



# **TortoiseHg マニュアル**

リリース **2.7.0**

**Steve Borho、他**

**2020 年 08 月 19 日**



# 目次

第 1 章	はじめに	3
1.1	対象読者	3
1.2	本書の読み方	3
1.3	ライセンス	4
1.4	コミュニティ	4
1.5	謝辞	4
1.6	表記規則	4
第 2 章	導入	7
2.1	TortoiseHg とは何か?	7
2.2	TortoiseHg のインストール	7
2.2.1	Windows の場合	7
2.2.2	Linux または Max OS の場合	9
第 3 章	TortoiseHg クイックスタート	11
3.1	TortoiseHg の設定	11
3.2	リポジトリの初期化	13
3.3	ファイルの追加	14
3.4	コミット	15
3.5	リポジトリの共有	15
3.6	共有リポジトリからの取得	17
3.7	リポジトリを使ったワークフロー	17
第 4 章	TortoiseHg ダイアログ	19
4.1	共通の機能	19
4.1.1	キーボードショートカット	19
4.1.2	差分表示	19
4.1.3	ツリービュー内検索	22
4.1.4	HG コマンドダイアログ	22
4.2	Windows エクスプローラ統合	23
4.2.1	アイコンオーバーレイ	23
4.2.2	右クリックメニュー	24
4.2.3	Nautilus	25

4.3	新しいリポジトリを作成する	26
4.3.1	コマンドラインからの実行	28
4.4	クローン	28
4.4.1	コマンドラインからの実行	29
4.5	コミット	30
4.5.1	機能	30
4.5.2	変更点の選択	33
4.5.3	キーボードショートカット	34
4.5.4	ファイルの右クリックメニュー	35
4.5.5	マージ作業	35
4.5.6	コミットメッセージパネル	36
4.5.7	MQ パッチ	36
4.5.8	QNew モード	37
4.5.9	オプション設定	37
4.5.10	コマンドラインからの実行	38
4.6	シェルフ	38
4.6.1	タブ	40
4.6.2	変更点をシェルフする	40
4.6.3	シェルフした変更点を戻す	40
4.6.4	キーボードショートカット	41
4.6.5	オプション設定	42
4.6.6	コマンドラインからの実行	42
4.7	チェンジログ	42
4.7.1	同期バー	44
4.7.2	フィルタバー	45
4.7.3	リビジョングラフ	46
4.7.4	実行速度への影響	46
4.7.5	リビジョンの右クリックメニュー	46
4.7.6	ファイルの右クリックメニュー	49
4.7.7	チェンジセットブラウザ	49
4.7.8	コミットメッセージの解析	50
4.7.9	Bisect	50
4.7.10	キーボードショートカット	51
4.7.11	オプション設定	51
4.7.12	コマンドラインからの実行	52
4.8	リポジトリ検索	52
4.8.1	検索タブ	52
4.8.2	コメントタブ	54
4.8.3	コマンドラインからの実行	55
4.9	同期ツール	55
4.9.1	プルした後	58

4.9.2	メール送信	58
4.9.3	コマンドラインからの実行	59
4.10	サーバ	59
4.10.1	複数のリポジトリ	60
4.10.2	コマンドラインからの実行	60
4.11	名前変更の推定	61
4.11.1	名前変更の検索	62
4.11.2	コピーの検索	62
4.11.3	推定結果	62
4.11.4	コマンドラインからの実行	62
4.12	無視フィルタ	62
4.12.1	コマンドラインからの実行	63
第 5 章	設定ダイアログ	65
5.1	設定ページ	66
5.1.1	TortoiseHg	67
5.1.2	コミット	67
5.1.3	チェンジログ	68
5.1.4	同期	68
5.1.5	サーバ	69
5.1.6	プロキシ	70
5.1.7	メール送信	70
5.1.8	差分	70
5.1.9	フォント設定	71
5.2	今回のリリースで削除された設定項目	71
5.3	キーボードショートカット	72
5.4	コマンドラインからの実行	72
第 6 章	リカバリ	73
第 7 章	パッチ	75
7.1	パッチとは	75
7.2	落とし穴	75
7.3	パッチのエクスポート	76
7.3.1	チェンジセット	76
7.3.2	リビジョン範囲	76
7.3.3	メール送信	76
7.3.4	チェンジセットのつまみ食い	77
7.4	パッチのインポート	78
7.5	MQ (パッチキュー)	79
第 8 章	拡張機能	81

8.1	Hgfold . . . . .	81
8.2	Hgcr-gui . . . . .	82
8.3	Perforce . . . . .	83
8.4	HGEOL . . . . .	83
8.5	Mercurial-Keyring . . . . .	83
第 9 章	他のバージョン管理システムとの相互運用	87
9.1	Perforce (Perforce) . . . . .	87
9.2	hgsubversion (SVN) . . . . .	89
9.3	hg-git (git) . . . . .	90
第 10 章	よくある質問	91
第 11 章	TortoiseHg のデバッグ	95
11.1	ダイアログ . . . . .	95
11.1.1	Windows . . . . .	95
11.1.2	Linux または MacOSX . . . . .	95
11.2	シェル拡張 . . . . .	96
11.2.1	Windows エクスプローラ . . . . .	96
11.2.2	Nautilus ファイルマネージャ . . . . .	96
第 12 章	索引	97
	Python モジュール索引	99
	索引	101

目次:





# 第 1 章

## はじめに

### 1.1 対象読者

本書は Mercurial を利用してファイルやデータを管理したいけど、コマンドラインから操作するのはやや不便だと感じる、ある程度コンピュータに慣れ親しんだ方のために書かれました。TortoiseHg は Windows のシェル拡張として提供されるため、ユーザは普段使っている Windows エクスプローラから利用することができます。

本書の最新版は [公式サイト](#) からダウンロード可能です。

### 1.2 本書の読み方

この章では本書の構成、開発や議論の中心となっているコミュニティ、利用および再配布のソフトウェアライセンスなど、TortoiseHg プロジェクトについて少し説明をしておきます。

**導入** では TortoiseHg とは何か、何ができるのか、どこからどうやってインストールすればいいのかなど、基本的な事柄について説明します。

**TortoiseHg クイックスタート** は TortoiseHg に初めて触れる人のための短いチュートリアルです。

**TortoiseHg ダイアログ** はこのマニュアルの本編で、TortoiseHg でよく利用するダイアログやコンポーネントについて解説します。

**設定ダイアログ** では TortoiseHg の設定ダイアログについて詳しく説明しています。

**リカバリ** はプロジェクトで行うリカバリ操作について書かれています。

**他のバージョン管理システムとの相互運用** では TortoiseHg を他のバージョン管理システムのクライアントとして使用する方法について述べています。

**よくある質問** には TortoiseHg 自体に関する質問や、多くの人がぶつかる問題の解決方法などがまとめてあります。

**TortoiseHg のデバッグ** は TortoiseHg のバグを見つけたときに、どのようにしてデバッグをすればいいのかについて書かれています。

## 1.3 ライセンス

TortoiseHg は [GPLv2](#) の下でリリースされています。好きなコンピュータに好きなだけインストールすることができますし、GPLv2 ライセンスに従う限り再配布も可能です。

## 1.4 コミュニティ

Bitbucket には [公式 Wiki](#) もあります。日本語 Wiki は [こちら](#) になります。

## 1.5 謝辞

TortoiseHg プロジェクトに貢献し、本当に便利なツールを作り上げたすべての開発者、翻訳者、利用者コミュニティ、特に貴重なバグレポートおよび機能要望をしていただいたユーザの方々に感謝の意を表します。

TortoiseHg の Windows 向けインストーラには [TortoiseSVN](#) プロジェクトによる TortoiseOverlays ハンドラが含まれています。

## 1.6 表記規則

本マニュアルでは以下の表記規則に従っています:

`Ctrl-A` 押すべきキーボードのキーまたはキーの組み合わせを示しています。

`コミット` ラベル、ボタンなど GUI に表示される文字列を示しています。

*TortoiseHg... → About* メニュー選択およびその組み合わせ、またタブ選択、GUI のラベルを示します。例えば *TortoiseHg... → ユーザ設定 → コミット → ユーザ名* はメニューから *TortoiseHg... → ユーザ設定* とたどり、開いたダイアログの `コミット` というページに切り替えて、その中にある `ユーザ名` というラベルの付いた場所では何かをする、という意味になります。

`.hg/hgrc` ファイル名やディレクトリ名を示します。

`hg tk log` コマンドプロンプトなどへの入力を示します。

`myproxy:8000` GUI のテキストボックスへの入力を示します。

---

注釈: 注釈です。

---

警告: 重要な注釈です。注意してください。



## 第 2 章

# 導入

### 2.1 TortoiseHg とは何か？

TortoiseHg は [Mercurial](#) 分散構成管理のための GUI ツールと Windows エクスプローラに統合するためのシェル拡張のパッケージです。

Windows では、TortoiseHg はエクスプローラにアイコンオーバーレイおよび右クリックメニューを組み込むシェル拡張と、TortoiseHg の各種ダイアログを起動するためのコマンドラインツール `hgtk.exe` から構成されます。Windows 用インストーラには Mercurial やマージツール、それ以外に必要なものすべてを含んでおり、これをインストールするだけですぐ使えるようになっています。

Linux では、TortoiseHg はコマンドラインから利用する `hgtk` スクリプトとファイルマネージャにアイコンオーバーレイと右クリックメニューを提供する Nautilus 拡張機能から構成されます。TortoiseHg を Linux で利用するには別途 Mercurial をインストールする必要がありますが、通常は TortoiseHg バイナリパッケージが Mercurial を依存ソフトウェアとしてリストアップするため、自動的にインストールされます。

TortoiseHg は (Windows のシェル拡張を除いて) 主に Python と PyGTK で書かれています。`hgtk` スクリプトと TortoiseHg の各種ダイアログは (例えば Mac OS など) PyGTK に対応した環境であればどこでも使うことができます。

### 2.2 TortoiseHg のインストール

#### 2.2.1 Windows の場合

TortoiseHg には簡単にセットアップできるようにインストーラが用意されています。ダウンロードしたインストーラをダブルクリックして起動し、後はインストーラに表示される手順に従って完了させてください。面倒な作業はすべてインストーラがやってくれます。

インストール後の再起動は必須なので注意してください。

注釈: バージョン 0.8 より古い TortoiseHg がインストールされている場合、インストーラは最初に古いバージョンをアンインストールするよう促します。アンインストールは通常のソフトウェアと同様に、スタートメニューからコントロールパネルを開き、その中の「プログラムの追加と削除」から可能です。

---

注釈: バージョン 0.8 以降の TortoiseHg がインストールされている場合、インストーラは新しいファイルをコピーするときに ThgTaskbar アプリケーションを終了します。

---

TortoiseHg は使いやすい MSI インストーラを採用しています。最新バージョンは TortoiseHg の [ウェブサイト](#) から入手できます。ダウンロードしたらインストーラをダブルクリックして起動し、表示される指示に従ってインストールを進めてください。

初回インストール後はアイコンオーバーレイを有効にするために通常は再ログインが必要です。

アップグレードインストールの場合は、インストーラが TortoiseHg のシェル拡張を読み込んでいるアプリケーションを検出し、それらを終了するか、もしくは再起動するか尋ねます。すべてのアプリケーションの終了を許可すれば、再起動・再ログインせずにインストールを完了できます。もし他のユーザがログインしていたり、終了できないアプリケーションがあった場合のみインストールを完了するために再起動が必要になります。

---

注釈: 古い TortoiseHg がインストールされている場合、TortoiseHg 1.0 のインストーラはそれをアンインストールするよう指示します。アンインストールはコントロールパネルの「プログラムの追加と削除」、またはスタートメニューから実行できます。

---

警告: アンインストール時、古いインストーラはホームディレクトリにある Mercurial.ini ファイルを削除する可能性があるため、事前にバックアップを取っておいてください。この不具合は新しい MSI インストーラでは問題ありません。

バージョン 1.0 以前のすべての TortoiseHg インストーラは InnoSetup を使用してビルドされていました。古いインストーラは TortoiseOverlay パッケージを独立したアプリケーションとしてインストールしていたため、コントロールパネルの「プログラムの追加と削除」において TortoiseHg と TortoiseOverlay の 2 つのアプリケーションが表示されていました (64 ビット環境では 32 ビット向けと 64 ビット向けに 2 つの TortoiseOverlay パッケージ)。

TortoiseHg 1.0 の新しい MSI インストーラは TortoiseOverlay を "マージモジュール" として収録しているため、今後は独立したアプリケーションとして表示されることはありません。バージョン 0.9.3 以前の TortoiseHg をアンインストールした後、もし独立した TortoiseOverlay パッケージを使用するアプリケーション (TortoiseCVS や TortoiseBZR) がインストールされていないのであれば、古い TortoiseOverlay をコントロールパネルの「プログラムの追加と削除」から安全にアンインストールすることができます。

---

注釈: TortoiseOverlay は Tortoise 系アプリケーション間でオーバーレイスロットを共有する仕組みを提供するパッケージです。Windows はオーバーレイスロットの数に限り (11 から 14) があるため、これは必要不可欠な仕組みです。TortoiseOverlay パッケージは TortoiseSVN 開発陣によって作成されました。

---

より完全な安全性を求めるのであれば 2 つの方法があります :

- 1) インストール済みの TortoiseOverlay パッケージをアンインストールせず、そのまま残しておく。残しておいても特に害はありません。
- 2) 一度すべての TortoiseOverlay パッケージをアンインストールしてから、Tortoise 系アプリケーションをすべて再インストールする。

### 言語設定

TortoiseHg のユーザインターフェイスは様々な言語に翻訳されています。すべての利用可能な言語はインストール済みなので、別途言語パックをダウンロードする必要はありません。C:\Program Files\TortoiseHg\locale に利用可能な言語が用意されています。環境変数 LANGUAGE に言語コードを設定するだけでその言語が利用可能になります。例えば日本語であれば `set LANGUAGE=ja` を設定してください。

---

注釈: LANGUAGE を設定しても *OK* や *Apply* のような GTK+ 標準のダイアログボタンのラベルが英語のままになっている場合、TortoiseHg のインストーラがその言語の GTK+ 翻訳ファイルを含んでいないことを意味します。これはリリース時にご利用の言語の TortoiseHg 翻訳が完了していなかったためです。

---

Windows シェル拡張の右クリックメニューはその翻訳文字列を Windows のレジストリから取得しています。すでに多くの言語向けに翻訳されており、それらは C:\Program Files\TortoiseHg\i18n\cmenu にインストールされています。使用したい言語のレジストリファイルをダブルクリックすると確認ダイアログが出ますので、指示に従ってインストールしてください。

## 2.2.2 Linux または Max OS の場合

Debian の DEB パッケージまたは Fedora 向けの RPM パッケージはダウンロード <<https://tortoisehg.bitbucket.io/download/linux.html>> ページに用意されています。

Max OS X 向けのパッケージは用意されていませんが、ソースコードからインストールすることで hgtk スクリプトを含むすべての TortoiseHg ダイアログが利用可能になります。詳しくは [MacOSX](#) をご覧ください。

---

注釈: TortoiseHg をソースコードからインストールしている場合、contrib/mergetools.rc ファイルを HGRC パスに

---

含める必要があります。例えば `%include` を使うことで `~/.hgrc` ファイルから読み込みます。

---

### 言語設定

TortoiseHg はローカライズに `gettext` ライブラリの Python バインディングを利用しています。ダイアログをローカライズするには環境変数 `LANGUAGE` を使用したい言語に設定するのが簡単です。



## 第 3 章

# TortoiseHg クイックスタート

TortoiseHg と Mercurial によろこ！ TortoiseHg は Mercurial 分散バージョン管理システム (DVCS) に Windows エクスプローラのシェル拡張と使い勝手の良い GUI フロントエンドを提供するアプリケーションです。すべての TortoiseHg の機能はエクスプローラの右クリックメニューからだけでなく、付属の `hg` コマンドラインツールからも起動することができます。Mercurial のコマンドも通常の `hg` コマンドで利用できます。

### 3.1 TortoiseHg の設定

まず最初にやることは TortoiseHg にあなた自身について教えることです。TortoiseHg のユーザ設定ダイアログから設定できます。プロジェクトディレクトリを右クリックして *TortoiseHg* → ユーザ設定 を選択します。

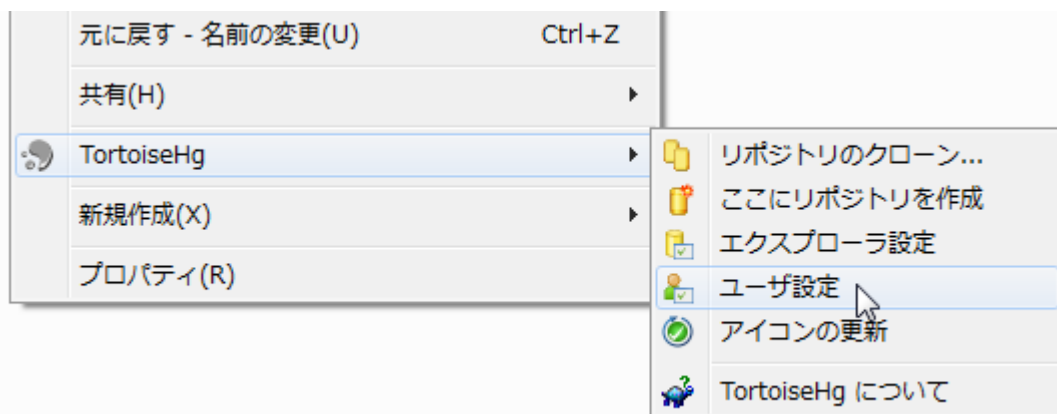


図 1 デスクトップからユーザ設定ダイアログを開く

このメニューは TortoiseHg 設定ダイアログを開いて、ホームディレクトリにあるあなたのユーザ設定を編集します。コマンドラインから使用しているのであれば `hg` `userconfig` でユーザ設定ダイアログを開くことができます。

そして コミット ページに切り替えて、ユーザ名 のテキストボックスに名前を入力してください。

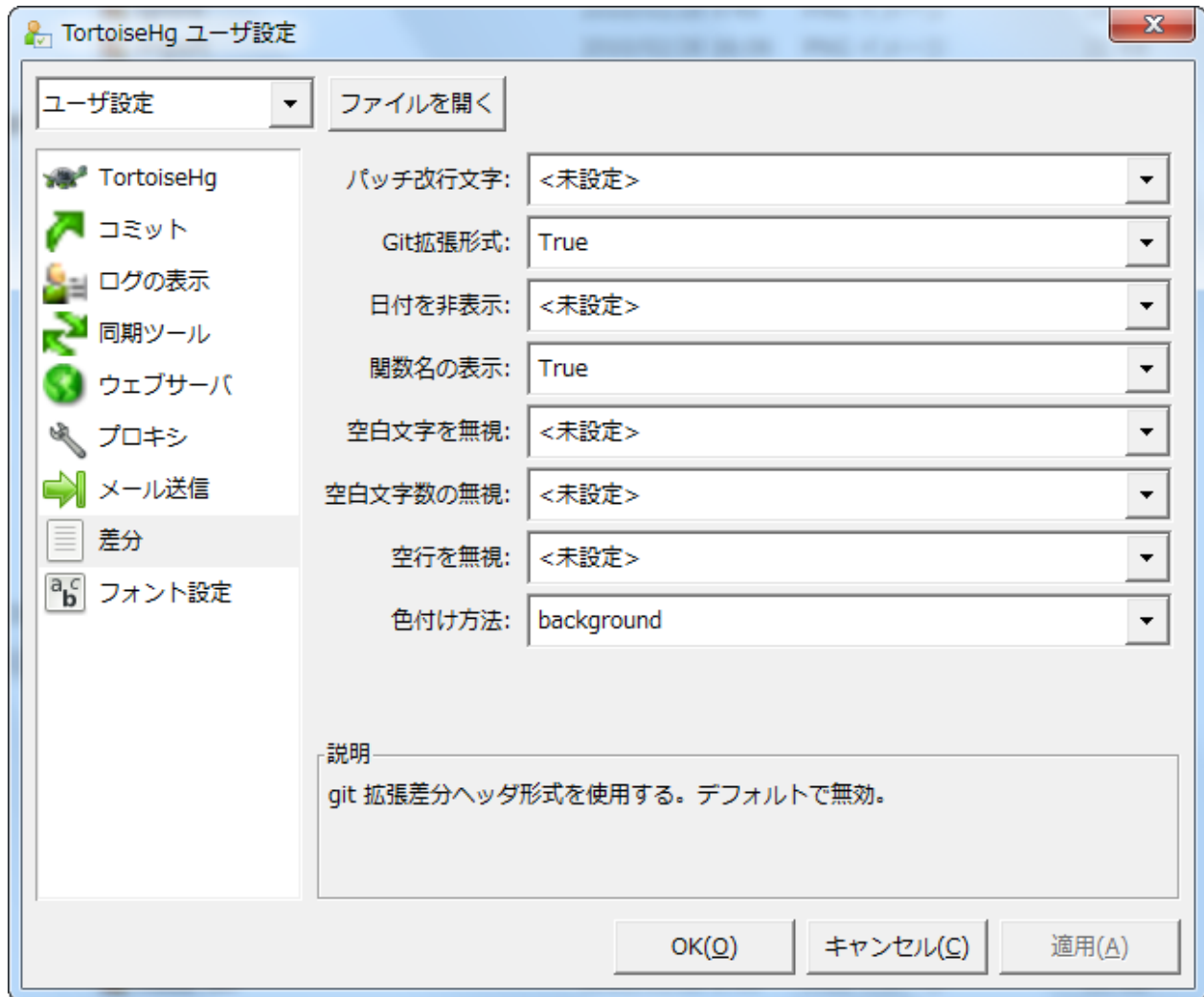


図 2 TortoiseHg 設定ダイアログ

注釈: ユーザ名の設定をし忘れた場合、TortoiseHg は必要に応じてコミット時にユーザ名の入力を促します。

注釈: ユーザ名として設定する名前の形式に厳密なルールがあるわけではありませんが、多くの人が次のような形式を採用しています:

氏名 <電子メールアドレス>

例えば:

Taro Yamada <taroyamada@example.net>

電子メールアドレスはチェンジログビューアで履歴を表示する際には削除され、Mercurial 内蔵のウェブサーバを

利用した場合はスパム対策のためメールアドレスの一部のみ表示されます。

---

次に *TortoiseHg* ページに切り替え、*3-way* マージツールの設定項目を選択します。このドロップダウンリストにはシステム内で検出されたすべての利用可能なマージツール (*kdiff3* は Windows 向けの *TortoiseHg* インストーラ同梱) と *Mercurial* 組み込みのマージ処理が含まれています。好みのマージツールを選んでください。

選択したマージツールを GUI 差分表示ツールとしても利用したい場合は *GUI* 差分表示ツールの設定項目を未設定のままにしておいてください。マージツールとは異なる他の差分ツールを利用したい場合はドロップダウンリストから好きなツールを選択してください。

ドロップダウンリストに選択肢がない場合は *mergetools.rc* がサポートしている差分/マージツールをインストールするか、未サポートのツールであれば *Mercurial.ini* ファイル、または *.hgrc* ファイルに設定を記述する必要があります。

---

注釈: *TortoiseHg* をソースコードからインストールしている場合、*contrib/mergetools.rc* ファイルを HGRC パスに含める必要があります。例えば `%include` を使うことで *~/.hgrc* ファイルから読み込みます。

---

他のユーザ設定も自由に設定してください。必要であればユーザ設定における設定値はリポジトリごとの設定で上書きすることができます。

OK ボタンを押して変更を保存して完了です。

---

注釈: ほとんどの *TortoiseHg* ダイアログは設定を反映するために再起動する必要があります (システムの再起動ではない)。

---

## 3.2 リポジトリの初期化

右クリックメニューの **ここにリポジトリを作成** を選択するか、コマンドラインから `hg tk init` を実行して最初のリポジトリをローカルディスク上に作成してみましょう。この作業はプロジェクトディレクトリのルートで一度だけ行ってください。

特に理由がなければ特殊ファイルを追加する (*.hgignore* など) にチェックを入れ、*Mercurial 1.0* 互換のリポジトリを作成のチェックは外した状態でリポジトリを作成すると良いでしょう。

作成 ボタンを押すと *Mercurial* はプロジェクトディレクトリに *.hg* という名前のディレクトリを作成します。ここには *Mercurial* のすべてのバージョン情報が保管され、一般に **リポジトリ**と呼ばれています。一方、実際のソースファイルなどが含まれているディレクトリを**作業ディレクトリ**と呼びます。コマンドを実行するときこの *.hg* ディレクトリを指定する必要はなく、作業ディレクトリのルートを指定するだけで大丈夫です。これは *Mercurial* がどのように動いているのかを理解するために覚えておくとも良いかもしれません。

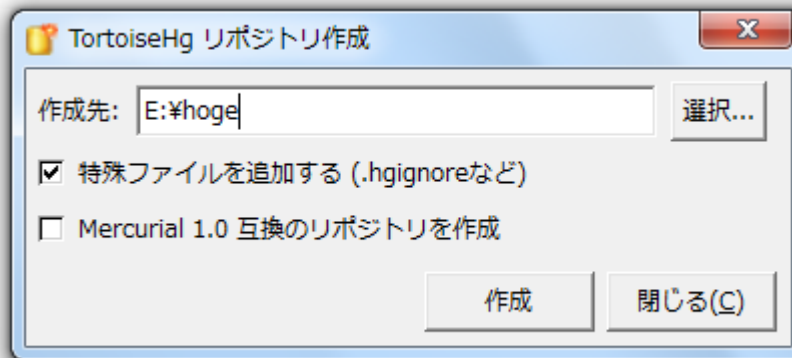


図 3 リポジトリ初期化ダイアログ

警告: `.hg` を直接編集することはリポジトリの破損を引き起こすため危険です。ただし `.hg/hgrc` のみ例外的に編集が認められています。

### 3.3 ファイルの追加

いよいよ Mercurial にどのファイルを管理下に置き、どのファイルを見捨てるべきなのかを指定します。これにはいくつかのやり方があります：

1. エクスプローラで追加したいファイルを右クリックしてメニューを開いて *TortoiseHg* → 追加... を選択します。するとファイル追加を確認するダイアログが表示されるので、それでよければ 追加 ボタンを押してください。
2. もしくは、ステータスツール (*TortoiseHg* → 変更点の確認またはコマンドラインから `hg tk status`) を開いて追加したいファイルのチェックボックスにチェックを入れ、追加 ボタンを押します。ステータスツールでは、不明なファイルの右クリックメニュー (無視) から無視フィルタツールを開くこともできます。
3. または、ファイルの追加を独立した作業として行わず、コミットツールで済ますことも可能です。コミットツールはステータスツールと非常に似ており、ステータスツールで可能な操作はすべてコミットツールでも行えます。このツールではファイルにチェックを入れて コミット ボタンを押すことで、まだ Mercurial 管理下に置かれていないファイルの追加およびコミットすることができます。
4. 無視ファイルを指定するために無視フィルタダイアログを開きます：右クリックメニューから *TortoiseHg* → 無視フィルタの編集を選択するか、コマンドラインから `hg tk hgignore` を実行します。一覧からファイルを選択するか直接 *Glob* または 正規表現 を入力して追加 ボタンを押してください。ここでの変更は即座に反映されます。

注釈: `.hgignore` ファイルは作業ディレクトリのルートに含まれており、通常は Mercurial のバージョン管理下

に置かれています。

---

注釈: 作業ディレクトリにたくさんの 不明 ファイルを放置したままにしないというのは、バージョン管理下に置くべきファイルの追加し忘れを未然に防ぐという意味でよい習慣と言えます。そのため `.hgignore` ファイルを極力最新の状態に保つよう努めてください。

---

## 3.4 コミット

プロジェクトディレクトリを右クリックして *Hg コミット ...* を選択するか、コマンドラインから `hg tk commit` を実行して変更点をローカルリポジトリにコミットします。コミットメッセージを記述し、コミットしたいファイルにチェックを入れてから コミット ボタンを押します。コミット後にコミットメッセージの内容やコミットしたファイルの間違いに気付いた場合は 取り消し ボタンで直前のコミットを取り消すことができます。入力したコミットメッセージはメッセージ履歴のドロップダウンリストに残るため、同じメッセージを再度入力し直す必要はありません。

---

注釈: コミットツールを閉じてしまうと直前のコミットを 取り消し ボタンで取り消すことは出来なくなります。

---

## 3.5 リポジトリの共有

これであなたの成果を共有するための準備が整いました。あとはこのリポジトリを他のメンバからも参照できる共用スペースにコピーするだけで OK です。Mercurial ではこのコピー操作を リポジトリのクローン (*cloning your repository*) と呼んでいます。リポジトリを共有ドライブにするには右クリックメニューから *TortoiseHg* → リポジトリのクローン を選択するか、コマンドラインから `hg tk clone` を実行します。

