたファイルが存在しなかったとき、空のファイルを作成して開きます。

[ファイル]-[開く]-[警告]-[長すぎる行で警告を出す]

Peggyは1行65000バイトまでしか扱うことができません。このオプションをONにすると、ファイル読み込み中にこの制限を越えた行があったことを知らせる警告ダイアログボックスを表示します。

[ファイル]-[開く]-[警告]-[不正な文字コードに警告を出す]

いろいろな漢字コードのファイルには、Shift JISでは表わせない文字を含んでいることがあります。Peggyは内部的にShift JISコードを使っているので、Shift JISに変換できない文字は失われてしまいます。このオプションをONにしておくと、明

らかに変換できない文字を含んだファイルを読みこむときに、警告ダイアログボックスを表示します。

【注意】Unicodeテキストファイル中にShift JISへ変換できない文字が含まれていた場合は、このオプションをONにしていても警告ダイアログボックスは表示されません。その代わり変換できなかった文字をクエスチョンマーク(?)と表示します。UTF-8では、UTF-8として不正なためUnicodeへ変換できない文字の警告は検出しますが、UnicodeからShift JISへ変換できなかった文字の警告ダイアログボックスは表示されません。

【注意】Shift JISファイルの読み込みには変換が不要なため、不正文字コードのチェックは行ないません。すべての情報をそのままの形で読み込みます。

[ファイル]-[開く]-[警告]-[変換できない文字に警告を出す]

ファイルを開くとき、UTF-7/8/8Nから内部コードへ変換できなかった文字があったことを警告します。

[ファイル]-[開く]-[警告]-[変換できない文字のあった行の背景色を変える]

ファイルを開くとき、UTF-7/8/8Nから内部コードへ変換できなかった文字があった行の背景色を変えます。

[ファイル]-[開く]-[警告]-[巨大なファイルに警告を出す]

ファイルサイズの大きなファイルを開こうとすると、時間がかかることがあります。間違って巨大なファイルを開いたとき待たなければいけません。それを防ぐための主記憶の半分を超える大きなファイルを開こうとしたときに、警告ダイアログボックスを表示します。

[ファイル]-[保存]-[保存のときに行末のタブとスペースを削除する]

行末の文字とリターンコードの間にスペースやタブがある場合、これらを削除してから保存します。

[ファイル]-[保存]-[保存のときに行末の改ページと垂直タブを削除する]

行末の文字とリターンコードの間に改ページコード(0x0c)や垂直タブ(0x0b)がある場合、これらを削除してから保存します。

[ファイル]-[保存]-[保存のときに行末の全角スペースを削除する]

行末の文字とリターンコードの間に全角スペースがある場合、これらを削除してから保存します。

[ファイル]-[保存]-[保存のとき選択されているフィルタの拡張子を追加する]

[名前を付けて保存]ダイアログボックスの[ファイル名]に拡張子なしの名前を指定したとき、選択しているフィルターの拡張子を自動的に追加します。

[ファイル]-[保存]-[保存確認のとき編集ウインドウを前面に表示する]

複数のファイルを保存、または複数の編集ウインドウを閉じるとき、未保存のファイルの処理を問い合わせるとき、編集ウインドウを前面に表示します。

[ファイル]-[保存]-[テンポラリファイルを使わず上書き保存する]

保存のとき、テンポラリファイルを使わず直接ファイルへ上書きします。テンポラリファイルを使って保存する方が安全ですが、それでは不都合のある場合、例えばSambaなどで共有しているファイルの実行パーミションなどをそのままにして保存したい場合にはチェックしてください。

[ファイル]-[保存]-[バックアップファイルを残す]

バックアップファイルを作成します。ファイル名は、元の<ファイル名+.bak>になります。

[ファイル]-[文字コード]-[JIS ファイルへ半角カナを保存するとき 7-bit に変換する (ESC, '(', 'I'))

JISコードで半角カナを含んだテキストを保存するとき、エスケープシーケンスを使って7ビットに変換します。このオプションをOFFにすると、半角カタカナを8ビットのまま保存します。

[ファイル]-[文字コード]-[JIS ファイルへ ASCII 文字を保存するとき JIS X 201 へシフトする]

このオプションをONにすると、JISコードのファイルを保存するとき、アスキーコードの部分でJIS X 201へシフトするエスケープシーケンス(ESC (J)を使います。OFFの場合は、USASCIIにシフトするESC (Bを使います。

[ファイル]-[文字コード]-[EUC/JIS ファイルへ保存するとき IBM 拡張漢字を変換する]

[ファイル]-[文字コード]-[EUC/JIS ファイルから読み込むとき IBM 拡張漢字へ変換する]

EUCやJISファイルへの保存と読み出しで、IBM拡張漢字コードをマイクロソフト拡張文字に変換してから保存/読み出しを行います。デフォルトは両方ともONです。ただし、Ⅰ(FA4A)～Ⅹ(FA53)、㈱(FA58)、№(FA59)、℡(FA5A)、∵(FA5B)は、保存の時のみ変換し、開くときは変換しません。Shift JISの￣(FA54)は、保存時に￣(81CA)に変換します。読み出し時の￣(EEF9)も￣(81CA)に変換します。￣(FA54)、￣(EEF9)を両方とも￣(81CA)に変換するのは、Shift JISとUnicode間の変換方法に近づけた結果です。

【注意】同じ文字に2つの文字コードが割り当てられているため、Shift JIS→EUC→Shift JISと変換したとき、元に戻るとは限らない(不可逆変換)ので注意してください。勝手な変換を止めたい場合は、これらのオプションをOFFにしてください。

[ファイル]-[文字コード]-[UTF-7 ファイルへ保存するとき ¥ や ~ もエンコードする]

UTF-7でファイルへ保存するとき、UTF-7の仕様に厳密に従い、円記号(¥)やチルダ(~)もエンコードします。

[ファイル]-[文字コード]-[UTF-7 を自動判定の対象外とする]

ASCII文字だけのテキストと見分けの付きにくいUTF-7ファイルを自動判定の対象外とします。

[ファイル]-[文字コード]-[UTF-8 ファイルへ保存するとき 0u0000 を 0xC0 0x80 にエンコードする]

UTF-8でファイルを保存するとき、テキスト中に0u0000が存在したら、それを0xC0 0x80にエンコードします。

[ファイル]-[文字コード]-[ASCII だけのファイルを開くときは新規ファイルの漢字コードを適用する]

ASCII文字だけのファイルを開くとき、同じ[編集]ページの[漢字コード]で指定されている漢字コードでファイルを開きます。

【注意】プロジェクトに登録されているファイルには適用されません。

[ファイル]-[文字コード]-[XML 宣言 <?xml ...?> も漢字コードを決めるヒントとして使う]

ASCII文字だけで構成されているファイルを開くとき、先頭行にXML宣言があれば、その中のencoding=”UTF-8”のような部分を使って漢字コードを判定します。XML宣言の中にencoding指定がない場合は、UTF-8であると判定します。

[ポップアップメニュー]-[ポップアップメニューをカーソルの近くに表示する]

キーボードからポップアップメニューを表示させたとき、キャレットの近くにメニューを表示します。このオプションをOFFにすると、マウスカーソルの近くに表示します。

[ポップアップメニュー]-[ポップアップメニューを1行下に表示する]

ポップアップメニューが編集している行に重ならないよう1行下に表示します。

[ポップアップメニュー]-[ユーザ定義メニューをできるだけカーソル位置に表示する]

ユーザ定義ポップアップメニューをできるだけキャレットの近くに表示します。このオプションがOFFか、アクティブな編集ウインドウがないときは、マウスカーソルの近くに表示します。

[ウインドウ]-[タイトルをフルパスで表示する]

ウインドウのタイトルバーに表示されるファイル名をフルパスで表示します。チェックをはずすと、ファイル名のみの表示になります。

[ウインドウ]-[タイトルにプロジェクト名を先に表示する]

タイトルバーの先頭にプロジェクト名を表示します。

[ウインドウ]-[できるだけ上下に大きなウインドウを開く(Shiftキーで取り消し)]

[ウインドウ]-[できるだけ左右に大きなウインドウを開く(Shiftキーで取り消し)]

このオプションをONにすると、それぞれ縦方向、横方向にできるだけ大きなサイズで編集ウインドウを開きます。大きく開く動作は、Shiftキーを押しながらファイルを開くことで逆転させることができます。つまり、このオプションをOFFにしていてもShiftキーを押しながらファイルを開くと大きく、逆にオプションONのときShiftキーを押しながら開くと通常の大きさになります。

[ウインドウ]-[アクティブなウインドウに重ねて開く]

このオプションをONにすると、新しいウインドウを開く直前にアクティブだったウインドウに重なるように新しいウインドウを開きます。

[ウインドウ]-[レイアウト変更を3通りにする(ウインドウ幅、固定幅、折り返しなし)]

レイアウト変更コマンドでウインドウ幅、折り返しなしに加えて、固定幅へも切り替えられるようにします。

[ウインドウ]-[ダイアログボックスの位置を記憶する]

多くのダイアログボックスの位置を記憶し、以前表示されたのと同じ場所へ表示します。OFFにすると、Peggyの中央に表示します。

[ウインドウ]-[Ctrl+F6で2つのウインドウを交互に切り替える]

Ctrl+F6キーで重なり順が上の2つの編集ウインドウを交互に切り替えられるようにします。OFFにすると、Ctrl+F6で重なり順の逆順で編集ウインドウをアクティブにします。

[ウインドウタブ]-[ソートする]

ウインドウタブを名前順にソートして表示します。チェックを外すと、開いた順に表示します。

[ウインドウタブ]-[下端に表示する]

ウインドウタブを下端に表示します。チェックを外すと上端に表示します。

[ウインドウタブ]-[アイコンを表示する]

ウインドウタブにドキュメントのアイコンも表示します。

[ウインドウタブ]-[固定幅（最大タブ幅）で表示する]

ウインドウタブを[最大タブ幅]で指定した固定幅で表示します。

[ウインドウタブ]-[変更ありを示すアンダーラインを表示する]

変更のあるファイルは、タブに表示されている名前にアンダーラインを付けます。

[ウインドウタブ]-[フルパスをツールチップ表示する]

タブにマウスを近づけると、フルパスまたはフルタイトル文字列をツールチップ表示します。

[ウインドウタブ]-[右マウスクリックでも選択する]

ミグマウスボタンでクリックしたときも、その編集ウインドウをアクティブにします。

[ウインドウタブ]-[Shift+Click でウインドウを閉じる]

Shiftキーを押しながらタブをクリックすると、編集ウインドウを閉じます。

[ウインドウタブ]-[ホイールクリックでウインドウを閉じる]

中マウスボタン(ホイール)でクリックしたとき、その編集ウインドウを閉じます。

[ウインドウタブ]-[ダブルクリックで最大化 / 最大化解除する]

タブをダブルクリックすると、編集ウインドウを最大化します。またすでに最大化されている編集ウインドウは、通常の大きさに戻ります。

[ウインドウタブ]-[タブの左右端が連続しているかのように扱う]

左右のタブを選択するコマンド、右隣/左隣のタブと入れ替えるコマンドで、左端と右端が連続しているかのように処理します。

[ウインドウタブ]-[左右端まで閉じるに「クリックしたウインドウ」も含める]

ウインドウタブポップアップメニューの[左端まで閉じる]、[右端まで閉じる]コマンドで、閉じる対象にクリックしたタブのウインドウ自身を含めるかどうかを指定します。

[ウインドウタブ]-[同時に複数のウインドウを閉じるときは確認する]

ONにすると、すべて閉じるや左右端まで閉じるコマンドで複数のウインドウが同時に閉じられる場合に確認メッセージが表示されます。

[その他]-[ツール実行の結果が Undo できない可能性のある場合に警告を出す]

スクリプトファイルなどを実行した結果が巨大な場合、アンドゥできないことがあります。そのおそれがある場合に、実行前に警告ダイアログボックスを表示します。

[その他]-[編集ウインドウの Esc キーでアウトプットウインドウを閉じる]

編集ウインドウでEscキーを押すと、アウトプットウインドウが閉じるようにします。キーボードから手を離さずにアウトプットウインドウを閉じることができます。

[その他]-[複数言語をインストールしたシステムで発生する描画問題を修正す

る]

日本語以外に複数言語をインストールしたWindows 2000システムでは、タブの部分が描画されず、背景がそのまま残ってしまう場合があります。このオプションをチェックすることで、タブの部分を塗りつぶすように強制します。

[その他]-[キャレットの移動跡を再描画する]

キャレット(カーソル)移動跡を再描画するようにします。ビデオカード、ドライバ、Peggyの相性などにより、万が一キャレット跡が残ってしまう場合に、ONにしてみてください。

[その他]-[0.5 秒以内の連続したビープを間引く]

各種操作時の警告ビープ音が0.5秒以内に連続するときそれを間引いて、1回のビープ音にします。

[その他]-[キーワードファイルの保存と同時にリロードする]

キーワードファイルを保存すると、即座にコンテキスト正規表現カラー表示(CREC)で再ロードします。保存すると、すぐにその効果を確かめることができます。

[その他]-[スクリプトからレジストリへの書き込みで警告を出す]

MocaScriptからレジストリへ変更を行う直前に警告ダイアログボックスを表示します。レジストリを誤って変更すると、最悪システムが不安定になったり、起動しなくなることがあり、その防止に役立ちます。ONのままにしておくことを推奨します。

[その他]-[必要に応じて <bin> フォルダのパスを Peggy プロセスの環境変数に追加する]

Peggyからコマンドライン版moca.exeを実行するときなどに、PATH環境変数に<bin>フォルダへのパスが設定されていない場合、PeggyのPATH環境変数に<bin>フォルダのパスを追加します。このオプションを有効にするには、Peggyを再起動してください。

[その他]-[マウスクリックでアクティブになるとき選択範囲を変更しない]

他のアプリケーションを使っているとき、Peggyの編集ウインドをクリックしてアクティブにしたとき、そのマウスクリックで選択範囲が変化しないようにします。

[その他]-[ブックマークをすべてクリアするときに確認メッセージを表示する]

作業中のファイルのブックマークをすべて削除するコマンドで、確認メッセージを表示します。

[実行前保存]

プロジェクトのマウスアクション、ツールアクションが[実行する]で、それを実行するとき、変更しているテキストの保存方法を選択します。

選択	動作
[保存しない]	実行時に、変更のあるファイルがあっても保存しません。

[保存する]	実行時に、ファイルが変更されていると、保存するかどうかの確認ダイアログボックスが表示されます。その実行に関係のないファイルもすべて問い合わせします。
[自動保存する]	実行前に、ファイルが変更されていると、自動的に保存します。

