
PERFORCE 2008.1

コマンド・リファレンス

2008年7月

This manual copyright 1999–2008 PERFORCE Software.

All rights reserved.

PERFORCE software and documentation is available from <http://www.perforce.com>. You may download and use PERFORCE programs, but you may not sell or redistribute them. You may download, print, copy, edit, and redistribute the documentation, but you may not sell it, or sell any documentation derived from it. You may not modify or attempt to reverse engineer the programs.

PERFORCE programs and documents are available from our Web site as is. No warranty or support is provided. Warranties and support, along with higher capacity servers, are sold by PERFORCE Software.

PERFORCE Software assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

By downloading and using our programs and documents you agree to these terms.

PERFORCE and *Inter-File Branching* are trademarks of PERFORCE Software. PERFORCE software includes software developed by the University of California, Berkeley and its contributors.

All other brands or product names are trademarks or registered trademarks of their respective companies or organizations.

© copyright 1999–2008 PERFORCE Software.

All rights reserved.

PERFORCE のソフトウェアおよび関連文書は <http://www.perforce.com> より入手できます。プログラムは、ダウンロードしてご利用になれますが、販売または再配布することは禁じます。関連文書は、ダウンロード、印刷、コピー、編集、再配布することを認めますが、販売することは禁じます。また、いかなるものであれ、本書を元にして作成した文書を販売することも禁じます。プログラムについては、変更を加えること、またリバース・エンジニアリングを試みることも禁じます。

当社 Web サイトより入手した PERFORCE プログラムおよび関連文書は無条件受け取りとなります。保証もサポートもいたしません。保証、サポートは、より高機能のサーバとともに、PERFORCE Software より有償で提供いたします。

PERFORCE Software は、本書中の誤りまたは不正確な記述について、いっさい責任も負担も負いません。当社のプログラムおよび関連文書をダウンロードして使用すると、以上の条件に同意なされたことになります。

PERFORCE、*Inter-File Branching* は、PERFORCE Software の商標です。PERFORCE のソフトウェアには、カリフォルニア大学バークレイ校およびその協力者によって開発されたソフトウェアが含まれています。

その他のブランドまたは製品名は、それぞれ当該各社または団体の商標または登録商標です。

本書について	7
p4 add.....	9
p4 admin	12
p4 annotate.....	14
p4 branch	16
p4 branches	18
p4 change	19
p4 changes.....	22
p4 changelist	25
p4 changelists.....	26
p4 client	27
p4 clients.....	34
p4 counter	35
p4 counters.....	37
p4 delete	38
p4 depot.....	40
p4 depots	43
p4 describe	44
p4 diff	46
p4 diff2.....	49
p4 dirs	52
p4 edit	54
p4 filelog	56
p4 files	58
p4 fix	60
p4 fixes	62
p4 flush	64
p4 fstat	67
p4 group.....	71
p4 groups	74
p4 have	76
p4 help.....	78
p4 info	79
p4 integrate	80
p4 integrated.....	84
p4 job	86
p4 jobs	88
p4 jobspec	93
p4 label	96
p4 labels.....	98
p4 labelsync	100
p4 license	102
p4 lock	103
p4 logger	104
p4 login	105
p4 logout	107
p4 monitor	108
p4 obliterate	110
p4 opened.....	112
p4 passwd.....	114
p4 print	116
p4 protect	118
p4 protects	127

p4 rename	128
p4 reopen	129
p4 resolve	131
p4 resolved	136
p4 revert	137
p4 review	139
p4 reviews	141
p4 set	143
p4 sizes	146
p4 submit	148
p4 sync	152
p4 tag	155
p4 tickets	157
p4 triggers	158
p4 typemap	165
p4 unlock	168
p4 user	169
p4 users	173
p4 verify	174
p4 where	175
p4 workspace	177
p4 workspaces	178
環境変数とレジストリ変数	179
P4AUDIT	180
P4CHARSET	181
P4COMMANDCHARSET	182
P4CLIENT	183
P4CONFIG	184
P4DEBUG	185
P4DIFF	186
P4DIFFUNICODE	187
P4EDITOR	188
P4HOST	189
P4JOURNAL	190
P4LANGUAGE	191
P4LOG	192
P4MERGE	193
P4MERGEUNICODE	194
P4PAGER	195
P4PASSWD	196
P4PCACHE	197
P4PFSIZE	198
P4OPTIONS	199
P4PORT	200
P4ROOT	201
P4TARGET	202
P4TICKETS	203
P4USER	204
PWD	205
TMP, TEMP	206

追加情報	207
グローバル・オプション	208
ファイル指定	211
ビュー	215
ファイルタイプ	218
索引	224

本書について

書名

本書は『PERFORCE 2008.1 コマンド・リファレンス』です。

内容

本書には、PERFORCE のすべてのコマンドと環境変数が記載されています。本書は、UNIX 形式の入力画面を使用して学習したいユーザや、PERFORCE の基本をすでに理解しており、個々のコマンドに関する情報を迅速に探し出す必要のあるユーザを対象としています。

以下の表では、機能ごとのコマンド・リファレンスのインデックスを示します。

機能	参照するページ
ヘルプ	p4 help, p4 info, ファイル指定, ビュー, グローバル・オプション, ファイルタイプ
クライアント・ワークスペース	p4 client, p4 clients, p4 flush, p4 have, p4 sync, p4 where, p4 workspace, p4 workspaces
ファイル	p4 add, p4 delete, p4 diff, p4 diff2, p4 dirs, p4 edit, p4 files, p4 fstat, p4 lock, p4 print, p4 rename, p4 revert, p4 sizes, p4 unlock
チェンジリスト	p4 change, p4 changelist, p4 changes, p4 changelists, p4 describe, p4 filelog, p4 opened, p4 reopen, p4 review, p4 submit
ジョブ	p4 fix, p4 fixes, p4 job, p4 jobs, p4 jobspec
ブランチとマージ	p4 branch, p4 branches, p4 integrate, p4 integrated, p4 label, p4 labels, p4 labelsync, p4 tag, p4 resolve, p4 resolved
管理	p4 admin, p4 counter, p4 counters, p4 depot, p4 depots, p4 license, p4 logger, p4 monitor, p4 obliterate, p4 reviews, p4 triggers, p4 typemap, p4 verify
セキュリティ	p4 group, p4 groups, p4 login, p4 logout, p4 passwd, p4 protect, p4 protects, p4 tickets, p4 user, p4 users
環境	p4 set, 環境変数とレジストリ変数, P4AUDIT, P4CHARSET, P4COMMANDCHARSET, P4CLIENT, P4CONFIG, P4DEBUG, P4DIFF, P4DIFFUNICODE, P4EDITOR, P4HOST, P4JOURNAL, P4LANGUAGE, P4LOG, P4MERGE, P4MERGEUNICODE, P4PAGER, P4PASSWD, P4PCACHE, P4PFSIZE, P4POPTIONS, P4PORT, P4ROOT, P4TARGET, P4TICKETS, P4USER, PWD, TMP, TEMP

PERFORCE の基礎となっているコンセプトを学びたい場合や、例やコマンドの具体的な使用方法の説明が多く盛り込まれている解説書をお望みの場合は、当社の Web サイト (<http://www.perforce.com>.) から入手できる『P4 ユーザ・ガイド』をお読みください。

本書に代わるもの

本書は PDF 形式および HTML 形式のファイルでもご覧になれます。

ご利用にあたって

本書の PDF ファイル版および HTML ファイル版は、広範囲に相互参照が行われています。オンラインで PDF ファイル版をご覧になっているときは、コマンド名をクリックするだけで、そのコマンドの解説ページに移ることができます。

こういう情報も含めた方がよいと思われることがありましたら、ご意見・ご希望を manual@perforce.com までお送りください。

p4 add

概要

ディポに追加するファイルをクライアント・ワークスペースで作業状態にします。

構文

```
p4 [g-opts] add [-c changelist#] [-f -n] [-t type] file...
```

解説

p4 add は、ディポに追加するファイルをクライアント・ワークスペースで作業状態にします。指定されたファイルはチェンジリストにリンクされます。ファイルは、チェンジリストが p4 submit でサーバに送られるまでは、実際にはディポに追加されません。追加するファイルは、ディポに存在しないファイルか、ディポに存在していても最新リビジョンで削除済みとしてマークされているファイルでなければなりません。

p4 add でファイルを作業状態にするには、そのファイルがクライアント・ワークスペースのビューに存在する必要がありますが、p4 add の実行時点でワークスペースに存在する必要はありません。ただし、p4 submit を実行するときには、ワークスペースに存在する必要があります。ファイルがワークスペースに存在しない場合、サブミットは失敗します。p4 add は、ワークスペース内のファイルの作成や上書きを行いません。ファイルが存在しない場合は、ユーザが作成しなければなりません。

デフォルトでは、指定されたファイルはデフォルトのチェンジリストにリンクされます。別のチェンジリストを指定するには、-c を使用します。

ファイルを追加するときには、PERFORCE はまずタイプマップ・テーブル (p4 typemap) を調べ、追加しようとしているファイルのファイルタイプをシステム管理者が定義しているかどうかを確かめます。合致するものが見つければ、そのファイルタイプがタイプマップ・テーブルの定義に従って設定されます。合致するものが 見つからない場合、PERFORCE は該当ファイルの先頭の 8192 バイトを調べ、それが text か binary かを判断した上、ファイルをディポに保存します。テキスト・ファイルのリビジョンはリバースデルタ形式で保存され、バイナリ・ファイルのリビジョンは完全なかたちで保存されます。