チームでコードを共有するための 中央リポジトリ に、あなたが持っているローカルのリポジトリのような作業ディレクトリは必要ありません。つまり、純粹にリポジトリだけあれば十分と言えます。このような場合、クローン後に作業ディレクトリを *HEAD* に更新しないにチェックを入れることで Mercurial は中央リポジトリのクローンから作業コピーをチェックアウトしません。その結果、プロジェクトのリビジョン履歴のすべてが格納されている `.hg` のみとなります。

他のチームメンバはこの中央リポジトリからクローンすることで作業できます。

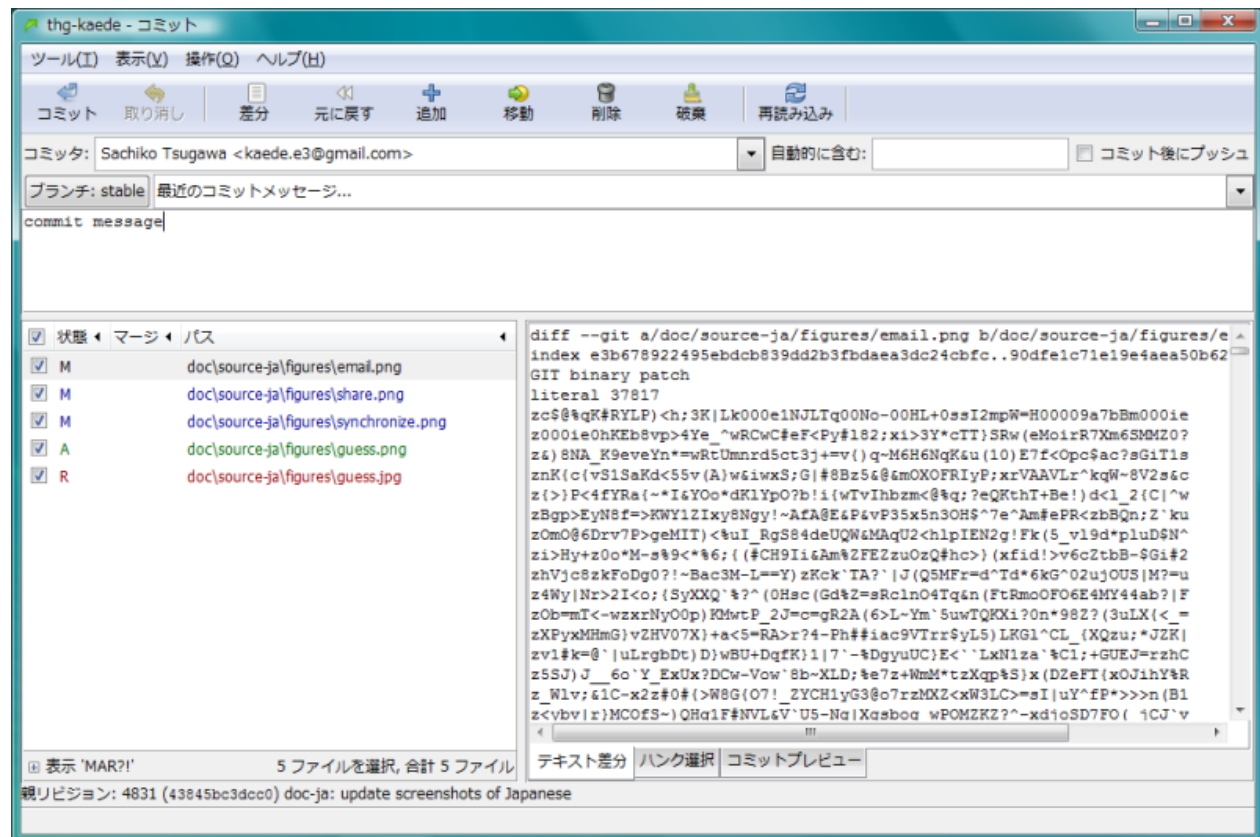


図 4 コミットツール

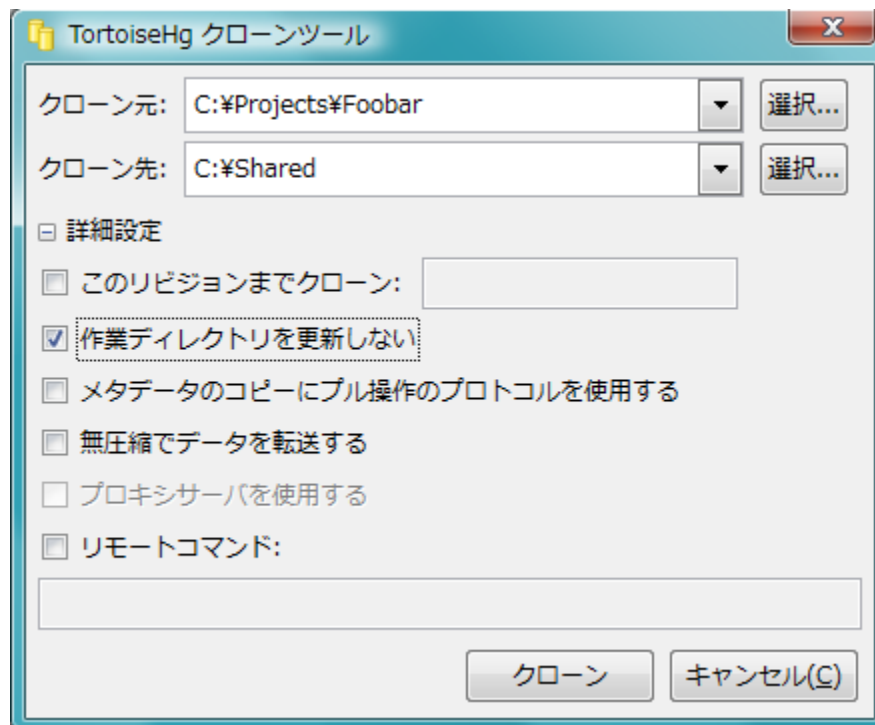


図 5 クローンダイアログ

### 3.6 共有リポジトリからの取得

今あなたがあるチームと共同作業を始めたいとします。彼らは X からリポジトリを取得してくれ と言っていますが、どういう意味なのでしょう？これは、X という場所に置かれているリポジトリをローカルにクローンしてくれ、という意味になります。Mercurial ではこれをクローン (cloning) と呼び、TortoiseHg には専用のダイアログが用意されています。クローンを作成したいディレクトリで右クリックメニューを開いて *TortoiseHg* → リポジトリのクローン を選択するかコマンドラインから `hg tk clone` を実行してください。

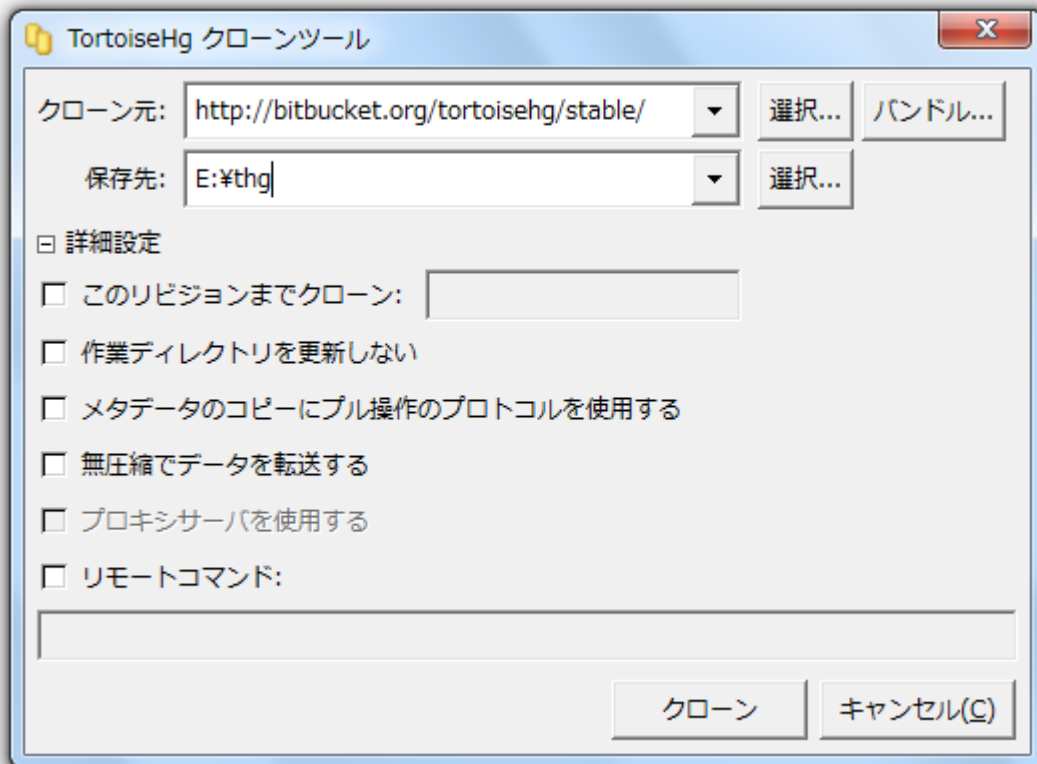


図 6 クローンダイアログ

今回はこれからクローンするリポジトリの作業ディレクトリでプロジェクトの作業をするため、クローン後に作業ディレクトリを *HEAD* に更新しないのチェックは外してください。これで Mercurial は作業ディレクトリを *tip* リビジョンに更新します。

### 3.7 リポジトリを使ったワークフロー

作業ディレクトリでいくつかの変更を加えたとします。エクスプローラでいくつかの変更が保留中であることをアイコンオーバーレイの変化から簡単に確認できます。より詳細な変更点の確認やそれらのコミットはエクスプローラから行うことができます。手っ取り早い方法はコミットツールを使うことです：

コミットツールは変更の差分を表示してくれます。もしくは普段使用している GUI の差分表示ツール (kdiff など)

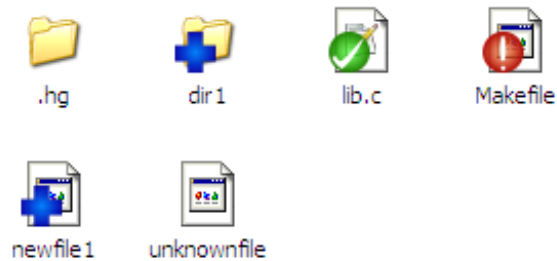


図 7 Windows Vista でのアイコンオーバーレイ表示

を使用することもできます。Mercurial のリポジトリは分散型になっているため、変更点を共有リポジトリと同期 (変更点の共有) するかどうか決める前にローカルリポジトリにコミットできます。

変更を共有リポジトリに公開する準備が整ったら、

1. ローカルのリポジトリに変更点をコミットします (前述)。
2. 共有リポジトリから変更をローカルリポジトリにプル (pull) します: 右クリックメニューから *TortoiseHg* → ログの表示 または コマンドラインから `hg tk log` を実行し、共有リポジトリの場所を同期バーで選択して、ツールバーの プル ボタンを押します。
3. いくつかのチェンジセットがプルされた場合、それらの変更とローカルでの変更をマージして、マージ結果をローカルリポジトリにコミットしてください。チェンジログビューアから (*TortoiseHg* → ログの表示 または `hg tk log`) マージしたいリビジョンの右クリックメニューを開いて マージ を選択します。マージダイアログが表示されるので、マージ ボタンを押してマージを実行し、コミットしてください。
4. マージが正しく行われたかどうかテストなどで検証してください。
5. 変更を共有リポジトリにプッシュ (push) するために、*TortoiseHg* → ログの表示 または `hg tk log` で同期ダイアログを開き、共有リポジトリの場所を選択してからプッシュ ボタンを押します。

文章にすると複雑に見えますが、実際にはコミットしてチェンジログビューアのボタンを押すだけです。

---

注釈: マージ作業に失敗したとしても安全に最初からマージ作業を始めることが可能です。

---

Mercurial は共同作業を簡単に、素早く、効率的にします。詳しくは Mercurial の [wiki](#) をご覧ください。



## 第 4 章

# TortoiseHg ダイアログ

### 4.1 共通の機能

これらの機能は多くの TortoiseHg ダイアログに共通しているため、ここにまとめておきます。

#### 4.1.1 キーボードショートカット

すべての TortoiseHg ダイアログで使用可能なキーボードショートカットがいくつか定義されています。

Ctrl-Q 開いているウィンドウすべてを閉じてアプリケーションを終了します

Ctrl-W 現在のウィンドウを閉じます (開いているウィンドウが 1 つだけの場合は Ctrl-Q と同じ動作)

Ctrl-D 現在選択されているファイルまたはチェンジセットの差分を表示します

Ctrl-Enter コマンドを実行します

Ctrl-. and Ctrl-, ファイル一覧で次および前のファイルを選択します

Ctrl-[ and Ctrl-] テキスト表示でページ単位で前後にスクロールします

F5, Ctrl-R 最新の状態に更新します

Mac OS X ではアップル (Command) キーを Ctrl キーモディファイアの代わりに使用します。ただしコピー & ペーストなど一部の機能については GTK+ 標準のキーボードショートカットが使用されるため、Ctrl キーを使わなければならない場合もあります。

#### 4.1.2 差分表示

TortoiseHg 1.0 では GUI 差分表示 (外部ツール) のベースが見直されました。新しいシステムはよく使われている差分ツール (インストーラ同梱の KDiff3 も含む) の検出とその選択に `mergetools.rc` を使用します。

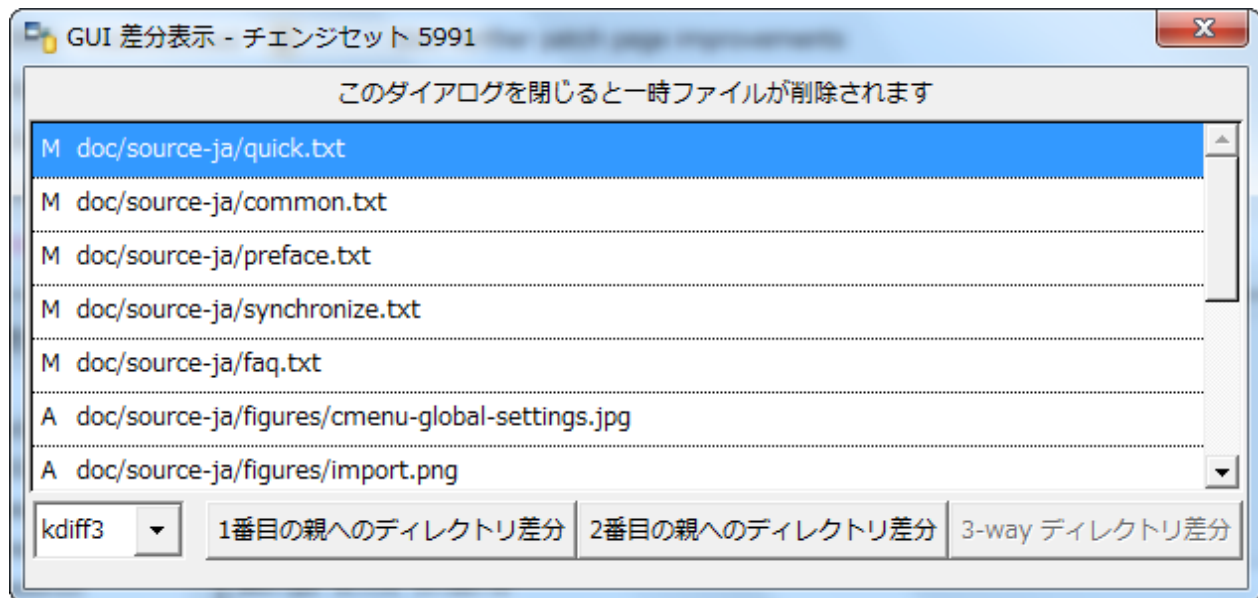


図 1 差分表示ウィンドウ

ユーザがマージツール (*TortoiseHg* → 3-way マージツール) を設定すると、ツール選択の処理を飛ばして、マージツールが GUI 差分表示にも使用されます。もちろん *TortoiseHg* → GUI 差分表示ツール を設定することで、これまで通りマージツールと差分表示ツールを別々に設定することもできます。

マージツール設定ファイルには各ツールごとに最適なコマンドライン設定を持っているため、面倒な追加設定などは不要です。使用したいツールを選択するだけで大丈夫です。

既存の *extdiff* 設定が見つかった場合は GUI 差分システムはそれを使います。しかし *extdiff* は 3-way 差分のラベル引数をサポートしていないため、より便利に使うために *extdiff* の設定を削除または無効にすることをお勧めします。

あえて *TortoiseHg* の GUI 差分ウィンドウを使おうとしない限り、GUI 差分システムは選択されている差分ツールを直接起動します。以下が想定している状況です:

- 1) 比較されるファイルが複数のツールを必要とする
- 2) 選択されたツールが別のバックグラウンドプロセスを起動する
- 3) 選択されたツールがディレクトリ差分をサポートしていない
- 4) 選択されたツールが 3-way 比較をサポートしていない
- 5) ファイルの変更が名前変更またはコピーを含む

GUI 差分ウィンドウが使用された場合、ウィンドウが閉じられたときに一時ファイルが削除されます。従って差分表示ツールを終了するまで閉じずに残しておく必要があります。直接差分表示ツールが起動された場合、そのツールが終了したときに一時ファイルが削除されます。

作業コピーのファイルの比較に差分表示ツールが起動された場合、ツールはその作業コピーのファイルを元に差分

を取るため、そのファイルをツール上で変更することができます。複数のファイルを比較する場合、GUI 差分システムは作業コピーのスナップショットを別の場所に作成し、そのファイルのサイズと日付を記録しておきます。差分表示ツールが閉じられたとき、さきほどのファイルサイズと日付を元に比較し、変更が加えられていた場合はそのファイルをオリジナルの作業コピーに書き戻します。このようにしてディレクトリ差分であっても差分表示ツール上からファイルの書き換えが可能になっています。

GUI 差分ウィンドウが作業コピーの比較に使用された場合、常に 1 つのファイルに対して操作されるため、作業コピーのファイルを元に差分を抽出します。

---

注釈: *TortoiseHg* → 差分ウィンドウをスキップのオプションは削除されました。

---

## ツールの追加

使っている GUI 差分表示ツールを TortoiseHg がサポートしていないときはホームディレクトリにある `Mercurial.ini` ファイルにツール設定を追加することができます。詳しい設定方法については Mercurial の [ドキュメント](#) を参照してください。それが終わったら TortoiseHg が使用する設定項目を追加してください：

```
diffargs: 2つのファイル比較に使用される引数です
diff3args: 3つのファイル比較に使用される引数です
dirdiff: 2つのディレクトリ比較をサポートするツールの場合に使用する引数です
dir3diff: 3つのディレクトリ比較をサポートするツールの場合に使用する引数です
```

コマンドライン引数には以下の変数を使用することができます：

```
$parent1: 1つ目の親リビジョンのファイルまたはディレクトリ
$parent2: 2つ目の親リビジョンのファイルまたはディレクトリ
$child: 比較元のリビジョンのファイルまたはディレクトリ
$parent: $parent1 の別名

$label1: 1つ目の親リビジョンのシンボル名
$label2: 2つ目の親リビジョンのシンボル名
$clabel: 比較元のリビジョンのシンボル名
```

もちろん `$parent2` と `$ancestor` は 3-way 差分を使用するときのみ意味を持ちます。使用している差分表示ツールが共通の祖先リビジョンを扱うことができないのであれば、`diff3args` から削除しても構いません。

---

注釈: Windows では *executable* 引数において環境変数を `${ProgramFiles}` という書き方で使用することができます。

---

未設定の場合、`diffargs` のデフォルト値は `'$parent $child'`、`diff3args` は `""` (空文字) になるため、デフォルトでは 3-way 比較が無効となります。

新しく設定を作成したり、既存のツールの設定を改良したときは是非私たちの開発メーリングリストに投稿してください。将来のリリースで利用させていただきます。

### Office ファイルの差分

TortoiseHg の Windows インストーラには様々なバイナリファイルを比較・マージするために TortoiseSVN のスクリプトを同梱しています。それらは TortoiseHg のインストールディレクトリにある `mergepatterns.rc` ファイルで設定されているため、使用にあたって特別な追加設定などは不要です。

ファイルの拡張子によって使用する差分表示ツールを変更するために、TortoiseHg は Mercurial の `merge-patterns` にならって `[diff-patterns]` セクションをサポートしています。

### 4.1.3 ツリービュー内検索

ステータス、コミット、シェルフ、チェンジログツールのファイル一覧、チェンジログビューアのグラフ表示パネル、リポジトリ検索ツールのコメントパネルなど多くの TortoiseHg ダイアログでツリービューを利用しています。

TortoiseHg のツリービューの大部分はインクリメンタル検索が可能です。どれか行を 1 つ選択するなどして、ツリービューがフォーカスされている状態で検索キーワードの入力を始めてください。入力した検索キーワードを含む小さなテキストボックスが表示されて、ツリービューは最初にマッチした行にジャンプします。さらに入力続けると検索結果が絞られます (エンターキーを押しても次のマッチ項目にジャンプしないので注意してください。検索ボックスが消えて検索が終了してしまいます)。

- `Ctrl-F` は明示的に検索テキストボックスを開きます
- `Ctrl-G` 次にマッチした行に移動します
- `Shift-Ctrl-G` 前にマッチした行に移動します
- マウスのホイールを回転させると前後のマッチ行に移動します

### 4.1.4 HG コマンドダイアログ

TortoiseHg ダイアログの多くは Mercurial のコマンド実行の状況をリアルタイムに表示するため `hgcmd` を使用しています。

---

注釈: エラーメッセージは目立つように赤い文字として表示されます

---

Mercurial コマンドの実行が終了したとき、ダイアログは 閉じる ボタンにフォーカスを移動させます。そのため `Enter` キーを押すだけでウィンドウを閉じることができます。

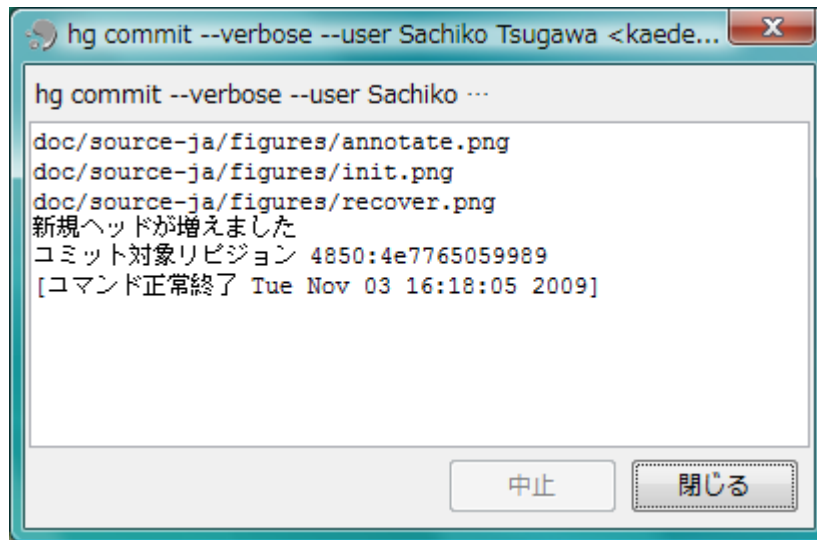


図 2 Mercurial コマンドダイアログ

## 4.2 Windows エクスプローラ統合

### 4.2.1 アイコンオーバーレイ

TortoiseHg は Windows エクスプローラのアイコンオーバーレイを通じてファイル状態の視覚的に表示します。これは TortoiseCVS や TortoiseSVN といった他の Tortoise クライアントのそれと似た機能です。

TortoiseHg は TortoiseOverlays (TortoiseSVN チームによるプロジェクト) を利用することでアイコンオーバーレイを TortoiseSVN (バージョン 1.5 系以降) や他の "Tortoise" プロジェクトと共有します。

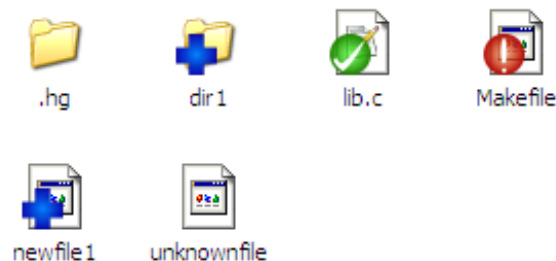


図 3 表示 → アイコン で表示したアイコンオーバーレイ (Windows XP)

右クリックメニューには アイコンの更新 メニューが用意されており、強制的に現在表示しているリポジトリまたはディレクトリのアイコンを最新の状態に更新できます。更新中はタスクバーアイコンが緑色になり、ディレクトリのアイコンオーバーレイは「？」マークが表示されます。

アイコンオーバーレイと右クリックメニューは設定可能です。どのフォルダ (もちろんデスクトップも可) でもいいのでアイコンのない場所で右クリックメニューを開いて、*TortoiseHg* → エクスプローラ設定を選択します。設定ダイアログではメニューを個別にトップに移動できます。

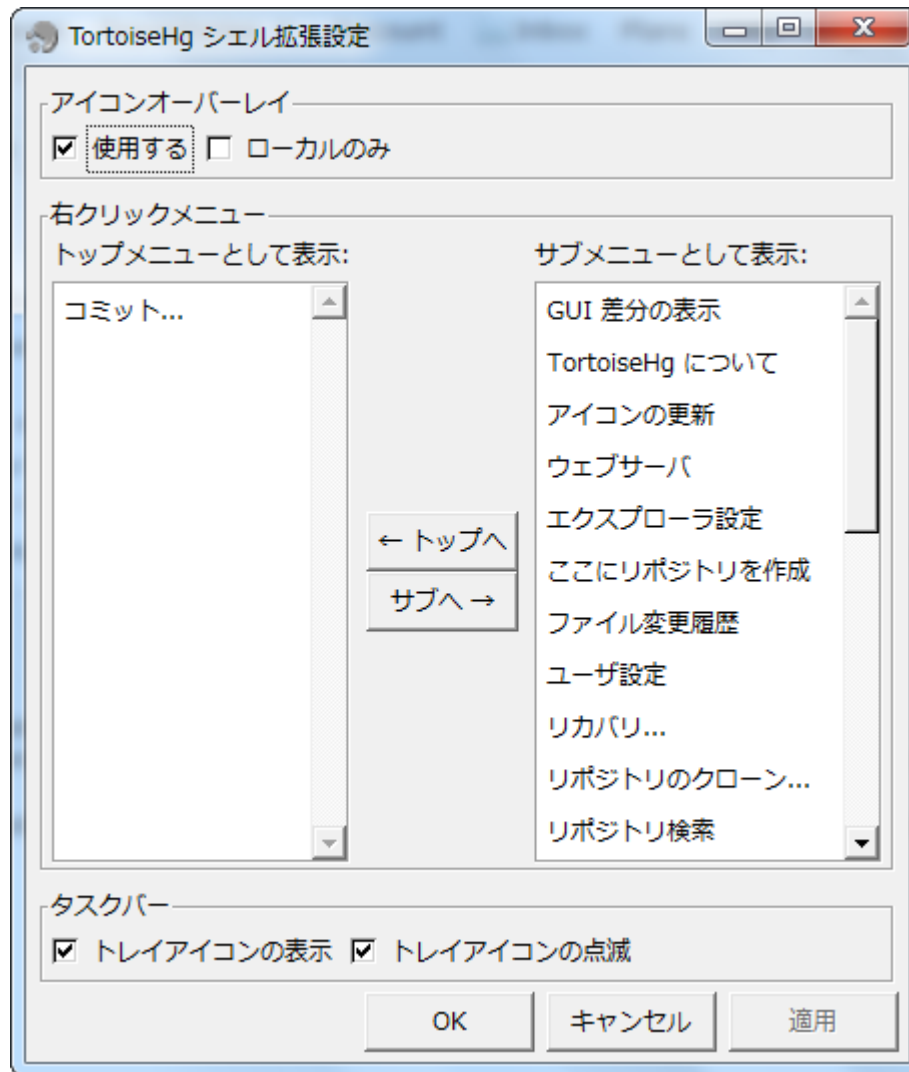


図 4 エクスプローラ設定ダイアログ

また `.hg\thgstatus` ファイルを編集することで特定のリポジトリだけでアイコンオーバーレイを無効にすることもできます。次の 1 行だけが書かれたファイルを作成してください：

```
@@noicons
```

## 4.2.2 右クリックメニュー

TortoiseHg の各種ダイアログを開くためのコマンドは普段使っているエクスプローラから実行します。次のスクリーンショットは Mercurial 管理下にあるディレクトリの右クリックメニューです：