[挿入モード]

エディタ文字入力の[挿入][上書]モードの切り替えを固定したいときに使います。

選択	動作
[INSキー切り替え]	Insertキーで、挿入、上書きモードを切り替えることができます。
[常に挿入モード]	編集中常に挿入モードになります。Insertキー切り替えは効きません。
[常に上書きモード]	編集中常に上書きモードになります。Insertキー切り替えは効きません。

[メール引用]

[編集]-[挿入と削除]-[メール引用符挿入]コマンドで選択範囲の各行の先頭に挿入する文字列を指定します。

新規ファイル

[漢字コード]

新規ファイルを作成したときの漢字コードを設定します。漢字コードは、ファイルに保存するときに変更することもできます。既存のファイルを開いたときの漢字コードには影響しません。

[改行コード]

新規ファイルを作成したときの改行コードを設定します。改行コードは、ファイルに保存するときに変更することもできます。既存のファイルを開いたときの改行コードには影響しません。

[タブ間隔]

新規ファイルを作成したときのタブ間隔を指定します。既存のファイルを開いたときのタブ間隔には影響しません。

[挿入カーソル]、[上書カーソル]

文字入力用カーソルの形状を変更します。[挿入][上書]モードにそれぞれ別形状のカーソルを使うことができます。

選択	意味	形状
[アイビーム]	アイビーム型カーソル	
[下線]	下線カーソル	

[文字下半分]	文字の下半分の大きさのカーソル	
[文字全体]	文字全体の大きさのカーソル	

[スクロール数]

インテリマウスのホイール回転に合わせて何行スクロールするかを設定します。

[加速スクロール]

上下スクロールを高速化する倍率を設定します。PCのビデオカード/ドライバによっては画面がチラついたり、設定した速度でスクロールできない場合もあります。

[加速遅延]

上下スクロールを始めてから何行スクロールした時点で高速化するかを指定します。1～100の範囲でスクロールの加速を遅らせることができます。

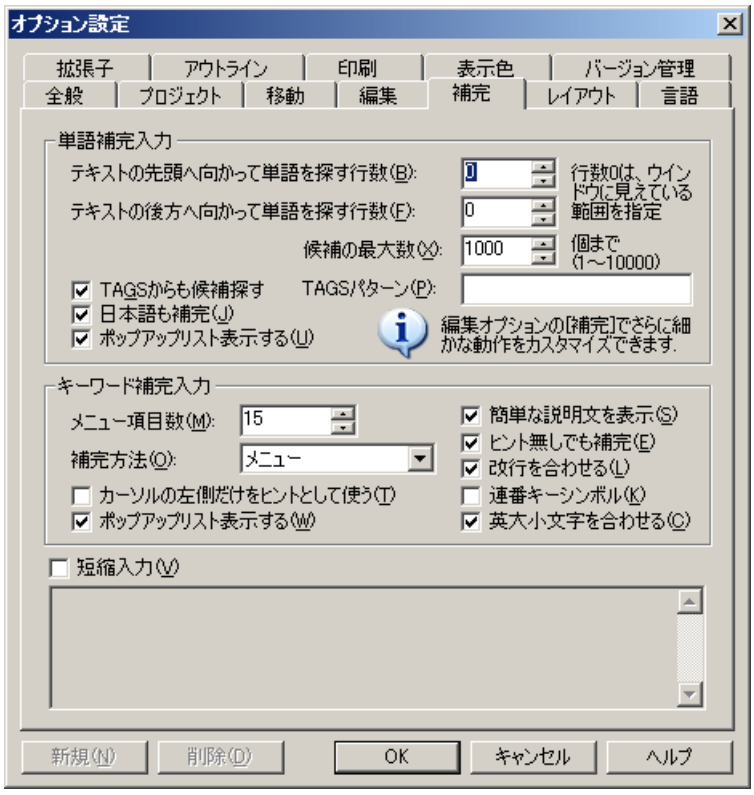
[上下スクロールマージン]

編集ウインドウの上下端からそれぞれ何行あけてスクロールさせるかを指定します。行数の少ないウインドウでは、ここで設定した行数よりもマージン行数よりも少なくなることがあります。

[最大タブ幅]

[固定幅(最大タブ幅)で表示する]オプションがONになっているときは、ここで設定した幅(ドット単位)でウインドウタブを表示します。OFFのときは可変タブ幅で表示しますが、ここで指定した幅までに制限されます。

[補完] ページ



単語補完入力

[テキストの先頭へ向かって単語を探す行数]

補完すべき文字列を何行前まで探すかを指定します。

[テキストの後方へ向かって単語を探す行数]

補完すべき文字列を何行後ろまで探すかを指定します。

[候補の最大数]

候補を何個まで探してメニュー表示するかを指定します。指定した前後の検索範囲行数内に多くの候補があるときは、近いものから順番に、ここで指定した個数までの候補をメニュー表示します。1を指定すると、メニューを表示せず、最初に見つかった候補で即補完します。

[TAGS からも候補を探す]

単語補完の候補をアウトプットウインドウの[TAGS]ページからもピックアップします。ピックアップ中に[候補の最大数]に達した場合は、そこで中止します。

[TAGS パターン]

アウトプットウインドウの[TAGS]ページから補完候補をピックアップするのに使う正規表現パターンを指定します。正規表現パターンには、必ず1対の括弧で候補として取り出す部分を指定してください。括弧が2対以上ある場合は、最初の括弧で示された部分の文字列が候補となります。正規表現を指定しないと、デフォルトで以下のパターンに一致する候補をピックアップします。

¥<(ヒント文字列[0-9A-Za-z_$¥-]+)[([,;]?[¥x01¥x7f]

ここで指定されたTAGSパターンは、アウトプットウインドウの[TAGS]ページの各行につき1回マッチするかどうかテストします。1回でピックアップしたい文字列へヒットするようなパターンを指定してください。さらに特別な方法で補完候補をピックアップするには、MocaScriptのEvent.onCompleteWord(theView,hint)イベントハンドラを定義してください。

[日本語も補完]

日本語の文字列の「漢字とカタカナの後にひらがなが続いたもの」も補完の対象とします。

[ポップアップリストを表示する]

メニューではなく補完ポップアップリストを使います。リストを表示したままキーを入力することで、リアルタイムに候補を絞り込むことが可能です。補完ポップアップリストの動作は、[編集]ページの[編集オプション]-[補完]で、さらに細かくカスタマイズすることができます。

キーワード補完入力

[メニュー項目数]

キーワード補完で複数の補完候補がある場合に表示されるポップアップメニューの最大項目数を制限します。候補がここで設定した数よりも多い場合、メニューを分割して見やすくします。

[補完方法]

補完コマンドの動作を以下から選択します。このオプションは、単語補完入力の動作にも影響します。

選択	動作
[メニュー]	補完候補をメニュー形式で表示します。
[予備補完+メニュー]	補完候補のうちすべてに共通する部分を予め編集中のテキストに挿入してから、補完候補をメニュー表示します。
[予備補完のみ]	補完候補のうちすべてに共通する部分を編集中のテキストに挿入し、終了します。メニューは表示されません。

[カーソルの左側だけをヒントとして使う]

カーソルの左側にある文字だけをヒントとして補完候補を検索します。

[ポップアップリストを表示する]

メニューではなく補完ポップアップリストを使います。リストを表示したままキーを入力することで、リアルタイムに候補を絞り込むことが可能です。補完ポップアップリストの動作は、[編集]ページの[編集オプション]-[補完]で、さらに細かくカスタマイズすることができます。

[簡単な説明文を表示]

キーワード補完で複数の補完候補がある場合に表示されるポップアップメニューに、キーワードの説明文を付けます。説明文は、キーワードチップで表示されるもとの同じです。説明文のないキーワードもあり、その場合はこのオプションをONにしても表示されません。

[ヒント無しでも補完]

カーソル位置にキーワードのヒントとなる文字列がない状態で単語補完コマンドを実行すると、すべての候補をメニュー表示します。

[改行を合わせる]

保管した文字列の一部に改行文字が含まれていた場合、それがどのようなものでも編集中のファイルの改行記号に合わせて変換してから挿入します。

[連番キーシンボル]

補完候補をメニュー表示するとき、連番キーシンボルを付けるか、補完候補内の一文字をキーシンボルとするかを選択します。チェックすると、メニューに1、2、・A9、A、B、・AZのようなキーシンボルを付けます。Offにすると、メニュー項目の一文字をキーシンボルとして選択します。このオプションは、単語補完入力の動作にも影響します。

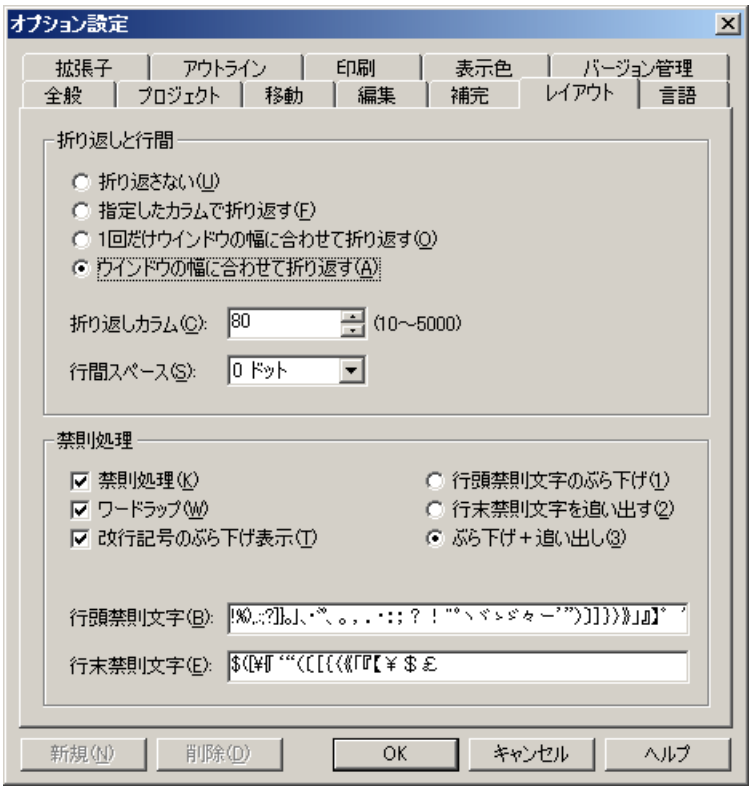
[英大小文字を合わせる]

大文字小文字を区別しない補完入力で、補完後の文字列の大文字小文字を選択した候補と同じになるように合わせます。

[短縮入力]

ワープロの単語登録のような機能です。短縮文字列の次にスペースを空けて、入力文字列を記述しておくと、短縮文字列をキー入力すれば登録文字列に変換することができます。この短縮入力を使うには、チェックをONにします。

[レイアウト] ページ



折り返しと行間

[折り返さない]

長い行でも折り返さず表示します。

[指定したカラムで折り返す]

長い行を[折り返しカラム]で指定したカラムで折り返して表示します。

[1 回だけウインドウの幅に合わせて折り返す]

ファイルを開いたときのウインドウの幅に合わせて折り返し表示しますが、それ以降ウインドウの幅を変えても折り返し位置は変化しません。

[ウインドウの幅に合わせて折り返す]

長い行をウインドウの幅に合わせて折り返し表示します。ウインドウをリサイズすると、それに合わせて折り返し位置も変化します。

【注意】何メガバイトもある巨大なファイルを編集中は、折り返し位置を再計算するのに多少時間がかかります。そのため、Windowsのフルドラッグ(内容を表示したままドラッグ)オプションを設定している場合は、ウインドウのリサイズがスムースではなくなることがあります。

[折り返しカラム]

[指定したカラムで折り返す]を選択しているときに、折り返す幅を指定します。

[行間スペース]

編集ウインドウの行間スペースを0～5ドットの範囲で指定します。上下に隙間のないフォントを使うときは、少し行間スペースを空けた方が見やすくなります。行間

スペースは、言語毎に間隔を変えることもできます。詳しくは、[オプション設定]ダイアログボックス[言語]ページを参照してください。

禁則処理

[禁則処理]

チェックすると禁則処理を行います。

[ワードラップ]

チェックすると、長い行の折り返し表示位置にかかってしまう単語を次行へ送ります。英文が読みやすくなります。

[改行記号のぶら下げ表示]

折り返し表示をしているとき、改行記号が折り返し位置からちょうどはみ出してしまう場合、改行記号を次行に送らず、ぶら下げ表示にします。

[行頭禁則文字のぶら下げ]、[行末禁則文字を追い出す]、[ぶら下げ＋追い出し]

禁則処理の方法を指定します。

[行頭禁則文字]

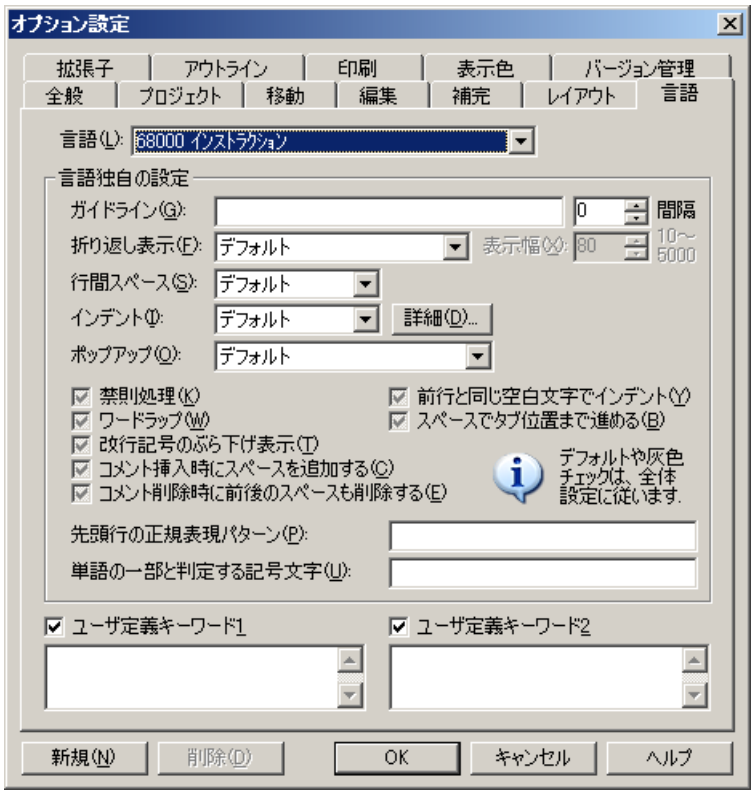
[禁則処理]をチェックしているとき、行の折り返し表示でここに指定した文字が次行の先頭へくる場合、その文字を次行へ送らず、折り返し位置を越えてぶら下げ表示します。