ファイルタイプを明示的に指定して、タイプマップ・テーブルと PERFORCE のデフォルト・ファイルタイプ検出機構の両方をオーバーライドするには、-t filetype オプションを使用します。

@、#、*、および % の文字が含まれたファイルを追加するには、-f オプションを使用します。このオプションを指定すると、通常は PERFORCE でワイルドカードとして使用される文字が強制的に文字解釈されます。

オプション

<code>-c changelist</code>	作業状態にして追加するファイルのリンク先 <u>チェンジリスト</u> を指定します。このオプションが付いていなければ、ファイルはデフォルトのチェンジリストにリンクされます。
<code>-t filetype</code>	追加するファイルの <u>ファイルタイプ</u> を指定します。 PERFORCE のファイルタイプについては、「ファイルタイプ」のセクションにあるリストをご覧ください。
<code>-f</code>	<code>-f</code> オプションを使用すると、ワイルドカードがファイル名に強制的に含められます。詳細については、「ファイル指定」をご覧ください。
<code>-n</code>	ファイルやメタデータを実際に変更せずに、追加目的で作業状態にするファイルをプレビューします。
<code>g_opts</code>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	open

- `p4 add` のファイル指定に含めた ワイルドカード は、PERFORCE サーバではなく、ローカルのオペレーティング・システムにより展開されます。例えば、`p4 add` にはワイルドカード「...」は使用できません。
- PERFORCE では、空のディポにファイルを追加する操作と、すでに他のファイルが含まれるディポにファイルを追加する操作に違いはありません。新しい空のディポにファイルを追加する場合も、`p4 add` を使用して、クライアント・ワークスペースのファイルを追加します。
- `p4 add -f` では特殊文字の ASCII 拡張は使用しないでください。ファイル `status@june.txt` を追加する場合は、次のように指定します。

```
p4 add -f status@june.txt
```

手動で `@` 記号を拡張して、ファイル `status%40june.txt` を追加しようとする、`%` 記号が文字解釈されて 16 進コード `%25` に拡張され、ファイル名は `status%2540june.txt` となります。

使用例

<code>p4 add -t binary file.pdf</code>	新しいファイルに特定のファイルタイプを割り当て、タイプマップ・テーブルの設定をオーバーライドします。
<code>p4 add -c 13 *</code>	ユーザのカレント・ディレクトリのすべてのファイルをディポに追加するために作業状態にします。これらのファイルはチェンジリスト 13 にリンクされます。
<code>p4 add README ~/src/*.c</code>	ユーザのディレクトリ <code>~/src</code> にあるすべての <code>*.c</code> ファイルをディポに追加するために作業状態にします。また、ユーザの現在の作業ディレクトリにある <code>README</code> ファイルもディポに追加するため、作業状態にします。これらのファイルはデフォルトのチェンジリストにリンクされます。
<code>p4 add -f *.c</code>	<code>*.c</code> という名前のファイルを追加のための作業状態にします。 このファイルをビューまたは他の PERFORCE コマンドで参照するために、その後はアスタリスクの代わりに 16 進拡張 <code>%2A</code> を使用する必要があります。詳細については、213 ページの「ファイル名およびエンティティに関する文字の制限」をご覧ください。

関連コマンド

編集するファイルを作業状態にする。	p4 edit
削除するファイルを作業状態にする。	p4 delete
すべての作業状態のファイルをディポにコピーする。	p4 submit
ディポからクライアント・ワークスペースへファイルを読み込む。	p4 sync
新しいチェンジリストを作成または編集する。	p4 change
すべての作業状態のファイルをリストする。	p4 opened
ファイルを非作業状態に戻す。	p4 revert
作業状態のファイルを別の作業中チェンジリストに移動する。	p4 reopen
作業状態のファイルのファイルタイプを変更する。	p4 reopen -t filetype

p4 admin

概要

サーバに対する管理操作を 実行します。

構文

```
p4 [g-opts] admin checkpoint [ -z ] [ prefix ]
p4 [g-opts] admin dbstat [ -h ] [ -a | dbtable... ]
p4 [g-opts] admin journal [ -z ] [ prefix ]
p4 [g-opts] admin lockstat
p4 [g-opts] admin stop
p4 [g-opts] admin updatespecdepot [ -a | -s type ]
```

解説

PERFORCE スーパー・ユーザは、現在作業中のマシンが PERFORCE サーバが稼動しているマシンとは異なる場合でも、p4 admin コマンドを使って管理タスクを実行することができます。

サーバを停止するには、p4 admin stop を使用します。このコマンドは、サーバの再起動時、データベースが一貫した状態にあるようにするためにデータベースをロックし、その上で PERFORCE のバックグラウンド・プロセスを停止します。(Windows において PERFORCE をサーバとして動作させている場合も、サービスとして動作させている場合も、同様に機能します。)

チェックポイントを作成するには、p4 admin checkpoint [prefix] を使用します。サーバ・マシンにログインし、p4d -jc [prefix] でチェックポイントを作成するのと同じことです。チェックポイントが作成され、ジャーナルが連番を付けたファイルにコピーされます。prefix を指定すれば、これらのファイルはそれぞれ prefix.ckp.n、prefix.jnl.n-1 という名前で保存されます。ここで、n は連番です。prefix の一部に任意のディレクトリを指定することにより、そのディレクトリにチェックポイントおよびジャーナルを格納することができます。(循環ジャーナルは現在のジャーナルの格納ディレクトリに関わらず、P4ROOT ディレクトリに格納されます。) "prefix" の指定がなければ、デフォルトのファイル名 checkpoint.n と journal.n-1 が使用されます。

p4 admin journal コマンドは、p4d -jj と実行するのと同様です。詳しくは、『システム管理者ガイド』をご覧ください。これらのファイルは、PERFORCE サーバ起動時の指定に従い、サーバ・ルートに作成されます。

p4 admin lockstat コマンドでは、現在読み取りまたは書き込みの処理のためにロックされているデータベース・テーブルが報告されます。

p4 admin dbstat コマンドは、PERFORCE サーバのデータベースの内部状態に関する統計情報を表示します。dbtable はお使いのサーバのルート・ディレクトリにある db.* ファイルに対応します。このコマンドは通常、PERFORCE テクニカル・サポートと共同でデータベースの順次スキャンによるディスクシークを測定する目的で使用されます。

p4 admin updatespecdepot コマンドを実行すると、PERFORCE サーバが保管フォーム (具体的には、client、depot、branch、label、typemap、group、user、および job の各フォーム) をスペック・ディポにアーカイブします。-a オプションを使用すると、すべての仕様タイプがアーカイブされます。-s オプションを使用すると、特定の type を持つフォームのみがアーカイブされます。まだアーカイブされていないフォームのみが作成されます。

オプション

-z	p4 admin checkpoint および p4 admin journal に付けると、チェックポイントとジャーナル・ファイルが拡張子 .gz を付けて圧縮 (gzip) フォーマットで保存されます。
-h	p4 admin dbstat に付けると、リーフ・ページ間の距離を示すヒストグラムが表示されます。
-a	p4 admin dbstat に付けると、すべてのテーブルの統計値が表示されます。 p4 admin updatespecdepot に付けると、スペック・ディポを現在のすべてのフォームで更新します。
dbtable	p4 admin dbstat に付けると、特定のテーブル (db.have、db.user など) の統計値だけを表示します。
-s type	p4 admin updatespecdepot に付けると、特定タイプのフォームを持つスペック・ディポを更新します。タイプは client、depot、branch、label、typemap、group、user、job のうち 1 つです。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	super

- p4 admin stop を実行すると PERFORCE サーバが停止するので、クライアントとサーバとの接続が予期せず切断されたというエラー・メッセージが表示される場合がありますが、このメッセージは無視してかまいません。
- p4 admin dbstat を実行するとテーブルのスキャン中にデータベースへの書き込みアクセスをブロックするため、このコマンドの使用には注意が必要です。ほとんどの場合、このコマンドは PERFORCE テクニカル・サポートと共に作業する際に使用します。
- p4 admin updatespecdepot を実行する前に、スペック・ディポが存在している必要があります。
- PERFORCE の管理について、詳しくは『システム管理者ガイド』をご覧ください。

使用例

p4 admin stop	PERFORCE サーバを停止します。
p4 admin checkpoint	checkpoint.n という名前のチェックポイントを作成し、かつ journal という名前の新しいジャーナルを開始します。以前のジャーナルの内容は journal.n-1 (n は連番) にコピーされます。
p4 admin checkpoint name	name.ckp.n という名前のチェックポイントを作成し、かつ journal という名前の新しいジャーナルを開始します。以前のジャーナルの内容は name.jn1.n-1 (n は連番) にコピーされます。

p4 annotate

概要

ファイルの行を、リビジョン番号とともに表示します。

構文

```
p4 [g-opts] annotate [ -a -c -i -q -dflag] file[revRange] ...
```

解説

p4 annotate は、ファイルのリビジョン（またはリビジョン範囲）に対して、各行が生成されたリビジョン番号を表示します。表示されたリビジョン番号を p4 filelog で調べれば、どのチェンジで誰が、いつ、なぜ変更したのかがわかります。

ファイルのその行が生成されたときのチェンジリスト番号を表示するには、-c オプションを指定します。

パラメタにリビジョン番号を指定した場合は、そのリビジョン番号までの行が表示されます。リビジョン範囲を指定した場合は、その範囲に該当する行が表示されます。

デフォルトでは、出力されるファイルの最初の行はヘッダ行となります。

```
filename#rev - action change num (type)
```