そしてこれはファイルまたはディレクトリを選択した状態で開いた右クリックメニューです：

TortoiseHg は普段よく使う Mercurial コマンドに対応したダイアログを提供します。一方で滅多に使わないコマン

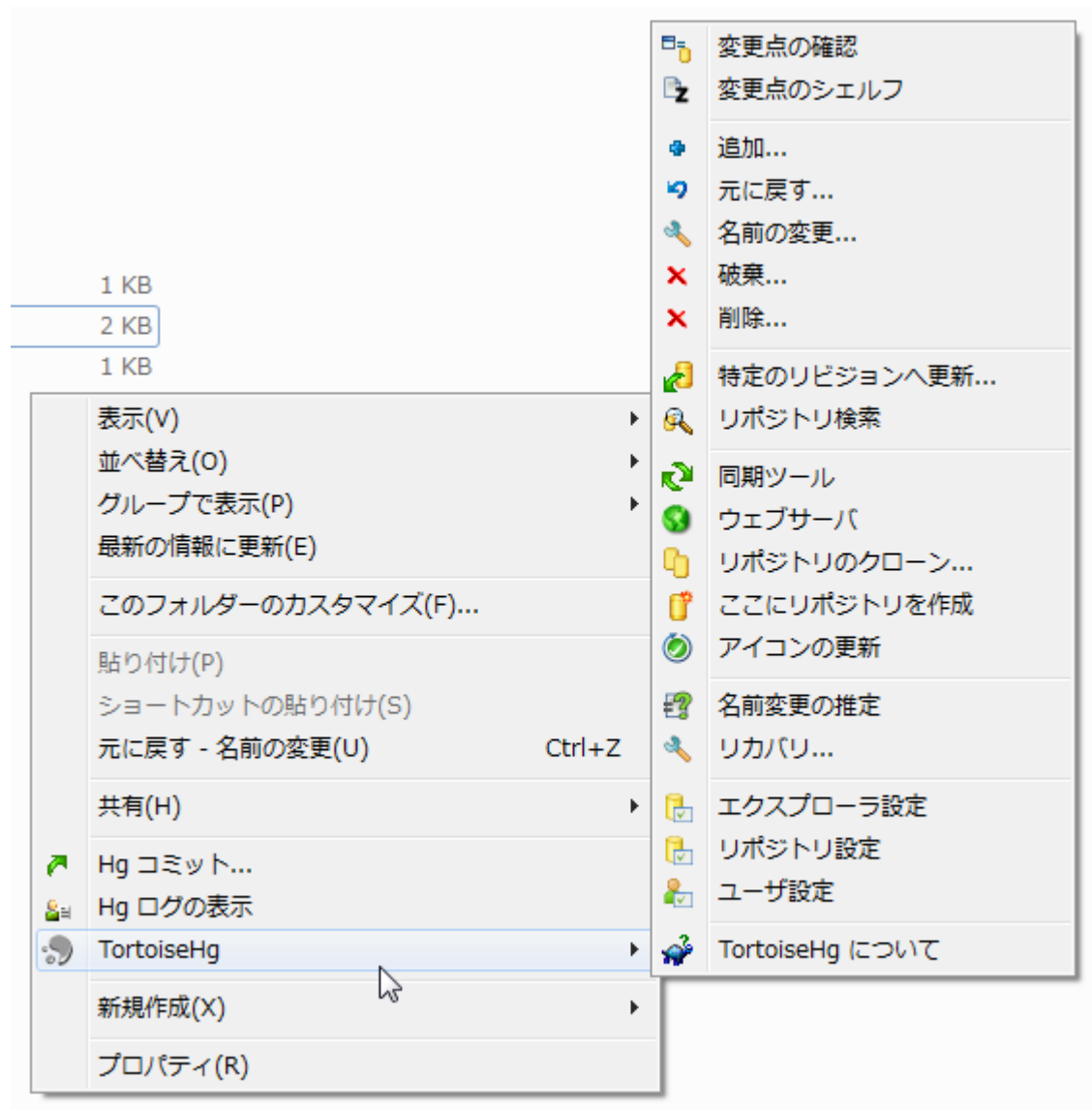


図 5 Mercurial 管理下にあるディレクトリの右クリックメニュー

ドや、比較的最近追加されたコマンドは `cmd.exe` などのコマンドライン (CLI) 経由で実行します。

### 4.2.3 Nautilus

TortoiseHg は `nautilus-python` プラグインを利用して GNOME デスクトップ環境のシェルとの統合も提供しています。頒布パッケージから TortoiseHg をインストールした場合はこの拡張機能も既に設定済みです。そうでない場合は Wiki を参考にしてください。

Nautilus 拡張機能のアイコンオーバーレイや右クリックメニューを設定するための GUI が用意されるまでの間は、`~/.hgrc` ファイルを編集することでメニュー項目をトップに持ってくる設定のみ可能です。以下の設定を追加してください:

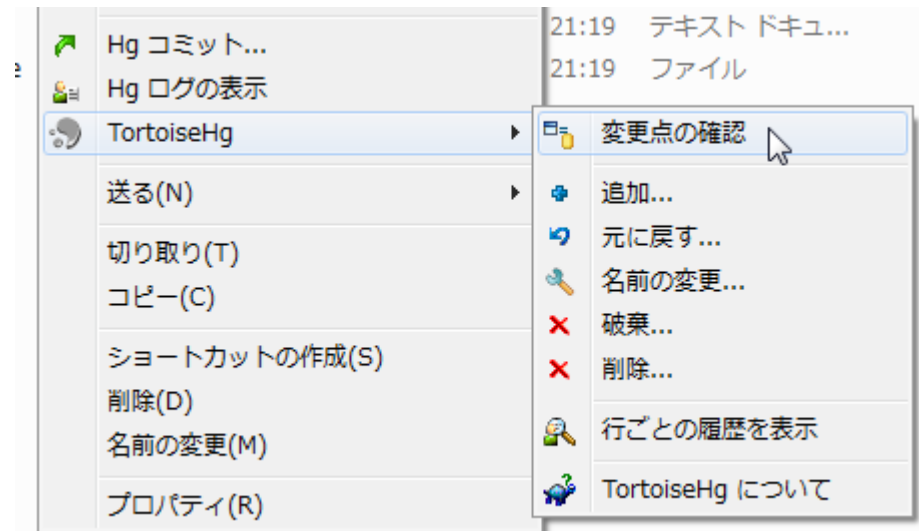


図 6 ファイルまたはディレクトリを選択した状態で開いた右クリックメニュー

```
[tortoisehg]
promoteditems = commit, log, synch
```

### 4.3 新しいリポジトリを作成する

既存のディレクトリ (プロジェクト) に新しくリポジトリを作成するには、リポジトリ初期化ダイアログを開きます。エクスプローラで、プロジェクトのルートディレクトリ内またはディレクトリそのものの右クリックメニューから *TortoiseHg...* → ここにリポジトリを作成 を選択するか、コマンドラインから `hg tk init` を実行してください。

**作成先** リポジトリを作成するディレクトリを指定してください。カレントディレクトリが初期ディレクトリとして指定されるため、作成したいディレクトリから起動すれば特に変更する必要はありません。

**特殊ファイルを追加する (.hgignore など)** これにチェックを入れると TortoiseHg は空の .hgignore ファイルを作業ディレクトリに作成します。

**Mercurial 1.0 互換のリポジトリを作成** これにチェックを入れると TortoiseHg は古い形式の Mercurial リポジトリを作成します。自分が何をしているのか十分に理解していて、どうしてもそうしなければならない場合のみチェックを入れてください。

リポジトリの新規作成は .hg ディレクトリの作成を意味し、このディレクトリで Mercurial はすべてのバージョン情報を保持しています。



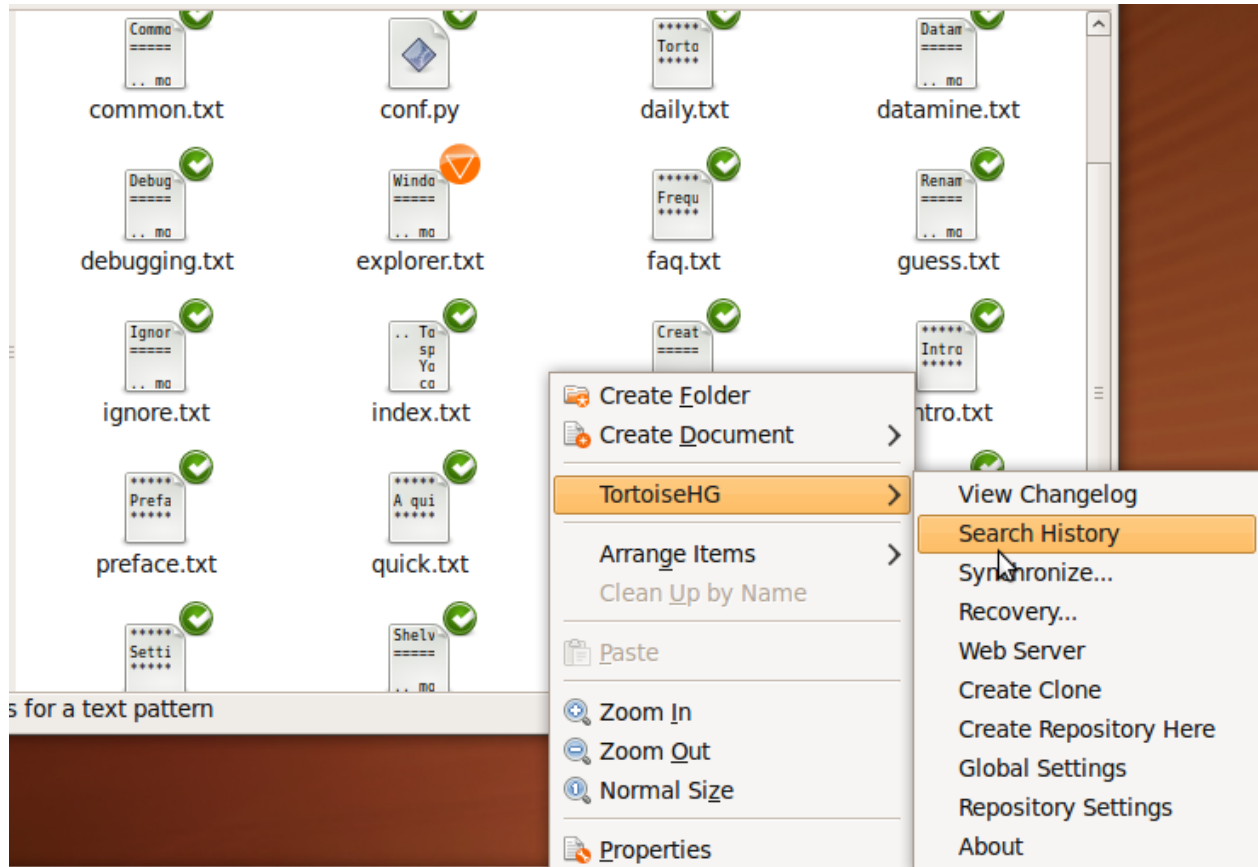


図 7 GNOME/Nautilus スクリーンショット

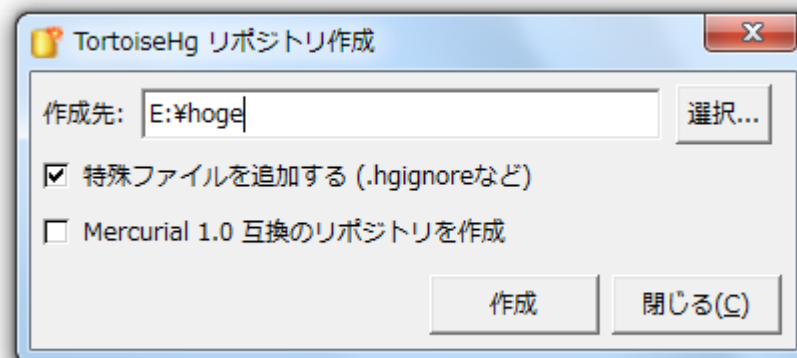


図 8 リポジトリ初期化ダイアログ

警告: .hg ディレクトリ内のファイルを編集するのは非常に危険です。リポジトリの破損を招きます。ただし .hg/hgrc のみ例外的に編集が認められています。

### 4.3.1 コマンドラインからの実行

リポジトリ初期化ツールはコマンドラインから起動することもできます:

```
hgtk init
```

コマンドの構文は:

```
hgtk init [DEST]
```

となっており、[DEST] には作成先のディレクトリを指定します。

## 4.4 クローン

リポジトリのクローンにはクローンダイアログを使います。エクスプローラで右クリックメニューを開いて *TortoiseHg...* → *リポジトリのクローン* を選択するか、コマンドラインから **hgtk clone** を実行してください。

**クローン元** クローン元のリポジトリパス (ディレクトリまたは URL) を指定してください。選択... ボタンを使えばローカルのディレクトリを選択できます。

**クローン先** クローン先のディレクトリを指定してください。このディレクトリにクローン元と同じ名前のディレクトリが作成されます。

詳細設定 には以下の設定項目が含まれています:

**リビジョン指定** ここで指定したリビジョンまでクローンするよう制限できます。指定したリビジョン以降のタグであればインポートされます。

**クローン後に作業ディレクトリを HEAD に更新しない** チェックを入れた場合、クローンが完了しても作業ディレクトリが空のままになります。これは共有リポジトリやバックアップなど、プルまたはプッシュしきないようなりポジトリのクローンに便利です。

**メタデータのコピーに Pull を使用する** クローン元とクローン先が同一ファイルシステムの時、Mercurial はハードリンクの使用を試みますが、AFS など一部の不完全なハードリンク実装を持つファイルシステムではエラー報告が出ません。このオプションを使用すればハードリンクの使用しないように明示的に指定できます。

**無圧縮で転送する** 無圧縮のデータ転送を使用します。有線 LAN ネットワークではその方が速い場合があります。

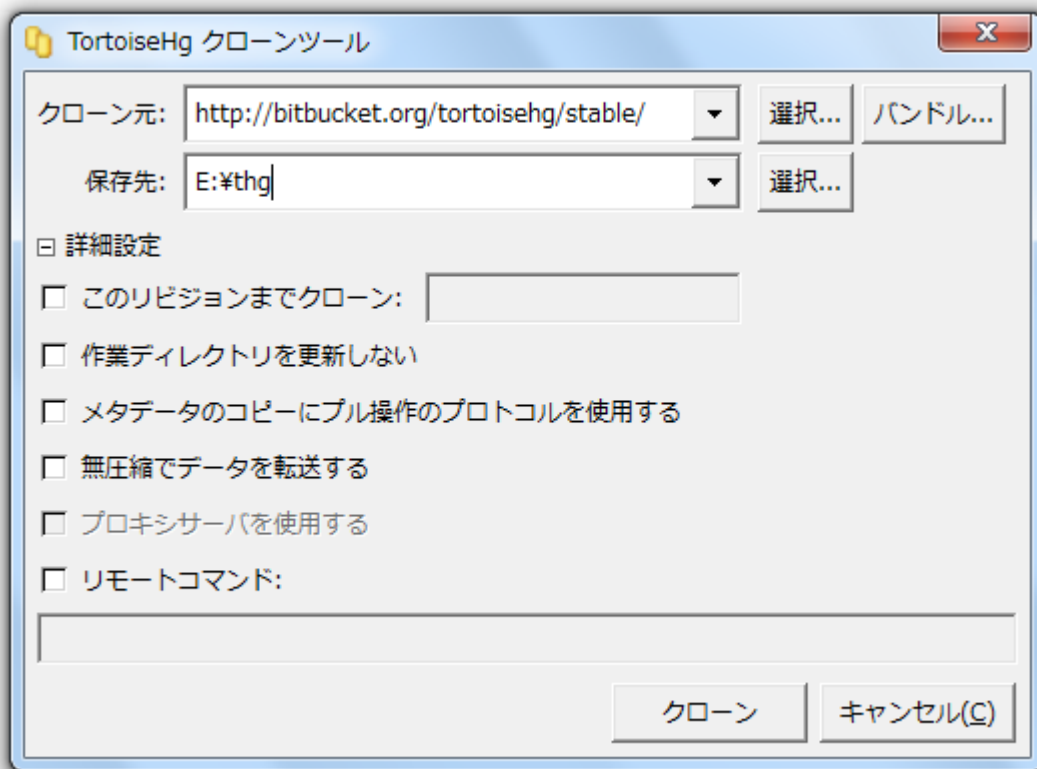


図9 クローンダイアログ

プロキシサーバを使用する *TortoiseHg...* → ユーザ設定 → プロキシ で設定したプロキシサーバを使用します。このオプションはプロキシサーバが設定されているときのみ有効になります。

リモートコマンド クローン元のリポジトリで実行する Mercurial コマンドを指定します。

#### 4.4.1 コマンドラインからの実行

クローンツールはコマンドラインから起動することもできます：

```
hg tk clone
```

コマンドの構文は：

```
hg tk clone [SOURCE] [DEST]
```

[SOURCE] と [DEST] は、それぞれクローン元リポジトリのパスとクローン先のディレクトリです。

## 4.5 コミット

警告: win32text 拡張機能はハンク選択で問題を引き起こす可能性があります。この問題は Mercurial 1.3 と TortoiseHg 0.8 において解決されましたが、適切な設定が必要です。詳しくは [issue #82](#) をご覧ください。

コミットツールは TortoiseHg に 2 つある主要アプリケーションのうちの 1 つです。リポジトリエクスプローラ (別名: チェンジログビューア) と共にこれら 2 つのツールは TortoiseHg が実装しているほぼすべての機能にアクセスできます。

コミットツールは変更点をコミットするだけではなく、作業ディレクトリの状態の確認や、その他にも様々な作業 (新規ファイルの追加、名前変更の記録、リポジトリの無視フィルタの編集、など) もコミットツール上で行なえます。

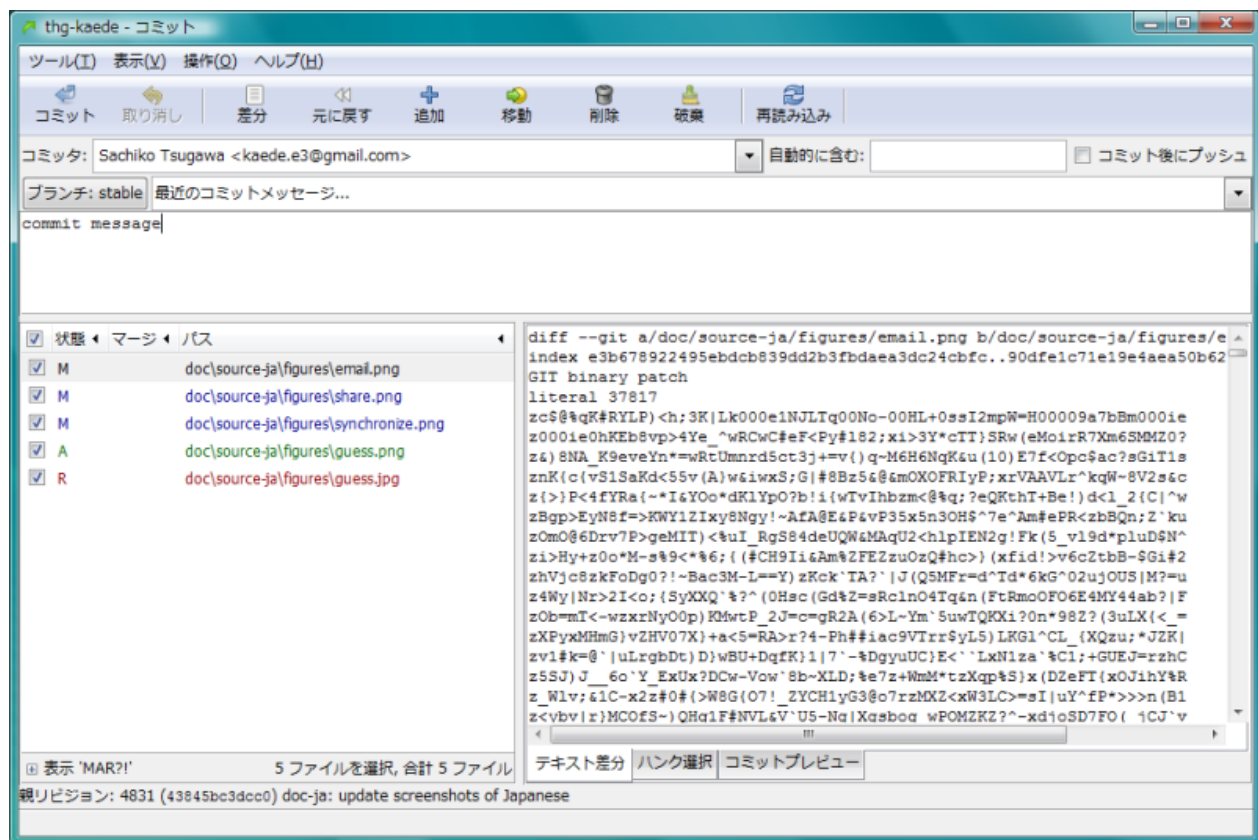


図 10 コミットダイアログ

### 4.5.1 機能

コミットツールの一番上にはバージョン 0.9 から導入されたメニューバーがあります。

**ツール** TortoiseHg の各種ツールを別プロセスで起動します。

**表示** オプション機能の表示切り替えと、作業ディレクトリの表示内容を再読み込みします。

**操作** ツールバーのボタンと同じ操作を実行できます。

**ヘルプ** システム既定のウェブブラウザでコミットツールに関するヘルプを開きます。また TortoiseHg のバージョン情報を表示します。

その下のツールバーにあるボタンを左から順に見ていきます：

**コミット** チェックされたファイルの選択されたハンクをコミットします。

**取り消し** 直前のコミットを取り消します。入力したコミットメッセージは最近コミットしたメッセージの一覧に追加されるため、コミットミスを修正した後に同じメッセージでコミットできます。

**差分** チェックされているファイルを GUI 差分ツールで表示します。

**元に戻す** チェックされているファイルを変更を加える前の状態に戻します。その変更がマージの場合はどちらの親リビジョンに戻すか選択できます。

**追加** チェックされている不明ファイル '?' または無視ファイル 'I' を追加します。

**移動** チェックされているファイルを指定されたディレクトリに移動します。

**削除** チェックされているファイルを削除します。

**破棄** チェックされている管理済みファイルを破棄します。

**再読み込み** 作業ディレクトリの状態を再読み込みします。再読み込み後もチェックボックスの選択状態は保たれます。

**MQ** MQ パネルの表示を切り替えます。このボタンは MQ 拡張機能がユーザによって有効化されているときのみ表示されます。

ツールバーの下にはいくつかの便利なウィジェットがあります：

**ブランチダイアログ** 作業ディレクトリの現在のブランチ名を表示します。通常はブランチ名の表示のみですが、このボタンを押すとブランチ操作のためのダイアログが表示されます。この機能は Mercurial の名前付き **ブランチ** を理解している場合のみ使用してください。

**最近コミットしたメッセージ** 最近のコミットメッセージ 10 件のドロップダウンリストです。このリストは最初に開いた時にコミットメッセージが読み込まれます。

**QNew** MQ 拡張機能を使用している場合、新しいパッチ名を入力するためのテキストボックスが表示されます。パッチ名を入力するとコミットツールは 'QNew' モードに切り替わります。

ファイル一覧には 4 つの列があります：

- 1) 各種操作のためにそのファイルが選択されているかどうかを示すチェックボックス。ツールバーボタンによる操作はチェックされているファイルにのみ行なわれます。部分選択状態のファイルについては特殊な

チェック状態となります。この列のヘッダを使えばすべてのファイルをチェックしたりチェックを外したりできます。

- 2) 状態 列には Mercurial の status コマンドで使用されているファイル状態を示す 'MARD?IC' のうちの 1 つが表示されます。'S' は未コミットの変更を含むサブリポジトリを意味します。
- 3) マージ 列はそのファイルのマージ状態を示しており、Mercurial の resolve コマンドで使用されている 'RU' のうちどれか 1 つが表示されます。
- 4) リポジトリルートを基準としたファイルパスが表示されます。

注釈: コミットツールがファイルパターンまたは特定のファイルを選択した状態で起動された場合、ファイル一覧の下にそれらのフィルタを削除して作業ディレクトリにおける変更すべてを表示するためのボタンが表示されます。

ファイル一覧の下に並んだチェックボックスは{M:変更, A:追加, R:削除, !:削除, ?:不明, C:クリーン, I:無視} のファイル状態ごとにファイル一覧に表示するかどうかを設定できます。このチェックボックスはコミットツールからファイルまたはディレクトリを指定された場合には無効になります。

R:削除 は管理済みファイルが Mercurial の管理から外されたことを示しています。一方で !:削除 は管理済みファイルが見当たらず、Mercurial がそのファイルを追跡できなくなったことを意味しています。例えば管理済みファイルの名前をエクスプローラで変更したとき、コミットツールのファイル一覧には元のファイル名が !:削除 として表示され、新しいファイル名が ?:不明 として表示されます。新しいファイル名の右クリックメニューから名前変更推定ダイアログを起動すればファイル内容の比較によって名前変更を検出し、古いファイルを R:削除 とマークし、新しいファイルを A:追加 とマークすることができます。

?:不明 は Mercurial によって管理されておらず、かつ、設定されている無視ファイルフィルタにマッチしないファイルに表示されます。リビジョン管理に追加する必要があるファイルなどはデフォルトで不明ファイルとして表示されます。無視ファイルフィルタを最新に保つことは良い習慣です。不明ファイルの右クリックメニューには無視ファイルフィルタを編集するツールを開くコマンドがあります。

C:クリーン は変更が加えられていない管理済みのファイルです。一方で I:無視 は設定されている無視ファイルフィルタに該当する管理外のファイルです。どちらのファイルもエクスプローラでの選択に含めるか、コマンドラインからそのファイル名を指定して起動しない限りデフォルトでは表示されません。

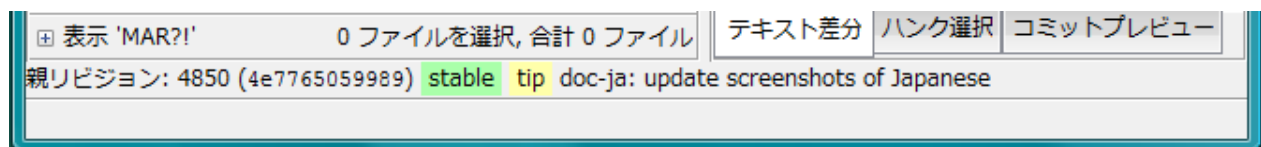


図 11 コミットツールの親リビジョンバー

ファイル一覧と差分表示パネルの下には 親リビジョンバー があります。ここには現在の作業ディレクトリの親にあたるチェンジセットが表示され、コミットによって新しくヘッドが作成されるかどうか見ることができます。こ

のバーは 表示 メニューから非表示にできます。

ファイル一覧の右側は差分表示パネルになります。バージョン 0.9 から差分表示パネルがタブ表示になりました。

**テキスト差分** 選択しているファイルの差分をテキストで表示します。

**ハンク選択** バージョン 0.7 から 0.8 まではこのタブは単純に内容を表示するだけでしたが、0.8 からはハンク選択機能が実装され、変更点の一部を選択してコミットできるようになりました。このタブには選択されているファイルの差分ハンクが表示されますが、バイナリファイルは表示されません。内容の確認は テキスト差分タブで行ってください。

**コミットプレビュー** このタブにはすべての選択されたハンクが表示されます。実質的にこのプレビューにはコミットボタンを押したときに何がコミットされるかが表示されます。

**パッチ内容** コミットツールがパッチ更新モードのときにのみ表示されます。ここには現在のパッチ内容が表示されます。



図 12 コミットツールの詳細設定バー

表示 メニュー で 詳細設定 を有効にした場合、ツールバーとコミットログ履歴の間に詳細設定バーが挿入されます。このバーには:

**コミッタ** コミットに使用するユーザ名です。この値は通常 Mercurial.ini ファイルから読み込まれますが、コマンドラインやパッチファイルから読み込んで指定することもできます。直接入力することで異なるユーザを指定することも可能です。

**自動的に含む** ファイルにチェックを入れたかどうかに関わらず、常にコミットに含めたいファイルをカンマ区切りで指定するためのテキストボックスです。この機能はリポジトリの pre-commit フックによってコミット時に自動更新されるファイル (例えば changelog ファイル) などを常にコミットに含めたい、というときに便利です。

**コミット後にプッシュ** コミットが正常に完了した後、デフォルトの URL にプッシュするかどうかを設定するチェックボックスです。

## 4.5.2 変更点の選択

さて、「チェックされたファイルの選択されたハンクをコミットする」と言ったとき、それは何を意味しているのでしょうか？ TortoiseHg のコミットツールには変更点の一部のみをコミットする機能が標準で備わっているため、単純にそれを使うという意味になります。つまり、管理下に置かれたファイルに対するすべての変更箇所は、どの部分をコミットに含めて、どの部分をコミットせずに (作業ディレクトリに) 残すのか個別に選択することができます。darc バージョン管理システム、または Mercurial の record 拡張機能のユーザであればすぐに分かるはずです。



### どんなとき必要になるのか？

関連性の低い複数の変更が作業ディレクトリにあり、それらを個別のチェンジセットとしてコミットしたいとします。この場合、コミット時にファイル一覧を指定することでうまく対処できるケースもありますが、同じソースコードに変更点が混ざり合っているケースではコミットする変更点を選択する機能が必要不可欠になります。