[行末禁則文字]

[禁則処理]をチェックしているとき、行の折り返し表示でここに指定した文字が行の最後になってしまう場合、次行へ送ります。

【ヒント】Peggyはエディタなので、最もシンプルな禁則のみを処理します。ワードプロセッサのように、高度な禁則処理は行いません。

[言語] ページ



[言語]

ユーザ定義キーワードを参照したり変更する言語を選択します。

言語独自の設定

編集ウインドウのレイアウトは、[レイアウト]ページで設定しますが、さらに言語ごとに変えることが可能です。

[ガイドライン]

ガイドライン(縦線)の桁位置(1～)を指定します。複数設定するときは、コンマで区切ります。ガイドラインを一定間隔で繰り返し表示したいときは、後ろの欄に何桁間隔で表示するかを指定してください。繰り返しガイドラインは、絶対桁位置で指定した最後の桁から等間隔で表示されます。繰り返しガイドラインに0を指定すると、繰り返しガイドラインは表示しません。

[折り返し表示]、[表示幅]

この言語のテキストを編集しているときの折り返し方法と、固定幅表示のときの折り返しカラムを指定します。[デフォルト]を選択すると、[レイアウト]ページの設定に従います。

[行間スペース]

この言語のテキストを編集しているときの行間スペースを設定します。[デフォルト]を選択すると、[レイアウト]ページの行間スペース設定に従いますが、それ以外の値を選ぶと、ここでの設定が優先します。

[インデント]、[詳細]

言語のインデント方法を、[デフォルト]、[なし]、[オート]、[スマート]から指定します。デフォルトを選択すると、[編集]ページの[編集オプション]の設定に従います。また、[詳細]ボタンをクリックすると、[スマートインデント詳細設定]ダイアログボックスにより、スマートインデントのスタイルをカスタマイズできます。カスタマイズできる項目は、言語毎に異なっています。

[ポップアップ]

編集ウインドウを右マウスボタンでクリックしたときに表示されるポップアップメニューを指定します。デフォルト(標準)メニューの他に、12種類のユーザ定義メニューから選択できます。ユーザ定義メニューの設定方法は、[カスタマイズ]ダイアログボックスの[メニュー]ページを参照してください。

[禁則処理]

禁則処理のON/OFFを言語独自に設定します。チェックボックスが灰色になっているときは、デフォルトとして[レイアウト]ページの設定に従います。

[ワードラップ]

ワードラップのON/OFFを言語独自に設定します。チェックボックスが灰色になっているときは、デフォルトとして[レイアウト]ページの設定に従います。

[改行記号のぶら下げ表示]

改行記号ぶら下げ表示のON/OFFを言語独自に設定します。チェックボックスが灰色になっているときは、デフォルトとして[レイアウト]ページの設定に従います。

[コメント挿入時にスペースを追加する]

[行コメント挿入]コマンドなどのコメント記号を挿入するコマンドで、コメント記号を挿入するときコメント記号とテキストとの間に空白を挿入します。チェックボックスが灰色になっているときは、デフォルトとして[編集]ページの[編集オプション]の[コメント]-[コメント挿入時にスペースを追加する]に従います。

[コメント削除時に前後のスペースも削除する]

[コメント削除]コマンドで、コメントを削除するとき、コメントと他のテキストとの間にある空白も一緒に削除します。チェックボックスが灰色になっているときは、デフォルトとして[編集]ページの[編集オプション]の[コメント]-[コメント削除時に前後のスペースも削除する]に従います。

[前行と同じ空白文字でインデント]

オートインデントのとき、前行と同じ空白文字を使ってインデントします。チェックボックスが灰色になっているときは、デフォルトとして[編集]ページの[編集オプション]の設定に従います。

[スペースでタブ位置まで進める]

タブの代りにスペースを使ってタブ位置まで進める機能を言語独自に設定します。チェックボックスが灰色になっているときは、デフォルトとして[編集]ページの[編集オプション]の設定に従います。

[先頭行の正規表現パターン]

拡張子マップで言語モードが決定できないときや、ファイル名に拡張子がない場合、Peggyはファイルの先頭行を見て、ここに書かれた正規表現と一致するパターンがあれば、この言語と判断します。

[単語の一部と判定する記号文字]

ダブルクリックや単語選択コマンドで選択する「単語」を構成する文字に、ここで指定した文字を追加します。

【注意】ここで指定した文字は、ダブルクリックや単語選択コマンドに影響するだけで、カラー表示には影響しません。

[ユーザ定義キーワード1]・[ユーザ定義キーワード2]

言語特有の予約語やキーワード以外に、ユーザ独自に定義した文字列をユーザ定義キーワードとして登録できます。ユーザ定義キーワードには2つのカテゴリを設けることができ、それぞれエディタ上での色分け表示が可能です。色は、[表示色]ページで行います。

各キーワードは空白や改行で区切って入力してください。1行に複数個のキーワードを入力してもかまいません。

[新規]

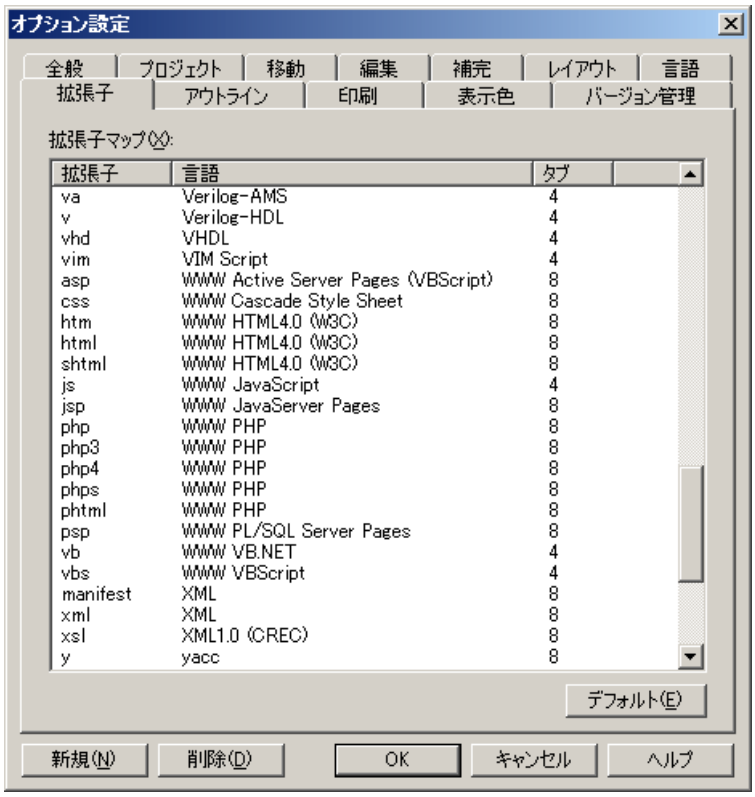
[言語]のリストにない言語を新たに追加したい場合、このボタンをクリックします。ユーザが独自に規定した拡張子を持つファイルに新規言語を対応させることで、ユーザ定義キーワードの色を変えてエディタ上に表示したり、印刷時に独自のフォーマットを指定することができるようになります。

ユーザ定義の言語では、ユーザ定義キーワードの指定、タブ間隔、印刷オプションの設定をすることができますが、独自の文法規則を定義することはできません。あくまで特定の文字列を強調表示したり印刷フォント、タブ間隔を変えるために使ってください。

[削除]

[新規]として追加したユーザ言語を削除するときに使います。最初から定義されている言語は、削除できません。

[拡張子] ページ



[拡張子マップ]

拡張子と言語モードとタブ設定の組み合わせを一覧できます。また、拡張子に対応する言語を変更したい場合は、拡張子に対応した言語をクリックします。現在登録されている言語のリストが表示されますから、選択してください。

[新規]

拡張子の欄に新規拡張子を入力するウインドウが表示されます。

[削除]

[新規]で追加した拡張子を削除するときに使います。Deleteキーでも削除できます。

[デフォルト]

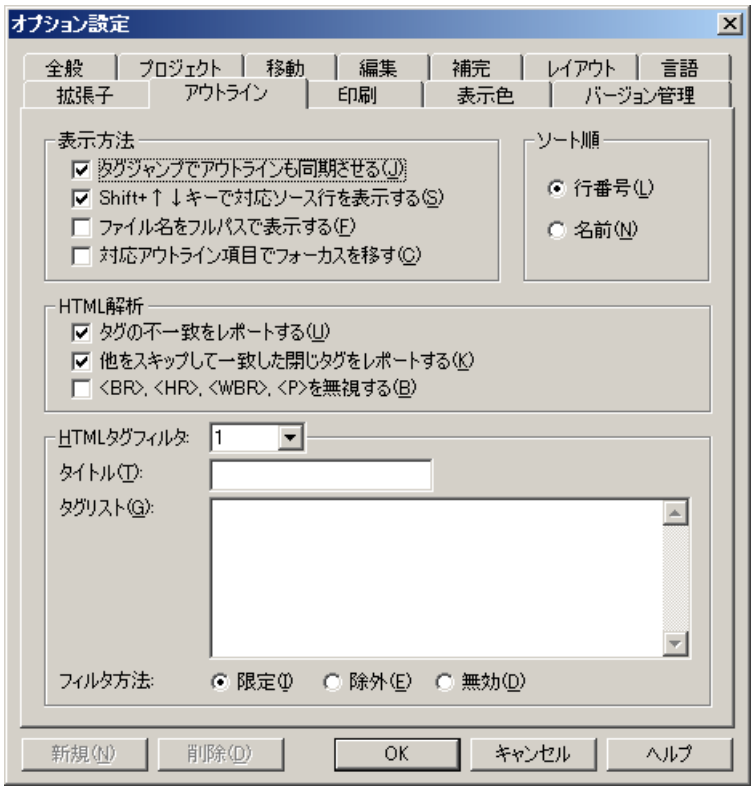
拡張子と言語モードの対応を変更しているときに[デフォルト]ボタンをクリックすると、初期設定状態に戻ります。

ソート

[拡張子マップ]の拡張子、言語、タブのそれぞれのヘッダをクリックすれば、リスト表示順をソートすることができます。ソートは、昇順、降順の2つのパターンです。

【参照】『第5章 言語モード/言語モードと拡張子マップ』

[アウトライン] ページ



同期表示

[タグジャンプでアウトラインも同期させる]

アウトライン解析結果のエラーなどに対するタグジャンプをするとき、プロジェクトウィンドウのアウトラインページの該当アイテムにもジャンプします。

[Shift+ ↑ ↓キーで対応ソース行を表示する]

プロジェクトウィンドウのアウトラインページ内にカーソルがあるとき、Shift+↑、Shift+↓で上下のアイテムに移動すると、そのアイテムに対応するソースファイルの位置を示します。

[ファイル名をフルパスで表示する]

アウトライン表示しているファイルをフルパスで表示します。

[対応アウトライン項目でフォーカスを移す]

このオプションをチェックすると、カーソルが編集ウインドウにあるとき、[ツール]-[アウトライン]-[対応項目]コマンドによってキーボードフォーカスがアウトラインウインドウへ移動します。チェックしないと、そのまま編集ウインドウに残るので、編集作業を続けることができます。

ソート順

[行番号]・[名前]

アウトライン解析結果を、ソースファイルの行番号順にするか、アルファベット順にするかを指定します。HTML解析では常に行番号順になります。

HTML 解析

[タグの不一致をレポートする]

対になるはずのタグに不一致があれば、タグジャンプできるようにアウトプットウィンドウにエラー表示します。

[他をスキップして一致した閉じタグをレポートする]

対になるタグの間に別のタグの不一致がある場合に、アウトプットウィンドウに警告を表示します。

**[
,<HR>,<WBR>,<P> を無視する]**

これらのタグをアウトライン解析から除外します。

[HTML タグフィルタ]

ここの数字がタグフィルタの識別番号です。15個のフィルタを設定できます。

[タイトル]

タグフィルタを識別しやすい名前を入力します。アウトラインページで右クリックしたときに出るポップアップメニューに表示されます。

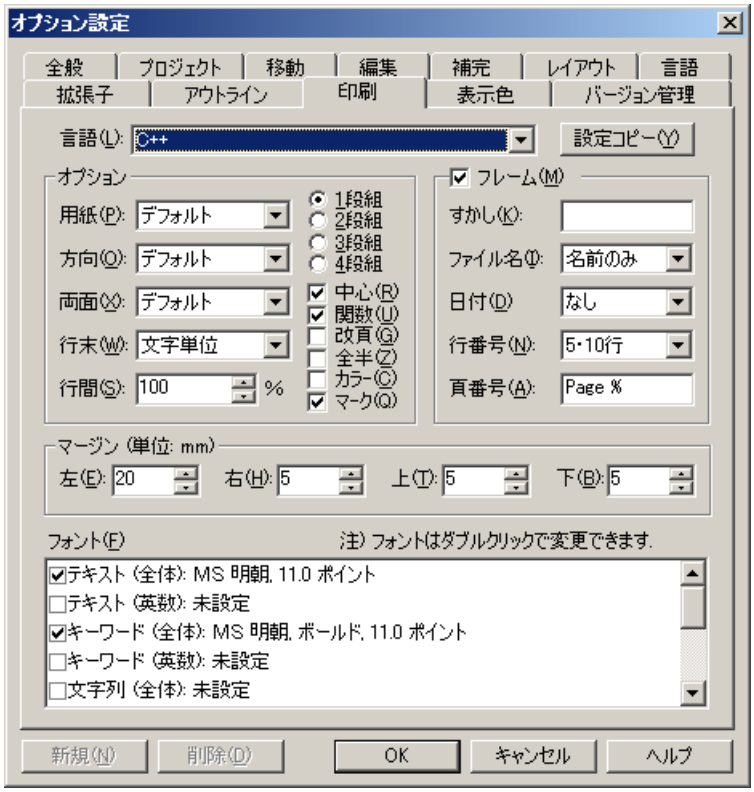
[タグリスト]

フィルタリング対象のHTMLタグを記述します。半角空白文字で区切って複数のタグを指定できます。

[フィルタ方法]

限定	タグリストのタグのみ解析対象にします。
除外	タグリストのタグを解析対象から除きます。
無効	一時的にフィルタを無効にします。アウトラインページのポップアップメニューにも表示されません。

[印刷] ページ



[言語]