ここで、*filename#rev* はファイル名とリビジョン指定子です。*action* はファイルが作業状態にされたときの操作で add、edit、delete、branch または integrate です。*num* はサブミットされたチェンジ番号であり、*type* はそのリビジョンのファイルタイプです。

ヘッダ行を抑制するには、-q (quiet) オプションを指定します。

すべての行（削除された行および（または）すでに最新リビジョンにない行）を表示するには、-a (all) オプションを指定します。

オプション

-a	すべての行を表示します。削除された行や、最新リビジョンではもはや存在しない行を含みます。 それぞれの行では、開始および終了リビジョンを含みます。
-c	各行を、リビジョン番号ではなく、チェンジリスト番号とともに表示します。 -a オプションと -c オプションをともに指定すると、それぞれの行に開始および終了チェンジリスト番号が含まれます。
-dflags	標準 UNIX の diff オプション・セットの 1 つを使って比較ルーチンを実行します。下の使用例にあるフラグのリストをご覧ください。
-i	ブランチをまたがってファイル履歴を追跡します。ファイルがブランチ機能により作成された場合、PERFORCE はブランチ・ポイントに至るまでのリビジョンを含めます。-i オプションは -c オプションを暗黙的に指定します。
-q	抑制モードです。それぞれのファイルに対して、ヘッダ行の表示を抑制します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	read

- p4 annotate の出力は、スクリプト作成などの自動化処理のフォームに適しています。
- p4 annotate でサポートされている差分フラグを以下に示します。

フラグ	名前
-db	空白についての変更を無視
-dw	すべての空白を無視

使用例

<code>p4 annotate file.c</code>	file.c のすべての行を表示し、先頭にはその行が記述されたリビジョン番号が示されます。
<code>p4 annotate -c file.c</code>	file.c のすべての行を表示し、先頭にはその行が記述されたチェンジリスト番号が示されます。
<code>p4 annotate -a file.c</code>	file.c のすべての行（削除された行も含む）を表示し、各行はリビジョン範囲で始まります。 リビジョン範囲は開始と終了のリビジョン番号を示します。
<code>P4 annotate -a -c file.c</code>	file.c のすべての行（削除された行も含む）を表示し、各行はチェンジリストの範囲で始まります。 ファイル内に存在する各行の開始および終了チェンジリスト番号を含みます。

p4 branch

概要

ブランチ仕様とそのビューの生成または編集を行います。

構文

```
p4 [g-opts] branch [ -f ] branchspec
p4 [g-opts] branch -o branchspec
p4 [g-opts] branch -d [ -f ] branchspec
p4 [g-opts] branch -i [ -f ]
```

解説

p4 branch により、p4 integrate で反映操作する 2 つのファイル・セット間のマッピングを作成することができます。ブランチ・ビューで、反映元のファイル (*fromFiles*) と反映先のファイル (*toFiles*) の間の関係を定義します。ディポ・シンタックスでビューの両側を指定します。

ブランチ仕様を生成して名前を付けた後、p4 integrate -b branchname と入力してファイルの反映操作を行います。ブランチ仕様は自動的にすべての *toFiles* をそれに対応する *fromFiles* にマッピングします。

p4 branch フォームを保存しても、ディポやクライアント・ワークスペースのファイルに直接的な影響はありません。ワークスペースにブランチされたファイルを作成し、チェンジリスト内でファイルを作業状態にするには、p4 integrate -b branchspecname を実行する必要があります。

フォーム・フィールド

フィールド名	タイプ	説明
Branch:	読み取り専用	コマンドラインで与えられたブランチ名。
Owner:	入力必須	ブランチ仕様の所有者。デフォルトではブランチを生成したユーザに設定されます。このフィールドは、[Option:] フィールドが locked になっていない限り重要ではありません。
Access:	読み取り専用	ブランチ仕様に最後にアクセスがあった日付。
Update:	読み取り専用	ブランチ仕様が最後に変更された日付。
Options:	入力必須	unlocked (デフォルト) または locked に設定します。 locked の場合には、[Owner:] だけがブランチ仕様を修正することができ、その仕様はこのフィールドが unlocked に設定されるまで削除することはできません。
Description:	任意入力	ブランチの目的を簡潔に説明するコメント。
View:	入力必須	ディポにある 1 つのファイル・セット (<u>反映元ファイル</u>) からディポにある別のファイル・セット (<u>反映先ファイル</u>) へのマッピングを記述します。ビューは、ディポの 1 つの場所から別の場所にマッピングします。クライアント・ワークスペースを参照することはできません。 例えば、ブランチ・ビュー //depot/main/... //depot/r2.1/... は //depot/main のすべてのファイルを //depot/r2.1 にマッピングします。

オプション

-d	名前を指定してブランチ仕様を削除します。ファイルは、この操作による影響を受けません。1つのコードラインから別のコードラインへの、保存されたマッピングだけが削除されます。通常、このオプションは該当するブランチ仕様を生成したユーザだけが使用できます。
-f	強制オプションです。PERFORCE 管理者が -d と組み合わせて使用すると、自分が所有者でないブランチを削除することができます。また、管理者がブランチ仕様の更新日を変更することも可能にします (-f を使用すると、[Update:] フィールドが書き込み可能になります)。
-i	エディタを立ち上げずに標準入力からブランチ仕様を読み取ります。
-o	エディタを立ち上げずに標準出力にブランチ仕様を出力します。
<code>g_opts</code>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	open

- ブランチ・ビューで、2つの関連するコードラインの関係を定義します。例えば、あるプロジェクトの開発ファイルが `//depot/project/dev/...` に保存されていて、そのプロジェクトのリリース 2.0 用の関連コードラインを `//depot/project/r2.0/...` に生成したいとすれば、ブランチ・ビューを次のように指定します。


```
//depot/project/dev/... //depot/project/r2.0/...
```

 ブランチ・ビューには、複数のマッピングを記述することができます。ビューの指定についてさらに詳しくは、「ビュー」のセクションをご覧ください。
- パスまたはファイル名に空白が含まれる場合は、パスを引用符で次のように囲みます。


```
//depot/project/dev/... "//depot/project/release 2.0/..."
```
- ブランチ・ビューは、`p4 diff2 -b branchname fromFiles` という構文により、`p4 diff2` でも利用することができます。これは、ブランチ・ビューで定義された `fromFiles` 対 `toFiles` のパターンに合致するファイルの差分を表示します。

関連コマンド

既存のブランチ仕様のリストを参照する。	<code>p4 branches</code>
1つのファイル・セットから別ファイル・セットへ変更をコピーする。	<code>p4 integrate</code>
2つのコードラインの間の違いを参照する。	<code>p4 diff2</code>

p4 branches

概要

既存のブランチ仕様のリストを出力します。

構文

```
p4 [g-opts] [ -u user ] [ -e namefilter -m max ] branches
```

解説

p4 branches は、システムが現在認識しているすべてのブランチ仕様のリストを出力します。ブランチ仕様の出力を max までに限定するには、-m max オプションを使用します。

namefilter というパターンに一致する名前を持つブランチに出力を限定するには、-e namefilter オプションを使用します。

指定されたユーザが所有しているブランチに出力を限定するには、-u user オプションを使用します。

オプション

-m max	max までのブランチ仕様のみを一覧表示します。
-e namefilter	namefilter に一致するブランチのみを一覧表示します。
-u user	user が所有しているブランチのみを一覧表示します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	list

関連コマンド

ブランチ仕様の生成または編集を行う。	p4 branch
--------------------	-----------

p4 change

概要

チェンジリスト仕様の生成または編集を行います。

構文

```
p4 [g-opts] change [ -s ] [ -f | -u ] [ changelist# ]
p4 [g-opts] change -d [ -f -s ] changelist#
p4 [g-opts] change -o [ -s ] [ changelist# ]
p4 [g-opts] change -i [ -s ] [ -f | -u ]
```

解説

p4 add、p4 delete、p4 edit、または p4 integrate でファイルを作業状態にすると、それらのファイルはチェンジリストに追加されます。ファイルの編集内容は、チェンジリストが p4 submit でディポへ送られるまで、ローカルのクライアント・ワークスペースに保存されます。デフォルトでは、ファイルはデフォルト・チェンジリスト内で作業状態になりますが、p4 change コマンドで複数のチェンジリストを生成、編集することができます。

p4 change は、環境変数またはレジストリ変数 P4EDITOR で定義されたエディタで編集または参照できるフォームを表示します。引数が与えられていなければ、このコマンドは新しい番号付チェンジリストを生成します。

チェンジリストには連番が割り当てられます。PERFORCE は、サブミット済チェンジリストの番号の並び順を日付時刻の並び順と同じにする目的で、サブミット時にチェンジリストの番号を振り直すことがあります。

作業中チェンジリストのコメントを編集するとき、またはサブミット済チェンジリストのフィールドを参照するときは、p4 change changelist を使用します。

p4 submit が失敗すると、デフォルトのチェンジリストに 1 つの番号が割り当てられます。以後、このチェンジリストを参照するには、番号を指定する必要があります。

p4 changelist は p4 change と同等に機能します。

フォーム・フィールド

フィールド名	タイプ	説明
Change:	読み取り専用	既存チェンジリストの編集時はその番号、新しいチェンジリストの生成時は new が表示されます。
Client:	読み取り専用	現在のクライアント・ワークスペースの名前。
Date:	読み取り専用	このチェンジリストが最後に変更された日付。
User:	読み取り専用	現在の PERFORCE ユーザの名前。
Status:	読み取り専用値	pending、submitted、new のいずれかが表示されます。ユーザによる編集はできません。チェンジリスト生成時が new、生成済みだが、まだ p4 submit でディポへサブミットしていないときが pending、すでに p4 submit でディポに内容が保存されている場合が submitted です。
Description:	書き込み可能、 入力必須	このチェンジリストに関するコメント。サブミット前に変更しなければなりません。