### どうやって使うのか？

差分表示パネルのハンク選択タブでチェンジハンクをダブルクリックすることでそのハンクの選択状態を切り換えることができます。厳密には選択状態はアクティベート操作によって切り替わるため、ダブルクリックではなくスペースキーを押すことでも可能です。ハンクが選択解除されると差分パネルの色分け表示が無効になり、背景色がグレーになります。それと同時にファイルの差分ヘッダが更新され、現在選択されているハンクの数と行数が表示されます。もう一度ダブルクリックしてハンクを選択することでコミットにその変更を含めることができます。

ファイル内の変更が部分的に選択されているとき、ファイル一覧に表示されるそのファイルのチェックボックス表示がラジオボタンのようになります。これにより、次のコミットで各ファイルの全体がコミットされるのか、部分的にコミットされるのか一目で分かります。

### コミットするとどうなるのか？

簡単に言ってしまうと、チェックの入ったファイル内の選択されたハンクがリポジトリにコミットされ、選択されなかった変更点は作業ディレクトリに残り、次のコミットに持ち越されます。

実際には少々複雑なことをしています。その裏側で何が起きているのかというと、選択されなかったハンクのバックアップを安全な場所に作成した後、作業ディレクトリの内容が最新リビジョンの状態に戻し、クリーンな状態にします。そして選択された変更点のみが作業ディレクトリに書き戻されて、この状態でコミットされます。最後にバックアップから選択されなかったハンクが作業ディレクトリに復元されます。ハンク選択されなかったファイルはバックアップ、書き戻し、コミット、復元の一連の処理はされません。

このハンク選択機能の内部処理に関する記述は、不運にも処理に失敗したときに役に立つかもしれません (Windows ではその可能性がゼロであると言い切れないため)。何らかのプログラム (アンチウイルスソフト、コンパイラなど) が処理の過程で生成された一時ファイルをロックしてしまった場合、書き戻しに失敗したというエラーが発生するかもしれません。しかしこれらのエラーは修復できます。リポジトリにある `.rej` ファイルを削除してからもう一度コミットを試みてください。

## 4.5.3 キーボードショートカット

`Ctrl-Enter` コミットを実行します

`Ctrl-C` 差分表示パネルでこのショートカットを押すと、ハイライト (選択ではない) されているハンクの内容がクリップボードにコピーされます。作業ディレクトリの変更から任意の変更点をテキストとしてどこかにペーストするのに使えます。



Alt-Q カーソルのある段落 (空行で区切られた行) を空白で連結して 1 行にします。このショートカットを使用するにはフォーマットポリシーの設定が必要です。

クリップボードにコピーされるハンクは気が利いていて、差分のヘッダ情報が付加されます。つまり、クリップボードの内容はいつでも正しいパッチとして使えるというわけです。

#### 4.5.4 ファイルの右クリックメニュー

ファイル一覧のファイルの上で右クリックすると、選択されたファイルに対するコマンドを含むメニューが表示されます。

不明ファイル? の右クリックメニューには名前変更を検出するコマンド (不明ファイルが管理下にあったファイルのコピーまたは名前変更されたものだと考えられる場合) や、リポジトリの無視フィルタを編集するコマンド (その不明ファイルを今後モリビジョン管理することではなく、Mercurial にそれを無視して欲しい場合) が提供されます。

#### 4.5.5 マージ作業

マージ状態のリポジトリを開いたとき (正確には現在の作業ディレクトリが 2 つの親リビジョンを持っているとき) コミットツールは特別なモードに切り替わります。ファイル一覧にはチェックボックスが表示されず、差分表示パネルのハンク選択 タブは非表示になります。マージ後に作業ディレクトリ全体をコミットしなければならないため、基本的にコミットマニフェストは不変です。

このモードではマージ状態を示す *ms* 列が特に役に立ちます。この *ms* 列で *R* というマークが付いたファイルは Mercurial またはユーザによって 2 つの親リビジョンから正しくマージ (衝突の解決) されたファイルです。 *U* マーク付いたファイルは未解決のファイルです。そのファイルのマージ作業を再開するには右クリックメニューの 衝突の解決 メニューを選択するか、編集 メニューでファイルを開いて手動で衝突を解決します。衝突の解決メニューからマージ作業を開始すると、その都度マージツールを変更できるため、ファイルごとに適切なマージツールを使用することができます。衝突を解決したら 解決済みとしてマークメニューを選択してファイルのマージ状態を手動で *R* に変更します。

Mercurial は 1 つでもマージ状態が *U* となっているファイルがあるとコミットを中止します。

ローカル はマージを始めたときにチェックアウトされていたリビジョンで、他のリビジョン はマージ対象のリビジョンです。

失敗したマージ作業を取り消すには 2 つ目の親リビジョンによる変更点を作業リポジトリから削除するために Mercurial に知らせなければなりません。これは通常 1 つ目の親リビジョンへのクリーンな (ローカルでの変更を上書き) 更新を意味します。マージツールにはまさにその操作のための 取り消しボタンがあります。これはリカバリツールの クリーン ボタンを押しても同じ結果になります。

作業ディレクトリをどちらかの親リビジョンに戻せば再びマージ作業を始めることができます。

### 4.5.6 コミットメッセージパネル

コミットメッセージパネルは以下の特別な右クリックメニューがあります:

ファイル名を貼り付け: チェックの入っているファイル名ををコミットメッセージのカーソル位置に貼り付けます。

フォーマットポリシーの適用: 設定されたメッセージの折り返しポリシーを現在のメッセージに適用します。

フォーマットポリシーの設定: 設定ダイアログの コミット ページを開きます。

プロジェクトにコミットメッセージのルールがある場合はそれらを設定ダイアログで変更できます。コミットツールは設定されたポリシーをコミット時にチェックし、コミット前でもポリシーに沿ったメッセージになっているか確認できます。設定ダイアログの コミット ページにはコミットメッセージポリシーの設定項目が 2 つあります:

要約行最大文字数: コミットログ最初の行 (要約行) の最大文字数。これが設定された場合、要約行が長すぎる、もしくは要約行とそれ以降のログが空行で区切られていないときに警告が表示されます。デフォルト: 0 (制限しません)。

コミットログ折り返し文字数: コミットログの折り返し文字数。これが設定された場合、コミットログ中の 1 行でも設定文字数を上回っていると警告が表示され、その文字数で強制的に折り返すためのポップアップメニューが表示されます。デフォルト: 0 (制限しません)。

### 4.5.7 MQ パッチ

多くの Mercurial 上級者はパッチの管理に MQ 拡張機能を使っていることでしょう。コミットツールはパッチが適用されているとき パッチ更新 モードに切り替わります。ダイアログのタイトルは "パッチ更新 パッチ名" となり、コミットメッセージパネルにはそのパッチのコメントが表示されます。

差分表示パネルの パッチ内容 タブには現在のパッチ内容がすべて表示されます。テキスト差分 と ハンク選択 タブにはパッチ内容に加えて作業ディレクトリでの変更点も合わせて表示されます。ハンク選択機能を使えば現在のパッチから変更点の一部を別のパッチに移動 (またその逆も) することができます。

これは実質的に `qdiff` コマンドを実行したときの内容が表示されます。コマンドラインから実行する `hg diff` のように作業ディレクトリにおける変更点のみを表示する方法はありませんが、新しいパッチ名を入力するテキストボックスに何か 1 文字でも入力すれば パッチ作成 モード (後述) に切り替わるので、作業ディレクトリにおける変更のみを見ることは一応できます。

コミット ボタンを押すことで (パッチ更新モードではラベルがパッチ更新 に変化します) は現在のパッチ内容を選択されている変更点にコミットメッセージと共に更新 (qrefresh) します。更新後も除外した変更点は作業ディレクトリに残り、ファイル一覧や差分表示パネルに表示される内容は何も変化しないため、最初は少し戸惑うかもしれません。

### 4.5.8 QNew モード

コミットツールから MQ のパッチを作成できます。MQ 拡張機能が有効になっている場合、ブランチボタンとコミットメッセージ履歴のドロップダウンリストの間にテキストボックスが表示されます。このテキストボックスにパッチ名を入力し始めるとコミットツールは コミットまたは パッチ更新 モードから パッチ作成 モードに切り替わり、差分表示パネルには現在の作業ディレクトリにおける変更点が表示されます。正常にモードが切り替わるとコミット ボタンが *QNew* ボタンに変化するので一目瞭然です。

*QNew* ボタンが押されると選択されている変更点のハンクが入力された名前の新しいパッチに書き込まれ、ダイアログが再読み込みされます。これにより、少なくとも 1 つ適用済みのパッチが存在するため、ダイアログは パッチ更新 モードに切り替わります。

パッチのコミットメッセージは パッチ作成 モードのときに入力することもできますし、作成後の パッチ更新 モードで更新することもできます。

パッチ作成時に変更点の一部を除外した場合、その変更点も差分表示パネルの新しいパッチの中に表示されますが、これは前述の通り パッチ更新 モードが現在のパッチ内容と作業ディレクトリの変更点をまとめて表示するためです。ここで再びテキストボックスにパッチ名を入力を始めるとコミットツールは再度 パッチ作成 モードに移行し、差分表示パネルには作業ディレクトリに残った (先ほどのパッチ作成で除外した) 変更点が表示されます。

### 4.5.9 オプション設定

コミット → ユーザ名 コミットに関連付けるユーザ名を設定します (*TortoiseHg* クイックスタート)

コミット → 要約文字数 要約行の文字数を制限するポリシーを設定します

コミット → メッセージ文字数 コミットメッセージ各行の文字数を制限するポリシーを設定します

これに加えて上級者向けの 3 つのオプションがあります。

コミット → コミット後にプッシュ: True にすることで コミット後にプッシュ チェックボックス の初期値として使用されます。

コミット → 自動コミットファイル: コミットに自動的に含めたいファイルをカンマ区切りで指定します。通常はユーザ設定ではなくリポジトリ設定で使います。

コミット → 自動除外ファイル: ステータス、コミット、シェルフツールにおいて自動的にチェックを外したいファイルをカンマ区切りで指定します。

*TortoiseHg* → 差分を画面下部に表示 差分表示パネルをファイル一覧の右側から下に切り替えます

*TortoiseHg* → 最大差分サイズ 表示する差分の上限サイズを設定します

外部ツールの設定はバージョン 0.9 で廃止されました。

## 4.5.10 コマンドラインからの実行

コミットツールはコマンドラインから起動することもできます :

```
hgtk commit [OPTIONS] [FILE]...

aliases: ci

commit tool

options:

-u --user record user as committer
-d --date record datecode as commit date

use "hgtk -v help commit" to show global options
```

日付の書式は :

```
hg help dates
```

から見るすることができます。

## 4.6 シェルフ

**警告:** win32text 拡張機能はハंक選択で問題を引き起こす可能性があります。この問題は Mercurial 1.3 と TortoiseHg 0.8 において解決されましたが、適切な設定が必要です。詳しくは [issue #82](#) をご覧ください。

このダイアログは作業ディレクトリにおける変更点のうち、必要な部分のみを選択して シェルフ (一時的に別の場所に保管しておく) するために使います。シェルフした変更点は特別なパッチとしてリポジトリに格納され、いつでもその変更点を作業ディレクトリに戻すことができます。

ツールバーボタンを左から順に見ていきます :

シェルフ チェックされたファイルの選択されたハंकをシェルフします。

シェルフから戻す シェルフされた変更点を作業ディレクトリに戻します。

差分 チェックの入っているファイルの GUI 差分を表示します。

元に戻す チェックされているファイルを変更を加える前の状態に戻します。その変更がマージの場合はどちらの親リビジョンに戻すか選択できます。

追加 チェックされている不明ファイル '?' または無視ファイル 'I' を追加します。

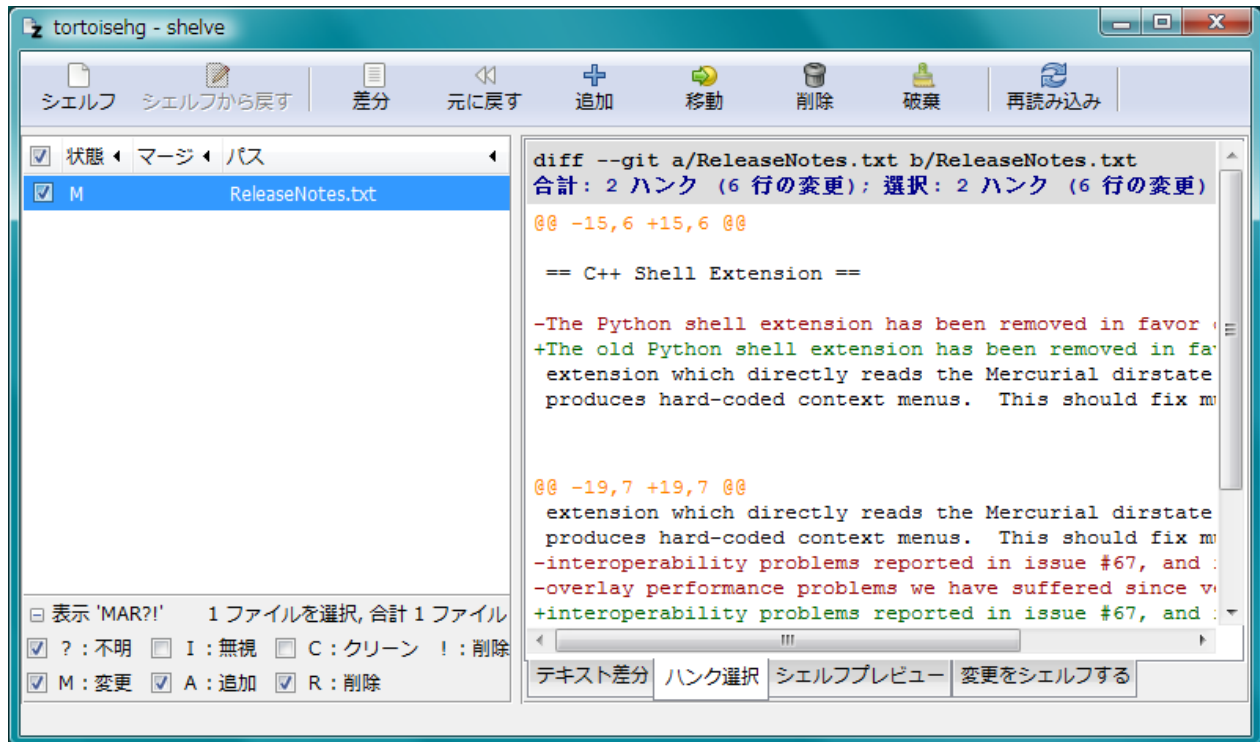


図 13 シェルフダイアログ

**移動** チェックされているファイルを指定されたディレクトリに移動します。

**削除** チェックされているファイルを削除します。

**破棄** チェックされている管理済みファイルを破棄します。

**再読み込み** 作業ディレクトリの状態を再読み込みします。再読み込み後もチェックボックスの選択状態は保たれます。

ファイル一覧には 4 つの列があります：

- 1) 各種操作のためにそのファイルが選択されているかどうかを示すチェックボックス。ツールバーボタンによる操作はチェックされているファイルにのみ行なわれます。部分選択状態のファイルについては特殊なチェック状態となります。この列のヘッダを使えばすべてのファイルをチェックしたりチェックを外したりできます。
- 2) 状態 列には Mercurial の status コマンドで使用されているファイル状態を示す 'MARD?IC' のうちの 1 つが表示されます。
- 3) マージ 列はそのファイルのマージ状態を示しており、Mercurial の resolve コマンドで使用されている 'RU' のうちどれか 1 つが表示されます。
- 4) リポジトリルートを基準としたファイルパスが表示されます。

ファイル一覧の下に並んだチェックボックスは {M:変更, A:追加, R:削除, !:削除, ?:不明, C:クリーン, I:無視} のフ

イル状態ごとにファイル一覧に表示するかどうかを設定できます。このチェックボックスはコミットツールからファイルまたはディレクトリを指定された場合には無効になります。

#### 4.6.1 タブ

シェルフツールの差分表示パネルには 4 つのタブがあります：

- 1) テキスト差分 - 選択されているファイルの差分が表示されます
- 2) ハンク選択 - 選択されているファイルのハンク選択状態を変更できます
- 3) シェルフプレビュー - すべての選択されている変更点が表示されます。このプレビュー に表示されている内容が作業ディレクトリから取り除かれてシェルフに格納されます。
- 4) シェルフ内容 - シェルフに格納されている内容が表示されます

#### 4.6.2 変更点をシェルフする

このダイアログはコミットツールのように TortoiseHg のハンク選択機能を採用しており、シェルフに移動したいファイルおよび変更点をユーザが選択できます。シェルフボタンを押すと、選択された変更点が作業ディレクトリから削除され、パッチファイルとして保存されます。既にシェルフされた変更点が存在する場合はその変更点と置き換えるか、もしくは新しい変更点をそれにマージするか尋ねられます。シェルフに変更点があるときのみシェルフから戻すボタンが有効になります。

#### 4.6.3 シェルフした変更点を戻す

シェルフから戻すボタンが押されると、シェルフした変更点が作業ディレクトリに再適用されます。

---

注釈: シェルフから戻した後、シェルフツールを再読み込みすると、シェルフから戻された変更点は作業ディレクトリへの変更として表示されます。

---

#### record/commit とはどう違うのか？

シェルフされた変更点はそれが戻されるまで実際に作業ディレクトリから取り除かれます。これはシェルフされている状態でもプロジェクトをビルドしたり、テストを実行したりできることを意味しています。コミットされた変更点が有効かどうか試すことができるので、ビルド時に変更点を選択するより安全と言えます。

変更点をシェルフすることは完成した作業結果からデバッグ用コードを取り除くことで動作に影響がないか確かめるのに便利です。

注意：シェルフされた変更点は作業ディレクトリの内容を基準としたパッチに保管されます。そのためもしシェルフ後の変更と衝突を起こした場合、そのパッチがキレイに再適用できるという保証はありません。

### MQ とはどう違うのか？

シェルフはチェンジセットに変換されない単一の無名 MQ パッチと見なすことができます。

しかしシェルフツールは MQ でパッチを管理していても便利に使えます。シェルフは MQ のパッチから変更点の一部を取得し、それらを別の MQ パッチ (または完全に新しいパッチ) に再適用できます。

例えば：

- 1) 分割したいパッチに変更をプッシュする
- 2) シェルフツールを開く (最上位のパッチがハンク選択可能になる)
- 3) そのパッチに残したいハンクの選択を解除して、シェルフ ボタンを押す
- 4) コマンドラインから `hg qrefresh` するかコミットツールの *QRefresh* ボタンで最上位のパッチを更新する
- 5) `qpush` または `qpop` してシェルフした変更を適用したいパッチに移動する
- 6) シェルフツールを開いて シェルフから戻す ボタンを押す
- 7) 最上位のパッチを更新する (手順 4 に戻る)

追加、削除、名前変更されたファイルをシェルフすることはできませんが、MQ ならそれが可能です。

### attic 拡張機能とはどう違うのか？

attic 拡張機能はシェルフ機能の上位版です。具体的には attic は独立した複数の名前付きシェルフ を保持することができます。

## 4.6.4 キーボードショートカット

*Ctrl-C* 差分表示パネルで現在ハイライト (選択ではない) されているハンクをクリップボードにコピーします。コピーしたハンクをテキストエディタに貼り付ければ、作業ディレクトリにおける変更点から任意のパッチを作り出すのに使えます。

クリップボードにコピーされたハンクは差分ヘッダが付加されるため、クリップボードの内容は常に有効なパッチとして利用できます。



### 4.6.5 オプション設定

- *TortoiseHg* → 差分を画面下部に表示
- *TortoiseHg* → タブ文字の幅
- *TortoiseHg* → 最大差分サイズ

### 4.6.6 コマンドラインからの実行

シェルフツールはコマンドラインから起動することもできます：

```
hgtk shelve

aliases: unshelve

shelve/unshelve tool

use "hgtk -v help shelve" to show global options
```

TortoiseHg のシェルフ機能を Mercurial のコマンドラインから使うには、Mercurial.ini ファイルで以下のように拡張機能を有効化してください：

```
[extensions]
tortoisehg.util.hgshelve=
```

これにより Mercurial に **shelve** および **unshelve** コマンドが追加されます。

## 4.7 チェンジログ

チェンジログビューア (別名:リポジトリエクスプローラ) はリポジトリのリビジョン履歴を可視化し、チェンジセットに対して様々な操作をするのに使用され、リビジョン履歴をグラフ化し、親/子リビジョンの関係を表示します。また各リビジョンにおいてどのファイルが変更されたのか、さらに変更内容についても見ることができます。

チェンジログビューアには各種機能を実行したり、他のツールを起動したりするためのメニューバーがあります。

ツール TortoiseHg の各種ツールを別プロセスで起動します。

表示 オプション機能の表示切り替えと、表示内容の再読み込みします。

リビジョン リポジトリ内の特定のチェンジセットを選択します。

同期 リポジトリ同期機能 (後述) に関する操作をします。



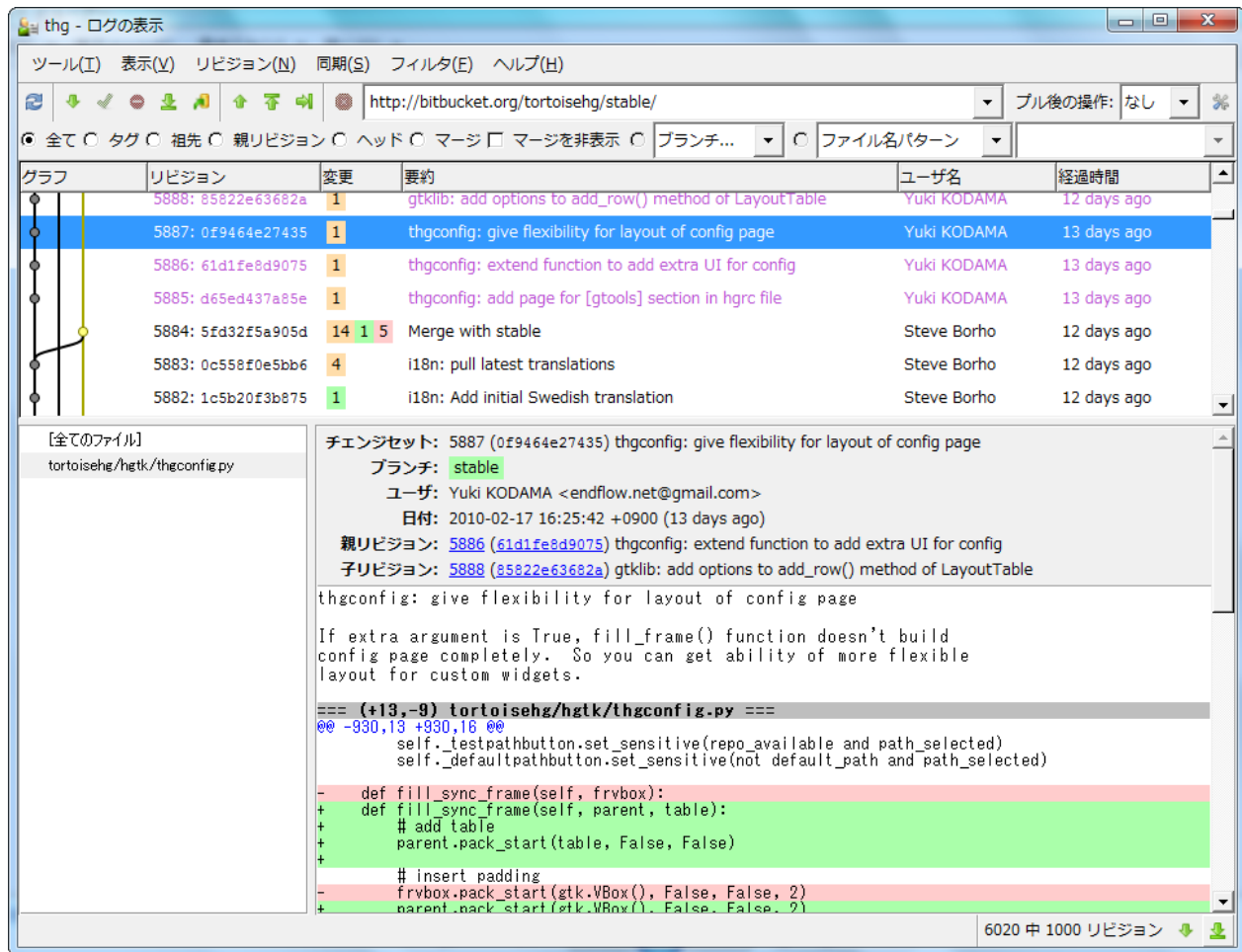


図 14 ツールバーを非表示にしたチェンジログビューア

**ヘルプ** システム既定のウェブブラウザでチェンジログビューアに関するヘルプを開きます。また TortoiseHg のバージョン情報を表示します。

**ツールバーボタン**を左から順に見ていきます：

**再読み込み** リビジョン履歴を再読み込みします (別のウィンドウでコミットした時など)。

**マークをリセット** リビジョングラフの新規・受信・送信リビジョンを表すマークを消して再読み込みします。

**同期ツール** 同期ツールを開きます。他のリポジトリとチェンジセットのやり取りできます。

**MQ** MQ パネルの表示を切り替えます。このボタンは MQ 拡張機能がユーザによって有効化されているときのみ表示されます。

**コミット** コミットダイアログを別プロセスで起動します。

**リポジトリ検索** リポジトリ検索ツールを別プロセスで起動します。

リカバリ リカバリツールを別プロセスで起動します。

ウェブサーバ ウェブサーバツールを別プロセスで起動します。

シェルフ シェルフダイアログを別プロセスで起動します。

さらにリビジョンを読み込む チェンジセット一覧に次の N 件のリビジョンを読み込みます。

全リビジョンを読み込む チェンジセット一覧に残りすべてのリビジョンを読み込みます。

#### 4.7.1 同期バー

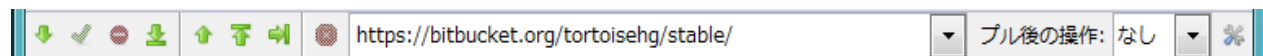


図 15 チェンジログビューアのリポジトリ同期機能

左から順に

**受信** 指定されたりポジトリからチェンジセットをダウンロードして一時的なバンドルファイルとして保存した後、プレビューモードに切り替えます。リビジョングラフでは受信したチェンジセットの横に下向きの矢印が表示されます。

**取り込み** プレビュー中のバンドルをローカルのリポジトリに取り込みます (プル)。このボタンはバンドルのプレビュー中にのみ有効になります。プル後の操作が設定されている場合、取り込み完了後に実行されます。

**破棄** プレビュー中のバンドルを破棄して、プレビューモードから抜けます。このボタンはバンドルのプレビュー中にのみ有効になります。

**プル** 指定されたりポジトリからチェンジセットをプルし、プル後の操作 (update, fetch, または rebase) が実行されます。

**インポート** インポートダイアログを開いてパッチを取り込みます。

**送信** 指定されたりポジトリに送信されるチェンジセットを確認します。リビジョングラフでは送信されるチェンジセットの横に上向きの矢印が表示されます。

**プッシュ** 指定されたりポジトリにチェンジセットをプッシュします。

**メール送信** 指定したリポジトリへのチェンジセットをメールで送信します。

**中止** 現在の操作を中止します。このボタンは送信チェンジセットの確認中 (incoming) にのみ有効になります。

**中止** ボタンの右側には閲覧中のリポジトリに設定されたすべてのリモートリポジトリを含むコンボボックスがあります。デフォルトのリポジトリパスが設定されている場合は起動時にそれが自動的に選択されます。リポジトリパス設定の詳細については <hg.1.html#urls> をご覧ください。

さらにその右側にはプル後の操作 コンボボックスがあり、同期バーによるプル操作が完了した後に実行する処理を選択することができます。更新 (update) 以外の操作として リベース (rebase) と 取得 (fetch) がありますが、それぞれ rebase 拡張機能と fetch 拡張機能を有効化しない限りリストには表示されません。

プル後の操作 コンボボックスの右側は 設定 ボタンになります。このボタンを押すとリポジトリ設定ダイアログの同期 タブが開かれ、素早くプル後の操作やりポジトリパスを設定することが可能になっています。

リポジトリ同期機能によってチェンジログビューアが起動した後に追加されたチェンジセットはリビジョングラフでは緑色の星印が表示されます。これにはコミット、プルされたチェンジセット、適用されたパッチなどを含みます。

---

注釈: 緑色の星印が表示された新規チェンジセットを通常表示に戻すには表示 -> マークをリセット を使用してください。

---

## 4.7.2 フィルタバー

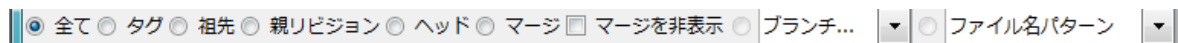


図 16 チェンジログビューアのフィルタ機能

フィルタバーを使えばチェンジセットに素早くフィルタをかけることができます。左から順に

**全て** すべてのチェンジセットを表示します。実質的にはすべてのフィルタを解除します。

**タグ** タグの付いたチェンジセットのみ表示します。

**祖先** 現在選択しているチェンジセットの祖先にあたるチェンジセットのみを表示します。このフィルタはチェンジセットが選択されているときのみ有効になります。

**親リビジョン** 作業ディレクトリの親リビジョンのみを表示します。マージ作業中でない限り常に 1 つのチェンジセットのみが表示されます。

**ヘッド** リポジトリのヘッドリビジョン (子リビジョンを持たないチェンジセット) のみを表示します。

**マージ** リポジトリのマージリビジョン (2 つの親リビジョンを持つチェンジセット) のみを表示します。

**マージを非表示** マージリビジョンの表示を切り替えるトグルボタンです。他のフィルタと併用できるようにするためラジオボタンではありません。

**ブランチ** 選択したブランチに属するリビジョンのみを表示するフィルタです。コンボボックスにはリポジトリ内のブランチ名が含まれますが、リポジトリ設定 -> チェンジログ -> 削除済みブランチの設定が適用されています。

**カスタムフィルタ** 最後のコンボボックスはその横にあるテキストボックスに入力するタイプのフィルタです。

カスタムフィルタはまず最初にフィルタの種類をコンボボックスで選択し、テキストボックスに条件を入力してエンターキーを押すことで適用されます。

**リビジョンの範囲** 入力されたテキストをリビジョンの範囲として解釈します。詳細は <hg.1.html#revisions> をご覧ください。

**ファイル名パターン** 入力されたテキストを glob パターンとして解釈します。 *regexp:* を先頭に付加することで正規表現パターンとして解釈されます。詳細は <hg.1.html#patterns> をご覧ください。

**キーワード** 入力されたテキストをチェンジセットのメタデータ (ユーザ名、コミットメッセージなど) にマッチするキーワードとして解釈します。

**日付** 入力されたテキストを日付の範囲として解釈します。詳細は <hg.1.html#dates> をご覧ください。

**ユーザ名** 入力されたテキストをユーザ名 (コミッタ) として解釈します。

フィルタ条件の入力エリアはコンボボックスになっており、ダイアログを開いている間は入力履歴を保持します。ドロップダウンリストを開いて項目を選択するとフィルタが適用されます。

### 4.7.3 リビジョングラフ

グラフ列はリポジトリのリビジョン間の親子関係を表示します。この列は読み込まれているチェンジセットを可視化するのに必要な幅が自動的に決定されます。チェンジログビュー全体の表示を保つために表示幅には制限が設けられていますが、読み込み完了後に変更することができます。

### 4.7.4 実行速度への影響

チェンジログビューには巨大なリポジトリで使うべきではない機能がいくつかあります。

There are some Repository Explorer features that should probably be avoided in large repositories.