印刷オプションを設定する言語を指定します。最初に表示されるのは、現在編集中のテキストの言語になります。他の言語の印刷オプションを参照/変更する場合は、まずその言語を選択してください。

[設定コピー]

このボタンをクリックすると、[印刷オプションのコピー]ダイアログボックスが表示され、選んでいる言語の印刷設定を他の言語へコピー、または他の言語の印刷設定をこの言語へコピーすることができます。

[用紙]

使用する用紙を指定します。特定の大きさの用紙を選ぶか、現在選んでいるプリンタのデフォルト用紙を使うかなどが設定できます。用紙は、[A3]、[A4]、[B4]、[B5]、[レター]、[リーガル]、[11″x17″]などから選べます。これ以外の大きさを指定したいときは、印刷時の[印刷]ダイアログボックスから指定してください。

[方向]

用紙を縦長に使う[ポートレイト]か、横長に使う[ランドスケープ]かを指定します。デフォルトとしておけば、現在のプリンタの設定をそのまま使います。

[両面]

両面印刷をサポートしているプリンタを使っているときだけ有効です。[片面]、[両面・横開き]、[両面・縦開き]から選ぶことができ、またデフォルトとしておけば、プリンタの設定をそのまま使います。

両面印刷を行うときは、奇数ページと偶数ページで左右(両面・横開き印刷時)、または上下(両面・縦開き印刷時)マージンを自動的に入れ替えて印刷を行います。両面印刷をサポートしていないプリンタの場合、この設定は無視され、上下左右マージンの入れ替えも行われません。

【注意】用紙、方向、両面にデフォルト以外を指定すると、印刷時にそのモードへ切り替え、以後Peggyを終了するまでそのモードがデフォルトとなります。システムで設定しているデフォルトの印刷方向がポートレイトでも、ランドスケープを明示的に指定してある言語のファイルを一度でも印刷すると、以後はランドスケープがデフォルトに変わってしまうので注意してください。

[行末]

1行の幅に収まりきらない長い行の処理を指定します。

選択	行末の処理
そのまま	長い行でも折り返さず、そのまま右マージンを超えてどこまでも印刷します。
切り捨て	1行の幅に入りきらない文字を切り捨てて印字しません。
文字単位	収まらなかった部分を文字単位で次の行へ折り返します。
単語単位	単語の途中で切れないように注意しながら単語単位で次の行へ折り返す点が、[文字単位]と異なっています。

[行間]

行と行の間隔を調整します。基準はテキストフォントの文字の高さで、それに対するパーセンテージで行間隔を指定します。50%～200%の範囲で任意の値を選ぶことができ、デフォルトは100%、文字の高さちょうどとなっています。

[1 段組]・[2 段組]・[3 段組]・[4 段組]

段組の段数を指定します。印刷は、1ページをそれぞれ1段、2段、3段、4段に分割して行を配置します。1段組以外は、自動的に文字の大きさを半分にした縮小印刷モードになります。段数を増やしたほうが、ページ当たりに印刷できる行数は増しますが、長い行が折れ曲がったり切れたりして見にくくなることがあります。紙の大きさと行の長さを考慮して決めてください。

[中心]

プリントアウトした用紙をバインダなどへ閉じるための穴をあける際に便利なよう、紙の中心に三角のマークを付けます。両面・横開きモードでは横に、両面・縦開きモードでは上下に、片面モードでは用紙の左か上のマージンの大きな方へ中心を示すマークが付きます。

[関数]

C/C++などのようにブロック構造を持ったプログラミング言語のソースを印刷するとき、ページや段組の終わりを関数がまたがないように調整します。具体的には、段組やページの下1/4の部分から始まっていて、その部分だけでは終わらず次の段組やページへ続いているような関数があれば、それを丸ごと次の段組/ページから始めるようにします。非常に長い関数などは、段組/ページの下端で切れてしまうこともあります。

[改頁]

テキスト中のCtrl-L文字で改ページします。2以上の段組を選択しているときは、次の段へ移ります。

［全半］

固定ピッチの日本語フォントを使う部分で、全角が半角のちょうど倍の幅となるように微調整を行います。このオプションを指定すると印刷速度が多少遅くなりますが、罫線などその文字幅に意味があるようなテキストを正確に印刷できます。

【ヒント】TrueTypeフォントなどのベクタフォントから実際に使う大きさの文字パターンに変換する際、1ドット未満となる部分をどう扱うかの丸め誤差により、全角の幅が必ずしも半角の倍にならないという問題が発生します。微調整を行わなくてもちょうど倍になる大きさもあれば、微調整なしでは文字毎に1～2ドットずれてしまう場合もあります。

【注意】このオプションで微調整されるのは、固定ピッチの日本語フォントだけです。プロポーショナルピッチのフォントや欧文フォントの文字幅には影響しません。

［カラー］

カラー印刷を行います。

［マーク］

ブックマークを印刷します。

［フレーム］

フレームをチェックすると、タイトル、ファイル名、日付などを入れるためのヘッダ、行番号を付けるための灰色の縦線などを印刷します。フレームに何々印刷するかは、以下のオプションで選択できます。

［すかし］

ヘッダ部分のバックグランドとして印刷する任意の文字列を指定します。プリンタをネットワークで共有しているような場合、すかし文字として所属や名前などを入れておけば、他の人の出力と混じってしまっても簡単に見分けることができます。

ただし、すかし文字機能が利用できるのは、グレーの文字を印字できるプリンタに限られます。グレーの文字を出力できないプリンタの多くは単に黒い文字として印字するため、ヘッダの他の文字と重なってしまい読めなくなります。グレー文字出力機能があるプリンタでのみ使ってください。

［ファイル名］

ファイル名をどのようにヘッダへ出力するかを選択します。

なし	ファイル名をプリントしません。
名前のみ	ファイル名と拡張子の部分だけがプリントされます。
フルパス(¥)	フォルダの区切り文字に¥を使ってフルパスで印刷します。
フルパス(/)	フォルダの区切り文字に/を使ってフルパスで印刷します。

［日付］

日付の入れ方を選択します。

なし	日付を印刷しません。
ファイル	ファイルの最終変更日付を印刷します。
今日	今日の日付を印刷します。

［行番号］

どのように行番号を印刷するかを指定します。

なし	行番号を付けません。
5･10行	10行毎の行番号と、その間の5行ごとにハイフン(-)を印字します。それに加えて、ページや段組の先頭にも行番号が付きます。
全行	すべての行に行番号を付けて印刷します。

[頁番号]

どのような書式でページ番号を入れるかを指定します。ここで指定した文字列の中の%が実際のページ番号と置き換わってヘッダ部分に右詰めで印刷されます。何も指定しなければ、ページ番号も印刷されません。デフォルトでは、"Page %"となっています。

[マージン（ 単位： ㎜)] ･ [左] ･ [右] ･ [上] ･ [下]

上下左右の余白をミリメートル単位で指定します。もしもプリンタの機械的制限などにより印刷することができない領域がここで指定したマージンよりも広い場合には、自動的にマージンを拡大し、文字やヘッダが切れないようにします。また両面印刷の偶数ページでは、綴じ方向によって自動的に左右、または上下のマージンを入れ替えます。

[フォント]

印刷に使う各種フォントを設定します。設定できるフォントは、11個もあります。この理由は、プログラミング言語の各要素を異なるフォントで出力できるようになっているためです。また日本語用、英数字用フォントを別々に設定できるようになっているのも一因です。

もちろんすべて設定しなければならないわけではありません。それぞれにはチェックボックスが付いていて、チェックを外すとデフォルトのフォントが使われます。基本的には数個指定しただけでも十分ですが、よりきれいに、好みに合わせて印刷したければ細かい指定もできるようになっています。

以下は設定できるフォントの一覧表です。フォントの設定を変えるには、そのフォントの上をマウスでダブルクリックしてください。

印刷フォント

	部分	デフォルト(チェックを外したとき)
1	テキスト(全体)	ＭＳ　明朝、11ポイント
2	テキスト(英数)	1　テキスト(全体)
3	キーワード(全体)	1　テキスト(全体)
4	キーワード(英数)	3　キーワード(全体)
5	文字列(全体)	1　テキスト(全体)
6	文字列(英数)	5　文字列(全体)
7	コメント(全体)	1　テキスト(全体)
8	コメント(英数)	7　コメント(全体)
9	タイトル	ＭＳ　Ｐゴシック、16ポイント
10	すかし	9　タイトル
11	行番号	2　テキスト(英数)

英数用フォントというのは、アスキー文字の印刷に使うフォントです。これを指定してチェックボックスをONにしておくと、漢字など2バイト文字や半角カタカナ、英数字や記号などを別々のフォントで印刷できるようになります。コメントの中には日本語を含んでいるけれども、英数字は「Times New Roman」などのようにより欧文らしいフォントで印刷したい場合に有効です。

行間の高さもフォントによって自動的に決まります。「テキスト(英数)」が指定されている場合はその文字セルの高さ、指定されていなければ「テキスト(全体)」の文字の高さをもとに決めています。従ってキーワードなど他で指定しているフォントに背の高いものがあると、行が重なってしまうこともあるので注意してください。背の高いフォントの文字が重なってしまう場合は、行間で調整できます。

【ヒント】テキスト編集中のエディタが使うスクリーンフォントはすべて固定ピッチのものしか指定できません。テキストの編集ではほとんどの場合それで十分です。逆にその方が桁位置が揃って読みやすくなります。

【ヒント】キーワードフォントを使って印刷する部分には、その言語のキーワード、ユーザ定義キーワード1と2、プリプロセッサ命令、レジスタ名、CPUインストラクション、HTMLエレメント名、HTMLアトリビュート名を含みます。

【注意】テキストなど文法による色分け表示機能のない言語のモードでは、「キーワード」、「文字列」、「コメント」のフォント(全体/英数とも)は使われません。テキストフォントの全体/英数のみを使います。

【注意】Peggyの印刷機能は、主にページプリンタの使用を前提に設計されています。すべてはWindowsのドライバ経由で印刷を行っているため、Windowsがサポートしているプリンタであれば、原則として機種を問わずすべて使用できるはずです。しかし特定の組み合わせでは印刷スピードが遅くなったり、不具合が発生する可能性もあります。正しく印刷できない場合、プリンタメーカーから新しいプリンタドライバなどが提供されているならば、それをインストールすると解消できることもありますので、試してみてください。

[印刷オプションのコピー] ダイアログボックス

[印刷]ページで選択している印刷オプションを他のすべての言語へコピーします。または、他の言語のコピーを選択している言語へコピーします。

[この言語の印刷オプションを他のすべての言語にコピーする]

これを選択してから[OK]をクリックすると、[印刷]ページで設定しているオプションを、他のすべての言語へコピーします。

[選択した言語の印刷オプションをこの言語にコピーする]

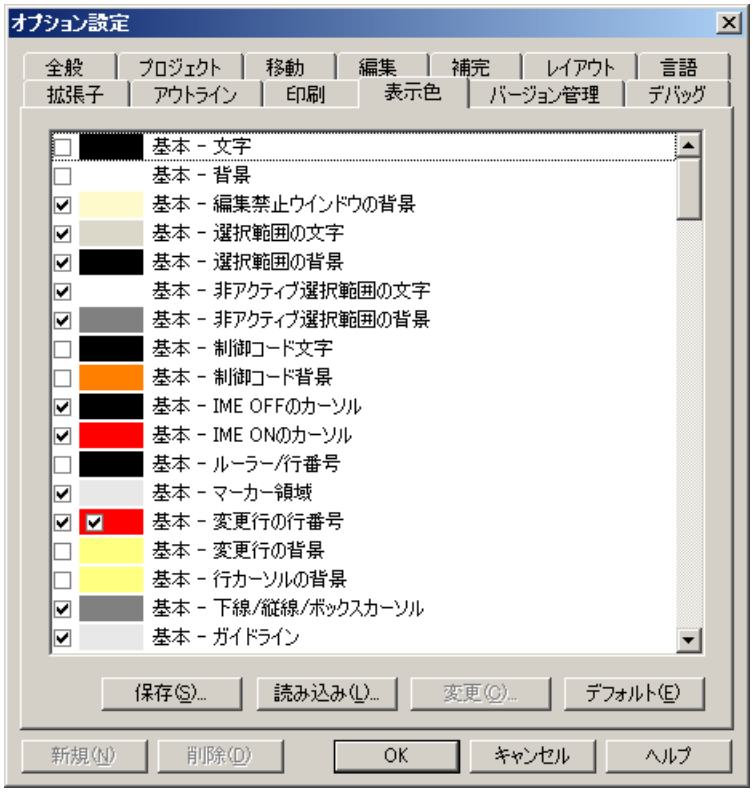
[コピー元]で選択している言語の印刷オプション設定をこの言語へコピーします。

[コピー元]

[選択した言語の印刷オプションをこの言語にコピーする]でコピーする言語を選択します。

[表示色] ページ

Peggy内の文字色や背景色を自由に変更することができます。左のチェックボックスがONの項目の色が有効になります。チェックがない場合はデフォルト色で表示されます。各設定をダブルクリックすると、色を変更することができます。



強調文字

色によっては、チェックが2つあるものがあります。2つめのチェックをONにすると、その部分を太字で強調表示します。

色の用途ラベル

[文法ー予備1～20]、[文法ー予備コメント1～5]、[文法ー予備文字列1～5]、[背景-サブ言語1～7]は、その色の用途をメモしておくことができます。その色を選択している状態で、[文法ー予備<番号>]のすぐ右をもう一度クリックするか、F2キーをタイプすることで、メモを編集できるモードになります。編集し終わると、Enterキーで確定させるか、Escキーでキャンセルすることができます。

[保存]

現在設定されている配色をファイルに保存します。デフォルトのファイル拡張子は、<.col>です。保存した表示色設定はテキストファイルです。テキストファイルを直接編集して色を変えることも可能です。

[読み込み]

ファイルに保存されている表示色設定を読み込みます。カラーの値に色の名前が使われていると、Peggyをインストールしたフォルダの<colordef.txt>というファイ

ルを参照してRGB値に変換します。<colordef.txt>には、Webブラウザなどで使われている一般的な色の名前とRGB値が定義してあります。

【ヒント】Peggyをインストールしたフォルダには、サンプルとしていくつかの<COL>ファイルが収められています。それを元にしてカスタマイズし、気に入った配色が完成したら、別の名前で保存しておきましょう。

[変更]