フィールド名	タイプ	説明
Jobs:	リスト	このチェンジリストで修正完了となるジョブのリスト。最初に表示されるジョブのリストは、p4 user のフォームの [JobView:] の設定によって制御されます。表示されるジョブは、削除または追加することができます。
Files:	リスト	このチェンジリストでサブミットされるファイルのリスト。表示されるファイルは削除することができ、デフォルトのチェンジリストに含まれているファイルを追加することもできます。

オプション

-d	指定のチェンジリストを削除します。通常はファイルが含まれていない作業中チェンジリストに対してのみ使用できますが、スーパー・ユーザはその他の場合でも -f オプションを付けてチェンジリストを削除することができます。
-f	強制オプションです。サブミット済みチェンジリストのコメントの編集を可能にします。サブミット済みチェンジリストを編集するには、admin もしくは super の権限が必要です。スーパーユーザおよび管理者は -f フラグを使用して読み取り専用フィールドを上書きすることができます。
-f -d	サブミット済みチェンジリストを強制的に削除します。PERFORCE 管理者またはスーパー・ユーザだけがこのコマンドを使うことができます。コマンド実行の前に、該当チェンジリスト内のすべてのファイルを p4 obliterate でシステムから削除しておく必要があります。
-o	チェンジリストの仕様を標準出力に書き出します。
-i	チェンジリストの仕様を標準入力から読み取ります。入力は p4 change のフォームで使用されているのと同じ形式でなければなりません。
-s	チェンジリストのサブミット時に、デフォルトの状態である closed ではなく任意の状態をジョブに割り当てられるようにします。ジョブを未変更のままにしておくには、特殊な状態である same を使用します。 新規のチェンジリストでは、修正完了の状態として ignore という特殊な状態が表示されます（状態が変更されなければ、ジョブはチェンジリストをサブミットしても修正完了になりません）。 このオプションは、p4 fix の -s と合わせて使用します。これは欠陥追跡システムと連携して使用することを目的としたオプションです。
-u	サブミット済みチェンジリストを更新します。[Jobs:] および [Description:] の両フィールドのみが更新可能であり、チェンジリストのサブミット実行者のみがこのチェンジリストを更新できます。-u と -f は相互に排他的です。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	open

- 論理的に異なる作業を処理するためにファイルを編集する場合、ユーザは複数のチェンジリストを生成する必要があります。例えば、同じバグを修正するためにファイル file1.c とファイル file2.c を編集し、一方、新しい機能を追加するためにファイル other.c を編集する場合は、file1.c と file2.c は同一のチェンジリストで作業状態にし、other.c は別のチェンジリストで作業状態にします。
- p4 change changelist# は、既存チェンジリストの仕様を編集するときに使用しますが、これを使用しても、そのチェンジリストにリンクしたファイルやジョブは表示されません。

特定のチェンジリストにリンクしたファイルのリストを見るには、`p4 opened -c changelist#`を使い、特定のチェンジリストにリンクしたジョブのリストを見るには、`p4 fixes -c changelist#`を使います。

- ファイルをあるチェンジリストから別のチェンジリストに移動するときには、`p4 reopen`を使うか、あるいは`p4 revert`を使ってすべての作業中チェンジリストから1つのファイルを削除します。

使用例

<code>p4 change</code>	新しいチェンジリストを生成します。
<code>p4 change -f 25</code>	サブミット済チェンジリスト 25 を編集します。管理者またはスーパー・ユーザのアクセス権限が必要です。
<code>p4 change -d 29</code>	チェンジリスト 29 を削除します。チェンジリスト 29 が作業中で、しかもファイルを全く含まない場合にのみ有効です。

関連コマンド

ディポへチェンジリストをサブミットする。	<code>p4 submit</code>
あるチェンジリストから別のチェンジリストへ1つのファイルを移動する。	<code>p4 reopen</code>
すべての作業中チェンジリストから1つのファイルを削除する。	<code>p4 revert</code>
特定の基準に合致するチェンジリストのリストを出力する。	<code>p4 changes</code>
作業状態のファイルのリストを出力する。	<code>p4 opened</code>
特定のチェンジリストにリンクされた修正完了のリストを出力する。	<code>p4 fixes</code>
特定のチェンジリストにジョブをリンクする。	<code>p4 fix</code>
特定のチェンジリストから1つのジョブを削除する。	<code>p4 fix -d</code>
チェンジリストに含まれるすべてのファイルのリストを出力する。	<code>p4 opened -c changelist#</code>
あるチェンジリスト内で変更されたファイルに関する詳細を取得する。	<code>p4 describe changelist#</code>

p4 changes

概要

サブミット済みおよび作業中チェンジリストのリストを出力します。

構文

```
p4 [g-opts] changes [-i -t -l -L -c client -m max -s status -u user]
[file[RevRange]...]
```

解説

p4 changes により、サブミット済みおよび作業中チェンジリストのリストを表示することができます。引数なしで p4 changes を使用すると、すべての番号付チェンジリストが表示されます。デフォルトのチェンジリストは表示されません。

デフォルトでは、各行の形式は次のとおりです。

```
Change num on date by user@client [status] description
```

-t オプションを使ってそれぞれのチェンジリストの時刻を表示すると、各行の形式は次のとおりです。

```
Change num on date hh:mm:ss by user@client [status] description
```

status (状態) の値は、そのチェンジリストが作業中の場合にのみ表示されます。description (コメント) は、オプションを指定しない場合、最初の 31 字に制限されます。-L オプションを指定すると最初の 250 文字、-l オプションを指定すると全テキストが表示されます。

ファイル・パターンを引数として与えれば、そのパターンに合致するファイルに影響を及ぼすサブミット済または作業中のチェンジリストのリストが表示されます。

ファイル・パターンには、リビジョン指定およびリビジョン範囲を含めることができます。リビジョン範囲を含めると、その範囲内のファイルに影響を及ぼすすべてのチェンジが表示されます。リビジョン指定子を 1 つ与えると、1 からその指定リビジョンまでのすべてのチェンジが表示されます。

-c client オプションおよび -u user オプションは、p4 changes の出力を、指定されたクライアント・ワークスペースまたはユーザによって作成されたチェンジリストに制限するために使用します。

-s status オプションは、p4 changes の出力を、指定された status 値 (pending または submitted) をとるチェンジリストに制限するために使用します。

オプションとファイル・パターンを組み合わせれば、表示されるチェンジリストをかなり絞ることができます。-m max オプションを使用すれば、出力対象を最新の max 個 のチェンジリストに制限することもできます。

p4 changelists は p4 changes と同等に機能します。

オプション

<code>-i</code>	指定ファイルに反映されたファイルに影響を及ぼしたチェンジリストを含め ます。
<code>-t</code>	各チェンジリストに対して、日付と同様に時刻も表示します。
<code>-l</code>	コマンドの出力に、各チェンジリストのコメントの全テキストを含めます。
<code>-L</code>	各チェンジリストのコメントを 250 文字に切り捨て、長い出力形式でリスト表示 します。
<code>-c client</code>	出力するチェンジリストを、指定されたクライアント・ワークスペースによって 作成されたものに限定します。
<code>-m max</code>	出力するチェンジリストを、最大番号を含め番号が大きい <code>max</code> 個（すなわち最新 の <code>max</code> 個の）のチェンジリストに限定します。
<code>-s status</code>	指定の状態（ <code>pending</code> または <code>submitted</code> ）のチェンジリストのみを出力します。
<code>-u user</code>	出力するチェンジリストを、指定されたユーザによって作成されたものに限定し ます。
<code>g_opts</code>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	<code>list</code>

使用例

<code>p4 changes -m 5 //depot/project/...</code>	ディレクトリ <code>project</code> 配下の ファイルを 1 つでも含むサブ ミット済チェンジリストのう ち、最新のもの 5 つを表示し ます。
<code>p4 changes -m 5 -c eds_elm</code>	クライアント・ワークスペー ス <code>eds_elm</code> からサブミットさ れたチェンジリストのうち、 最新のもの 5 つを表示します。
<code>p4 changes -m 5 -u edk</code>	ユーザ <code>edk</code> がサブミットした チェンジリストのうち、最新 のもの 5 つを表示します。
<code>p4 changes file.c@2000/05/01,2000/06/01</code>	クライアント・ビューを介し てディポへマッピングされ た、ファイル <code>file.c</code> を含む すべてのチェンジリストのう ち、日付が 2000 年 5 月のもの を表示します。
<code>p4 changes -m 1 -s submitted</code>	サブミット済チェンジリスト うち、最新のもの 1 つだけ を表示します。
<code>p4 changes @2001/04/01,@now</code>	2001 年 4 月 1 日から現在まで にサブミットされたチェンジ リストをすべて表示します。
<code>p4 changes @2001/04/01</code>	2001 年 4 月 1 日より前にサブ ミットされたチェンジリスト をすべて表示します。