- 表示 -> ブランチで色分け このオプションはグラフ描画のために各リビジョンごとにブランチ名を取得する必要があり、リビジョン一覧の読み込み速度の低下を招きます。
- 表示 -> コンパクトグラフ このオプションを有効にすると通常のグラフ描画速度より遅くなります。またグラフにおけるリビジョン間の接続が不正確になります。その意味でグラフの正確さがグラフの横幅の節約のどちらを取るかによります。
- リビジョン一覧の 変更 列 巨大な作業コピーを持つリポジトリにおいてこの列を表示することは実行速度の面で高くつきます。チェンジログビューの起動だけでなく、スクロール速度も低下します。

### 4.7.5 リビジョンの右クリックメニュー

グラフパネルのリビジョンを選択して右クリックするとリビジョンの右クリックメニューが表示されます：

GUI 差分ツールで表示 指定された GUI 差分表示ツールでこのチェンジセットを開きます。

変更点の表示 チェンジセットブラウザでこのチェンジセットを開きます。

ローカルとの差分 GUI 差分表示ツールでこのチェンジセットと作業ディレクトリの内容の差分を表示します。

ハッシュ値のコピー このリビジョンの完全なハッシュ値をクリップボードにコピーします。

更新... 作業ディレクトリをこのリビジョンにおける内容に更新します。<sup>\*1</sup>

マージ... このリビジョンとマージします。<sup>\*2</sup>

バックアウト... 選択中のリビジョンをバックアウトするチェンジセットを作成します。

元に戻す 作業ディレクトリの親リビジョンを変更せずに、このリビジョンの内容に戻します。十分注意して使用してください。

## エクスポート

パッチの作成... このリビジョンの変更点を含んだパッチを作成します。

メールで送信... このリビジョンの変更点をメールで送信します。<sup>\*3</sup>

バンドルの作成... 選択中のリビジョンから tip リビジョンまでのすべてのリビジョンを含むバンドルを作成します。

アーカイブに保存... このリビジョンでアーカイブダイアログを開き、リポジトリのバックアップを作成します。

## タグ

タグの追加/削除... TortoiseHg のタグ編集ダイアログを開きます。

ブックマークの追加/移動/削除... TortoiseHg のブックマークダイアログを選択中のリビジョンで開きます。このメニューは *bookmarks* 拡張機能が有効なときにのみ表示されます。

ブックマークの名前変更... TortoiseHg のブックマーク名変更ダイアログを開きます。このメニューは *bookmarks* 拡張機能が有効なときにのみ表示されます。

## Mercurial Queue

MQ にインポート 選択中のリビジョンを MQ にインポートします。qbase リビジョンまたはヘッドリビジョンにのみ有効です。このメニューは *mq* 拡張機能が有効なときにのみ表示されます。

<sup>\*1</sup> 選択したリビジョンが指定された状態で TortoiseHg の更新ダイアログが開きます。

<sup>\*2</sup> 選択したリビジョンが指定された状態で TortoiseHg のマージダイアログが開きます。

<sup>\*3</sup> 選択したリビジョンが指定された状態で TortoiseHg のメール送信ダイアログが開きます。

削除... 選択中のリビジョンとそのすべての子孫にあたるリビジョンをリポジトリから削除します。<sup>\*4</sup>

このメニューは *mq* 拡張機能が有効なときにのみ表示されます。

ローカルに移植 選択中のリビジョンを現在の作業ディレクトリの親リビジョンの上に移植 (transplant) します。このメニューは *transplant* 拡張機能が有効なときにのみ表示されます。

### Bisect

リセット Bisect の状態をリセットします。詳しくは後述の *bisect* のセクションを参照してください。

Good としてマーク 選択中のリビジョンを Good としてマークします。

Bad としてマーク 選択中のリビジョンを Bad としてマークします。

テストをスキップ 選択中のリビジョンのテストをスキップします。

リビジョンを選択した後、別のリビジョンの上で右クリックすると先ほどとは異なる右クリックメニューが表示されます。このメニューにはリビジョン範囲を扱うコマンドが含まれています。

選択中のリビジョンとの差分 リビジョン範囲で累積された変更をステータスビューで開きます。これは変更点の中から特定のファイルへの変更だけを取り出して保存するのに便利です。

選択中のリビジョンとの GUI 差分表示<sup>\*5</sup> リビジョン範囲で累積された変更を GUI 差分表示ツールで開きます。

この範囲をメールで送信 リビジョン範囲でメール送信ダイアログを開きます。

この範囲をバンドル リビジョン範囲を含むバンドルファイルを作成します。

この範囲をパッチとして保存... リビジョン範囲のチェンジセットを個別のパッチとして保存します。

マージ...<sup>\*6</sup> このリビジョンと選択中のリビジョンをマージします。もしどちらのリビジョンもチェックアウトされていない場合はマージ作業を始める前に最初に選択したリビジョンに更新されます。作業ディレクトリがクリーンでない場合、この操作は失敗します。

この範囲をローカルに移植 リビジョン範囲のチェンジセットを現在の作業ディレクトリの親リビジョンの上に移植 (transplant) します。このメニューは *transplant* 拡張機能が有効なときにのみ表示されます。

選択中のリビジョンの上にリベースする このリビジョンを選択中のリビジョンの上にリベースします。このメニューは *rebase* 拡張機能が有効なときにのみ表示されます。

この範囲を MQ にインポート リビジョン範囲のチェンジセットを MQ にインポートします。このメニューは *mq* 拡張機能が有効なときにのみ表示されます。

---

<sup>\*4</sup> strip コマンドは削除したリビジョンを後で再適用できるようにバンドルファイルとして保存します。詳細は [こちら](#) を参照してください。

<sup>\*5</sup> ユーザ設定 → TortoiseHg → GUI 差分表示コマンド

<sup>\*6</sup> 選択中のリビジョンが現在の作業ディレクトリの親リビジョンであるときのみ有効になります。

#### 4.7.6 ファイルの右クリックメニュー

左下にあるファイル一覧のファイル名の上で右クリックすると、そのファイルの右クリックメニューが表示されます：

GUI 差分表示 このファイルを GUI の差分表示ツールで開きます。

ローカルとの差分 このファイルと作業ディレクトリの該当ファイルの差分を視覚的に表示します。

エディタで表示 このファイルを GUI のエディタで開きます。<sup>\*7</sup>

ファイルに保存... このファイルを指定された場所に保存します。

履歴の表示 このファイルが変更されたすべてのリビジョンを表示します。<sup>\*8</sup>

行ごとの履歴を表示 このファイルの行ごとの変更履歴を表示します。

元に戻す このファイルを作業ディレクトリに取り出します。<sup>\*9</sup>

#### 4.7.7 チェンジセットブラウザ

チェンジセットブラウザは動作速度向上のため通常 1 つのファイルの差分のみ表示します。一度にすべてのファイルの差分を表示したい場合は [すべてのファイル] をクリックしてください。またチェンジセットブラウザは最大差分サイズを超えるファイルは表示せず飛ばします。詳しくはユーザ設定 → *TortoiseHg* → 最大差分サイズを参照してください。表示-> 最大差分サイズ制限を無視 を使用すると一時的にこのオプションを無効化できます。

チェンジログビューアとリポジトリ検索ツールは単一のリビジョンまたはいくつかのリビジョン範囲が連結された差分を表示するためにチェンジセットブラウザを開きます。チェンジセットブラウザはコミットツールやシェルフツールにとっても良く似ています。左側には変更されたすべてのファイルが一覧表示され、右側の差分パネルには変更箇所が行ごとに表示されます。

‘変更差分モード’で開いた場合、表示しているチェンジセットからファイルまたはファイル内のハンクを選択して特定の変更点を抽出し、ツールバーの名前を付けて保存 ボタンからそれらをパッチとして保存できます。これはリポジトリから必要な変更のみを取り出すのに非常に効率的な方法です。チェンジセットブラウザはキーボードショートカットの `Ctrl-C` に対応しているため、ハイライトされているハンクをクリップボードにコピーできます。

残念ながら *TortoiseHg* にはまだ変更点を取り込むためのダイアログが用意されていないため、これらについてはコマンドラインから `hg import` を実行してください。

<sup>\*7</sup> ユーザ設定 → *TortoiseHg* → GUI エディタ

<sup>\*8</sup> ファイルの履歴を変更しない単純な変更のみ表示し、削除されたファイルなどは表示されません。

<sup>\*9</sup> 新しいファイル内容は作業ディレクトリへの変更として反映されるので、コミットが必要になります。



### 4.7.8 コミットメッセージの解析

TortoiseHg 1.0 からチェンジログビューアはコミットメッセージ内のチェンジセットのハッシュ値、ウェブサイトの URL、バグ管理システムの識別番号を認識して下線を引きます。下線が引かれた部分はクリック可能なリンクになります。

16 進数 (0-9a-f) の 12 文字または 40 文字の連続した文字列はすべてチェンジセットのハッシュ値と見なされ、そのチェンジセットへのリンクになります。開いているリポジトリにそのチェンジセットが存在すれば、クリックすることで移動できます。

ウェブサイトの URL も同様にクリック可能なリンクになり、システム既定のブラウザで開かれます。

バグ番号のリンクは設定ファイルの `tortoisehg` セクションが設定されているときに有効になります。今のところ 1 度に 1 つのバグ管理システムしか登録できませんが、通常はリポジトリの `.hg/hgrc` ファイルに記述します。設定項目は `issue.regex` と `issue.link` の 2 つです。 `issue.regex` でバグ番号を切り出すための正規表現を定義し、 `issue.link` でどのような URL を生成するかを定義します。

`issue.regex` では正規表現のグループ機能を使うことができ、グループでマッチした文字列は `{n}` (`n` は非負整数) を `issue.link` に含めることで展開されます。 `{0}` は `issue.regex` によるマッチ全体、 `{1}` は最初のグループを参照します。 `{n}` トークンが `issue.link` に含まれていない場合、マッチ全体が `issue.link` 文字列の最後に追加されます。

設定例：

```
BitBucket:
issue.regex = #(\d+)\b
issue.link = https://bitbucket.org/<your project and repo>/issue/{1}/

Mercurial:
issue.regex = \bissue(\d+)\b
issue.link = https://bz.mercurial-scm.org/show_bug.cgi?id={1}
```

### 4.7.9 Bisect

TortoiseHg 1.0 から、どのチェンジセットにおいてバグが入り込んだのかを調べる Bisect 機能に対応しました。まず、バグが確認されているもっとも新しいチェンジセットに "Bad" のマークを付け、次にバグがないとわかっているもっとも古いチェンジセットに "Good" のマークを付けます。そしてテストを実行して作業ディレクトリの親リビジョンが "Good" なのか "Bad" なのかをマークします。すると Bisect は別の可能性のあるチェンジセットに更新します。もしそれ以上可能性のあるリビジョンがない場合は問題のリビジョンが見つかったことを知らせます。

リビジョンに "Good" または "Bad" のマークを付ける際、そのチェンジセットをチェックアウトする必要はありません。

より自動的な Bisect を実現するには Mercurial をコマンドラインから使用し、作業ディレクトリの構成でビルドしてテストを実行し、0 (成功)、125 (スキップ)、127 (中断)、または "Bad" (0、125、127 以外) を返す自動化されたテストを提供する必要があります。詳しくは `hg bisect` コマンドのヘルプを参照してください。



#### 4.7.10 キーボードショートカット

Ctrl-P 作業ディレクトリの親リビジョンに移動します

Ctrl-D GUI の差分表示ツールで選択中のチェンジセットを表示します

Ctrl-R 表示内容を再読み込みします

Ctrl-G 指定リビジョンに移動します

#### 4.7.11 オプション設定

チェンジログビューアには TortoiseHg 設定ダイアログのチェンジログタブから変更できるいくつかのオプション項目があります。

コミッタの色分け チェンジセットをコミッタごとに色分けします

改行を無視してログを表示 コミットメッセージの改行を無視して、最大 80 文字まで表示します

読み込むログの件数 一度に読み込むリビジョンの件数

ハッシュ値のコピー 選択時にリビジョンのチェンジセット ID (ハッシュ値) をクリップボードにコピーします。廃止予定。

削除済みブランチ リポジトリでブランチ名一覧を作成する際に無視するブランチ名をカンマ区切りで指定します。

ブランチの色: ブランチ名とその色を "branch:#XXXXXX" の形式で空白文字区切りで指定。ブランチ名に含まれる空白文字とコロンの ":" はバックスラッシュ ("\") でエスケープする必要があります。同様に他の文字についても、例えば "u0040" はアットマーク ("@" ) にデコードされ、"n" は改行になります。

非表示タグ: 非表示にしたいタグ名を空白区切りで指定。便利な使用例: "qbase qparent qtip" を設定することで MQ 拡張機能が使用するタグを隠すことができます。

折り畳み チェンジセットのヘッダ情報を折り畳み可能にします。このオプションを有効にした場合、ヘッダ情報の部分は固定表示になります。

特定のコミットに指定した色を割り当てたい場合は Mercurial.ini ファイルに次の設定を追加してください:

```
[tortoisehg]
authorcolor. コミッタ名 = color
```

またチェンジログビューアは TortoiseHg ページの以下の設定も反映されます:

タブ文字の幅 コメント情報の表示でタブ文字 1 つあたりいくつの空白に展開するかを指定します

最大差分サイズ 表示する最大の差分ファイルサイズ

差分を画面下部に表示 ファイル一覧の下に差分表示パネルを配置する

#### 4.7.12 コマンドラインからの実行

チェンジログビューアはコマンドラインから起動することもできます :

```
hgtk log [OPTIONS] [FILE]

aliases: history

changelog viewer

options:

  -l --limit limit number of changes displayed

use "hgtk -v help log" to show global options
```

### 4.8 リポジトリ検索

リポジトリ検索ツールはリポジトリのリビジョン履歴を調べるのに使うことができます。検索 と コメント の 2 種類のタブを持つアプリケーションです。

#### 4.8.1 検索タブ

検索タブはすべてのリビジョン履歴からキーワード、変数名、関数名などで検索 (grep) できます。

各テキストボックスは以下の意味があります :

正規表現 正規表現検索に使用する条件。

次を含む 検索対象に含めるパスをカンマ区切りで指定。指定されていない場合は、リポジトリ全体が対象となります。つまりパスを指定して初めて検索対象が絞り込まれます。

次を含まない 検索対象から外すパスをカンマ区切りで指定。この条件より先に 次を含む の条件が適用されます。

テキストボックスの下のチェックボックスは以下のようになります :

コピーと名前変更を追跡 次を含む の条件が指定されていないとき、コピーまたは名前変更を考慮して検索します

大文字小文字を区別しない アルファベットの大文字小文字を区別せず検索を実行します

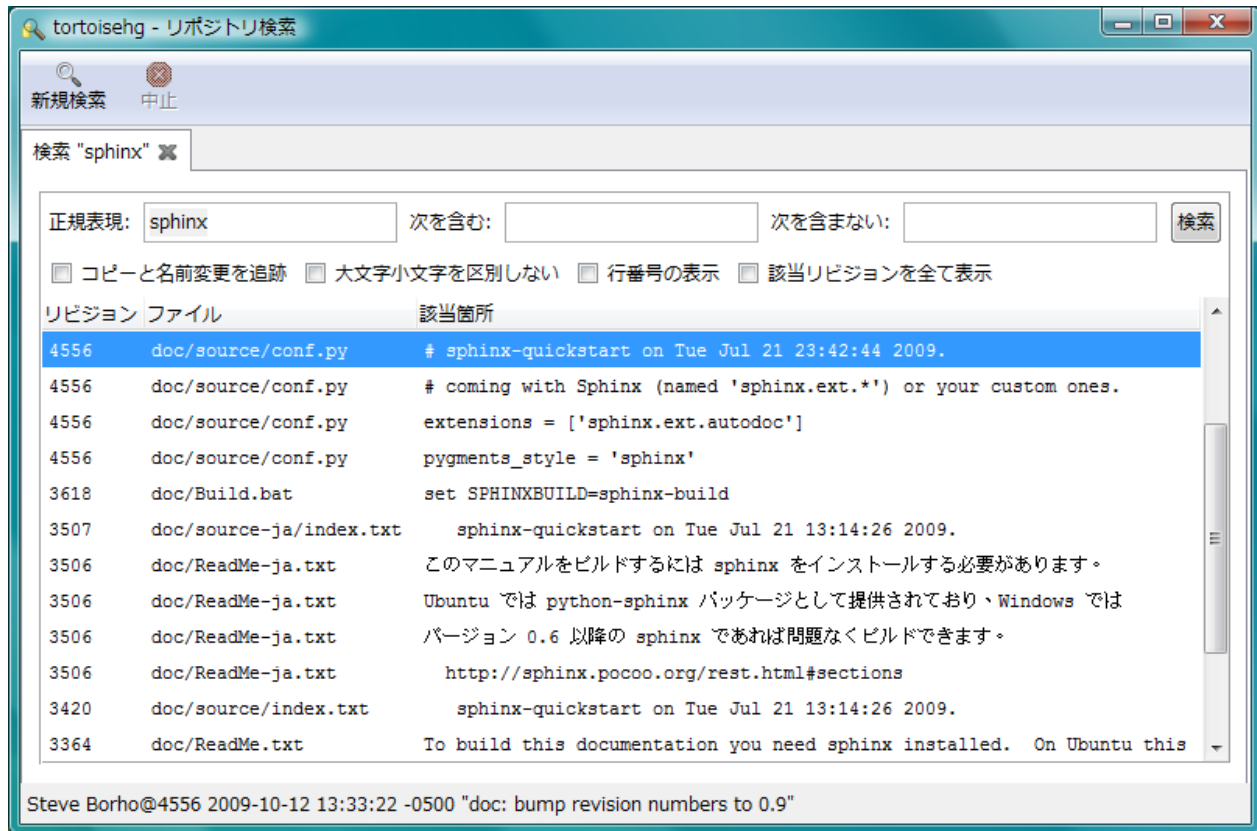


図 17 検索タブ

行番号の表示 検索結果に該当個所の行番号を表示します

該当リビジョンをすべて表示 検索条件にマッチした直近のリビジョンだけでなく、すべてのリビジョンを表示します。該当行が追加されたのか削除されたのかを示す +/- 記号が表示されます。

検索タブは検索を開始するとそのキーワードにタブ名が変更されます。ツールバーの 中止 ボタンを押せば処理中の検索が停止しますが (中止 ボタンは検索中にのみ押せる状態になります)、新規検索 ボタンを押せば新しい検索タブを開くことができます。

## 検索結果

検索結果は該当チェンジセットへのリンクとなり、マウスカーソルを乗せることでツールチップ (コミット、日時、コミットログ) が表示されます。検索結果の各行で右クリックをすると次の機能を持つ右クリックメニューが表示されます:

変更点の表示 このチェンジセットをチェンジセットダイアログで開き、すべての変更点を確認します

コメントする このリビジョンのファイルをコメントタブで開きます

履歴の表示 チェンジログビューアでこのファイルのリビジョン履歴を表示します

ファイルの表示 そのファイルをあらかじめ設定された GUI エディタで開きます

## 4.8.2 コメントタブ

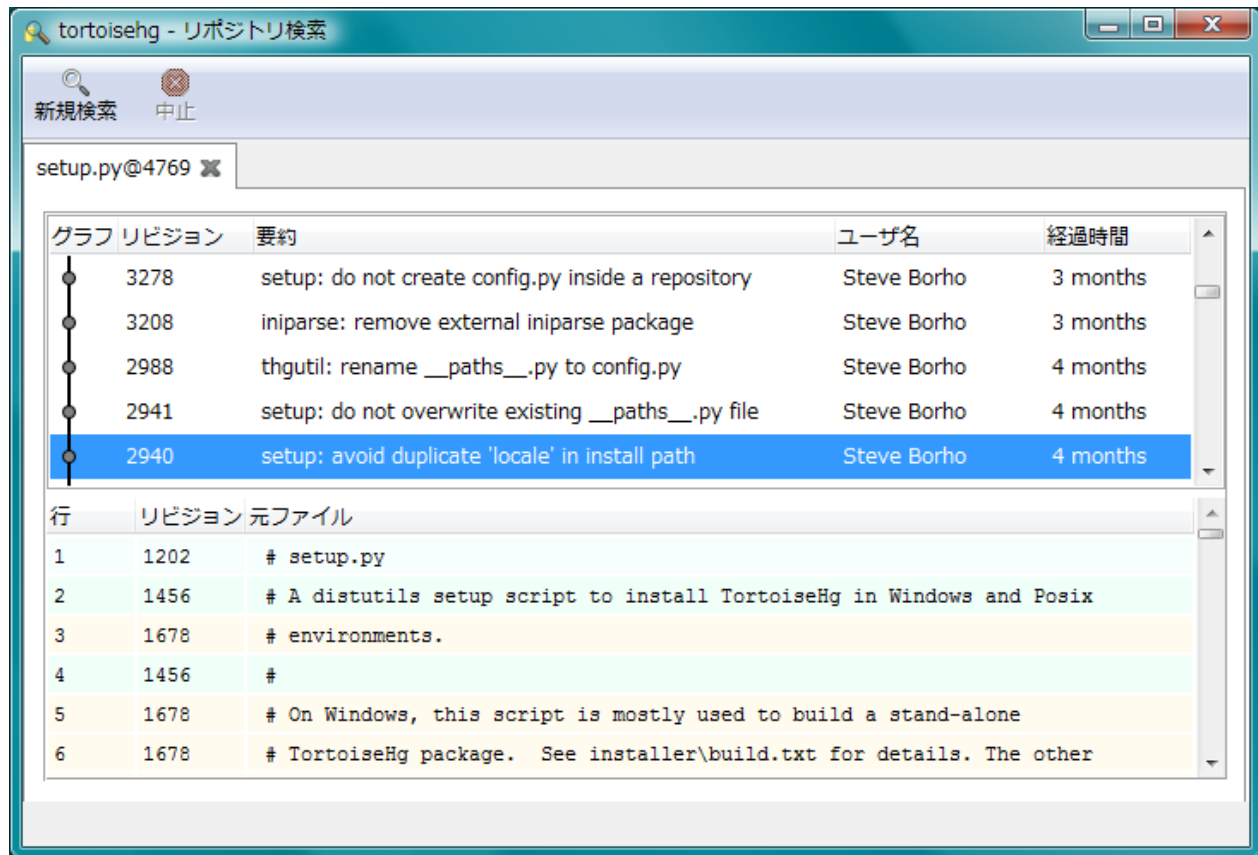


図 18 コメントタブ

リビジョングラフパネル (上半分) にはそのチェンジセットにおける変更全体をチェンジセットダイアログで表示するための右クリックメニューが用意されています。リビジョングラフパネルのリビジョンをダブルクリックすると下半分のパネルにファイルのコメント情報が更新されます。

コメントパネル (下半分) は現在のコメント情報です。パネル内の各行も同様にチェンジセットへのリンクになっています。行をダブルクリックすると上半分のパネルが更新され、その行が追加または変更されたチェンジセットが表示され、フォーカスが移動します。

コメントパネル内の各行は 2 つの指標に基いて色分けされており、コミットが色相 (色の違い)、最後の変更からの経過時間が彩度になっています。つまり、古い変更になるほど色が薄くなるわけです。

コメントパネルの列ヘッダ部分 (行、リビジョン、ソース) で右クリックすると表示される右クリックメニューからデフォルトでは表示されていないファイル名とユーザ列を表示できます。

## 名前変更の追跡

コメント情報のデータはコードの各行が作成された初期チェンジセットを探し出すために自動的にコピーや名前変更の履歴をさかのぼって収集されます。

一方で、リビジョングラフは自動的に名前変更やコピーをたどりません。その代わりに、リビジョニー覧の名前変更またはコピーを含むチェンジセット行をクリックすると、その履歴をたどったグラフを表示するためのボタンが表示されます。そのボタンを押すと同じチェンジセットの古いファイル名でコメントタブが開きます。リビジョングラフも同様に名前変更とコピーをたどって表示されるため、異なるファイル名を含むチェンジセット同士が接続される場合があります。その場合はグラフの色が変化します。

## オプション設定

コメントタブは以下の他のツールの設定項目を使用しています：

チェンジログ → コミッタの色分け   チェンジセットをコミッタごとに色分けします

チェンジログ → 改行を無視してログを表示   コミットメッセージの改行を無視して、最大 80 文字まで表示します

*TortoiseHg* → タブ文字の幅   コメント情報の表示でタブ文字 1 つあたりいくつの空白に展開するかを指定します

### 4.8.3 コマンドラインからの実行

リポジトリ検索ツールはコマンドラインから起動することもできます：

```
hg tk datamine

aliases: annotate, blame

repository search and annotate tool

use "hg tk -v help datamine" to show global options
```