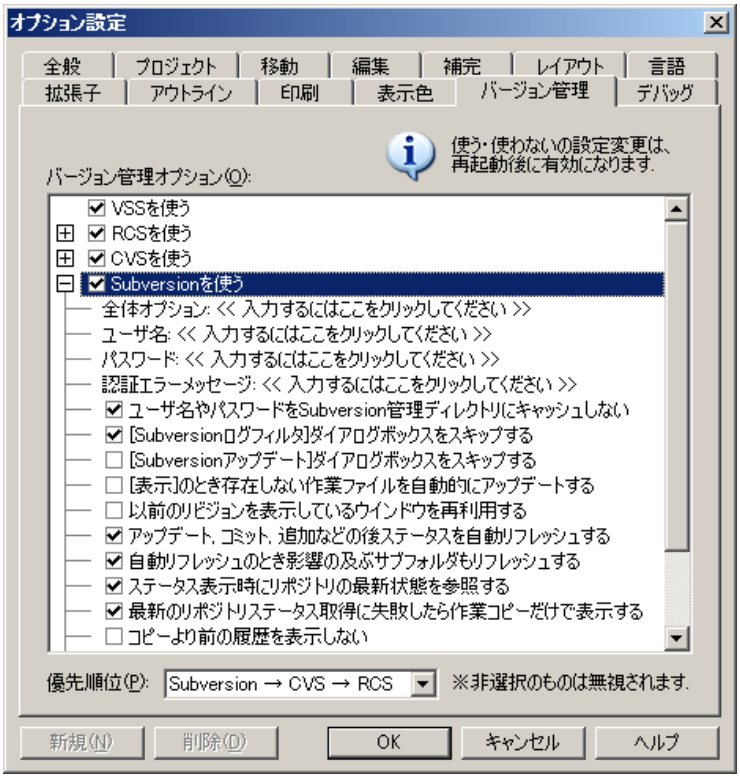
変更するときは目的の色を選択してから[変更]ボタンをクリックします。[色の設定]ダイアログボックスが表示されますから、任意の色を選択してください。リスト中の変更したい項目上を左マウスボタンでダブルクリックしても同じです。[色の設定]ダイアログボックスを閉じると、自動的にチェックされます。

[デフォルト]

デフォルトボタンをクリックすると、デフォルト状態に戻します。確認ダイアログボックスも出ませんので注意してください。

【注意】プロジェクトウインドウの文字、線、背景色の設定は、Windows 98/Me/2000/XP、またはIE4以上がインストールされているWindows NT 4.0でのみ可能です。もしもプロジェクトウインドウに正しい色が設定できない場合は、チェックをOFFにし、デフォルト色に戻してください。

[バージョン管理] ページ



[VSS を使う]

VSSバージョン管理システムを使う場合はチェックしてください。

[RCS を使う]

RCSのメニューやコマンドを有効にします。

[RCS を使う]-[ログメッセージを EUC に変換する]

ログメッセージをEUCに変換します。Unix/Linuxで使われていたRCSファイルのログメッセージが正しく読み書きできるようになります。

[RCS を使う]-[ファイル名の最後のバイトが 5C で終わる場合はドットを追加する]

RCSは漢字のファイル名を正しく扱えないことがあります。特にファイル名の最後の文字が全角文字(2バイト文字)で、その2バイト目が5C(16進)である場合はエラーになります。

[CVSを使う]

CVSのメニューやコマンドを有効にします。

[CVS を使う]-[全体オプション]

CVSコマンドの全体に共通して付ける任意のオプションを指定できます。例えば、-qとすると、cvs.exeコマンドが出力するメッセージを幾分押さえるように指定できます。

[CVS を使う]-[CVS ログフィルタダイアログボックスをスキップする]

チェックすると、[CVSログフィルタ]ダイアログボックスの表示をスキップし、即座に[CVSリビジョンログ]ダイアログボックスを表示します。

[CVS を使う]-[CVS アップデートダイアログボックスをスキップする]

チェックすると、[CVSアップデート]ダイアログボックスの表示をスキップし、即座にアップデートコマンドを実行します。

[CVS を使う]-[表示のとき存在しない CVS 作業ファイルを自動的にアップデートする]

[CVSリビジョンログ]、[CVSステータス]ダイアログボックスで[表示]ボタンをクリックしたけれども実際の作業ファイルがなかった場合、自動的にリポジトリからアップデートします。

[CVS を使う]-[以前のリビジョンを表示しているウインドウを再利用する]

過去のリビジョンを表示するウインドウをできるだけ再利用して、いくつも開かないようにします。

[CVS を使う]-[アップデート , コミット , 追加 , 削除の後ステータスをリフレッシュする]

[CVSステータス]ダイアログボックスの[アップデート]、[コミット]、[追加]、[削除]ボタンをクリックしてそれぞれの操作を実行したあと、CVSステータスを自動更新します。このチェックを外すと、[更新]ボタンをクリックするまで更新されません。

[CVS を使う]-[ログメッセージを EUC に変換する]

cvs.exeコマンドとのログメッセージの受け渡しをEUC漢字コードで行います。

[CVS を使う]-[ログメッセージを UTF-8 に変換する]

cvs.exeコマンドとのログメッセージの受け渡しをUTF-8漢字コードで行います。

【ヒント】Unix/Windowsが混在しているネットワーク環境では、ログメッセージの漢字コードをEUC、UTF-8、Shift JISのどれか統一しなければ混乱が生じます。これらオプションをONにすることで、Windowsからコミットしたときのログメッセージもも EUC、UTF-8で記録できます。また、ログを表示するときも、EUC、UTF-8からShift JISへ変換してから表示します。EUCとUTF-8オプションを同時に指定することはできません。

【ヒント】このオプションは、ファイル内のテキストの漢字コードを変換するものではありません。Peggyは各種漢字コードを扱えるので、ファイルの漢字コードがEUCであってもそのまま編集できます。

[CVS を使う]-[毎回アウトプットウインドウをクリアする]

ONにすると、個々のCVS操作を実行する前に[アウトプットウインドウ]の[バージョン管理]ページをクリアします。

[CVS を使う]-[CVS ステータスダイアログボックスの [タグ] ボタンを有効にする]

OFFにすると、[CVSステータス]ダイアログボックスの[タグ]ボタンが無効になります。タブの設定はあまり頻繁に行う操作ではないため、誤ってクリックすることを防ぐ安全策です。

[CVS を使う]-[ログの日付指定オプションで -d の後にスペースを空けない]

ログの日付の範囲でフィルタするオプション-dの後に、スペースを空けずに日付、時刻を表す文字列を続ける。cvs.exeのバージョンによっては、スペースでエラーになるものがあるので、その場合はこのオプションをONにしてください。

[CVS を使う]-[ログの日時を現地時間で表示する]

リビジョンログに表示される日時をローカルタイムで表示するようにします。

[Subversion を使う]

Subversionのメニューやコマンドを有効にします。

[Subversion を使う]-[全体オプション]

Subversionの全コマンドに共通して設定したいオプションをを指定します。

[Subversion を使う]-[ユーザ名]、[パスワード]

リポジトリへのアクセスにユーザ名やパスワードが必要なときは、ここへ設定します。

【重要】パスワードは簡単に暗号化された状態でレジストリへ保存されます。高度な暗号化ではないため、厳重なセキュリティを必要とするリポジトリのパスワードは指定しないでください。

【注意】パスワードが必要なリポジトリへパスワード無しでアクセスすると、[Subversionユーザ認証]ダイアログボックスが表示されます。

【メモ】パスワードが設定されているときは、アウトプットウインドウのバージョン管理ページへコマンド行(svn.exe ...)が表示されなくなります。

[Subversion を使う]-[認証エラーメッセージ]

認証エラーが発生したことを調べるためのSubversionの認証エラーメッセージを指定します。デフォルトでも 2 種類のエラーメッセージ「認証が失敗」、「AUTHORIZATION FAILED」で認証エラーが発生したと見なしますが、これ以外のメッ

セージが表示される場合は、ここへ指定してください。大文字、小文字は区別しません。

[Subversion を使う]-[ユーザ名やパスワードを Subversion 管理ディレクトリにキャッシュしない]

Subversionのコマンドラインプログラムsvn.exeは、接続に成功した[ユーザ名]と[パスワード]をシステム上にキャッシュします。キャッシュしないようにするには、このオプションをONにしてください。

[Subversion を使う]-[Subversion ログフィルタダイアログボックスをスキップする]

チェックすると、[Subversionログフィルタ]ダイアログボックスの表示をスキップし、即座に[Subversionリビジョンログ]ダイアログボックスを表示します。

[Subversion を使う]-[Subversion アップデートダイアログボックスをスキップする]

チェックすると、[Subversionアップデート]ダイアログボックスの表示をスキップし、即座にアップデートコマンドを実行します。

[Subversion を使う]-[表示のとき存在しない作業ファイルを自動的にアップデートする]

[Subversionリビジョンログ]、[Subversionステータス]ダイアログボックスで[表示]ボタンをクリックしたけれども実際の作業ファイルがなかった場合、自動的にリポジトリからアップデートします。

[Subversion を使う]-[以前のリビジョンを表示しているウインドウを再利用する]

過去のリビジョンを表示するウインドウをできるだけ再利用して、いくつも開かないようにします。

[Subversion を使う]-[アップデート , コミット , 追加などの後ステータスを自動リフレッシュする]

[Subversionステータス]ダイアログボックスの[アップデート]、[コミット]、[追加]、[削除] などのボタンをクリックしてそれぞれの操作を実行したあと、Subversionステータスを自動更新します。このチェックを外すと、[更新]ボタンをクリックするまで更新されません。

[Subversion を使う]-[自動リフレッシュのとき影響の及ぶサブフォルダもリフレッシュする]

[Subversionステータス]ダイアログボックスの[アップデート]、[コミット]などに伴う自動リフレッシュで、影響が予想されるサブフォルダも自動的にリフレッシュします。

[Subversion を使う]-[ステータス表示時にリポジトリの最新状態を参照する]

[Subversionステータス]ダイアログボックスの表示や更新時に、リポジトリの状態を参照します。

[Subversion を使う]-[最新のリポジトリステータス取得に失敗したら作業コピーだけで表示する]

[ステータス表示時にリポジトリの最新状態を参照する]オプションがONのときにサーバーへの接続に失敗した場合、作業コピーからステータスを取得して表示します。

[Subversion を使う]-[コピーより前の履歴を表示しない]

[Subversionリビジョンログ]ダイアログボックスで、コピーされた時点より以前の履歴を表示しません。

[Subversion を使う]-[毎回アウトプットウインドウをクリアする]

ONにすると、個々のSubversion操作を実行する前に、アウトプットウインドウの[バージョン管理]ページをクリアします。もしも以前の操作を記録として残したままにしておきたい場合は、OFFにしてください。最大10,000行までSubversionコマンドの実行記録が残ります。

[Subversion を使う]-[ログの日時を現地時間で表示する]

Subversionログに表示される日時をローカルタイムで表示するようにします。

[Subversion を使う]-[リポジトリブラウザを開くと同時に最初の URL を表示する]

ONにすると、[Subversionリポジトリブラウザ]ダイアログボックスを開くときにURLで指定されているリポジトリの内容を自動的に表示します。OFFの場合は、Enterキーが押されるまで表示しません。

[Subversion を使う]-[外部ビューアで開くファイル]

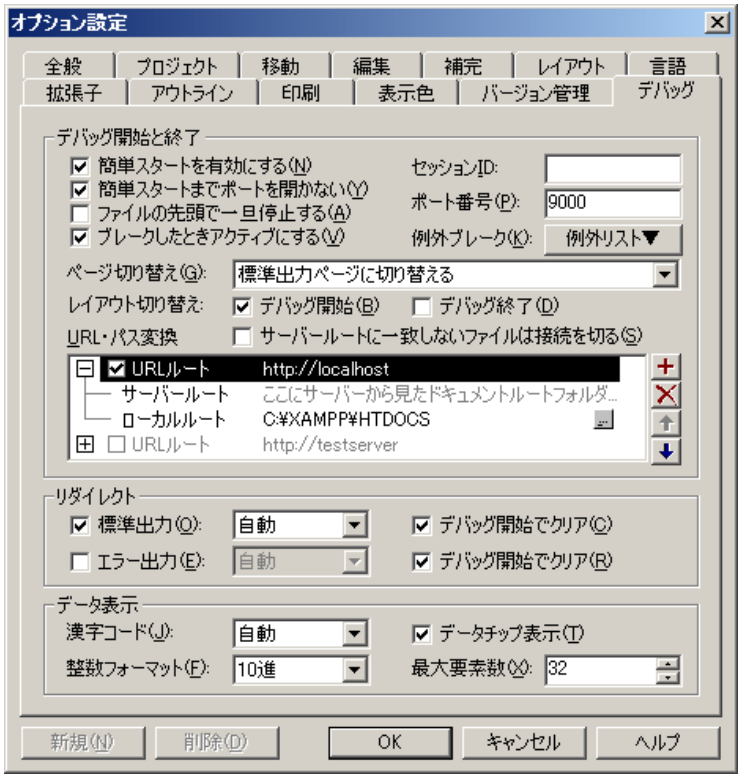
Subversionのファイルを表示するとき、外部ビューアを使うべきファイルパターンを指定します。ファイルパターンは、スペースやコンマで区切って幾つでも指定できます。

【例】*.doc *.xls *.pdf

[優先順位]

RCS、CVS、Subversion共通で使われるコマンド[ツール]-[リビジョンログ]、[ツール]-[リビジョン比較]を実行したとき、どのバージョン管理システムを優先して使うかを設定します。

[デバッグ] ページ



PHPプログラムのデバッグに関するオプションを設定します。

【注意】[デバッグ]ページは、Peggy Pro for PHP以外にはありません。

[デバッグ開始と終了]

以下は、デバッグ開始と終了の動作をカスタマイズするオプションです。

[簡単スタートを有効にする]

デバッグしたいソースファイルを開いてるときに、[実行]、[ステップイン]、[ステップオーバー]、[カーソル位置まで実行]などのボタンをクリックするだけで、簡単にデバッグを開始できるようになります。

また、カーソルがURL上にあるときは、[実行]、[ステップイン]、[ステップオーバー]ボタンで、そのURLに対してデバッグ開始を試みます。

【重要】簡単スタートでは、以下の[URL・パス変換]が正しく設定されていることが重要です。例えば[実行]ボタンをクリックすると、[URL・パス変換]に設定された情報に従ってファイル名からURLを作成しWebブラウザを起動します。[URL・パス変換]が設定されていないか誤りがあるときは、デバッグを開始できません。

[簡単スタートまでポートを開かない]

ONにすると、簡単スタート機能によりデバッグを開始するときまで、リモートデバッグの待ち受けポートをオープンしません。OFFにすると、Peggy Pro for PHPの起動と同時に待ち受けポートをオープンします。このオプションを有効にするには、Peggy Pro for PHPを再起動してください。