関連コマンド

作業中チェンジリストをサブミットする。	p4 submit
新しい作業中チェンジリストを生成する。	p4 change
特定のチェンジリストに関する詳細なレポートを読み出す。	p4 describe

p4 changelist

概要

チェンジリスト仕様の生成または編集を行います。

構文

```
p4 [g-opts] changelist [-f -s] [changelist#]
p4 [g-opts] changelist -d [-f -s] [changelist#]
p4 [g-opts] changelist -o [-s] [changelist#]
p4 [g-opts] changelist -i [-f -s]
```

解説

p4 changelist は p4 change と同等に機能します。

p4 changelists

概要

サブミット済みおよび作業中チェンジリストのリストを出力します。

構文

```
p4 [g-opts] changelists [-i -t -l -L -c client -m max -s status -u  
user] [file[RevRange]....]  
p4 [g-opts] changelists [-i -t -l -L -c client -m max -s pending -u  
user]
```

解説

p4 changelists は p4 changes と同等に機能します。

p4 client

概要

クライアント・ワークスペース仕様およびそのビューの生成または編集を行います。

構文

```
p4 [g-opts] client [-f -t template] [clientname]
p4 [g-opts] client -o [-t template] [clientname]
p4 [g-opts] client -d [-f] clientname
p4 [g-opts] client -i [-f]
```

解説

PERFORCE クライアント・ワークスペースは、ユーザのマシン上にあるファイルの集合で、ディポ内にあるファイルの集合の一部に対応しています。p4 client コマンドは、クライアント・ワークスペース仕様の生成または編集に使用します。このコマンドを実行すると、PERFORCE がクライアント・ワークスペース管理のために必要とする情報の入力フォームが表示されます。

クライアント・ワークスペースのファイルとディポのファイルは、常に 1 対 1 のマッピングで関連づけられていますが、これらのファイルは、同じ相対的位置に保存されている必要もなければ、同じ名前を持っている必要もありません。p4 client フォームの [View:] (ビュー:) フィールドで指定される クライアント・ビュー が、クライアント・ワークスペースのファイルからディポへのマッピング経路、およびその逆のマッピング経路を指定します。

引数 *clientname* なしで実行すると、p4 client は、環境変数 P4CLIENT または同等の変数で指定されたクライアント・ワークスペースに対して動作します。ロックされたクライアントを引数 *clientname* に指定して実行すると、クライアント仕様が読み取り専用で表示されます。

p4 client が完了すると、新しく生成された、または変更されたクライアント・ワークスペース仕様が PERFORCE データベースに保存されます。この時点では、クライアント・ワークスペースのファイルは影響を受けません。新しいクライアント・ビューは、次に p4 sync が実行されて初めて有効になります。

p4 workspace は p4 client と同等に機能します。

フォーム・フィールド

フィールド名	タイプ	説明
Client:	読み取り専用	環境変数 P4CLIENT または同等の変数で指定されたクライアント・ワークスペース名。
Owner:	書き込み可能	クライアント・ワークスペースを所有するユーザの PERFORCE ユーザ名。デフォルトは、そのクライアント・ワークスペースを生成したユーザです。
Update:	読み取り専用	クライアント・ワークスペース仕様が最後に更新された日付。
Access:	読み取り専用	クライアント・ワークスペース仕様のいずれかの部分が最後に PERFORCE コマンドによってアクセスされた日付と時刻。

フィールド名	タイプ	説明
Host:	書き込み可能、任意入力	<p>クライアント・ワークスペースが常駐するホスト・マシンの名前。指定されていれば、このクライアント・ワークスペースは上記ホストからのみ操作できます。</p> <p>ホスト名は、そのホストから <code>p4 info</code> を実行したときの出力と正確に一致していなければなりません。</p> <p>このフィールドは、クライアント・ワークスペースが誤って別のホスト上で使用されるのを防ぐためにあります。ただし、ホスト名の実効値は <code>p4 コマンド -H オプション</code> または環境変数 <code>P4HOST</code> でオーバーライドすることができるので、このフィールドはセキュリティを保証するものではありません。これと同様の機能でありながらセキュリティを保証するものとしては、<code>p4 protect</code> の IP アドレス制限機能があります。</p>
Description:	書き込み可能、任意入力	<p>クライアント・ワークスペースに関するコメント。デフォルトは <code>Created by owner</code> です。</p>
Root:	書き込み可能、入力必須	<p>[View:] に記述されているすべてのファイルの相対位置の基準となる（ローカル・ホスト上の）ディレクトリ。デフォルトは現在の作業ディレクトリです。</p>
AltRoots:	書き込み可能、任意入力	<p>2 つまで任意で指定するクライアント・ルートの別名（代理）です。PERFORCE のクライアント・プログラムでは、最初にメインのクライアント・ルート、次に別名のクライアント・ルートを検査することによって、カレント・ワーキング・ディレクトリと合致させます。</p> <p>これによりユーザは、異なるディレクトリ名の変換をすることができ、複数プラットフォーム上で同じ PERFORCE クライアント仕様を使用することができます。</p> <p>クライアント・ルートのいずれかで Windows ディレクトリを使用している場合には、その Windows ディレクトリをメインのクライアント・ルートとして指定し、他のクライアント・ルートのディレクトリを <code>AltRoots:</code> フィールドに指定しなければなりません。</p> <p>例えば、複数プラットフォーム上で製品をビルドしているエンジニアは、Windows 上でのビルド用としてメインのクライアント・ルートを <code>C:\Projects\Build</code>、UNIX 上でのビルドに関する作業用として別名のルートを <code>/staff/userid/projects/build</code> と指定します。</p>
Options:	書き込み可能、入力必須	<p>クライアント・オプションを制御する 7 つのスイッチを指定します。オプションの内容については、後述の「使用上の留意点」をご覧ください。</p>

フィールド名	タイプ	説明
SubmitOptions :	書き込み可能、入力必須	<p>p4 submit のデフォルトの動作を制御するオプション。</p> <ul style="list-style-type: none"> • submitunchanged すべての作業状態のファイル (変更の有無にかかわらず) がディポにサブミットされます。これは PERFORCE のデフォルトの動作です。 • submitunchanged+reopen すべての作業状態のファイル (変更の有無にかかわらず) がディポにサブミットされ、すべてのファイルが自動的にデフォルト・チェンジリストで再度作業状態にされます。 • revertunchanged 内容またはファイルタイプが変更されたファイルのみがディポにサブミットされます。未変更のファイルは元に戻します。 • revertunchanged+reopen 内容またはファイルタイプが変更されたファイルのみがディポにサブミットされ、デフォルト・チェンジリストで再度作業状態にされます。未変更のファイルは元に戻しますが、デフォルト・チェンジリストでは作業状態にされません。 • leaveunchanged 内容またはファイルタイプが変更されたファイルのみがディポにサブミットされます。未変更のファイルはデフォルト・チェンジリストに移動されます。 • leaveunchanged+reopen 内容またはファイルタイプが変更されたファイルのみがディポにサブミットされます。未変更のファイルはデフォルト・チェンジリストに移動され、変更済みのファイルはデフォルト・チェンジリストで再度作業状態にされます。このオプションは submitunchanged+reopen と似ていますが、未変更のファイルはディポにサブミットされません。
LineEnd:	書き込み可能、入力必須	<p>復帰 / 改行 (CR/LF) の変換を制御する 4 つのスイッチを指定します。設定については、後述の「使用上の留意点」をご覧ください。</p>
View:	書き込み可能、複数行	<p>ディポのファイルとクライアント・ワークスペースのファイルとの間のマッピングを指定します。詳しくは、「ビュー」のセクションをご覧ください。</p>

オプション

-t <i>clientname</i>	<p>クライアント・ワークスペースの [View:] フィールドと [Option:] フィールドに、別のクライアント・ワークスペース <i>clientname</i> のビューとオプションをコピーします。(すなわち、<i>clientname</i> の [View:] を 1 つのテンプレートとして使用します。)</p>
-f	<p>通常は読み取り専用になっている最終更新日を設定可能にします。スーパー・ユーザはこのオプションを使用して他のユーザが所有しているロックされたワークスペースを削除または修正することができ、スーパーユーザ以外のユーザはこのオプションを使用して自分が所有しているロックされたワークスペースの削除または修正を行うことができます。</p>
-d <i>clientname</i>	<p>指定されたクライアント・ワークスペースを削除します。ただし、該当クライアントが、このコマンドを実行するユーザによって所有されているか、またはロックされていないことが条件となります。(-f オプションにより、PERFORCE スーパー・ユーザが自分の所有でないクライアント・ワークスペースを削除したり、スーパーユーザ以外のユーザが自分が所有しているロックされたワークスペースを削除または修正することができます。)</p>

-i	クライアントの仕様を標準入力から読み込みます。
-o	クライアント仕様を標準出力へ書き出します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	list

- スペースを含むファイルまたはディレクトリ名のディポ・サイドまたはクライアント・サイドのマッピングは引用符で囲みます。
- クライアント・ワークスペース名に含まれるスペースはアンダーバーと解釈されます。例えば、p4 client "my client" というコマンドを入力すると、my_client というクライアント・ワークスペースが生成されます。
- [Options:] (オプション:) フィールドには、6 つの設定項目がスペースで区切って表示されます。これらの設定項目には、それぞれ2つの設定値があり、その設定値と意味を次の表に示します。