## 4.9 同期ツール

注釈：同期ツールはバージョン 0.9 において廃止予定となり、以降のリリースで削除されます。そのため同期操作はチェンジログビューアで行うことを推奨しています。

同期ツールはリポジトリ間でチェンジセットを転送したり、メールを送信するのに使います。

受信 指定したリポジトリからプルされるチェンジセットを表示します。それらはローカルのリポジトリではなく、指定したリポジトリにあるチェンジセットです。

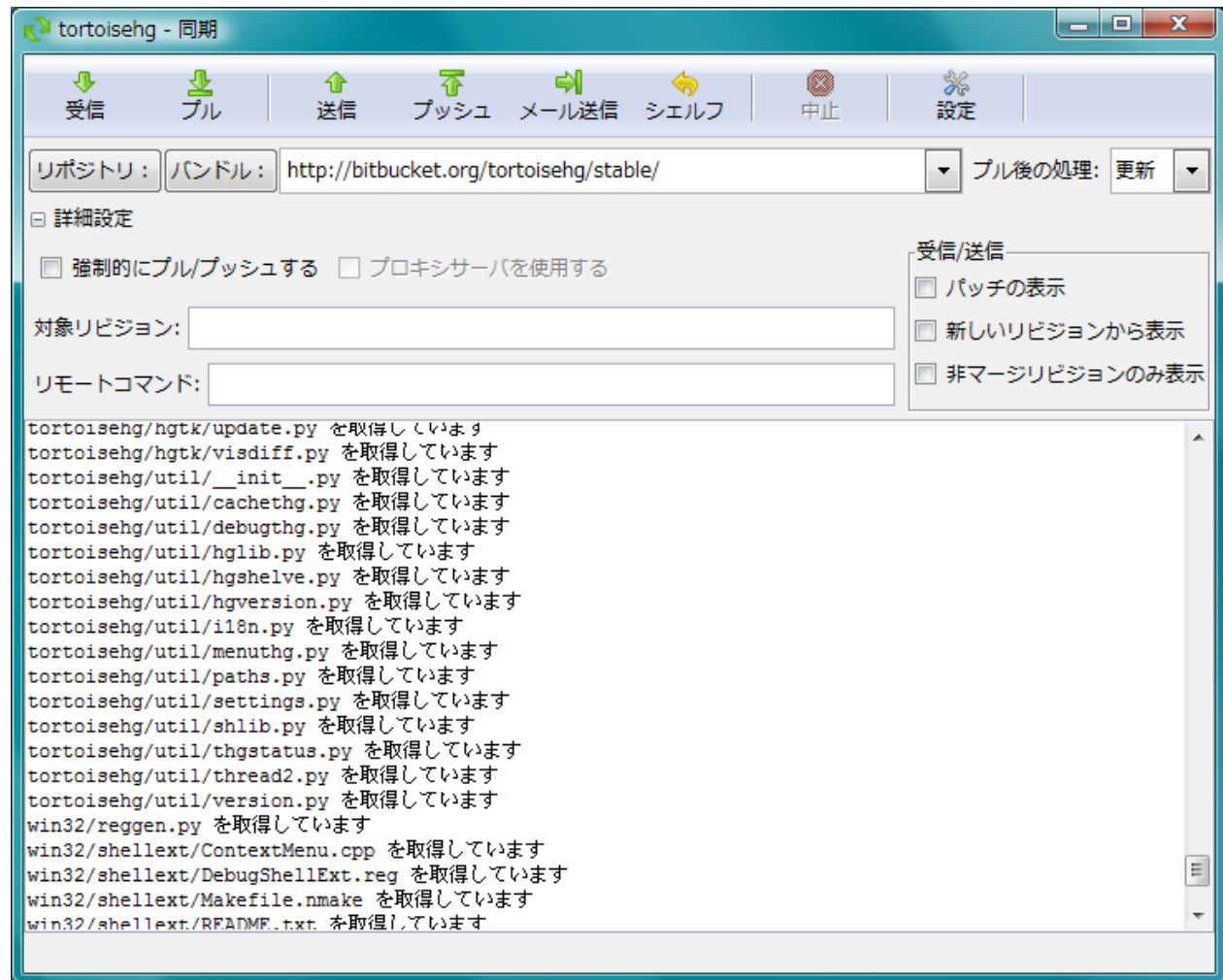


図 19 同期ダイアログ

**プル** チェンジセットを指定したリポジトリからプルします

**送信** 指定したリポジトリにプッシュするチェンジセットを表示します。それらはローカルにあって、指定したリポジトリにはないチェンジセットです。

**プッシュ** チェンジセットを指定したリポジトリにプッシュします。その結果、ローカルにおける *tip* リビジョンが指定したリポジトリの新しい *tip* リビジョンになります。

**メール送信** 指定したリポジトリへのチェンジセットをメールで送信します

**シェルフ** いくつかの同期操作は作業ディレクトリがクリーンであることを前提としているため、シェルフツールを起動して作業ディレクトリにおける変更点を一時的にシェルフに退避します。

**中止** 現在の操作を中止します

**設定** リポジトリパス (とそのエイリアス) の設定ダイアログを開きます

ツールバーの下ボタンはそれぞれ：

リポジトリ: 同期に使用するローカルリポジトリを指定します

バンドル: プルの対象となるローカルのバンドルファイルを指定します

その隣のコンボボックスには同期の対象となるリポジトリのパスを入力または選択します。ドロップダウンリストにはこのリポジトリに設定されているパスのエイリアスが表示されます。

プル後の処理 のフレームにはプル操作が完了した後に実行する処理を選択するラジオボタンがあります。この動作はツールバーの 設定 ボタンから変更可能です。ユーザ設定を上書きすることでリポジトリごとにデフォルトの動作を設定できます。

なし プル後に何もしません。下のログパネルでプルされたチェンジセットを確認できます。 *tip* リビジョンに更新するためのボタンが表示されることもあります。

更新 ローカルリポジトリに新しいリビジョンがプルされた場合に自動的にブランチの *tip* リビジョンに更新します。これによってプルした変更とローカルのまだコミットされていない変更が衝突した場合にマージが発生する可能性があります。

取得 `hg fetch` と同じ操作になります。動作の詳細については `fetch` 拡張機能のヘルプを参照してください。この機能は `fetch` 拡張機能がユーザによって有効化されているときのみ表示されます。

リベース `hg pull --rebase` と同じ操作になります。動作の詳細については `rebase` 拡張機能のヘルプを参照してください。この機能は `rebase` 拡張機能がユーザによって有効化されているときのみ表示されます。

プロキシサーバを使用する チェックボックスは同期操作ごとにプロキシサーバの設定を無効化するのに便利です。このボタンはプロキシ設定がされている場合にのみ有効になります。

すべての同期操作はユーザからの入力を促すダイアログによる認証を必要とします。TortoiseHg は `ssh:` 接続とその認証のために TortoisePlink ツール (TortoiseSVN からの提供) を使用します。SSH サーバへの接続に問題がある場合は [よくある質問](#) を参考にしてみてください。

折り畳める 詳細設定 パネルにはプル・プッシュ操作に関連した設定がいくつか用意されています:

強制的にプッシュ/プルする 無関係なリポジトリまたは複数のヘッドがあるなどの警告を無視して実行します

対象リビジョン すべてのリビジョンを送らないようにします

リモートコマンド `-e` オプションを使います

パッチの表示 チェンジセット送受信時の差分を表示します

最新のものを最初に表示 チェンジセットを逆順に表示します

マージリビジョン以外を表示 結果表示からマージリビジョンを除外します (プッシュ・プル操作に影響はありません)

### 4.9.1 プルした後

チェンジセットがローカルのリポジトリにプルされた後、ダイアログの下部にボタンが表示されます：

*tip* リビジョンに更新 作業ディレクトリを現在のブランチの *tip* リビジョンに更新します

ボタンは使用可能なときにのみ表示されます。

### 4.9.2 メール送信

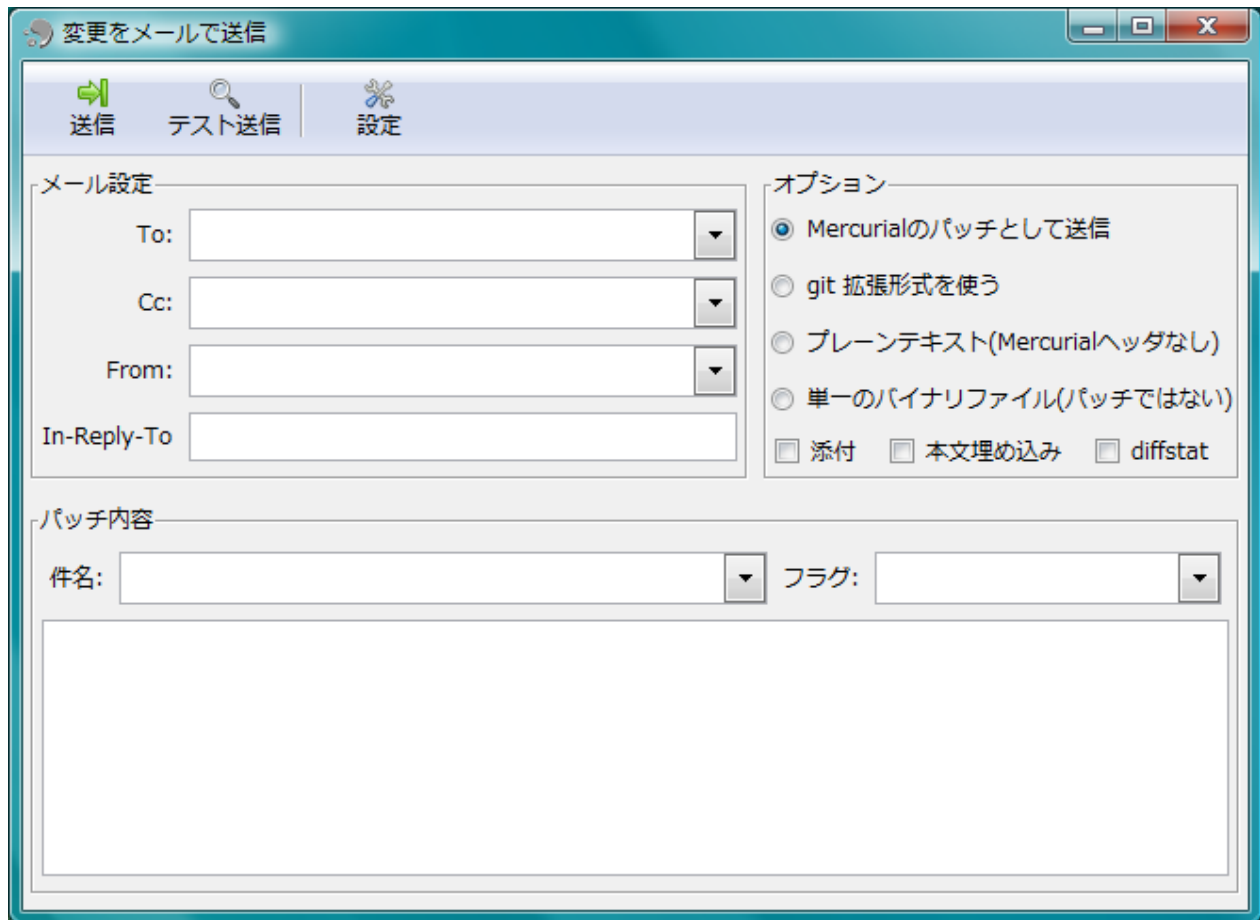


図 20 メールダイアログ

メールダイアログは 2 つの TortoiseHg ツールから起動できます。

- 1) チェンジログビューア：単一または複数リビジョンをメールで送信したい場合。
- 2) 同期ツール：指定したリポジトリに送信可能なすべてのチェンジセットをメールで送信したい場合（メールダイアログを起動する前に 送信 ボタンでどのチェンジセットが送信対象になるのかを確認するのはよい習慣です）。



送信 ボタンはそのまま、設定 ボタンを押すと TortoiseHg の設定ダイアログにある メール送信 ページを開きます。そこでは SMTP 設定や *To:* アドレスと *From:* アドレスのデフォルトを指定することができます。

*In-Reply-To:* は送信するパッチがメーリングリストで確実にスレッドとして認識されるために使われます。

ブレーンパッチ、Mercurial パッチ、GIT パッチ、バンドルの違いについては Mercurial のマニュアルを参照してください。

### 4.9.3 コマンドラインからの実行

同期ツールはコマンドラインから起動することもできます：

```
hgtk synch

aliases: pull, push, incoming, outgoing, email

repository synchronization tool

use "hgtk -v help synch" to show global options
```

コマンド構文は共通オプションを除けばオプションや引数を必要としないためシンプルになっています。**push**、**outgoing**、**email** のエイリアスで同期ツールを起動した場合、自動的に *default-push* URL が選択されます。それ以外のエイリアスの場合は *default* URL が選択されます。選択された URL が見つからない場合は最初に見つかったパスが使用されます。

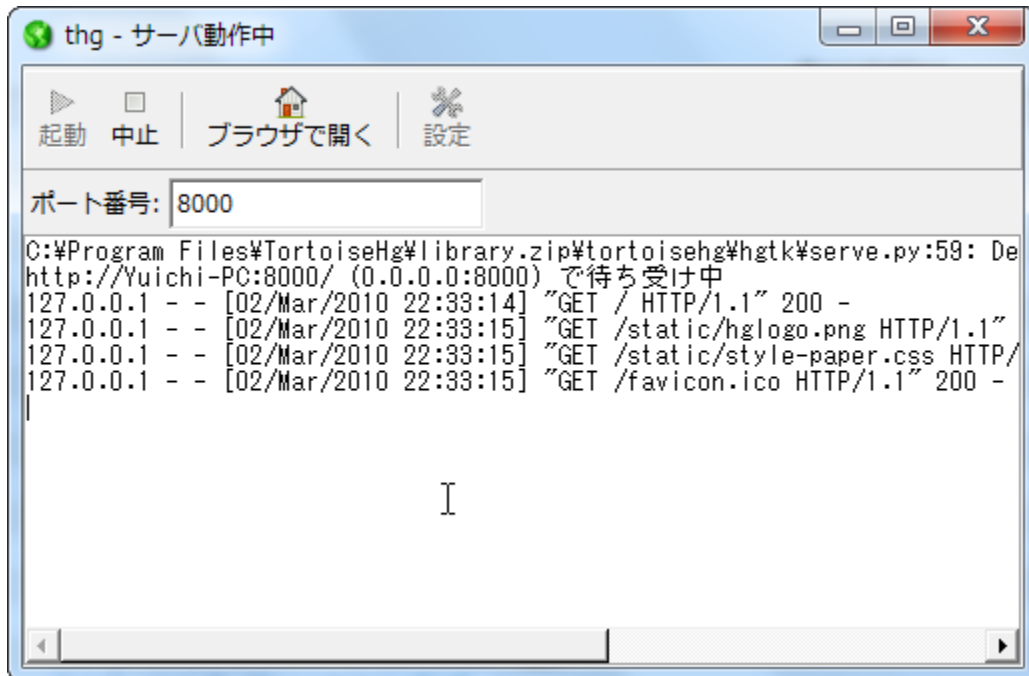
## 4.10 サーバ

サーバツールは Mercurial 内蔵のウェブサーバのラッパーです。起動すればどんなコンピュータでも HTTP ポートでサービスを開始し、リポジトリの閲覧や、クローン、プルだけでなく、サーバ設定を変更して許可することでプッシュも可能です。

ツールバーボタン：

```
:guilabel: `開始`
           ウェブサーバを開始します
:guilabel: `中止`
           ウェブサーバを停止します
:guilabel: `ブラウザで開く`
           既定のブラウザを起動してリポジトリを表示します
:guilabel: `設定`
           リポジトリのテーマ、説明、アクセスポリシーの設定をします
```

設定 ボタンを押すとそのリポジトリの設定ダイアログが開きます。サーバ設定の詳細については Mercurial の Wiki を参照してください。



#### 4.10.1 複数のリポジトリ

1 つのウェブサーバプロセスで複数のリポジトリを運用したい場合は `hgwebdir.conf` という名前のテキストファイルを作成し、そのファイルに次のような設定を記述してください:

```
[paths]
/ = /path/to/repositories/*
```

左辺の `/` はウェブサーバにアクセスするときのパスで、右辺はローカルリポジトリのディレクトリパスです。ここに複数の項目を記述することができます。

この設定ファイルを使用するためにウェブサーバダイアログを右クリックメニューからではなく、コマンドラインから `hggtk serve --webdir-conf=hgwebdir.conf` を実行して起動してください。

#### 4.10.2 コマンドラインからの実行

サーバツールはコマンドラインから起動することもできます:

```
hggtk serve [OPTION]...

web server

options:

    --webdir-conf name of the webdir config file
```

(次のページに続く)

(前のページからの続き)

use "hgtk -v help serve" to show **global** options

## 4.11 名前変更の推定

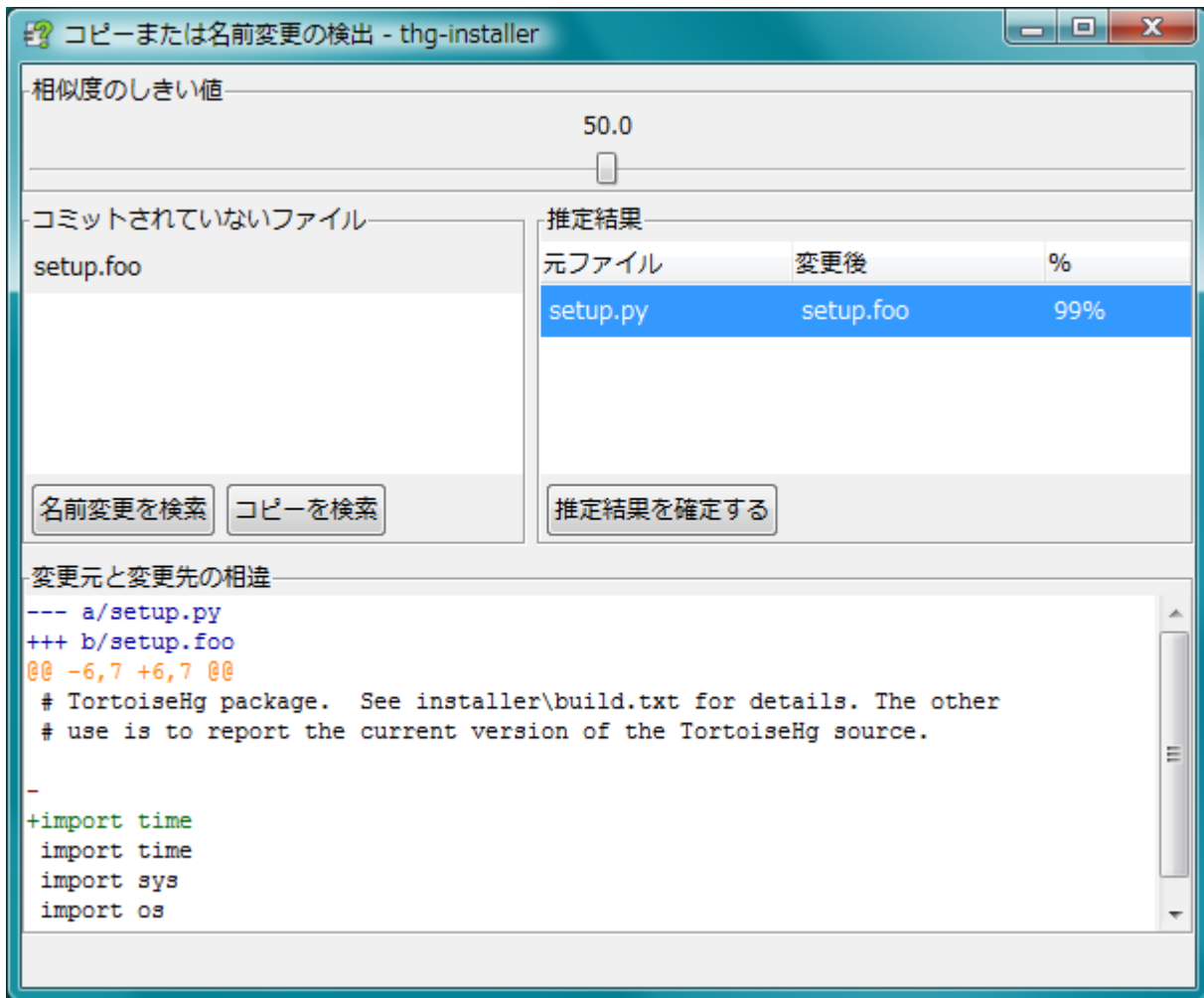


図 21 名前変更推定ダイアログ

このダイアログはファイルの名前変更、移動およびコピーを Mercurial からの情報を使わずに検索するのに使います。エクスプローラの右クリックメニューや、ステータス・コミットツールのファイル一覧に表示された不明ファイルの右クリックメニューから起動できます。

基本的な使い方は以下になります：

- 1) コミットされていないファイル から 1 つまたは複数のファイルを選択する
- 2) 相似度のパーセンテージをスライダを使って設定する

- 3) 名前変更の検索 または コピーの検索 ボタンを押す
- 4) 右側の推定結果から適切な項目を選択して 推定結果を確定する ボタンを押します
- 5) すべての未コミットファイルが解決するまで以上を繰り返します

#### 4.11.1 名前変更の検索

この機能はリポジトリから見あたらないファイル (管理されているファイルだったけど今は無い) を検索します。見あたらないファイルごとにリポジトリで管理されているファイルの最新リビジョンと比較し、相似度のしきい値の設定を上回るファイルを推定結果の一覧にそのペアを追加します。

#### 4.11.2 コピーの検索

この機能は見あたらないファイルごとにリポジトリで管理されているファイルと完全に一致するファイルを探し出します。

#### 4.11.3 推定結果

この一覧に表示された項目を選択すると、その差分が下のパネルに表示されます。推定結果を確定する ボタンを押すと Mercurial に名前変更、またはコピーとして記録されます。

#### 4.11.4 コマンドラインからの実行

推定ツールはコマンドラインから起動することもできます：

```
hgtk guess

guess previous renames or copies

use "hgtk -v help guess" to show global options
```

### 4.12 無視フィルタ

無視フィルタダイアログはリポジトリのルートにある `.hgignore` ファイルのような Mercurial の無視ファイルフィルタの編集に使われます。エクスプローラの右クリックメニューや、ステータス・コミットツールのファイル一覧に表示された不明ファイルの右クリックメニューから起動できます。

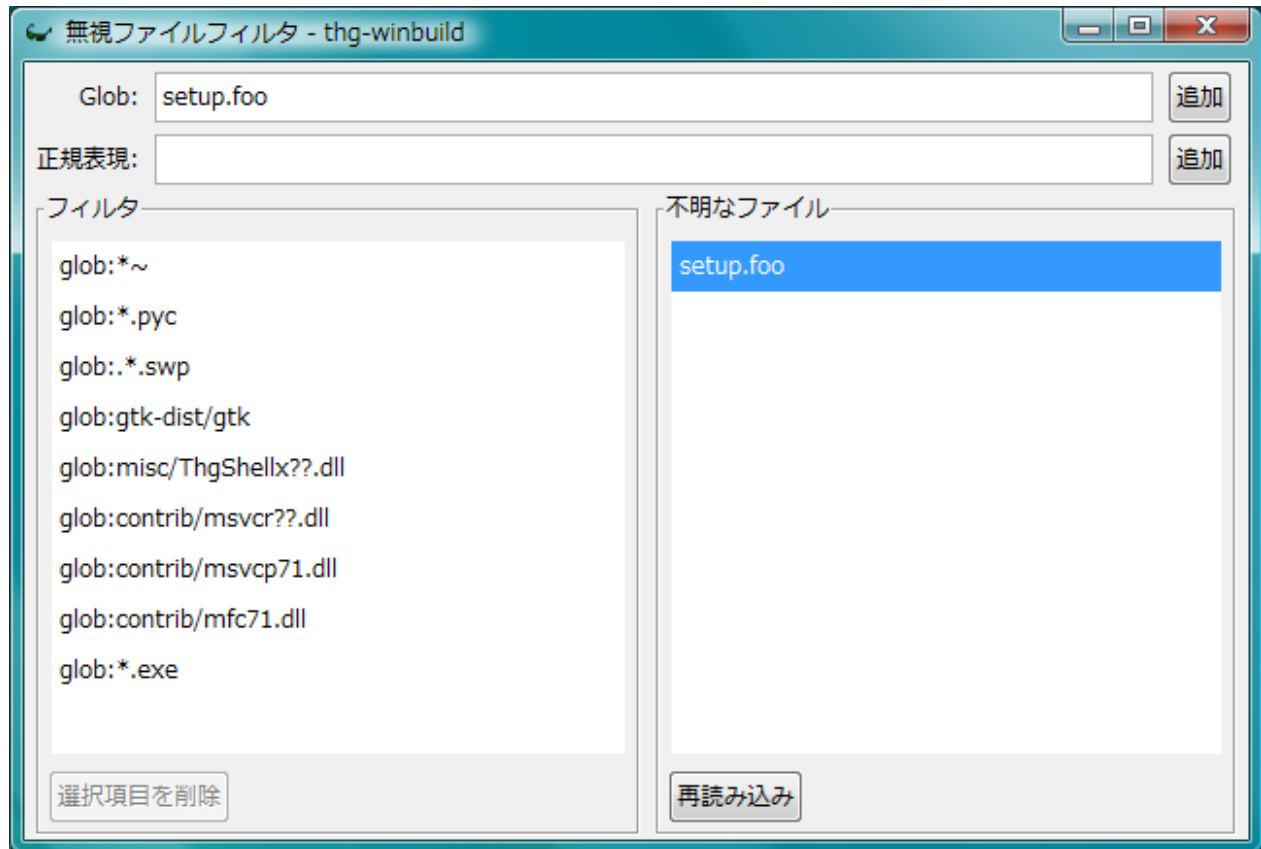


図 22 無視フィルタダイアログ

#### 4.12.1 コマンドラインからの実行

無視フィルタツールはコマンドラインから起動することもできます：

```
hgtk hgignore [FILE]

aliases: ignore, filter

ignore filter editor

use "hgtk -v help hgignore" to show global options
```



## 第 5 章

# 設定ダイアログ

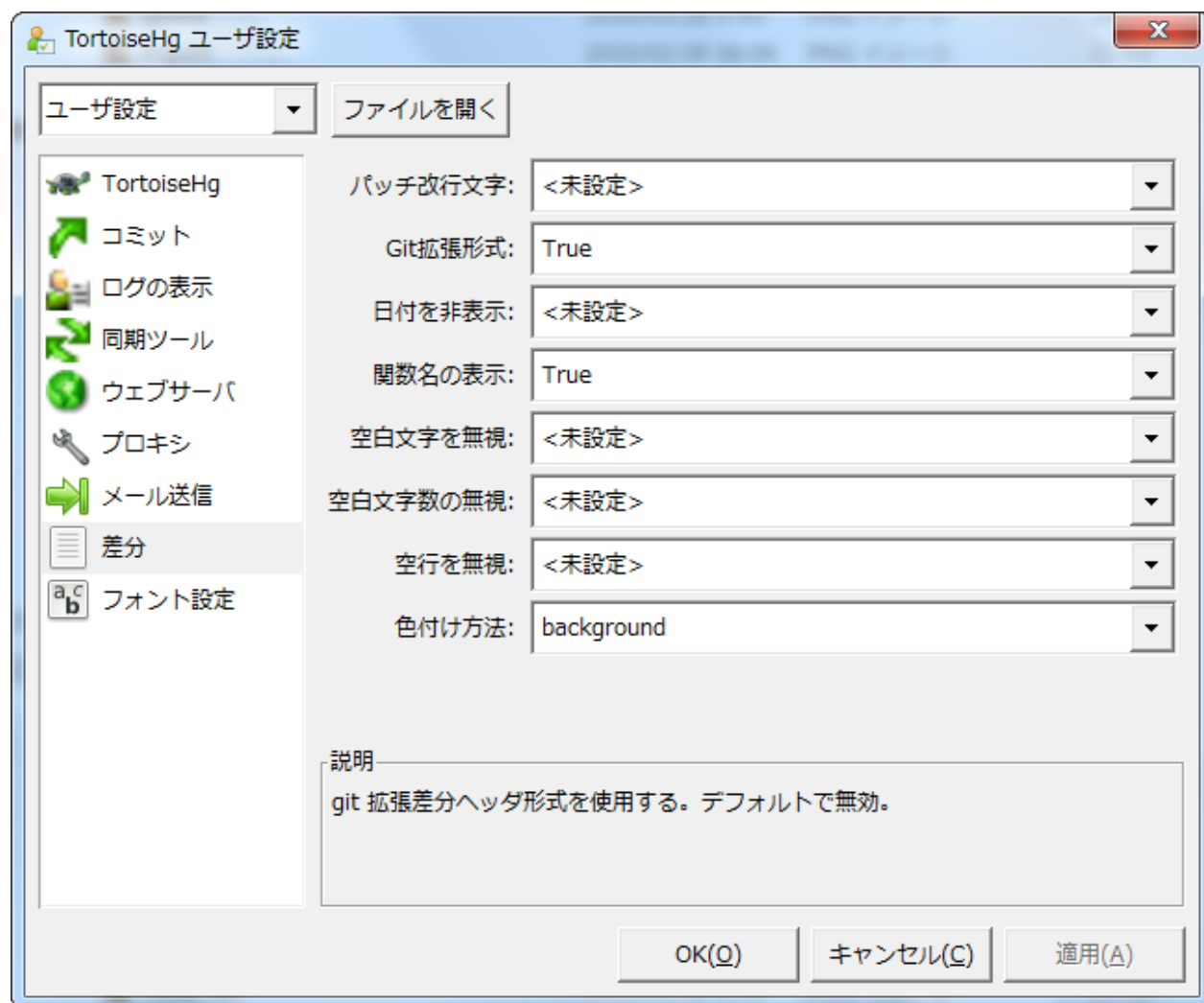


図 1 設定ダイアログ

設定ダイアログでは TortoiseHg と Mercurial 本体の両方の設定ができます。TortoiseHg は Mercurial の設定システ

ムを利用して設定情報を保存・取得しているため、本質的に同じになります。

Windows の Mercurial には 3 つの設定ファイルがあります。

- 1) グローバル設定ファイル : `C:\Program Files\TortoiseHg\Mercurial.ini` このファイルは最初  
に読み込まれるため、最も優先順位が低くなります。
- 2) ユーザ設定ファイル : `C:\Documents and Settings\ユーザ名\Mercurial.ini` このファイルは 2  
番目に読み込まれるため、前述のグローバル設定ファイルを上書きできます。
- 3) リポジトリ設定ファイル : リポジトリのルート `\.hg\hgrc` このファイルは最後に読み込まれるため、グ  
ローバルおよびユーザ設定ファイルを上書きできます。