【ヒント】JIT(ジャスト・イン・タイム)デバッグのような使い方をするには、このオプションをOFFにし、Peggy Pro for PHPを起動してから最小化しておきます。Webサーバーの PHPにエラーが発生してPeggy Pro for PHPのポートに接続すると、ポップアップしてエラー発生箇所をデバッグできます。

【注意】初めてポートを開くときに、Windowsファイヤーウォールから警告ダイアログボックスが表示されることがあります。その場合は、[ブロックを解除する]をクリックしてブロックしないようにしてください。詳しくは、『第10章 PHPプログラムのデバッグ／デバッグの準備』を参照してください。

[ファイルの先頭で一旦停止する]

デバッグ開始したとき、ソースファイルの実行を開始する直前で一旦停止します。

[ブレークしたときアクティブにする]

ブレークしたときにPeggy Pro for PHPをアクティブにします。

[セッション ID]

デバッグセッションの識別名を指定します。空欄にすると、環境変数USERNAMEの値が使われます。

[ポート番号]

Webサーバーからデバッグ開始の接続を待ち受けるポート番号を指定します。デフォルトは、9000です。

【注意】変更したポート番号を有効にするには、Peggy Pro for PHPを再起動してください。

[例外ブレーク]

例外が発生したときブレークするかどうかを選択します。[条件選択]ボタンをクリックすると、例外の種類がメニュー表示されます。チェックが付いている例外でブレークします。メニューから例外を選択すると、ブレークする/しないを切り替えられます。

【注意】このオプションは、次のデバッグセッションから有効になります。

[ページ切り替え]

デバッグ開始時、アウトプットウインドウのページを自動的に切り替えるオプションです。

[レイアウト切り替え]

ツールバー、ツールボタンのレイアウトには、レギュラーレイアウトとデバッグレイアウトがあります。このオプションをチェックすると、デバッグ開始時、デバッグ終了時に自動的にレイアウトを切り替えます。

【ヒント】[デバッグレイアウト]コマンドにより、手動で切り替えることもできます。

[URL・パス変換]

以下は、デバッグ対象のURLと、それに対応するWebサーバー、ローカルコンピュータそれぞれのファイルの場所を指定するオプションです。この組み合わせは複数登録できます。複数登録されているときは、上から順番に一致するものを検索します。[ ]、[ ]、[]、[ ]ボタンにより、追加、削除、上へ移動、下へ移動できます。

[URLルート]

対象となるWebサーバーのトップレベルURLを指定します。チェックを外すと、一時的にOFFにできます。

[サーバールート]

Webサーバーからみたソースファイル群が置いてあるトップレベルのフォルダ(ドキュメントルート等)を指定します。PeggyとWebサーバーが同じコンピュータ上にあるときは省略できます。省略すると、[ローカルルート]と同じと見なします。

[ローカルルート]

Peggyからみたソースファイル群が置いてあるトップレベルフォルダ(ドキュメントルート等)のパスを指定します。ここで指定したフォルダ以下にあるファイルを「ソースファイル」と見なします。

【ヒント】URL・パス変換は、複数設定できます。

【ヒント】URL・パス変換は、プロジェクトのプロパティにも設定できます。プロジェクトにURL・パス変換がされている場合は、プロジェクトの設定が優先します。詳しくは、『第6章 プロジェクト機能／プロジェクトプロパティページ』を参照してください。

【参照】詳しい設定方法に関しては、『第10章 PHPプログラムのデバッグ／デバッグの準備』を参照してください。

[リダイレクト]

以下は、リダイレクト方法を指定するオプションです。

[標準出力]

デバッグしているスクリプトの標準出力をアウトプットウインドウの[標準出力]ページにリダイレクトします。スクリプトからの出力は、Webサーバーからブラウザへ送られると同時に、アウトプットウインドウにも表示されます。

リダイレクトで送られてくるテキストの漢字コードも選択できます。[自動]を選択すると、ソースファイルと同じ漢字コードと仮定して表示します。ソースファイルの漢字コードとスクリプトの出力の漢字コードが異なる場合は、明示的に指定してください。

[デバッグ開始でクリア]をチェックすると、デバッグ開始時に[標準出力]ページをクリアします。

[エラー出力]

デバッグしているスクリプトの標準エラー出力をアウトプットウインドウの[標準エラー出力]ページにリダイレクトします。

リダイレクトで送られてくるテキストの漢字コードも選択できます。[自動]を選択すると、ソースファイルと同じ漢字コードと仮定して表示します。ソースファイルの漢字コードとスクリプトのエラー出力の漢字コードが異なる場合は、明示的に指定してください。

[デバッグ開始でクリア]をチェックすると、デバッグ開始時に[標準エラー出力]ページをクリアします。

【ヒント】漢字コードは、標準出力、標準エラー出力ページのポップアップメニューからも選択できます。

【注意】Webサーバーに組み込まれているデバッグモジュールには、リダイレクトをサポートしていものもあります。その場合アウトプットウインドウには何も表示されません。

[データ表示]

以下は、変数値などの表示方法を指定するオプションです。

[漢字コード]

変数ウインドウやチップ表示で文字列データを表示するときの漢字コードを指定します。[自動]を選択すると、ソースファイルと同じ漢字コードと仮定して表示します。

[整数フォーマット]

変数ウインドウやチップ表示で整数データを表示するときのフォーマットを指定します。

[データチップ表示]

チップ表示を有効にします。マウスカーソルをソースファイルの変数上へ移動させるだけで、変数値を確認できます。

【ヒント】漢字コード、整数フォーマットは、変数ウインドウのポップアップメニューからも選択できます。

[最大要素数]

配列など子を持った変数を表示するときの最大要素数を10～1000の範囲で指定します。

【注意】多くの要素を表示するように設定すると、変数ウインドウを更新する毎に大量の通信が発生するため、ステップ動作が遅くなることがあります。

APPENDIX

付録 A コマンド一覧

ショートカットキーに割り当てることのできるコマンドの一覧です。またエディタのキーボードマクロに登録できるかどうかも「マクロ」欄にチェックしています。

【注意】表のキー操作欄は、デフォルトショートカットキーを表します。

ファイル

ボタン	機能	キー操作	マクロ
	新規作成	Ctrl+N	
	テンプレートから新規作成		
	開く	Ctrl+O	
	Shift JISとして開き直す		
	EUCとして開き直す		
	JISとして開き直す		
	UTF-7として開き直す		
	UTF-8として開き直す		
	Unicodeとして開き直す		
	ダイレクトオープン	Ctrl+Shift+O	
	関連ファイルを開く		
	インクルードファイルを開く		
	閉じる	Ctrl+F4	
	上書き保存	Ctrl+S	
	名前を付けて保存		
	すべて保存		
	印刷	Ctrl+P	
	ダイレクト印刷		
	印刷プレビュー		
	印刷オプション		
	プリンタの設定		
	ヒストリ	Ctrl+H	
	最近使ったファイル		
	最近使ったフォルダ		
	最近使ったプロジェクト		
	URLをブラウズする	Ctrl+B	
	スクリプト実行		
	関連アプリケーション起動		
	Peggyの終了	Alt+F4	

編集

ボタン	機能	キー操作	マクロ
	元に戻す	Ctrl+Z	○
	やり直し	Ctrl+Y	○
	編集操作リスト		○
	再変換		
	切り取り	Ctrl+X/Shift+Delete	○

	切り取って追加	Ctrl+Shift+X	○
	単語切り取り		○
	単語を切り取って追加		○
	行切り取り※		○
	ファイル行切り取り		○
	表示行切り取り		○
	行を切り取って追加※		○
	ファイル行を切り取って追加		○
	表示行を切り取って追加		○
	コピー	Ctrl+C/Ctrl+Insert	○
	コピーして追加	Ctrl+Shift+C	○
	単語コピー		○
	単語をコピーして追加		○
	行をコピー※		○
	ファイル行をコピー		○
	表示行をコピー		○
	行をコピーして追加※		○
	ファイル行をコピーして追加		○
	表示行をコピーして追加		○
	貼り付け	Ctrl+V/ Shift+Insert	○
	行として貼り付け	Ctrl+Shift+V	○
	カーソル前後の文字を入れ替える	Ctrl+Alt+T	○
	カーソル前後の単語を入れ替える	Ctrl+Shift+T	○
	カーソル前後の行を入れ替える	Alt+Shift+T	○
	インデントを深く	選択+Tab	○
	インデントを浅く	選択+Shift+Tab	○
	選択範囲のテキストを左揃えする	F9 ＊	○
	選択範囲のテキストを右揃えする	Shift+F9	○
	選択範囲のテキストを中央揃えする	Ctrl+F9	○
	選択範囲をスマートインデント	Ctrl+I	○
	ファイル全体をスマートインデント		○
	挿入モード切り替え	Insert	○
	編集禁止モード切り替え		○
	プロパティ	Alt+Enter	

※の付いたコマンドは、行番号のモード(ファイル行/表示行)によって、それぞれファイル行/表示行を基準に動作します。
＊の付いたショートカットキーは、Peggy Pro for PHPではデフォルトで割り当てられていません。

選択

ボタン	機能	キー操作	マクロ
	文字選択モード	F8	○
	矩形選択モード	Ctrl+Shift+F8	○
	カーソル位置の単語を選択	Shift+F8	○
	カーソル位置の行全体を選択※	Ctrl+F8	○
	カーソル位置のファイル行全体を選択		○
	カーソル位置の表示行全体を選択		○

	括弧内部の選択		○
	括弧と内部の選択		○
	コメントを選択	Alt+F8	○
	コメント内部を選択	Ctrl+Alt+F8	○
	すべてを選択	Ctrl+A	○
	選択解除	Esc	○
	選択範囲を出力		

※の付いたコマンドは、行番号のモード(ファイル行/表示行)によって、それぞれファイル行/表示行を基準に動作します。

挿入

ボタン	機能	キー操作	マクロ
	制御コード挿入	Ctrl+Q	
	改行コード挿入		○
	RGB値挿入		
	日付を挿入	Ctrl+D	○
	時間を挿入		○
	日時を挿入		○
	次のガイドライン位置まで空白で埋める		○
	上に空白行挿入	Ctrl+Shift+N	○
	下に空白行挿入		○
	カーソル行を複製して下に挿入		○
	行コメント挿入		○
	行コメント(2)挿入		○
	ブロックコメント挿入		○
	単語をコメント化		○
	行をコメント化		○
	メール引用符挿入	Alt+.	○

削除

ボタン	機能	キー操作	マクロ
	メール引用符削除	Alt+,	○
	バックスペース	Backspace/ Shift+Backspace	○
	1文字削除	Delete	○
	すべて削除		○
	単語削除	Ctrl+Shift+Delete	○
	右へ単語削除	Ctrl+Delete	○
	左へ単語削除	Ctrl+Backspace	○
	行末まで削除※	Ctrl+K	○
	ファイル行の行末まで削除		○
	表示行の行末まで削除		○
	行頭まで削除※	Ctrl+Shift+K	○
	ファイル行の行頭まで削除		○
	表示行の行頭まで削除		○
	改行コード削除		○
	複数の空白を1つにまとめる		○
	現在行を削除※		○

	現在のファイル行を削除		○
	現在の表示行を削除		○
	連続した空白行削除		○
	行末の空白を削除	Ctrl+Shift+S	○
	コメント削除	Ctrl+Shift+F10	○
	コメント内部を削除		○
	コメント解除	Shift+F10	○

※の付いたコマンドは、行番号のモード(ファイル行/表示行)によって、それぞれファイル行/表示行を基準に動作します。

文字変換

ボタン	機能	キー操作	マクロ
	選択範囲を大文字に変換	Ctrl+Shift+U	○
	選択範囲を小文字に変換	Ctrl+U	○
	選択範囲の大文字小文字を反転		○
	選択範囲の頭文字を大文字に変換		○
	単語を大文字に変換		○
	単語を小文字に変換		○
	単語の大文字小文字を反転		○
	単語の頭文字だけ大文字に変換		○
	選択範囲のひらかなをカタカナに変換		○
	選択範囲のカタカナをひらかなに変換		○
	選択範囲の半角文字の種類を選択し全角に変換		
	選択範囲のすべての半角文字を全角に変換		○
	選択範囲の半角英字を全角に変換		○
	選択範囲の半角数字を全角に変換		○
	選択範囲の半角英数字を全角に変換		○
	選択範囲の半角非空白文字を全角に変換		○
	選択範囲の半角空白文字を全角に変換		○
	選択範囲の全角文字の種類を指定し半角に変換		
	選択範囲のすべての全角文字を半角に変換		○
	選択範囲の全角英字を半角に変換		○
	選択範囲の全角数字を半角に変換		○
	選択範囲の全角英数字を半角に変換		○
	選択範囲の全角非空白文字を半角に変換		○
	選択範囲の全角空白文字を半角に変換		○
	行頭スペースをタブに変換		○
	行頭タブをスペースに変換	Ctrl+Shift+Space	○
	スペースをタブに変換		○
	タブをスペースに変換		○
	改行をスペースに変換		○
	選択範囲の行を昇順にソートする		○
	選択範囲の行を降順にソートする		○
	行ソートで行頭の空白文字を無視するかどうか切り替える		○
	選択範囲の行の順番を逆転する		○
	選択範囲の文字コードを変換する		○
	選択範囲のEUC文字を変換する		○
	選択範囲のJIS文字列を変換する		○

	選択範囲のUTF-7文字列を変換する		○
	選択範囲のUTF-8文字列を変換する		○
	選択範囲のUnicode(LH順)文字列を変換する		○
	選択範囲のUnicode(HL順)文字列を変換する		○
	ファイルの漢字コードをShift JISに変更する		○
	ファイルの漢字コードをEUCに変更する		○
	ファイルの漢字コードをJISに変更する		○
	ファイルの漢字コードをUTF-7に変更する		○
	ファイルの漢字コードをUTF-8に変更する		○
	ファイルの漢字コードをUTF-8Nに変更する		○
	ファイルの漢字コードをUnicodeに変更する		○
	ファイルの改行コードをCR-LFに変更する		○
	ファイルの改行コードをLFに変更する		○
	ファイルの改行コードをCRに変更する		○

補完入力

ボタン	機能	キー操作	マクロ
	前後の単語で補完	Ctrl+Space	○
	前の行から単語を探して補完		○
	後の行から単語を探して補完		○
	キーワード補完入力	Ctrl+/	○
	短縮入力		○