設定値	意味	デフォルト
[no]allwrite	allwrite に設定すると、クライアント上の作業状態でないファイルが書き込み可能な状態になります。	noallwrite
[no]clobber	clobber に設定すると、p4 sync 実行時にクライアントで新たに同期したファイルと同じ名前を持つ、書き込み可能だが作業状態でないファイルが上書き (clobber) されます。	noclobber
[no]compress	compress に設定すると、クライアントとサーバの間のデータ・ストリームが圧縮されます。(クライアントとサーバのバージョンがともに 99.1 以降でない場合、この設定は無視されます。)	nocompress
[no]crlf	注: 2000.2 以前のバージョンのみ! Windows では、crlf に設定すると、ディポとクライアント・ワークスペースとの間でファイルがコピーされるときに、CR/LF 変換が自動的に実行されます。	crlf
[un]locked	他のユーザにクライアント仕様を編集するパーミッションを認めるか否かを設定します。(locked のクライアント仕様を効果的に使用するには、p4 passwd で該当クライアントの所有者のパスワードも設定しておく必要があります。) locked に設定されている場合は、所有者だけがクライアント仕様を使用、編集、または削除できます。PERFORCE 管理者は、p4 client に -f (強制) オプションを付けてそのロックをオーバーライドすることができます。	unlocked

設定値	意味	デフォルト
[no]modtime	<p>ファイルタイプ修飾子 +m (modtime) の <u>付いていない</u> ファイル:</p> <ul style="list-style-type: none"> バージョンが 99.2 以前の PERFORCE クライアントの場合、modtime に設定されていれば、新たに同期したファイルの (ローカルのファイルシステムにおける) 更新日は、そのファイルがディポへサブミットされたときの <u>サーバにおける</u> 日付と時刻になります。 バージョンが 2000.1 以降の PERFORCE クライアントの場合、modtime に設定されていれば、新たに同期したファイルの (ローカルのファイルシステムにおける) 更新日は、そのファイルが最後に更新されたときの <u>そのファイルにおける</u> 日付スタンプになります。 nomodtime に設定されていれば、PERFORCE クライアントのバージョンに関係なく、更新日は同期化の日付と時刻になります。 <p>ファイルタイプ修飾子 +m (modtime) の <u>付いている</u> ファイル:</p> <ul style="list-style-type: none"> バージョンが 99.2 以前の PERFORCE クライアントの場合、+m 修飾子は無視され、modtime 設定時と nomodtime 設定時の動作は上記のようになります。 バージョンが 2000.1 以降の PERFORCE クライアントの場合、新たに同期したファイルの (ローカルのファイルシステムにおける) 更新日は、クライアントにおける modtime/nomodtime の設定に <u>関係なく</u>、ファイルがディポへサブミットされたときの <u>ファイルにおける</u> 日付スタンプになります。 	nomodtime (すなわち、大半のファイルでは同期の日付と時刻) ファイルタイプ修飾子 +m が付いているファイルでは無視。
[no]rmdir	rmdir に設定して p4 sync を実行すると、クライアントにある空のディレクトリが削除されます。	normdir

- デフォルトでは、どのユーザも p4 client -c clientname ですべてのワークスペースを編集できます。そうした事態を防ぐには、[Options:] フィールドで locked を設定し、p4 passwd でクライアント・ワークスペース所有者のパスワードを生成します。
- compress オプションは、伝送するデータ量を減らして、低速リンクを介したクライアント / サーバ間通信を高速化します。高速リンクの場合には、圧縮プロセスそのものに、節約される伝送時間より多くの時間がかかることがあります。一般に、compress は T1 以下の伝送速度のときに設定すべきで、それ以外の場合は設定しない方がよいでしょう。距離による違いは様々です。
- [LineEnd:] (行末:) フィールドは、クライアント・ワークスペースのテキスト・ファイルに使用される行末文字を制御します。

注 [LineEnd:] フィールドは、PERFORCE 2002.1 に新たに追加されたものです。[Options:] フィールドで crlf または nocrlf を指定していた従来の方式に代わるものです。
LineEnd: win および Options: crlf という互いに矛盾した設定が行われている場合の動作は未定義です。

[LineEnd:] フィールドは、次の 5 種類の設定値のいずれかを受け付けます。

設定値	意味
local	クライアントのネイティブ・モード (デフォルト) を使用します。
unix	UNIX 形式 (および MAC OS X) の行末識別を LFで行います。
mac	OS X より前のバージョンの Macintosh の行末識別を CRで行います。

設定値	意味
win	CR、LF で Windows 形式の行末識別を行います。
share	<p>share オプションは、混在した行末コードを UNIX の行末形式に標準化します。share オプションはクライアント・ワークスペースに同期されたファイルには影響しません。しかし、ファイルが PERFORCE サーバにサブミットして戻される際、share オプションにより Windows 形式の行末である CR/LF および Mac 形式の行末である CR はすべて UNIX 形式の LF に変換され、LF のみがそのままの状態に残されます。</p> <p>クライアント・ワークスペースの同期の際、行末は LF に設定されます。ファイルを Windows マシンで編集する場合、エディタにより各 LF の前に CR が挿入されても、追加された CR はアーカイブ・ファイル内には現れません。</p> <p>一般に share の設定を利用するのに適しているのは、UNIX のホーム・ディレクトリをネットワーク・ドライブとしてマウントした Windows ワークステーションのユーザが、ファイルの同期は UNIX から行い、編集は Windows マシン上で行うような場合です。</p>

PERFORCE による行末設定の使用方法に関して詳しくは、PERFORCE Web サイトのテクニカル・ノート 63 を参照してください。

<http://www.perforce.com/perforce/technotes/note063.html>

- デフォルトでは、クライアント・ワークスペースに空のディレクトリがあっても（例えば、そのディレクトリにマッピングされるすべてのファイルが最後の同期以後に削除されているような場合）、p4 sync はそのディレクトリをそのまま残します。ただし、[Options:] フィールドで rmdir が設定されていれば、p4 sync はクライアント・ワークスペースの空のディレクトリを削除します。

[Options:] フィールドで rmdir が設定されていると、p4 sync で現在の作業ディレクトリが削除されることがあります。その場合には、作業を続行する前に既存のディレクトリに切り換えてください。

- ファイルタイプ修飾子 modtime (+m) が付いているファイルは主に、ファイルのオリジナル・タイムスタンプを保存する必要がある開発者用です。+m を使用すると、[Options:] フィールドの modtime または nomodtime の設定がオーバーライドされます。+m 修飾子について詳しくは、「ファイルタイプ」のセクションをご覧ください。
- 別名のクライアント・ルート ([AltRoots:] フィールドを使用している場合は、p4 info で表示される Client root: によって、どのクライアント・ルートが有効になっているかをいつでも知ることができます。
- 複数ドライブにまたがる Windows 上の PERFORCE クライアントを指定するには、クライアント・ワークスペース・ビューで [Root:] フィールドを null に設定し、個々のドライブを指定します。例えば、下のようにクライアント・ルートを null にしてクライアント仕様を指定すると、//depot/main/... から C ドライブの領域へ、他の release から D ドライブへマッピングされます。

```
Client: eds_win
Owner: edk
Description:
    Ed's Windows Workspace
Root: null
Options:      nomodtime noclobber
SubmitOptions: submitunchanged
View:
    //depot/main/...      "//eds_win/C:/Current Release/..."
    //depot/rel1.0/...    //eds_win/D:/old/rel1.0/...
    //depot/rel2.0/...    //eds_win/D:/old/rel2.0/...
```

複数ドライブにまたがるワークスペースの個々のドライブは小文字で指定します。

使用例

<code>p4 client</code>	P4CLIENTまたは同等の方法によって指定された名前のクライアント・ワークスペースの仕様を編集または生成します。
<code>p4 client -t sue joe</code>	デフォルトとしてクライアント・ワークスペース <code>sue</code> の設定値とオプションを含むフォームを表示して、クライアント・ワークスペース <code>joe</code> の生成または編集を行います。
<code>p4 client -d release1</code>	クライアント・ワークスペース <code>release1</code> を削除します。

関連コマンド

システムが認識しているクライアント・ワークスペースのリストを表示する。	<code>p4 clients</code>
ディポからクライアント・ワークスペースへファイルを読み込む。	<code>p4 sync</code>
クライアント・ワークスペースでディポに追加する新しいファイルを作業状態にする	<code>p4 add</code>
クライアント・ワークスペースで編集するファイルを作業状態にする。	<code>p4 edit</code>
クライアント・ワークスペースで削除するファイルを作業状態にする。	<code>p4 delete</code>
クライアント・ワークスペースのファイルに対する変更をディポに書き込む。	<code>p4 submit</code>

p4 clients

概要

現在システムが認識しているすべてのクライアント・ワークスペースのリストを表示します。

構文

```
p4 [g-opts] clients [ -u user ] [ -e namefilter -m max ]
```

解説

p4 clients は、PERFORCE サーバが認識しているすべてのクライアント・ワークスペースのリストを表示します。ワークスペースは1行に1つずつ、次の形式で表示されます。

```
Client clientname moddate root clientroot description
```

例えば、

```
Client paris 1999/02/19 root /usr/src 'Joe's client'
```

は、1999年2月19日に最後に更新され、/usr/srcをルート・ディレクトリとする、parisという名前のクライアント・ワークスペースを表します。ワークスペースに関するコメント(description)は、p4 clientのフォームで入力されたものがそのまま表示され、この例ではJoe's clientです。