グローバル設定ファイルは TortoiseHg のアップグレード時に上書きされる可能性があるため、このファイルを編  
集することはお勧めできません。その代わり、ホームディレクトリにあるユーザ設定ファイル `Mercurial.ini`  
または `hgrc` ファイルを変更した方が良いでしょう。TortoiseHg 設定ダイアログもこれに従い、次の 2 つのフ  
ァイルにのみ変更を行います :

ユーザ ユーザ設定ファイル `Mercurial.ini` または `hgrc` を編集します

リポジトリ リポジトリ設定ファイル `.hg/hgrc` を編集します

どちらの設定ファイルを編集するかはダイアログ上部のコンボボックスから切り替えることができます。さらにあ  
らかじめ設定されている GUI エディタで直接編集をすることもできます。

これら 2 種類の設定ファイルの使い分けについては、リポジトリ設定ファイルは他の設定ファイルより優先される  
ため、すべてのリポジトリに共通する設定をユーザ設定ファイルにまとめて保存しておき、リポジトリ設定フ  
ァイルにはリポジトリパスのエイリアスや、ウェブサーバ設定といったリポジトリ固有の設定を保存する、といった感  
じになります。ユーザおよびリポジトリ設定ファイルは自分で作成するか、初めて設定ダイアログを開くまで存在  
しませんので注意してください。

## 5.1 設定ページ

設定ツールは複数のページに分かれており、各ページは `Mercurial.ini` ファイルの各セクションに対応してい  
ます。

同期 ページを除けばすべてのページは同じレイアウトになっており、設定項目がドロップダウンコンボボックス  
と共に一覧になっています。コンボボックスには設定可能な値の他、ユーザが過去に入力した値の履歴も含みま  
す。設定項目のラベルにはツールチップが設定してあるため、マウスカーソルを乗せるとその項目の詳しい説明を  
見ることができます。コンボボックスをクリックするなどしてアクティブにしても、ダイアログ下部の説明表示用  
のフレームに同じ説明が表示されます。

設定項目についてのより詳細な情報については Mercurial の公式 Wiki を参照してください。ただし、最初の 3 つ  
のページ (TortoiseHg, コミット, チェンジログ) については TortoiseHg 固有の設定になるため、Mercurial の Wiki  
には該当する情報がありません。



### 5.1.1 TortoiseHg

**3-way マージツール:** マージ時の衝突を解決する GUI のマージプログラム。指定されなかった場合、Mercurial はシステムで最初に見つかった内蔵マージツールを衝突マーカを取り除くために使用します。強制的に衝突マーカを表示するには *internal:merge* を指定し、常に local か other を選択できるようにするには *internal:prompt* を指定し、手作業でマージするために作業ディレクトリにファイルを書き出ししておくには *internal:dump* を指定してください。

**GUI 差分表示ツール:** Mercurial 設定ファイルの [merge-tools] セクションに記述されている GUI 差分表示ツールを指定。未設定の場合は選択されている 3-way マージツールを使用します。3-way マージツールが選択されていない場合は TortoiseHg が利用可能なツールを探して使います。

**差分表示をスキップ:** TortoiseHg の GUI 差分表示ダイアログを表示せず、設定されている GUI 差分表示ツールのディレクトリ比較機能を使います。この機能は GUI 差分表示ツールが正しい *extdiff* 設定を持っている場合のみ有効にしてください。デフォルト: False。

**GUI エディタ:** ファイルを表示するための GUI エディタを指定します。

**コマンドラインエディタ:** コミット時や Mercurial がユーザから複数行のテキスト入力が必要としている場合に使用するエディタです。CLI コマンドにのみ使用されます。

**タブ文字の幅:** TortoiseHg の各種ダイアログにおいてテキスト中のタブ文字をいくつかの空白文字に展開するかを指定してください。デフォルト: タブ文字を展開しない。

**最大差分サイズ:** チェンジログ・ステータス・コミットダイアログで表示する最大ファイルサイズを KB 単位で指定してください。0 の場合は無制限となります。デフォルト: 1024。

**差分を画面下部に表示:** 差分表示パネルをステータス・シェルフ・コミットダイアログのファイル一覧の下に配置します。デフォルト: False (ファイル一覧の右側に表示)。

**標準エラーの補足:** 標準エラー出力の実行時エラーを補足して表示します。デフォルト: True。

**ダイアログプロセスの生成:** hgtk をコマンドラインから起動したとき、ダイアログ用のバックグラウンドプロセスを生成するかどうかを設定します。デフォルト: True。

**絶対パス表示:** 各種ダイアログのタイトルにリポジトリのディレクトリ名のかわりにフルパスを表示します。デフォルト: False。

**スペルチェック言語:** スペルチェックに使用するデフォルトの言語を指定します。指定されなかった場合はシステム既定の言語が使用されます。例: en, en\_GB, en\_US。スペルチェックには *gtkspell* が必要になるため、GNOME デスクトップを採用している環境でのみ使用可能です。

### 5.1.2 コミット

**ユーザ名:** コミット時に使用する名前。

要約行最大文字数: コミットログ最初の行 (要約行) の最大文字数。これが設定された場合、要約行が長すぎる、もしくは要約行とそれ以降のログが空行で区切られていないときに警告が表示されます。デフォルト: 0 (制限しません)。

コミットログ折り返し文字数: コミットログの折り返し文字数。これが設定された場合、コミットログ中の 1 行でも設定文字数を上回っていると警告が表示され、その文字数で強制的に折り返すためのポップアップメニューが表示されます。デフォルト: 0 (制限しません)。

コミット後にプッシュ: True にすることで コミット後にプッシュ チェックボックス の初期値として使用されず。デフォルト: False。

自動コミットリスト: コミットに自動的に含めたいファイルをカンマ区切りで指定します。通常はユーザ設定ではなくリポジトリ設定で使います。デフォルト: 指定無し。

自動除外リスト: ステータス、コミット、シェルフツールにおいて自動的にチェックを外したいファイルをカンマ区切りで指定します。デフォルト: 指定無し。

### 5.1.3 チェンジログ

コミッタで色分け: コミッタ名でチェンジセットを色分けします。このオプションが無効の場合、マージリビジョンは緑、親リビジョンが離れているリビジョンは赤、それ以外の通常のリビジョンは黒に色分けされます。デフォルト: False。

長めの要約行: この設定が有効の場合、最大 80 文字になるまでコミットログの改行を無視して 1 行として表示します。デフォルト: False。

ログ読み込み件数: 1 度の処理で読み込みおよび表示するリビジョンの件数。デフォルト: 500。

削除済みブランチ: リポジトリでブランチ名一覧を作成する際に無視するブランチ名をカンマ区切りで指定。デフォルト: 指定無し。

ブランチの色: ブランチ名とその色を "branch:#XXXXXX" の形式で空白文字区切りで指定。ブランチ名に含まれる空白文字とコロン (":") はバックスラッシュ ("\") でエスケープする必要があります。同様に他の文字についても、例えば "u0040" はアットマーク ("@") にデコードされ、"n" は改行になります。デフォルト: 指定無し。

非表示タグ: 非表示にしたいタグ名を空白区切りで指定。便利な使用例: "qbase qparent qtip" を設定することで MQ 拡張機能が使用するタグを隠すことができます。デフォルト: 指定無し。

### 5.1.4 同期

同期 ページではリポジトリに関連するパス (URL やディレクトリパス) を保存することができます。これらのパスをユーザ設定やユーザ設定ファイルに保存することはあまりなく、多くの場合リポジトリ設定ファイルにのみ保

存します。Mercurial には 2 つの特別なパス名 (エイリアス) があり、それらはいくつかの Mercurial コマンドでデフォルトのパスとして使用されます。

- *default* - ブル元のデフォルトのパス。普通はクローンしたときのパス。
- *default-push* - コマンドラインを使用したときのデフォルトのプッシュ先のパス。

ブル後の処理: ブル操作が正常に完了するとそのまま続けて実行されます。*update* はコマンドラインの **pull --update**、*fetch* は *fetch* 拡張機能、*rebase* は **pull --rebase** と同等の操作です。デフォルト: なし。

リモートリポジトリパス このパネルでは頻繁に利用するリポジトリのパスとそのエイリアスを設定できます。クローンを実行すると Mercurial は自動的にそのリポジトリを *default* というエイリアスで登録します。ここで設定したパスはすべて同期ツールのコンボボックスに表示され、コマンドラインから使用する場合でもショートカットとして利用できます。

### 5.1.5 サーバ

リポジトリ名: ウェブブラウザを使ったりリポジトリ表示で使用する名前。デフォルト: 作業ディレクトリ名。

説明: リポジトリの目的または内容に関する説明文。

連絡先: このリポジトリの管理者の名前またはメールアドレス。

テーマ: 使用するテンプレート名。

アーカイブ形式: ダウンロード可能にするアーカイブ形式をカンマ区切りで指定。

ポート番号: サーバの待ち受けポート番号。

SSL プッシュ: パスワード傍受などを防ぐため、プッシュ操作に SSL 通信の使用を必須とするかどうか。

ストライプ: 複数行を出力するとき、何行ごとに色を変えるか。デフォルト: 1 (0 でストライプ無効)。

最大ファイル表示件数: チェンジセットごとに表示するファイルの最大件数。

最大チェンジセット表示件数: チェンジログ一覧に表示するチェンジセットの最大件数。

プッシュ許可リスト: リポジトリへのプッシュを許可するかどうか。空欄または未設定の場合、プッシュは許可されません。特別な設定値 "\*" の場合は認証されていないユーザも含め、誰でもプッシュ可能になります。特定のユーザのみ許可したい場合はユーザ名をカンマ区切り (空白文字でも可) で指定してください。それによってユーザ認証が必要になります。ここで指定した許可ユーザー一覧は拒否ユーザー一覧の後に適用されます。

プッシュ拒否リスト:

リポジトリへのプッシュを拒否するかどうか。空欄または未設定の場合、プッシュは拒否されません。特別な設定値 "\*" の場合はすべてのユーザがプッシュを拒否されます。特定のユーザのみ拒否したい場合はユー

ザ名をカンマ区切り (空白文字でも可) で指定してください。ここで指定した拒否ユーザー一覧は許可ユーザー一覧の前に適用されます。

文字エンコーディング: 文字エンコーディング名。

### 5.1.6 プロキシ

ホスト名: プロキシサーバのホスト名とポート番号。例えば `myproxy:8000` など。

除外ホスト: 必要であればプロキシサーバを経由しないホスト名をカンマ区切りで指定。

ユーザ名: 必要であればプロキシサーバ認証用のユーザ名を指定。

パスワード: 必要であればプロキシサーバ認証用のパスワードを指定。

### 5.1.7 メール送信

送信者: SMTP の "From" ヘッダに使用する送信元メールアドレス。

受取人: 送信先のメールアドレスをカンマ区切りで指定。

Cc: CC のメールアドレスをカンマ区切りで指定。

Bcc: BCC のメールアドレスをカンマ区切りで指定。

メール送信手段: 必要ならメールを送信する手段を指定。デフォルトの "smtp" は SMTP (以下の項目で設定可能) を用いてメールを送信します。もしくは `sendmail` (`-f` オプションで送信者、引数として受取人一覧、メール本文は標準入力) のようなメール送信プログラムを直接指定することもできます。通常は `sendmail` または `/usr/sbin/sendmail` と指定するだけで送信できます。

SMTP ホスト名: メールサーバのホスト名。

SMTP ポート番号: メールサーバへの接続ポート番号。デフォルト: 25。

SMTP TLS 設定: TLS を使用してメールサーバに接続します。デフォルト: False。

SMTP ユーザ名: メールサーバ認証用のユーザ名。

SMTP パスワード: メールサーバ認証用のパスワード。

ローカルホスト名: 送信者自身をメールサーバに識別させるためのホスト名。

### 5.1.8 差分

パッチ改行文字: パッチファイルの改行文字を LF または CR+LF に正規化します。Strict の場合は正規化されません。デフォルト: Strict。

**Git 拡張形式:** GIT の拡張差分ヘッダ形式を使用します。デフォルト: False。

**日付を非表示:** 差分ヘッダに日付情報を付加しないようにします。デフォルト: False。

**関数名の表示:** どの関数に変更されたのかを表示。デフォルト: False。

**空白文字の無視:** 行比較の際に空白文字の有無を無視します。デフォルト: False。

**空白文字の数の違いを無視:** 空白文字の数の違いを無視します。デフォルト: False。

**空行を無視:** 比較時に空行の有無を無視します。デフォルト: False。

**色付け方法:** 差分表示の色付け方法を指定。foreground は文字自体に色を付け、background は背景色を変更する。  
none にすることで差分の色付けを無効にできます。デフォルト: foreground。

### 5.1.9 フォント設定

**テーマ標準フォント** 現在の GTK テーマのフォント設定を使用します。

**推奨フォント:** 各言語・環境向けに最適なフォント設定を選びます。

**個別設定フォント:** 使用するフォント名およびサイズをその使用目的ごとに個別に設定します。

このグループに含まれるドロップダウンコンボは 個別設定フォント: ラジオボタンを選択したときに有効化されます。

**コミットメッセージ:** チェンジセットの詳細表示およびコミットダイアログのログ入力エリアで使用するフォント。デフォルトは monospace 10 です。

**差分表示:** チェンジセットの詳細表示およびコミットダイアログの差分表示で使用するフォント。デフォルトは monospace 10 です。

**ファイル一覧:** コミットダイアログのファイル一覧で使用するフォント。デフォルトは sans 9 です。

**コマンド出力:** コマンド出力ウィンドウで使用するフォント。デフォルトは monospace 10 です。

## 5.2 今回のリリースで削除された設定項目

**ハッシュ値のコピー:** チェンジログビューアでそのとき選択しているチェンジセットのチェンジセット ID をクリップボードにコピーできるようにします。廃止予定。デフォルト: False。

チェンジセットのハッシュ値をクリップボードにコピーするには、チェンジセットの右クリックメニューから ハッシュのコピー を使用してください。

## 5.3 キーボードショートカット

Ctrl-Enter 変更を保存してダイアログを閉じます。これは OK ボタンを押したときと同じ動作になります。

## 5.4 コマンドラインからの実行

設定ダイアログはコマンドラインから起動することもできます。リポジトリ設定 (.hg/hgrc file) の場合は :

```
hgtk repoconfig
```

ユーザ設定 (Mercurial.ini file) の場合は :

```
hgtk userconfig
```

コマンド構文はこんなにシンプルで共通オプションを除けばコマンドラインオプションを指定する必要ありません。

## 第 6 章

# リカバリ

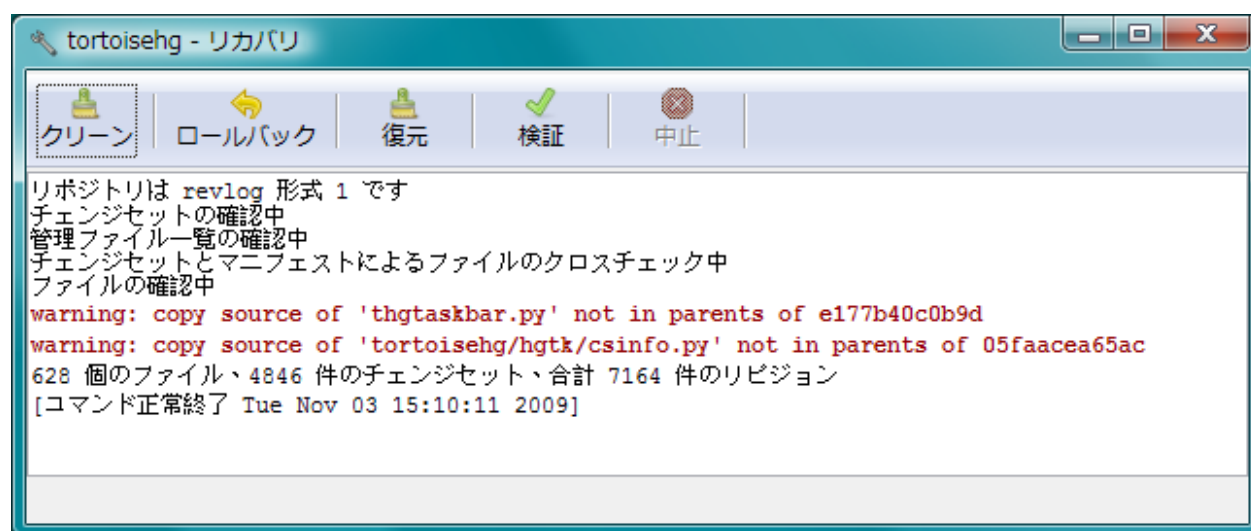


図 1 リカバリダイアログ

ツールバーのボタンは Mercurial コマンドそのものになっています：

**クリーン** `hg update --clean` - 現在の親リビジョンそのものを作業ディレクトリにクリーンな状態でチェックアウトします。マージ作業を途中で中止した場合や部分的にしか完了していないマージ結果を取り消すことができます。ただしこれは作業ディレクトリにおける変更を破棄することになるため十分注意してください。そのためシェルフ機能を使って変更を保管してからこのコマンドを使用すべきです。

**ロールバック** `hg rollback` - コミットやブルなど、リポジトリに対する最後の操作を取り消します。ただし、ロールバックの結果どのような操作が取り消されるのかを事前に知ることはできないため、最後に何をしたのか分かっている状況でのみ使用してください。

**復元** `hg recover` - トランザクションの中断などで破損したリポジトリを復元します。このコマンドは滅多に使いませんが、する必要のあるときは Mercurial が教えてくれます。

**検証** `hg verify` - リポジトリの整合性チェックをします。この操作はリポジトリ内容の変更を一切伴わないため安全です。





## 第 7 章

# パッチ

### 7.1 パッチとは

以下のリンクはパッチの本質と歴史を理解し、Mercurial ではどのように使われているのかを理解するのによい資料です。

- [パッチ管理問題](#)
- [パッチを理解する](#)
- [さらにパッチについて](#)

### 7.2 落とし穴

標準的なパッチ形式はバイナリファイルや名前変更、コピー、パーミッションの変更を記述できません。そのためまず GIT 形式のパッチを有効にする必要があります：

```
[diff]
git=True
```

Mercurial 1.5 ではこの振る舞いが改善され、GIT 形式の差分が必要な場合はユーザに警告を出すか、自動的に GIT 形式に変更します。詳しくは hgrc マニュアルの [差分](#)のセクションを参照してください。

Mercurial のパッチ処理は改行文字に一貫性のないソースファイルとパッチをうまく取り扱うことができないため、その解決方法として `patch.eol` 設定がバージョン 1.3 で導入されました：

```
[patch]
eol = auto #strict, lf, or crlf
```

hgcol 拡張機能もこの領域の問題の改善に取り組んでいます。おそらく Mercurial 1.5 で解決します。詳しくは 詳しくは hgrc マニュアルの [パッチ](#)のセクションを参照してください。

実のところパッチの適用というのはそれほど簡単な作業ではありません。もしソースファイルがパッチ作成時から枝分かれしてしまっていたら、パッチ適用時に衝突が起こり、`.rej` という拡張子のファイルが生成されます。TortoiseHg 1.0 からこれらの除外されたハンクをうまく適用してくれる実験的な `hg tk mpatch` コマンドが導入されました。このコマンドは Chris Mason による `mpatch` ツールが元になっています。この `mpatch` でも適用できなかったハンクは手動で適用するしかありません。

## 7.3 パッチのエクスポート

### 7.3.1 チェンジセット

チェンジセットをパッチとして保存するには、チェンジログビューアでチェンジセットの右クリックメニューからエクスポート → パッチの作成 を選択してください。ファイル保存ダイアログが表示されます。

### 7.3.2 リビジョン範囲

チェンジログビューアでリビジョンの範囲を選択してください。始点 (base) となるチェンジセットを左クリックで選択し、次に終点 (target) のチェンジセットの上で右クリックすると、このリビジョン範囲に対するコマンドを含んだ右クリックメニューが表示され、パッチの生成・バンドルの保存、メール送信、そして選択範囲のチェンジセットの変更点を積み重ねた差分を見ることができます。

始点と終点の選択には制限がないため、この操作は非常に強力な機能です。

### 7.3.3 メール送信

単独のチェンジセットを送るにはチェンジログビューアでチェンジセットの右クリックメニューを使用してください: エクスポート → メールで送信... 。選択するとメール送信ダイアログが開きます。

リビジョン範囲をまとめて送る場合はリビジョン範囲の右クリックメニューからこの範囲をメールで送信... を選択してください。

最後に、選択中のリモートリポジトリにすべてのチェンジセットを送るために同期ツールバーのメール送信 ボタンを使うこともできます。

---

注釈: パッチをメールで送信するには事前に `SMTP` の設定が必要です。

---

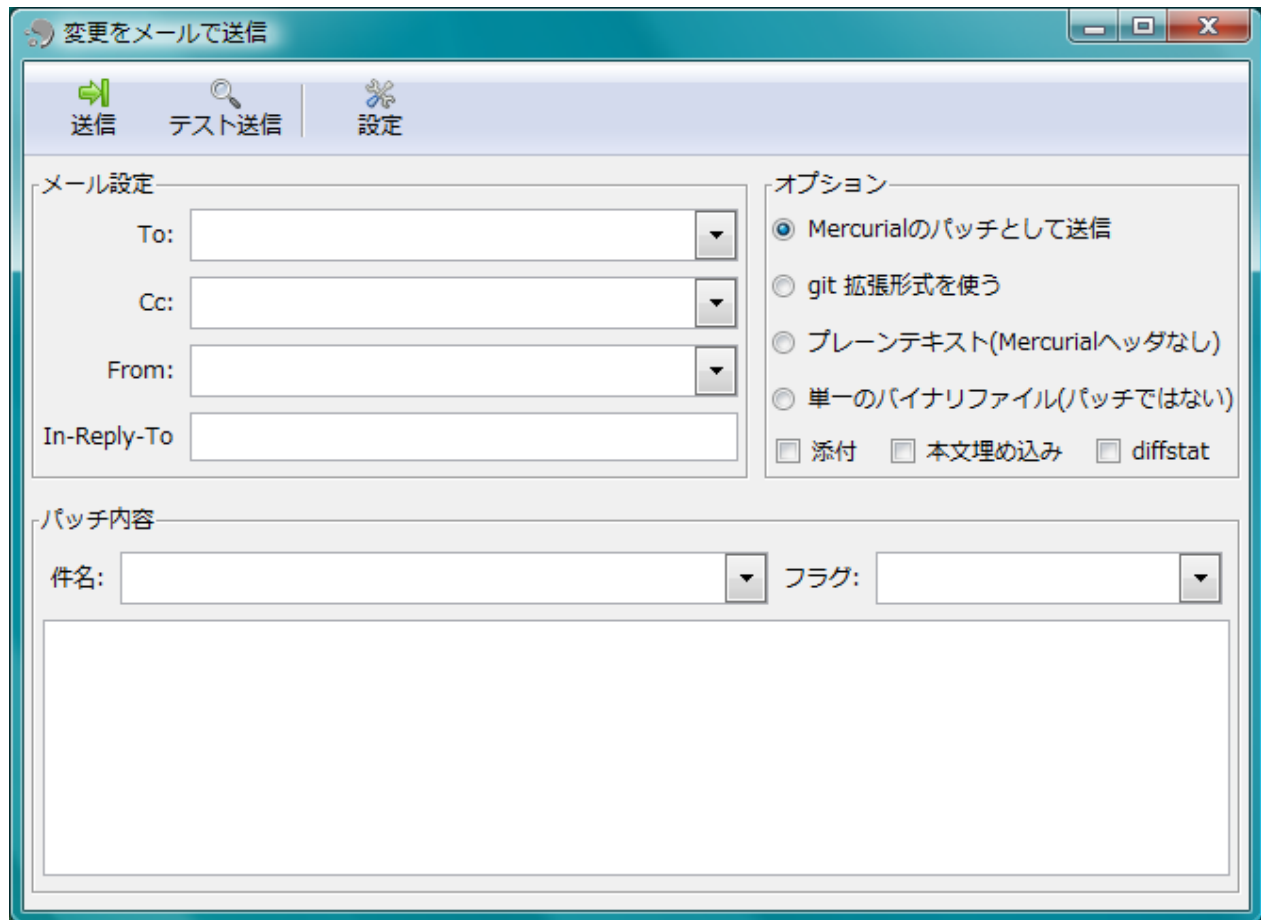


図 1 メール送信ダイアログ

### 7.3.4 チェンジセットのつまみ食い

チェンジログビューアでリビジョン範囲の右クリックメニューの選択中のリビジョンとの差分を使用してください。始点から終点までのすべてのリビジョンにおける変更点を積み重ねた差分がステータスダイアログに表示されます。

ステータスダイアログではコミットツールと同じようにファイルおよびハンク選択機能を使うことができ、最終的な結果はプレビュータブで見ることができます。名前を付けて保存 ボタンを押せば選択されている変更点をパッチファイルとして保存できます。

より手軽な方法としてはハンク選択画面で必要なハンクをハイライト(選択ではない)してから CTRL-C すればクリップボードに差分をコピーできます。

---

注釈: 選択する順番を逆にすると、生成されるパッチの順番も逆になります。

---

## 7.4 パッチのインポート

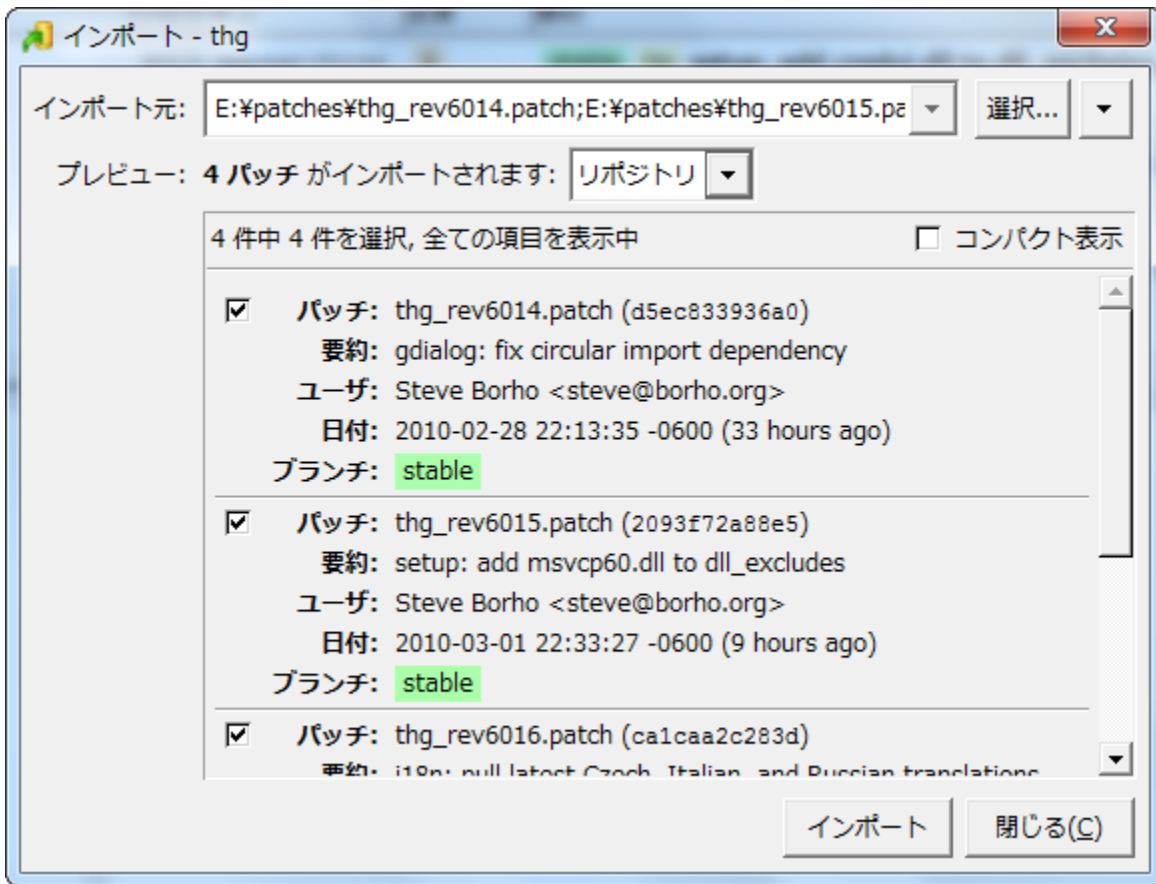


図2 インポートダイアログ

インポートダイアログはログビューアの同期バーまたはメニューから、もしくはコマンドラインから `hg tk import` で開くことができます。このダイアログはファイルマネージャからのファイルおよびディレクトリのドラッグ&ドロップ操作に対応しています。選択 ボタンの右隣にあるドロップダウンメニューには、パッチを含むディレクトリを選択するためのディレクトリの選択... メニューと、クリップボードから取り込むクリップボードからインポート メニューがあります。