ジャンプ

ボタン	機能	キー操作	マクロ
	指定行	Ctrl+G	
	最終変更個所	Ctrl+Shift+L	○
	対応括弧	Ctrl+]	○
	選択範囲を拡大して対応括弧	Ctrl+Shift+]	○
	見かけの対応括弧		○
	選択範囲を拡大して見かけの対応括弧		○
	次のタグ	F4	
	前のタグ	Shift+F4	
	カーソル行のタグ	Ctrl+T	
	次のラベル/ブロック	Ctrl+.	○
	前のラベル/ブロック	Ctrl+,	○
	次の対応する条件コンパイル文	Alt+]	○
	前の対応する条件コンパイル文	Alt+[	○
	次の変更行	Ctrl+Shift+9	○
	前の変更行	Ctrl+Shift+8	○
	変更行情報クリア	Ctrl+Shift+0	○

カーソル移動

ボタン	機能	キー操作	マクロ
	1文字左へ	←	○
	1文字右へ	→	○
	1行上へ	↑	○
	1行下へ	↓	○
	選択範囲を1文字左へ	Shift+←	○

	選択範囲を1文字右へ	Shift+→	○
	選択範囲を1行上へ	Shift+↑	○
	選択範囲を1行下へ	Shift+↓	○
	1単語左へ	Ctrl+←	○
	1単語右へ	Ctrl+←	○
	選択範囲を1単語左へ	Ctrl+Shift+←	○
	選択範囲を1単語右へ	Ctrl+Shift+←	○
	左の単語境界へ		○
	右の単語境界へ		○
	選択範囲を左の単語境界へ		○
	選択範囲を右の単語境界へ		○
	行の先頭へ※	Home	○
	ファイル行の先頭へ		○
	表示行の先頭へ		○
	行の最後へ※	End	○
	ファイル行の最後へ		○
	表示行の最後へ		○
	選択範囲を行の先頭へ※	Shift+Home	○
	選択範囲を行のファイル行の先頭へ		○
	選択範囲を行の表示行の先頭へ		○
	選択範囲を行の最後へ※	Shift+End	○
	選択範囲をファイル行の最後へ		○
	選択範囲を表示行の最後へ		○
	半ページ上へ		○
	半ページ下へ		○
	選択範囲を半ページ上へ		○
	選択範囲を半ページ下へ		○
	1ページ上へ	PageUp	○
	1ページ下へ	PageDown	○
	選択範囲を1ページ上へ	Shift+PageUp	○
	選択範囲を1ページ下へ	Shift+PageDown	○
	2つのウインドウを同時に1ページ上へ	Ctrl+Alt+PageUp	○
	2つのウインドウを同時に1ページ下へ	Ctrl+Alt+PageDown	○
	テキストの先頭へ	Ctrl+Home	○
	テキストの最後へ	Ctrl+End	○
	選択範囲をテキストの先頭へ	Ctrl+Shift+Home	○
	選択範囲をテキストの最後へ	Ctrl+Shift+End	○
	ウインドウの最上行へ		○
	ウインドウの最下行へ		○
	ウインドウの中央へ		○
	アンカー/カレント入れ替え	Ctrl+Shift+A	○

※の付いたコマンドは、行番号のモード(ファイル行/表示行)によって、それぞれファイル行/表示行を基準に動作します。

スクロール

ボタン	機能	キー操作	マクロ
	左スクロール	Ctrl+PageUp	○
	右スクロール	Ctrl+PageDown	○
	スクロールアップ	Ctrl+↓	○
	スクロールダウン	Ctrl+↑	○
	同時に左スクロール	Ctrl+Alt+←	○
	同時に右スクロール	Ctrl+Alt+→	○
	同時にスクロールアップ	Ctrl+Alt+↓	○

	同時にスクロールダウン	Ctrl+Alt+↑	○
	高速スクロールアップ	Ctrl+Shift+↓	○
	高速スクロールダウン	Ctrl+Shift+↑	○

罫線

ボタン	機能	キー操作	マクロ
	罫線を左へ	Alt+←	○
	罫線を右へ	Alt+→	○
	罫線を上へ	Alt+↑	○
	罫線を下へ	Alt+↓	○
	罫線を左へ向かって消す	Alt+Shift+←	○
	罫線を右へ向かって消す	Alt+Shift+→	○
	罫線を上へ向かって消す	Alt+Shift+↑	○
	罫線を下へ向かって消す	Alt+Shift+↓	○
	罫線の種類を変更する	Alt+Home	○
	文字(+－|)で罫線を描く		○
	細い全角罫線文字で描く		○
	太い全角罫線文字で描く		○
	NECの旧罫線文字を変換		○

キーボードマクロ

ボタン	機能	キー操作	マクロ
	キーボードマクロ記録	Ctrl+Shift+R	
	キーボードマクロ一時停止		
	キーボードマクロ終了		
	キーボードマクロ再生	Ctrl+Shift+P	
	プロジェクトへ保存		

HTML

ボタン	機能	キー操作	マクロ
	 と改行を挿入		○
	次のHTMLタグ	Ctrl+Shift+.	○
	前のHTMLタグ	Ctrl+Shift+,	○
	対応するHTMLタグへジャンプ	Ctrl+Shift+/	○
	HTMLタグ選択		○
	HTMLタグ間のテキストを選択		○
	一対のHTMLタグの内側を選択		○
	一対のHTMLタグの外側を選択	Ctrl+F7	○
	HTMLタグ削除	Shift+F7	○
	一対のHTMLタグを削除	Ctrl+Shift+F7	○
	エレメント/属性名を大文字化		○
	エレメント/属性名を小文字化		○
	文字からエンティティへ変換	Alt+F7	○
	エンティティから文字へ変換	Alt+Shift+F7	○
	HTMLからテキストへ変換		○

ブックマーク

ボタン	機能	キー操作	マクロ
	ブックマーク一覧	Alt+F2	
	設定/解除	Ctrl+F2	○
	ブックマーク1を設定/解除		○
	ブックマーク2を設定/解除		○
	ブックマーク3を設定/解除		○
	次へジャンプ	F2	○
	前へジャンプ	Shift+F2	○
	すべてクリア	Ctrl+Shift+F2	○

検索

ボタン	機能	キー操作	マクロ
	検索	Alt+F3	
	検索ツール	Ctrl+F	
	次を検索	F3	○
	前を検索	Shift+F3	○
	別ペインで次を検索	F5 ＊	○
	別ペインで前を検索	Shift+F5 ＊	○
	次の単語を検索	Ctrl+F3	○
	前の単語を検索	Ctrl+Shift+F3	○
	別ペインで次の単語を検索	Ctrl+F5	○
	別ペインで前の単語を検索	Ctrl+Shift+F5	○
	ファイルから検索	Ctrl+Shift+G	
	ファイル比較	Ctrl+Shift+D	
	クイック比較	Ctrl+Shift+Q	
	左から右へコピー		
	右から左へコピー		
	このファイルから検索		
	このファイルから追加検索		
	カレントグループから検索	F11	
	カレントグループから追加検索	Shift+F11	
	カレントグループ以下から検索	Ctrl+F11	
	カレントグループ以下から追加検索	Ctrl+Shift+F11	
	プロジェクト全体から検索	Alt+F11	
	プロジェクト全体から追加検索	Alt+Shift+F11	
	シンボル定義位置	F12	
	最初のシンボル定義位置		
	最後のシンボル定義位置		
	次のシンボル定義位置	Ctrl+F12	
	前のシンボル定義位置	Ctrl+Shift+F12	

	前の定義ジャンプ位置	Ctrl+Shift+B	
	次の定義ジャンプ位置	Ctrl+Shift+F	
	TAGS作成	Shift+F12	
	定義位置ブラウザ	Alt+F12	
	置換	Ctrl+R	
	検索条件の保存		
	ファイルから検索条件の保存		
	ファイル比較条件の保存		
aA	大文字小文字区別を切り替え	Ctrl+E	○
W	単語単位検索の切り替え	Ctrl+W	○
[^]	正規表現検索の切り替え	Ctrl+Shift+E	○
	あいまい検索		○
	検索強調表示	Ctrl+Alt+F3	
	検索強調クリア		○
	検索強調全クリア	Alt+Shift+F3	

＊ の付いたショートカットキーは、Peggy Pro for PHP ではデフォルトで割り当てられていません。

プロジェクト

ボタン	機能	キー操作	マクロ
	新規プロジェクト		
	プロジェクトを保存する		
	プロジェクトを閉じる		
	新規グループ		
	ファイル挿入		
	フルパス表示		
	位置の確認	Ctrl+Shift+I	
	浅くソート		
	深くソート		
	存在しないファイルの削除		
	ごみ箱を空にする		
	実行中止	Shift+Esc	

デバッグ

ボタン	機能	キー操作	マクロ
	実行	F5	
	中止	Shift+F5	
	ステップイン		
	ステップオーバー		
	ステップアウト		
	カーソル位置まで実行		

	ブレークポイント	F9	
	全ブレークポイント削除		
	ブレークポイント有効切替		
	全ブレークポイントの無効化		
	ブレークポイントリスト	Alt+F9	
	変数ウインドウ		
	スタックウインドウ		
	デバッグレイアウト		

デバッグコマンドは、Peggy Pro for PHP 専用です。

ユーザ定義ツール

ボタン	機能	キー操作	マクロ
	ユーザ定義メニュー 1		
	...省略...		
	ユーザ定義メニュー 12		
	ユーザ定義ツール1	Ctrl+1	
	ユーザ定義ツール2	Ctrl+2	
	ユーザ定義ツール3	Ctrl+3	
	ユーザ定義ツール4	Ctrl+4	
	ユーザ定義ツール5	Ctrl+5	
	ユーザ定義ツール6	Ctrl+6	
	ユーザ定義ツール7	Ctrl+7	
	ユーザ定義ツール8	Ctrl+8	
	ユーザ定義ツール9	Ctrl+9	
	ユーザ定義ツール10	Ctrl+0	
	...省略...		
	ユーザ定義ツール24		
	DBツール1＃		
	...省略...		
	DBツール8＃		
	DBツールの設定＃		
	ツールリスト		
	アスキーコード表		
	カスタマイズ		
	ショートカット		
	オプション		

＃印の付いたコマンドは、Peggy Pro for PHPのみでサポートされています。

スクリプト

ボタン	機能	キー操作	マクロ
	スクリプト１		
	...省略...		
	スクリプト72		
	リロード		
	選択範囲を実行	Alt+X	

アウトライン

ボタン	機能	キー操作	マクロ
	アウトライン解析	Alt+Shift+Enter	
	アウトライン対応項目	Ctrl+Shift+Enter	
	アウトラインプログラム解析		
	アウトライン見出し解析		
	アウトラインHTML解析		

バージョン管理

ボタン	機能	キー操作	マクロ
	Subversionアップデート#		
	Subversionコミット#		
	Subversionステータス#		
	Subversionリビジョンログ#		
	Subversionリビジョン比較#		
	Subversionチェックアウト#		
	Subversionリポジトリブラウザ#		
	CVSアップデート#		
	CVSコミット#		
	CVSステータス#		
	CVSリビジョンログ#		
	CVSリビジョン比較#		
	CVS設定#		
	RCS取得		
	RCSチェックアウト		
	RCSチェックイン		
	RCSチェックアウト取消		
	RCSリビジョンログ		
	RCSリビジョン比較		
	リビジョンログ#		
	リビジョン比較#		
	VSSチェックイン#		
	VSSチェックアウト#		

	VSSチェックアウト取消#		
	VSS取得#		
	VSSソース管理へ追加#		
	VSSソース管理から削除#		
	VSS履歴の表示#		
	VSS相違点の表示#		
	VSSプロパティ #		

#印の付いたコマンドは、Peggy Proのみでサポートされています。

表示

ボタン	機能	キー操作	マクロ
	プロジェクトウインドウ	Alt+1	
	プロジェクトページ	Alt+7	
	ウインドウリストページ	Alt+8	
	アウトラインページ	Alt+9	
	アウトプットウインドウ	Alt+2	
	実行結果ページを開く	Ctrl+Shift+1	
	ファイルから検索ページを開く	Ctrl+Shift+2	
	フォルダ比較ページを開く	Ctrl+Shift+3	
	ファイル比較ページを開く	Ctrl+Shift+4	
	バージョン管理ページを開く	Ctrl+Shift+5	
	TAGSページを開く	Ctrl+Shift+6	
	スクリプトページを開く	Ctrl+Shift+7	
	デバッグページを開く ＊		
	標準出力ページを開く ＊		
	標準エラー出力ページを開く ＊		
	左のアウトプットページ		
	右のアウトプットページ		
	ウインドウリスト	Alt+3	
	ウインドウタブ	Alt+4	
	改行/EOF記号表示の切り替え		○
	タブ表示の切り替え		○
	半角スペース表示の切り替え		
	全角スペース表示の切り替え		○
	キーワードチップ表示の切り替え		○
	ルーラー表示の切り替え		○
	テキスト外領域表示の切り替え		○
	行番号表示の切り替え		○
	行カーソルの切り替え		○
	桁カーソルの切り替え		○
	変更行表示の切り替え		○
	ファイル行単位で行番号を表示		○
	表示行単位で行番号を表示		○
	ガイドライン表示/非表示を切り替え		
	カーソル位置にガイドラインを設定/解除		○

	言語に設定されているガイドラインをクリア		○
	すべての言語に設定されてるすべてのガイドラインをクリア		
	ツールバー		
	ステータスバー		
	リフレッシュ	Ctrl+L	
	レイアウト変更	Ctrl+Shift+Enter	○
	ファイル比較背景のクリア		

＊の付いたコマンドは、Peggy Pro for PHP の専用コマンドです。

ウインドウ

ボタン	機能	キー定義	マクロ
	新しいウインドウを開く		
	ウインドウを閉じる		
	ウインドウ選択	Ctrl+Enter	
	最大化して表示		
	最小化して表示		
	元の大きさで表示		
	大きく表示	Ctrl+Shift+W	
	上下に大きく表示		
	左右に大きく表示		
	上半分に表示		
	下半分に表示		
	左半分に表示		
	右半分に表示		
	2つのウインドウを上下に表示		
	2つのウインドウを左右に表示		
	分割		
	上下に分割	Ctrl+Shift+H	
	左右に分割		
	分割解除		
	リサイクル		
	重ねて表示		
	上下に並べて表示		
	左右に並べて表示		
	アイコンの整列		
	すべて閉じる		
	次のウインドウ	Ctrl+F6	
	前のウインドウ	Ctrl+Shift+F6	
	次のペイン	F6	
	前のペイン	Shift+F6	
	ペインを縦に拡大	Alt+F6	
	ペインを縦に縮小	Alt+Shift+F6	
	ペインを横に拡大	Ctrl+Alt+F6	