クライアント・ワークスペースの出力をmaxまでに限定するには、-m max オプションを使用します。

namefilter というパターンに一致する名前を持つラベルに出力を限定するには、-e namefilter オプションを使用します。

指定されたユーザが所有しているワークスペースに出力を限定するには、-u user オプションを使用します。

p4 workspaces は p4 clients と同等に機能します。

オプション

-m max	max までのクライアント・ワークスペースのみを一覧表示します。
-e namefilter	namefilter に一致するラベルのみを一覧表示します。
-u user	user が所有しているクライアント・ワークスペースのみを一覧表示します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	list

関連コマンド

クライアント・ワークスペース仕様を編集または参照する。	p4 client
クライアント・ワークスペース名その他の有用なデータを見る。	p4 info
PERFORCE ユーザのリストを参照する。	p4 users

p4 counter

概要

持続的に使用する変数のアクセス、設定、削除を行います。

構文

```
p4 [g-opts] counter countername
p4 [g-opts] counter countername value
p4 [g-opts] counter -d countername
p4 [g-opts] counter -f [ change|job|journal|monitor|security] value
```

解説

カウンタは、PERFORCE にアクセスするスクリプトのために継続的に使用する変数を保存します。例えば、PERFORCE レビュー・デーモンは、最後に処理されたチェンジリストの番号を保存する1つのカウンタ (review) を利用します。

p4 counter countername の形式で使用されると、変数 countername の値が返されます。p4 counter countername value の形式で使用されると、変数 countername の値が value に設定されます。変数 countername が存在しない場合、作成されます。

PERFORCE サーバは、正規の動作において change、job、journal の3つのカウンタを使用します。スーパー・ユーザは、-f オプションを用いてこれらのカウンタを強制的に変更できます。これらのカウンタの変更は、リスクを伴います。これらのカウンタの手動リセットを行うのに適した状況については、『リリース・ノート』をご覧ください。

カウンタ monitor を 0 (監視を無効にする)、または 1 (アクティブなプロセスの監視を有効にする)、または 2 (アクティブなプロセスと休止中のプロセスの監視を有効にする) に設定することによって、サーバ・プロセスの監視を制御することができます。カウンタの変更を有効にするには、PERFORCE サーバを停止し再起動する必要があります。プロセスの監視を有効にすると、PERFORCE サーバでの動作を p4 monitor コマンドで監視することができます。

パスワード強度の要件を設定する場合や、チケット・ベースの認証機構の使用を必要とする場合は、security カウンタを必要なレベルに設定します。設定を有効にするには、PERFORCE サーバを停止し再起動する必要があります。詳細については、『システム管理者ガイド』をご覧ください。

オプション

-d countername	変数 countername を PERFORCE サーバから削除します。
-f [change job journal]	PERFORCE が使用する3つの内部カウンタのいずれかに変更を強制します。通常、このオプションを使う必要はほとんどありません。
-f monitor [0 1 2]	サーバ・プロセスの監視を無効にする、アクティブなプロセスを監視する、アクティブなプロセスと休止中のプロセスの監視を有効にする 詳しくは、p4 monitor のページをご覧ください。
-f security [0 1 2 3]	サーバのセキュリティ・レベルを設定します。詳細については、『システム管理者ガイド』をご覧ください。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	list : カウンタの値を表示する場合 review : 新しい値を設定する場合 super : -f オプションを使用する場合

- カウンタが存在しなければ、その値はゼロとして返されます。カウンタ名は、ゼロ以外の値がセットされるまではデータベースに保存されません。
- PERFORCE サーバが認識している最後のチェンジリスト番号(p4 counter change の出力)は、ユーザによって生成されたがまだディポにはサブミットされていない作業中チェンジリストも含めての番号です。チェンジ・レビュー・デーモンのスクリプトを作成する上で、最後にサブミットされたチェンジリストの番号を確認したいときは、`p4 changes -m 1 -s submitted` コマンドを実行してください。このコマンドの出力の第 2 フィールドで、番号を確認することができます。
- カウンタの値は、内部的に文字列で表現されています。(リリース 2008.1 より前は、符号付き整数で格納されていました。)

使用例

<code>p4 counter mycounter 123</code>	カウンタ mycounter の値を 123 に設定します。Mycounter が存在しない場合は作成されます。 実行には Review 権限が必要です。
<code>p4 counters mycounter</code>	カウンタ mycounter の値を表示します。Mycounter が存在しない場合、値は 0 と表示されます。 実行には List 権限が必要です。

関連コマンド

すべてのカウンタとその値のリストを表示する。	<code>p4 counters</code>
チェンジリストをリスト表示し、追跡する。	<code>p4 review</code>
あるファイルをレビューできるように登録しているすべてのユーザの一覧を表示する。	<code>p4 reviews</code>

p4 counters

概要

PERFORCE および関連スクリプトによって継続的に使用される変数のリストを表示します。

構文

```
p4 [g-opts] counters
```

解説

PERFORCE サーバは、最後にサブミットされたチェンジリストの番号や次のジョブの番号を保存する変数として、カウンタを使用します。ユーザは、カウンタを使用する独自のスクリプトを生成することができます。p4 counters は、現在のカウンタのリストをその値とともに表示します。

オプション

`g_opts` 「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	list

関連コマンド

カウンタの値を参照または変更する。 `p4 counter`

p4 delete

概要

クライアント・ワークスペースのファイルをディポから削除する目的で作業状態にします。

構文

```
p4 [g-opts] delete [-c changelist#] [-n] file...
```

解説

p4 delete コマンドは、クライアント・ワークスペースのファイルをディポから削除する目的で作業状態にします。ファイルは、直ちにクライアント・ワークスペースから削除されますが、対応するチェンジリストが p4 submit でサーバに送られるまで、ディポからは削除されません。

削除されたファイルは、ディポからも削除されたように見えますが、同じファイルの古いリビジョンへのアクセスはいつでも可能ですので、決して完全に削除されたわけではありません。実際には、該当ファイルの削除を示す最新リビジョンが新たに生成されることとなります。p4 sync を使って該当ファイルの最新リビジョンを別のワークスペースに呼び出すと、そのワークスペースからは該当ファイルが削除されます。

削除目的で作業状態にされたファイルは、該当クライアントの所有リストには表示されません。

オプション

-c change#	指定されたチェンジリストに対して、ファイルを削除目的で作業状態にします。このオプションが付いていない場合、ファイルはデフォルト・チェンジリストにリンクされます。
-n	ファイルやメタデータを実際に変更せずに、削除目的で作業状態にするファイルをプレビューします。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	open

- p4 delete でクライアント・ワークスペースから削除されたファイルも、p4 revert でもう一度そのクライアント・ワークスペースに戻し、作業中チェンジリストから除外することができます。ただし、p4 revert は、チェンジリストがサブミットされる前に実行する必要があります。
- PERFORCEでは、すでに作業状態になっているファイルを別のユーザも作業状態にすることができます。デフォルトのスキームでは、複数のユーザが同じファイルを同時に作業状態にして、後で p4 resolve を用いてファイル衝突を解消することができます。あるユーザが作業状態にしているファイルを別のユーザが同時に作業状態にすることを防ぐには、p4 lock を使います。特定のファイルをすでに別のユーザが作業状態にしているかどうかを調べるには、p4 opened -a file を使います。

使用例

<code>p4 delete //depot/README</code>	ディポの最上位ディレクトリにある README というファイルを削除目的で作業状態にします。クライアント・ワークスペースにある対応のファイルは直ちに削除されますが、ディポのファイルはデフォルトのチェンジリストがサブミットされるまで削除されません。
<code>p4 delete -c 40 file</code>	カレント・クライアント・ワークスペースのすべてのファイルを削除目的で作業状態にします。ファイルは、クライアント・ワークスペースからは直ちに削除されますが、 <code>p4 submit</code> によってチェンジリスト 40 がサーバに送られるまで、ディポからは削除されません。

関連コマンド

ディポに追加するファイルを作業状態にする。	<code>p4 add</code>
編集目的でファイルを作業状態にする。	<code>p4 edit</code>
すべての作業状態のファイルをディポにコピーする。	<code>p4 submit</code>
ディポからクライアント・ワークスペースへファイルを読み出す。	<code>p4 sync</code>
新しいチェンジリストを生成または編集する。	<code>p4 change</code>
作業状態にあるすべてのファイルをリスト表示する。	<code>p4 opened</code>
ファイルを非作業状態に戻す。	<code>p4 revert</code>
作業状態のファイルを別のチェンジリストに移す。	<code>p4 reopen</code>

p4 depot

概要

ディポ仕様を生成または編集します。

構文

```
p4 [g-opts] depot depotname
p4 [g-opts] depot -d depotname
p4 [g-opts] depot -o depotname
p4 [g-opts] depot -i
```

解説

PERFORCE サーバ上のファイルはディポに保存されます。デフォルトでは、各 PERFORCE サーバ上に1つずつディポがあり、その名前は depot です。ディポを生成または編集するには、`p4 depot depotname` コマンドを使用してフォーム内のフィールドを編集します。ディポのタイプは local、remote、spec のうちいずれかになります。

別に生成した Local のディポはデフォルトのディポ depot と全く同様に使用できます。例えば、ディポ new のディレクトリ rel2 にある README という名前のファイルを同期するには、クライアント・ワークスペースマッピングの左側に `//new/rel2/...` を追加し、`p4 sync //new/rel2/README` を実行します。

リモート・ディポをお使いの場合、PERFORCE サーバ（すなわち、P4PORT で指定されたマシン）の設定により、PERFORCE クライアント・プログラムが別の PERFORCE サーバ上のファイルを読み取る権限が与えられています。リモート・ディポの使用は読み取り専用アクセスに限られます。つまり、PERFORCE クライアント・プログラムは、他のサーバ上のディポにあるファイルを加、edit、delete、または integrate することはできません。リモート・ディポについて詳しくは、『システム管理者ガイド』をご覧ください。