リポジトリに直接インポートするか、MQ にインポートするか選択できます。

注釈: パッチのインポートを正しく行うには作業ディレクトリがクリーンでなければなりません。インポートする前に未コミットの変更点をコミット、元に戻す、またはシェルフしてください。

警告: コミットメッセージのないパッチをインポートすると Mercurial はエディタを起動して入力を促します。そのため `ui.editor` 設定値として GUI エディタが設定されている必要があります。

## 7.5 MQ (パッチキュー)

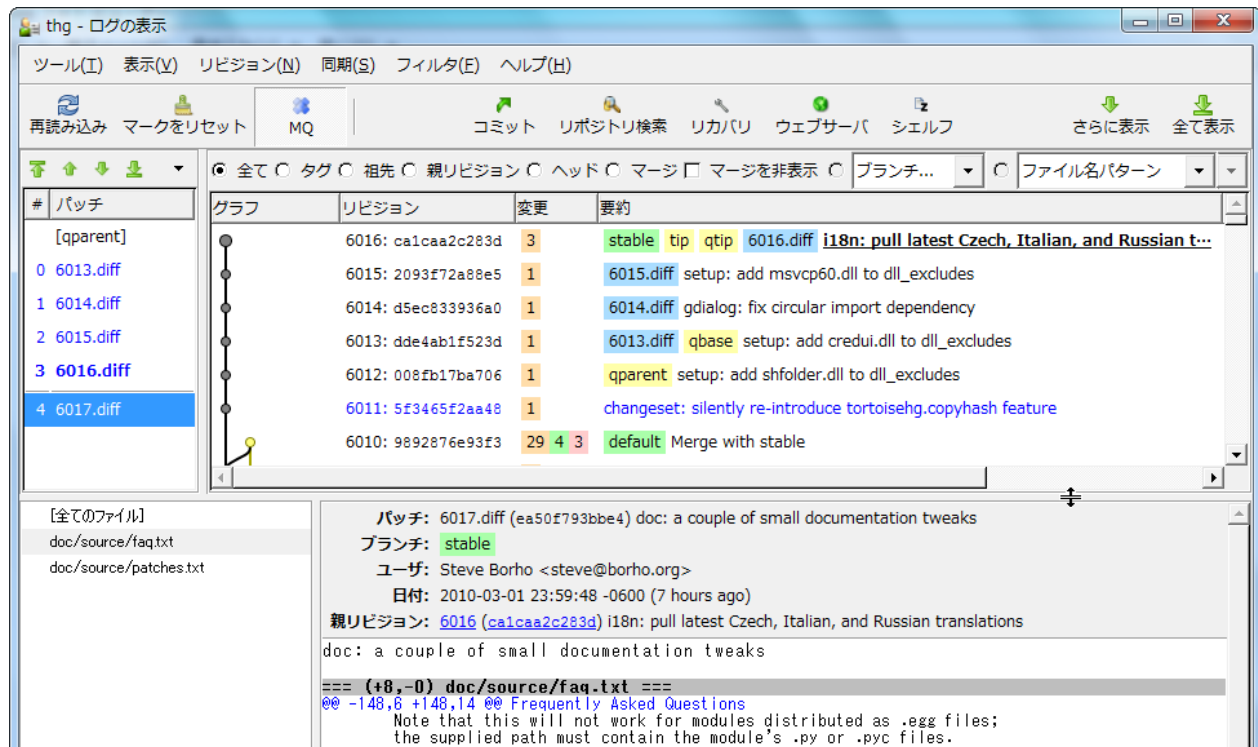


図3 ログビューアのMQパネル

リポジトリでMQ拡張機能が有効化されていると、ログビューアとコミットツールでMQパネルが利用可能になります。MQパネルではqpush、qpop、qrename、qfinishを含むほとんどのパッチ操作を行えます。MQパネルを使う前にMQ拡張機能について学んでおくことをお勧めします。



## 第 8 章

# 拡張機能

この章では Windows 向けの TortoiseHg バイナリパッケージに標準添付されている Mercurial の拡張機能について述べます。これらの拡張機能は TortoiseHg をより便利に使ってもらうためで、必要になったらすぐに簡単に有効にできます。

### 8.1 Hgfold

`hgfold` は VFAT および NTFS 上におけるファイル名の大文字小文字の違いによる名前衝突を回避したい Windows ユーザ向けの Mercurial 拡張機能です。

この拡張機能は以下の Mercurial コマンドにオプションを追加します。詳しいヘルプは `hg help <command>` で見ることができます：

```
up      - ファイル名の衝突を解決して特定のリビジョンへ更新
merge   - ファイル名の衝突を解決してマージ
```

#### インストール

この拡張機能をテストするためにコマンドラインから次のように実行します：

```
hg --config "extensions.fold=" status
```

今後も継続して使用したい場合は Mercurial.ini またはリポジトリの hgrc に次のように記述してください：

```
[extensions]
fold=
```

この設定をすればコマンドラインで `--config` 引数を指定する必要はありません。

#### 注意

通常のマージコマンドと同様、fold.py は作業ツリーの親を変更する必要があります。まだ実験的な拡張機能ですので十分注意して使用してください。

`hg recover` を実行後に何かエラーが発生した場合は `hg debugsetparents <number of tip revision>` を試してみてください。tip リビジョンの番号は `hg log -l 2` で調べることができます。

## 8.2 Hgcr-gui

### コードレビュー 管理ツール

- この拡張機能はプロジェクトのコードレビューを管理します。
- レビュー管理を Mercurial 内で完結させます。
- One can add files to the review, remove them and notify reviewr that files are ready for review.
- レビューアはコードに "完了" とマークして、開発者にメッセージを返します。
- プロジェクトマネージャはレビュー状況を確認できます - どのファイルがレビュー済みでどれがまだなのか。
- hgcr は自動的に最後にレビューした時点から変更されたファイルを見つけ出して通知します。
- この拡張機能は TortoiseHg の GUI を利用しますが、コマンドラインから使うことができます。
- コードレビューのデータベースはリポジトリのルートディレクトリに .code-review という名前のファイルとして保存されます。

使い方:

```
hg cr [OPTIONS] [FILES]

Code Review Plugin (requires Mercurial 1.3.x and TortoiseHg 0.9)

options:

-c --complete  Mark CR as complete
-a --add       Add files to CR list
-r --remove    Remove files from CR list
-l --list      Print files in CR list

use "hg -v help cr" to show global options
```

---

注釈: オプションなしで実行すると GUI が起動します。

---

詳細説明



- I've implemented the review around files and not changesets, because at the end, I want to be able to tell for the specific project if all the files went through code review process or not - the project status.
- おそらくあなたは多くの開発者と管理しなければならない複数のプロジェクトを持っていて、開発者のコードをレビューするグループがあります。
- 開発者のように変更点を追いつけるのは非常に難しいことですが、どのファイルが(レビューによって)レビュー済みでどれがまだなのかを知るのは簡単です。
- この拡張機能を使えば開発者は作業が終わった段階でレビューが必要なファイルにマークを付けて、レビューに知らせることができます。
- コードレビューのデータベースにチェンジセットが保管されているので、レビューはチェンジセットをひらいてきてコードレビューを始めます (開発者のコードに注釈を付けられる)。
- そしてコードレビューが終わり次第レビューは "レビュー完了" のマークを付けて開発者に知らせます。
- プロジェクトマネージャはいつでもプロジェクトがどうなっているのか確認できます。

#### インストール

Mercurial.ini またはリポジトリの hgcr に次のように記述してください：

```
[extensions]
hgcr-gui=
```

## 8.3 Perfance

#### ウェブサイト

この拡張機能については [こちら](#) で解説しています。

## 8.4 HGEOL

hgeol は将来 win32text 拡張機能に取って代わる拡張機能で、より完全で確実な方法で改行文字問題の解決を試みます。まだ未確定な部分も多く、頻繁に内容が書き変わることが予想されるため、ここに詳細な説明を書くかわりにリンクを張っておきます。

- [EolExtension](#)

## 8.5 Mercurial-Keyring

- [Mercurial Keyring](#) ウェブサイト

- [Keyring 拡張機能 Wiki](#)

keyring 拡張機能は安全に認証パスワード (HTTP/HTTPS and SMTP) をシステム固有のパスワードデータベース (Gnome Keyring, KDE KWallet, OSXKeyChain, Win32, コマンドライン) に保存するために keyring ライブラリのサービスを使います。

### 何をするのか

この拡張機能は初回のリモートリポジトリとのプル/プッシュ操作時に HTTP パスワードの入力を促し、それをパスワードデータベースに保存します (ユーザ名とリポジトリ URL の組み合わせのキーで)。それ以後は .hg/hgrc のユーザ名を確認して適切なパスワードをパスワードデータベースから取り出して使用します。

同様に認証を必要とする SMTP 経由のメール送信についても初回のみ入力を要求し、次回からはパスワードデータベースのパスワードを再利用します。

パスワード認証に失敗した場合は再び入力を求めます。

### インストール

まずは拡張機能を Mercurial.ini ファイルで有効にしてください：

```
[extensions]
mercurial_keyring=
```

### バックエンドの設定

使用しているシステムに最適なバックエンドは自動的に選択されますが、必要に応じて ~/keyringrc.cfg ファイル (ユーザのホームディレクトリの keyringrc.cfg ファイル) を設定することで指定可能です。設定に関する詳細は [keyring ドキュメント](#) を参照してください。

---

注釈: Windows XP 以降の Windows では、暗号化されたパスワードは [CredRead](#) と [CredWrite](#) を用いて、Credentials Subsystem に保管されます。

---

---

注釈: Windows 2000 では、暗号化されたパスワードはシステムレジストリ HKCU\Software\Mercurial\Keyring 以下に保管されます。

---

### リポジトリの設定 (HTTP)

リポジトリローカルの .hg/hgrc ファイルにリモートリポジトリの URL とユーザ名を記述して保存します。パスワードは keyring から提供されるためここでは記述しないでください：

```
[paths]
myremote = https://my.server.com/hgrepo/someproject
```

(次のページに続く)

(前のページからの続き)

```
[auth]
myremote.schemes = http https
myremote.prefix = my.server.com/hgrepo
myremote.username = mekk
```

次のようにリポジトリ URL にユーザ名を含める形式を使用すれば、より簡潔に記述できます：

```
[paths]
bitbucket = https://User@bitbucket.org/User/project_name/
```

注釈: .hg/hgrc ファイル内にユーザ名とパスワードの両方を記述した場合、keyring 拡張機能はパスワードデータベースを使用せず、記述してあるパスワードを使用します。また、ユーザ名が記述されていない場合は認証が必要になるたびにユーザ名とパスワードの入力を要求しますが、それらはパスワードデータベースに保管されません。つまりこれら両方の状況は keyring 拡張機能を使用していない、Mercurial 標準の動作になるということです。

Mercurial の認証に関するより詳しい説明はマニュアルの [\[auth\]](#) セクションにあります。

### リポジトリの設定 (SMTP)

リポジトリローカルの .hg/hgrc ファイル、またはホームディレクトリの hgrc ファイル (メール送信の認証情報はリポジトリ共通で使えるので通常は後者) に SMTP パスワード以外の SMTP メール設定を記述してください。例えば：

```
[email]
method = smtp
from = Joe Doe <Joe.Doe@remote.com>

[smtp]
host = smtp.gmail.com
port = 587
username = JoeDoe@gmail.com
tls = true
```

前述の HTTP の設定と同様に、ユーザ名だけを設定してパスワードは設定しません。パスワードを記述した場合の動作は Mercurial 標準の振る舞いになります。

### 使い方

上記設定を終えたら、あとは普通にプルやプッシュ (またはメール送信) をするだけです。(ユーザ名とリポジトリ URL の組み合わせで) 初回のみパスワードを要求されます。



## 第 9 章

# 他のバージョン管理システムとの相互運用

この章では他のバージョン管理システムと相互運用するために使用される 3 つの Mercurial 拡張機能について述べます。リポジトリの変換も参照してください。

### 9.1 Perforce (Perforce)

- Perforce ウェブサイト
- Perforce ユーザのための Mercurial

この拡張機能は Perforce サーバ上の名前付き Perforce クライアント仕様に基いて

```
p4://p4server[:port]/clientname
```

のようリモートリポジトリパスを持つリポジトリにおける同期処理を変更します。

この拡張機能を使用する前にサーバ上にクライアント仕様を用意しておく必要があります。

Making changes to the client specification Views causes problems when synchronizing the repositories, and should be avoided.

5 つの Mercurial 標準コマンドが上書きされます。

outgoing:

送信先リポジトリのパスが p4:// で始まっている場合、ローカルリポジトリにあって Perforce リポジトリにないリビジョンのファイルを表示します。

push:

プッシュ先リポジトリのパスが p4:// で始まっている場合、変更点をローカルリポジトリから Perforce リポジトリにエクスポートします。

(次のページに続く)

(前のページからの続き)

リビジョンを指定せずに実行したときは、最後のプッシュした時点からのすべてのリビジョンが送信されます。どちらの場合も単一の *Perforce* 変更リストにまとめられた状態でプッシュされます。

結果の変更リストを *Perforce* サーバにサブミットするかどうかは `--submit` オプションを使用するか、 `**perforce.submit**` 設定を `True` にしてください。

`**perforce.keep**` が `False` のとき、ファイルのサブミットに成功すると、*Perforce* 作業エリアが削除されます。

pull:

ブル元リポジトリのパスが `p4://` で始まっている場合、*Perforce* リポジトリから変更点をインポートし、`hg push` で自動的にサブミット済みの変更リストとマージされます。 `**perforce.keep**` 設定が `False` の場合は *Perforce* 作業エリアからファイルが削除され、`True` のときは *Perforce* 作業エリアは新しいファイルで更新されます。

incoming:

受信元リポジトリのパスが `p4://` で始まっている場合、*Perforce* リポジトリにあって、まだローカルにない変更点を表示します。

clone:

クローン元リポジトリのパスが `p4://` で始まっている場合、クローン先のリポジトリを作成し、すべての変更点を *Perforce* リポジトリからプルします。

## TortoiseHg との統合

*perforce* 拡張機能が有効化されると、クローンダイアログに開始リビジョン オプション設定が追加され、チェンジログビューアのメニューバーには *Perforce* メニューが追加されます：

identity:

*Perforce* 変更リストの `tip` リビジョンがローカルリポジトリにおけるどのリビジョンなのかを探して、そのリビジョンを選択します。

pending:

*Perforce* クライアントにプッシュしているが、まだサブミットしていない、または取り止めた保留中の *Perforce* 変更リストを検出します。

実行すると保留中の変更リストが表示され、それらをサブミットするか取り止めるか決めることができます。

## インストール

拡張機能を `Mercurial.ini` ファイルまたはリポジトリの `hgrc` ファイルで有効にしてください：

```
[extensions]
perforce=
```

## 9.2 hgsubversion (SVN)

- [hgsubversion ウェブサイト](#)
- [hgsubversion 拡張機能 Wiki](#)
- [Subversion リポジトリを使う](#)

hgsubversion はその名前の通り、Mercurial を Subversion サーバのクライアントとして使用するための拡張機能です。また Subversion リポジトリを Mercurial のリポジトリに変換するのにも使えます。

### インストール

TortoiseHg の Windows インストーラは hgsubversion が必要とする Python 用の SVN ライブラリを同梱しているため、以下の hgsubversion リポジトリをローカルにクローンするだけで使えます：

```
hg clone https://bitbucket.org/durin42/hgsubversion/ C:\hgsvn
```

そして次のようにしてクローンしたリポジトリの中を指すようにして hgsubversion 拡張機能を Mercurial.ini で有効にします：

```
[extensions]
hgsubversion = C:\hgsvn\hgsubversion
```

正しくインストールできたかどうかは **hg help hgsubversion** をコマンドラインから実行して確認してください。

より詳細な説明については hgsubversion の Wiki をご覧ください。

**警告：** Subversion サーバからクローンする際は、最初のいくつかのリビジョンだけをクローンしてから、残りのリビジョンをプルすることをお勧めします。クローンに失敗すると不完全なクローンリポジトリ全体が削除されてしまうためです。その点、プル操作はより寛容です。

### TortoiseHg におけるサポート

Subversion リポジトリからインポートしたチェンジセットには、Subversion におけるバージョン番号がログビューアに表示されます。

## 9.3 hg-git (git)

- [hg-git ウェブサイト](#)
- [hg-git 拡張機能 Wiki](#)
- [Git ユーザのための Mercurial](#)

hg-git はその名前の通り、Mercurial を git サーバのクライアントとして使用するための拡張機能です。また git リポジトリを Mercurial のリポジトリに変換するのにも使えます。

### インストール

TortoiseHg の Windows インストーラは hg-git が必要とする Python 用の git ライブラリ (dulwich) を同梱しているため、以下の hg-git リポジトリをローカルにクローンするだけで使えます：

```
hg clone https://bitbucket.org/durin42/hg-git/ C:\hg-git
```

Mercurial.ini ファイルで hggit と bookmarks 拡張機能を有効にしてください：

```
[extensions]
bookmarks =
hggit = C:\hg-git\hggit
```

正しくインストールできたかどうかは **hg help hggit** をコマンドラインから実行して確認してください。

より詳細な説明については hg-git の Wiki をご覧ください。

'incoming' コマンドは git リポジトリとのやり取りがうまくいっておらず、'outgoing' コマンドはあまり有用な情報を表示しません。他のバージョン管理システムと相互運用する拡張機能では珍しくないことですが、実質的に使用できるのはプルとプッシュ操作だけになります。



## 第 10 章

# よくある質問

*TortoiseHg* って何？

Mercurial 分散構成管理システムのための Windows シェル拡張です。Subversion における TortoiseSVN や、CVS における TortoiseCVS のようなクライアントアプリケーションです。TortoiseHg は他のプラットフォーム向けにコマンドラインから起動するための hgtk スクリプトも含んでいます。

Windows 向けの *TortoiseHg* インストーラには何が含まれていますか？

Mercurial, kdiff3, TortoisePlink, に加えて 5 つの拡張機能: hgfold, hgcr-gui, perforce, hgeol, mercurial-keyring を同梱しています。また hgsubversion 拡張機能およびリポジトリ変換拡張機能のために python-svn、hg-git 拡張機能のために dulwich も含めています。詳細は `extension-versions.txt` をご覧ください。

Windows 上の Mercurial はインデックスサービスやアンチウイルスソフトウェアと競合しませんか？

競合する可能性があります。TortoiseSVN と同様、リポジトリとその作業コピーのディレクトリではインデックスサービスを無効にし、アンチウイルスソフトウェアのスキャン対象外に指定してください。

エクスプローラの右クリックメニューは翻訳されないのですか？

右クリックメニューの翻訳ファイルは TortoiseHg のインストールディレクトリ `C:\Program Files\TortoiseHg\i18n\cmenu` にあります。使用したい言語の Windows レジストリファイルをダブルクリックして、表示される確認ダイアログの指示に従ってインストールしてください。

*TortoiseHg* ダイアログのツールバー表示を設定できますか？

TortoiseHg のダイアログは PyGTK のアプリケーションなので、`C:\Program Files\TortoiseHg\gtk\etc\gtk-2.0\gtkrc` にある `gtkrc` ファイルを変更することで設定可能です：

```
gtk-toolbar-icon-size = GTK_ICON_SIZE_LARGE_TOOLBAR
# Pick an icon size from:
# GTK_ICON_SIZE_MENU
# GTK_ICON_SIZE_SMALL_TOOLBAR
```

(次のページに続く)

(前のページからの続き)

```
# GTK_ICON_SIZE_LARGE_TOOLBAR    (default)
# GTK_ICON_SIZE_BUTTON
# GTK_ICON_SIZE_DND
# GTK_ICON_SIZE_DIALOG
gtk-toolbar-style = GTK_TOOLBAR_BOTH
# Pick a toolbar style from:
# GTK_TOOLBAR_ICONS
# GTK_TOOLBAR_TEXT
# GTK_TOOLBAR_BOTH                (default)
# GTK_TOOLBAR_BOTH_HORIZ
```

これらの設定は TortoiseHg 全体に適用されます。

マージや任意のリビジョンへの更新はどうやるの？

チェンジログツールからマージや特定リビジョンへの更新が行えます。エクスプローラの右クリックメニューの ログの表示から起動してください。

TortoiseHg の *shelve* 拡張機能をコマンドラインから使うにはどうすればいいですか？

Mercurial.ini ファイルで以下のように拡張機能を有効化してください：

```
[extensions]
tortoisehg.util.hgshelve=
```

SSH サーバに接続できない (TortoisePlink.exe - "...cannot execute specified..." というエラーメッセージ)

SSH 接続 ページを参照してください。

～差分表示ツールを使うにはどうすればいいの？

バージョン 1.0 から TortoiseHg はよく使われている GUI 差分表示ツールを自動的に検出し、それらを設定画面で選択可能な状態にしてくれます。

シェル拡張を無効にする方法は？ (アイコンオーバーレイと右クリックメニュー)

TortoiseHg インストーラを使って、シェル拡張機能を削除してください。

TortoiseHg はどこに設定を保存しているの？

TortoiseHg は 2 種類の設定システムを利用しています。

1. Mercurial の設定システムで、3 つのレイヤがあります
  1. グローバル設定：  
%ProgramFiles%\TortoiseHg にある Mercurial.ini ファイル
  2. ユーザ設定：  
%UserProfile% にある Mercurial.ini ファイル
  3. リポジトリ設定：*repo-root.hg* にある *hgrc* ファイル
2. ダイアログの位置や状態などを保存している %APPDATA%\Tortoisehg

以下は Mercurial の設定システムに保存される設定項目です :

```
[tortoisehg]
vdiff = vdiff
editor = gvim
tabwidth = 4
longsummary = True
graphlimit = 500
authorcolor = True
authorcolor.steve = blue
```

フォントの変更はできますか？

一部可能です。TortoiseHg の各種ダイアログ (コミット、変更点の表示、シェルフ) のベースとなっている gtools はフォント設定が可能です :

```
[gtools]
# font used in changeset viewer and commit log text
fontcomment = courier 10
# font used for diffs in status and commit tools
fontdiff = courier 10
# font used in command output window
fontlog = courier 10
```

GTK のテーマを変更するにはどうすればいいの？

テーマをダウンロードしてそれを `gtk\share\themes` にコピーし、`gtk\etc\gtk-2.0\gtkrc` ファイルで変更してください。

Windows の TortoiseHg はどこ Python モジュールを参照しているの？

TortoiseHg は Python の実行に必要な環境を DLL の形で同梱しています。標準ライブラリはすべて `C:\Program Files\TortoiseHg` の `library.zip` ファイルに格納されています。

そのため、非標準の Python モジュールに依存した Mercurial 拡張機能を使用するとそのモジュールが見つからないため読み込みに失敗します。例えば ReviewBoard 拡張機能は Python の標準ライブラリに含まれていない `simplejson` モジュールをインポートします。

そのような拡張機能を動作させるにはちょっとした変更が必要です。拡張機能のソースコードを開いて非標準のモジュールインポートの前にそのモジュールのインストール場所を追記します :

```
import sys
sys.path.append(r'C:\path\to\module')
```

この方法はモジュールが `.egg` ファイルとして配布されている場合はうまくいきません。指定するパスは `.py` または `.pyc` ファイルを含んでいる必要があります。

もしいくつかの拡張機能またはフックスクリプトを使用していて、同じ Python パッケージを共有したい場

合は明示的に sys.path を変更する拡張機能を作ることができます。最初に読み込まれるような単純な名前 (つまりアルファベット順で) にしてください。例えば :

```
[extensions]
00setSysPath = C:\path\to\setsyspath.py
```

ダイアログのぎこちないフォント描画を直すにはどうすればいいの ?

バージョン 0.8 以降では複数の GTK+ テーマを同梱せず、デフォルトのテーマとして MS-Windows のみを同梱しています。その結果、MS-Windows テーマが原因でフォント描画がおかしくなる場合があります。これを直すには以下の設定を TortoiseHg の `gtkrc` ファイルに追加してください :

```
style "msw-default"
{
    font_name = "MS UI Gothic 9"
}
```

`gtkrc` ファイルは TortoiseHg のインストールディレクトリにあります。C:\Program Files\TortoiseHg\gtk\etc\gtk-2.0\gtkrc

## 第 11 章

# TortoiseHg のデバッグ

### 11.1 ダイアログ

標準エラー出力はプログラムの終了時点でバッファに補足されます。何らかの深刻なエラーが標準エラー出力で見つかった場合、ユーザがバグを報告できるようにそのすべての内容がバグレポートツールに送信されます。出力されていないエラーがあるかもしれない場合は環境変数 **THGDEBUG** を設定して標準エラー出力のバッファリングを無効化してください。

多少 Python の経験があれば：

```
hgtk --debugger <command>
```

を使うこともできます。

hgtk によるダイアログ用の別プロセス生成を無効にするには環境変数 **THG\_HGTK\_SPAWN** を設定するか、コマンドラインから **'-nofork'** オプションを使って起動してください。

#### 11.1.1 Windows

例えばチェンジログビューアをデバッグする場合、コマンドプロンプト (**cmd.exe**) を起動してリポジトリに移動し、次のコマンドを実行してください：

```
set THGDEBUG=1  
hgtk log
```

#### 11.1.2 Linux または MacOSX

例えばチェンジログビューアをデバッグする場合、ターミナルでリポジトリに移動し、次のコマンドを実行してください：

```
export THGDEBUG=1
hgtk log
```

## 11.2 シェル拡張

デバッグ方法はプラットフォームに依存します。

### 11.2.1 Windows エクスプローラ

[https://msdn.microsoft.com/en-us/library/cc144064\(VS.85\).aspx](https://msdn.microsoft.com/en-us/library/cc144064(VS.85).aspx) に Windows シェル拡張を実行・デバッグするための情報が掲載されています。

SysInternals による **DbgView** ツールはシェル拡張からのデバッグ情報を補足することができます。

**ThgTaskbar** アプリケーションの設定ダイアログにはエラーログが表示されるタブがあります。このダイアログを開いた状態でエクスプローラを使ってリポジトリを閲覧すると各種デバッグメッセージを取得することができます。

### 11.2.2 Nautilus ファイルマネージャ

環境変数 `DEBUG_THG` を設定して、

- 個別のプロセスをテストします：

```
DEBUG_THG=Ne TMPDIR=/tmp/anydir/ --no-desktop nautilus [path]
```

- メインインスタンスをテストします：

```
nautilus -q
DEBUG_THG=NOe nautilus
```

- 常にデバッグ状態で起動したい場合は環境変数 `DEBUG_THG` をセッション開始と共に読み込まれる設定ファイル (`~/.profile`, `~/.xprofile` など) に設定してください。

Upper case characters in `DEBUG_THG` specify modules. Only *O* and *N* for *OverlayCache* and *Nautilus*, respectively, are supported module names. Lower case characters imply parts. Only *e* is supported, implying *error* messages.

Nautilus を再起動は次のどちらかの方法で可能です：

- 1) すべての Nautilus プロセスを削除する (セッションが自動的に Nautilus を再起動し、標準入出力先は `~/.xsession-errors`)
- 2) コマンドラインから **nautilus -q; nautilus** を実行 (標準入出力先はコンソール)

## 第 12 章

# 索引

- `genindex`
- `modindex`
- `search`





# Python モジュール索引

## C

changelog.dialog, 42  
changelog.settings, 68  
clone.dialog, 28  
commit.dialog, 30  
commit.settings, 67  
common.dialog, 19

## d

datamine.dialog, 52  
debugging, 95  
diff.settings, 70

## e

email.settings, 70  
explorer, 23  
extensions, 81

## f

font.settings, 71

## g

guess.dialog, 61

## i

ignore.dialog, 62  
init.dialog, 26  
introduction, 7

## n

nonhg, 87

## p

patches, 75  
preface, 3  
proxy.settings, 70

## r

recovery.dialog, 73

## s

serve.dialog, 59  
settings.dialog, 65  
shelve.dialog, 38  
synchronize.dialog, 55  
synchronize.settings, 68

## t

TortoiseHg.settings, 66  
tour, 11

## w

web.settings, 69



# 索引

changelog.dialog (モジュール), 42  
changelog.settings (モジュール), 68  
clone.dialog (モジュール), 28  
commit.dialog (モジュール), 30  
commit.settings (モジュール), 67  
common.dialog (モジュール), 19

datamine.dialog (モジュール), 52  
debugging (モジュール), 95  
diff.settings (モジュール), 70

email.settings (モジュール), 70  
explorer (モジュール), 23  
extensions (モジュール), 81

font.settings (モジュール), 71

guess.dialog (モジュール), 61

ignore.dialog (モジュール), 62  
init.dialog (モジュール), 26  
introduction (モジュール), 7

nonhg (モジュール), 87

patches (モジュール), 75  
preface (モジュール), 3  
proxy.settings (モジュール), 70

recovery.dialog (モジュール), 73

serve.dialog (モジュール), 59  
settings.dialog (モジュール), 65  
shelve.dialog (モジュール), 38  
synchronize.dialog (モジュール), 55  
synchronize.settings (モジュール), 68

TortoiseHg.settings (モジュール), 66  
tour (モジュール), 11

web.settings (モジュール), 69