	ペインを横に縮小	Ctrl+Alt+Shift+F6	
	左のタブを選択		
	右のタブを選択		
	左端のタブを選択		
	右端のタブを選択		
	左隣のタブと入れ替え		
	右隣のタブと入れ替え		
	ウインドウへ保存		

ヘルプ

ボタン	機能	キー操作	マクロ
	ヘルプ	F1	
	状況ヘルプ	Shift+F1	
	目次		
	トピックの検索		
	キーワードヘルプ	Ctrl+F1	
	ユーザーズマニュアル		
	バージョン情報		

その他

ボタン	機能	キー操作	マクロ
	何もしないコマンド		

索引

Symbols

&amp 89
&gt 89
&lt 89
&quot 89
()、[]、{}、「」、〈〉、()、{}、《》、【】、〔〕、[]、「」、『』、"" 65,74

A

awk.exe 172

C

cmd.exe 40
COLORREF 72,304
CR 55,94,311
CREC 3,100,143,145,318
CR-LF 29,55,94,311
CSV ファイル 256
ctags 80,118,121,122
CVS 198,204,339
CVS アップデート 214
CVS コマンド 204
CVS コミット 215
CVS ステータス 206
CVS タグ設定 218
CVS ファイル削除 217
CVS ファイル追加 217
CVS フォルダ選択 218
CVS リビジョンログ 210
CVS ログフィルタ 212,339

D

DBMS_OUTPUT 255
DB タイプ 255
DB ツール 254

E

etags 80,118,121,122
EUC 29,53,93,94,171,315
EXEC 261

H

Hello, world 188
HTML 65,72,74,80,87,307
HTML 解析 81

I

IME 307
initial.peg 150

J

JavaScript 2,65,74
JIS 29,53,93,94,171,315

L

LF 29,55,94,311

M

make.exe 40,172
Makefile 42
MIFES 風 296
moca.exe 172
MocaScript 2,12,36,187

N

NEC 罫線文字 82

O

ODBC 2,254,255
ODBC ドライバ 254
Oracle 2,8,12,254,255
Oracle クライアント 254

P

Peggy の終了 56
Peggy の特徴 12
perl.exe 172
PHP 168,264,343
php.ini 267,274
phpinfo() 268
PHP デバッガの設定 169
PL/SQL 260
PRINT 261

R

RCS 198,339
RCS コマンド 202
RCS チェックイン 203
RCS リビジョンログ 210
RGB 値 72,303
RRGGBB 72
ruby.exe 172

S

sed.exe 172
Shift JIS 29,53,93,94,171
SQL*NET 8,254
startup.ms 193
Subversion 198,219,340
Subversion アップデート 230
Subversion アンロック 235
Subversion コピー 233
Subversion コマンド 219
Subversion コミット 231
Subversion 削除 232
Subversion 衝突解除 234
Subversion ステータス 220
Subversion チェックアウト 229
Subversion 追加 232
Subversion フォルダ選択 218
Subversion 戻す 234
Subversion リビジョンログ 224
Subversion リポジトリブラウザ 228
Subversion ログフィルタ 226,341
Subversion ロック 235
svn.exe 219,226,340

T

TAGS 120,124,168,321
TAGS のインポート 124
Times New Roman 336
TNS 名 255
TRIGGER 261

U

Unicode 29,53,93,94,171
URL・パス変換 169,270,343,344
URL ルート 169,270,344
URL をブラウズ 58
UTF-7 29,53,93,94,171,315
UTF-8 29,53,93,94,171,315
UTF-8N 94

V

Visual SourceSafe..... 237
VSS 168, 338
VSS コマンド 242
VSS 最新のバージョンを取得 246
VSS 相違点表示オプション 249
VSS ソース管理へ追加 247
VSS ソースコード管理から削除..... 248
VSS チェックアウト 244
VSS チェックアウトの取り消し..... 245
VSS チェックイン..... 243
VSS ファイルの履歴表示オプション 249
VSS プロパティ..... 251

W

Web サーバー 169, 264, 266, 344

X

XAMPP..... 265, 270, 272
Xdebug..... 266, 274
XDEBUG_SESSION_START 271
XML 80, 315

あ

アイコンの整列 98
アイテムを削除 155
あいまい検索..... 106, 108, 114, 132
アウトプットウインドウ 162
アウトライン..... 80
アウトラインページ..... 330
新しいウインドウを開く 96
新しいショートカットキー..... 294
アドホックパラメタ..... 177
アトリビュート名 88
アンインストール 9
アンロック 223, 235

い

移動ページ 302
インクリメンタルサーチ 108
インクルードファイルを開く 59
印刷..... 33, 60
印刷オプション 61
印刷フォント..... 335
印刷プレビュー 61
印刷ページ 332
インジケータ..... 15, 312
インストール..... 8, 254
インテリマウス 64, 306
インデント 86, 101, 308, 326

う

ウインドウ選択 98
ウインドウタブ 316
ウインドウに保存 98
ウインドウメニュー..... 95
上書カーソル..... 319
上書きモード..... 303
上書きモードインジケータ..... 16

え

エディタプロパティ..... 101
エラー出力 345
エレメント名..... 88
エンティティ..... 88

お

オートインデント 308
大文字化 90
大文字と小文字を区別する.... 108, 115, 132
オプション設定..... 296

か

カーソル..... 15
カーソル位置まで実行..... 277
カーソル移動 62
カーソル位置まで実行..... 271
改行コード 29, 52, 55, 72, 94, 319
改行をスペースに変換..... 91
ガイドライン 312, 326
外部スクリプト..... 40
外部ビューア 342
改頁 333
拡張子ページ 329
拡張子マップ 143, 329
重ねて表示 98
頭文字化..... 90
カスタマイズ 17, 48, 283, 285, 292
カスタムフィルタ 298
加速スクロール..... 320
加速遅延..... 320
カタカナへ変換..... 90
カテゴリ..... 286
カラー印刷 334
カラーコーディング 2, 142
環境変数..... 318
漢字コード... 29, 52, 55, 93, 171, 319, 346
漢字コードインジケータ..... 16
関数 333
簡単スタート 271, 275, 343
関連アプリケーション..... 59
関連ファイルを開く 59

き

キーボードマクロ 27, 79, 158, 307
キーボードマクロインジケータ..... 16
キーボードマクロプロパティページ 180
キーワードチップ..... 15, 312
キーワード定義ファイル..... 9
キーワードヘルプ..... 100
キーワード補完..... 85, 322
逆転 95
行カーソル 15, 306
行間 333
行間スペース 324, 326
行コメント 86
行順序 95
強制保存..... 55
行選択 65
行挿入 70
行番号 21, 312, 334
行末 333
行列インジケータ 16, 62
切り取り..... 67, 155
禁則処理..... 325, 327

く

クイック検索 111
クイックスクロール 64
クイック比較 125
矩形選択..... 65, 66
クリップボード..... 66, 67, 155, 306
グループ..... 152
グループから検索 116
グループデフォルト 163, 168
グループプロパティページ 167

け

警告 313
罫線 26, 309
罫線機能..... 81

桁カーソル 15, 306
言語 332
言語インジケータ 16
言語ページ 326
言語モード 143, 163
検索 23, 106, 107, 309
検索強調表示 117, 310
検索条件 106, 159
検索ツール 109, 309
検索方法プロパティページ 183
言語サポート DLL 9

こ

降順 95
更新しないホスト 298
コーディング 192
コードインジケータ 16
ゴシック 335
コピー 67, 155
コマンド検索パス 301
コマンド宣言 189
コマンドライン引数 102
コミット 256
ごみ箱 156, 299
コメント 85, 309
小文字化 90
コンパイル 192

さ

サーバールート 169, 270, 345
最近使ったファイル 58, 298
最近使ったフォルダ 58, 298
最近使ったプロジェクト 58, 298
再検索 111
最大化 103, 317
最大要素数 346
再変換 70, 307
作業コピー 202, 215, 220, 341
削除 70, 72
左右に並べて表示 98
右に分割 98

し

実行終了ベル 301
実行中止 158
実行フォルダ 173
実行方法プロパティページ 172
実行前保存 318
指定行ジャンプ 21, 73
自動的に復元するファイル数 299
ジャンプ 21, 73
状況ヘルプ 100
上下に並べて表示 98
上下に分割 98
条件コンパイル文 75
条件保存 134
昇順 95
ショートカット 288
ショートカットキー 293
除外フォルダ 114
シリアル番号 101
新規グループ 152
新規ツールバー 286
新規ファイル 319
新規プロジェクト 149
シンボル検索 117

す

すかし 334
スクリプト言語 2
スクリプトコマンド 288
スクリプト実行 58
スクリプトの実行 188, 193
スクリプトファイルの実行 162
スクリプトボタン 288
スクロール 63
スクロールバー 15, 306
スクロールマージン 320
スタックウインドウ 281
ステップアウト 277
ステップイン 271, 277
ステップオーバー 271, 277
スプリッタ 15
スペースをタブに変換 90
すべて閉じる 98
すべて保存 55
スマートインデント 87, 308

せ

正規表現 106, 108, 132, 134
正規表現とあいまい検索 139
制御コード 70
整数フォーマット 346
セッション ID 344
設定コピー 332
セットアップ 8
セパレータ 256
全角スペース 304, 308, 311, 314
全角に変換 91
選択 65
選択範囲 306
全半 334
全般ページ 297

そ

相違箇所のコピー 126
挿入 70
挿入カーソル 319
挿入モード 319
ソート 95, 154

た

対応括弧ジャンプ 22
対応括弧の強調表示 312
対応括弧へジャンプ 74
対応するタグ 88
大小文字の区別 106
ダイレクト印刷 60
ダイレクトオープン 59
タグジャンプ 22, 74, 163, 259, 309
タブ間隔 319
ダブルクリック 304
タブをスペースに変換 90
段組 333
単語単位で探す 108, 115, 132
単語として検索 106
単語補完 83
単語補完入力 321
短縮入力 84, 323

ち

チェックイン 200
遅延時間 290
置換 25, 131
置換の範囲 132
中央揃え 87
中心 333

つ

追加検索 114
ツールアクション .....
42, 161, 165, 171, 180, 182, 183, 185

ツールタイトル 255
ツールとの組み合わせ 45
ツールバー 284, 285
ツールボタン 42
ツールボタンの追加 285
ツールリスト 186

て

定義位置ブラウザ 120
データチップ 346
テキスト外領域 15
テキスト領域 15
デバッグ 193, 264
デバッグインジケータ 16
デバッグ実行 277
デバッグ中止 277
デバッグページ 343
デバッグレイアウト 281, 344
テンプレート 52, 298
テンプレートパス 298

と

ドキュメントルート 169, 270, 345
特殊タグパターン 75, 168, 298
ドッキング 285
トピックの検索 100
ドラッグ＆ドロップ 68, 307
取り消しとやり直し 69
トリプルクリック 66, 304
トレース実行 193

に

二重起動 297

は

バージョン管理 3, 198, 204, 219, 237
バージョン管理ページ 338
バージョン情報 101
排他制御 56
バインド変数 260
パスカウント 278
パスワード 240, 255, 340
バックアップ 314
バックスペース 72
バッチファイル 163
貼り付け 67, 155
半角スペース 311
半角に変換 91

ひ

比較 31, 124
比較条件 159
比較方法プロパティページ 185
引数 173
日付 334
左から右へコピー 127
左揃え 87
日付と時刻 71
ビットマップ 289, 292
秀丸風 296
表示位置 96
表示色ページ 337
標準出力 103, 174, 273, 345
ひらがなへ変換 90
ビルトインスクリプト 187

ふ

ファイヤーウォール 266
ファイルから検索 112
ファイルから検索プロパティページ 182
ファイル検索パス 167
ファイル挿入 151
ファイル比較 125
ファイル比較プロパティページ 184
ファイルヒストリ 56
ファイルプロパティページ 170
ファイルメニュー 50
フィルタ 109
フォルダ取得 114
フォルダ比較 127
フォント 299, 335
ブックマーク 14, 21, 76, 108, 318, 334
ブックマーク一覧 78
部分印刷 35, 60
ぶら下げ表示 325
プリンタの設定 61
古いリビジョンを見る 201
ブレークポイント 2, 277
プレースバー 313
フレーム 334
フローティング 284
プログラム解析 81
プロジェクト 149
プロジェクト機能 3, 148
プロジェクトデフォルト 163, 168
プロジェクトから検索 116
プロジェクトファイル 37
プロジェクトプロパティページ 167
プロジェクトページ 299
プロジェクトを移動 153
ブロックコメント 86
プロパティ 166
分割 97
分割解除 98

へ

頁番号 335
ヘルプ 100
ヘルプメニュー 99
変更箇所を確認 202
変更行へジャンプ 21, 76
編集 49
編集ウインドウ 15
編集禁止 101, 171
編集禁止インジケータ 16
編集操作リスト 69
編集メニュー 61
変数ウインドウ 279

ほ

方向 332
ポート番号 267, 344
補完 28, 310
補完入力 83
補完ページ 321
ボタンギャラリー 254, 289
ポップアップメニュー 316

ま

マーカー領域 15, 65, 304
マーク設定 108
マージン 335
マウスアクション 161, 171, 172, 180, 181, 182, 184
マニュアルの構成 5

み

右から左へコピー 127
右揃え 87
見出し解析 81
明朝 335

め
メール引用符 71, 319

も
目次 100
文字クラス 135, 139
文字コード 315
文字の入力 66
文字変換 89
文字列検索プロパティページ 181

ゆ
ユーザーズマニュアル 100
ユーザ定義キーワード 143, 328
ユーザ定義ツール 186
ユーザ定義メニュー 290, 316, 327
ユーザ名 239, 255, 340

よ
用紙 332

ら
ライセンスキー 9, 101
ラベル行ジャンプ 75

り
リサイクル 79, 98, 309
リダイレクト 176, 256, 345
リビジョン 201, 207, 222
リポジトリ 204, 212, 228
リポジトリブラウザ 342
両面 332

る
ルーラー 15, 311, 312

れ
レイアウト切り替え 344
レイアウトページ 324
例外ブレーク 344
レギュラーレイアウト 344

ろ
ローカルルート 169, 270, 345
ログメッセージ 203
ログを見る 201
ロック 223, 235

わ
ワードラップ 325, 327
ワンキーコマンド 279

行番号 73, 102, 163, 334

2009年9月5日　第１版
2010年1月5日　第２版

Programmer's Edit Gear

Peggy シリーズ Ver.4.6
ユーザーズマニュアル

発行 株式会社エスコア

落丁・乱丁はお取り替えいたします。