スペック・ディポが存在する場合、クライアント・ワークスペース仕様、ジョブ、ブランチ仕様などのユーザ編集フォームに対する変更を追跡します。スペック・ディポは一つのサーバに対して一つしか作成できません。スペック・ディポにあるファイルは自動的にサーバによって生成され、PERFORCE 構文は次のように表記されます。

```
//specdepotname/formtype/objectname[suffix]
```

例えば、名前が spec であり、拡張子が .p4s であるスペック・ディポが存在する場合、以下のようして `job000123` に対する変更履歴を取得することができます。

```
p4 filelog //spec/job/job000123.p4s
```

スペック・ディポの設定について詳しくは、『システム管理者ガイド』をご覧ください。

スペック・ディポを作成したら、`p4 admin updatespecdepot` を使用して、現在のクライアント、ディポ、ブランチ、ラベル、タイプマップ、グループ、ユーザ、ジョブの各フォーム集合からデータを読み込みます。

フォーム・フィールド

フィールド名	タイプ	説明
Depot:	読み取り専用	p4 depot depotname で指定されたディポの名前。
Owner:	書き込み可能	そのディポを所有するユーザの名前。デフォルトでは、そのディポを生成したユーザの名前です。

フィールド名	タイプ	説明
Description:	書き込み可能	ディポの目的に関する短いコメント。書き込み必須ではありません。
Type:	書き込み可能	local または remote です。ローカル・ディポは書き込み可能ですが、リモート・ディポは他のサーバ上にあるディポのプロキシなので、書き込みはできません。
Address:	書き込み可能	[Type:] が local の場合は、subdir と入力します。 [Type:] が remote の場合は、リモート・サーバの P4PORT アドレスを入力します。
Suffix:	書き込み可能	[Type:] が spec の場合、このフィールドには spec ディポ内のオブジェクトの生成パスに付加する拡張子を設定できます。 デフォルトの拡張子は .p4s です。spec ディポ使用の際、拡張子は必須ではありませんが、ファイル拡張子を PERFORCE サーバのバージョン化仕様に対して設定すれば GUI クライアントソフトウェアのユーザが PERFORCE 仕様と任意のテキスト・エディタとを関連付けることが可能です。 [Type:] が local または remote の場合、このフィールドの設定は無視されます。
Map:	書き込み可能	[Type:] が local の場合は、PERFORCE サーバの P4ROOT に対するディポ・サブディレクトリの相対位置をマッピングします。ワイルドカード「...」を含んでいなければなりません。例えば、ローカル・ディポ new の [Map:] は new/... のように記述します。 [Type:] が remote の場合は、リモート・ディポの物理的ネームスペースの位置を、例えば //depot/new/rel2/... のようにマッピングします。このディレクトリがローカルで見えているリモート・ディポのルートとなります。

オプション

-d depotname	ディポ depotname を削除します。削除するディポは、ファイルを含んでいてはなりません。PERFORCE スーパー・ユーザは p4 obliterate でファイルを削除できます。 削除するディポが remote であっても、やはり p4 obliterate を実行する必要があります。ファイルは削除されませんが、削除するディポに関して残っているクライアントまたはラベルのレコードが除去されます。
-i	ディポ仕様を標準入力から読み込みます。
-o depotname	ディポ仕様を標準出力へ書き出します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	super

- p4 depot でディポを生成しても、そのディポは、p4 add でファイルが追加されるまではサーバ上に物理的に生成されることはありません。
- ユーザは、p4 depot で新たに生成されたディポがあっても、p4 protect でアクセスのパーミッションが認められるまでは、そのディポにアクセスすることはできません。

- リモート・ディポは常に `remote` という仮想ユーザによってアクセスされます。また、デフォルトでは、どのPERFORCEサーバ上のどのファイルもリモートからアクセスできるようになっています。特定のサーバへのリモート・アクセスを制限もしくは排除するには、`p4 protect` を用いて、そのサーバのユーザ `remote` に与えるパーミッションを設定します。例えば、ある特定のサーバ上のすべてのディポに保存されたすべてのファイルに対するリモート・アクセスを排除するには、そのサーバに関するパーミッションを次のように設定します。

```
read user remote * -/...
```

リモート・ディポはもともと `read` アクセス専用なので、`write` または `super` のアクセス権限を解除する必要はありません。

仮想ユーザ `remote` は PERFORCE のライセンスを必要としません。

- デフォルトでは、ローカル・ディポの `[Map:]` フィールドは、使用しているサーバのサーバ・ルート (P4ROOT) の設定値からの相対位置を示す、ディポ名と一致するディポ・ディレクトリを指します。ディポのバージョン管理されたファイルを別のボリュームやドライブに保存するには、`[Map:]` フィールドに絶対パスを指定してください。指定するパスは、P4ROOT の下にある必要はありません。
- Windows 上で `[Map:]` フィールドに絶対パスを指定する場合は、ディポ・フォームでスラッシュ (`d:/newdepot/` など) を使用する必要があります。

関連コマンド

PERFORCE サーバが認識しているすべてのディポのリストを参照する。	<code>p4 depots</code>
新しいディポにファイルを入れる。	<code>p4 add</code>
既存のクライアント・ワークスペースから新しいディポへのマッピングをディポに追加する。	<code>p4 client</code>
ディポから 1 つのファイルのあらゆる痕跡を除去する。	<code>p4 obliterate</code>
ディポへのリモート・アクセスを制限する。	<code>p4 protect</code>

p4 depots

概要

PERFORCE サーバが認識している全ディポのリストを表示します。

構文

```
p4 [g-opts] depots
```

解説

PERFORCE サーバが認識している全リモートおよびローカル・ディポが、次のフォーマットで表示されます。

```
Depot name date type address map description
```

ただし、*name*、*date*、*type*、*address*、*map*、*description* の定義は、p4 depot のフォームの定義と同じです。

オプション

`g_opts` 「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか?	ファイル引数に リビジョン範囲を使えるか?	最低限必要なアクセス・レベル
適用外	適用外	list

関連コマンド

リモート・ディポまたは新しいローカル・ディポを生成する。	p4 depot
あるファイルの痕跡をすべてディポから取り除く。	p4 obliterate

p4 describe

概要

チェンジリストおよびそのチェンジリストに含まれるファイルに関する情報を表示します。

構文

```
p4 [g-opts] describe [-dflag -s] changelist#...
```

解説

p4 describe は、チェンジリストの詳細を表示します。出力には、チェンジリスト番号、チェンジリスト作成者、クライアント・ワークスペース名、チェンジリスト作成日、チェンジリストのコメントが含まれます。

チェンジリストが submitted (サブミット済) であれば、デフォルトの出力には影響を受けたファイルのリストとそれらのファイルを前のリビジョンと比較した差分が含まれます。

チェンジリストが pending (作業中) の場合は、出力にその旨を示すフラグが付き、作業中ファイルのリストが表示されます。(ファイルはまだサーバにサブミットされていないため、作業中チェンジリストに対する差分は表示されません。)

デフォルトのチェンジリストに対しては p4 describe を実行できません。

p4 describe の実行中、サーバは PERFORCE の内部差分サブルーチンを使用します。環境変数 P4DIFF はこのコマンドに全く影響しません。

オプション

-s	ファイルの差分を含まない短縮された出力を表示します。
-dflag	標準的な UNIX の差分オプションの 1 つで差分ルーチンを実行します。使用可能なオプションについては、次の「使用上の留意点」のセクションをご覧ください。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	read ただし、p4 describe -s には list

p4 describe がサポートしている差分オプション

オプション	意味
-dn	RCS 形式
-dc	context 形式
-ds	要約 (summary)
-du	unified 形式
-db	空白に加えられた変更を無視する
-dw	すべての空白を無視する

関連コマンド

チェンジリストのリストを参照する。	p4 changes
作業状態にある全ファイルのリストを参照する。	p4 opened
ディポに保存されている2つのファイルのリビジョンを比較する。	p4 diff2
クライアント内で変更されたファイルを、ディポに保存されているファイルのリビジョンと比較する。	p4 diff

p4 diff

概要

クライアント・ワークスペースのファイルをディポ内のリビジョンと比較します。

構文

```
p4 [g-opts] diff [-dflag -f -m max-sa -sb -sd -se -sr -sl -t]
[file[rev#]...]
```

解説

p4 diff は、PERFORCE クライアントに対して差分プログラムを実行し、クライアント・ワークスペース内のファイルをディポに保存されているリビジョンと比較します。

このコマンドはファイル引数を付けることができます。引数にはリビジョン指定子を含めることができます。引数にリビジョン指定子が含まれていると、クライアントと指定されたリビジョンとの差分を調べることができます。引数にリビジョン指定子が含まれていない場合、クライアント・ワークスペースのファイルは、現在の編集元のリビジョン（通常は最新リビジョン）と比較されます。いずれの場合でも、クライアントのファイルはコマンド edit で作業状態にされていなければなりません。または、クライアントのファイルが最後に同期したリビジョン以外のリビジョンに対して、この比較が行われなければなりません。

ファイル引数にワイルドカードが含まれている場合は、そのファイル・パターンに適合するすべての作業状態のファイルについて差分を調べられます。ファイル引数が与えられていない場合には、すべての作業状態のファイルについてディポの対応するファイルとの差分を調べられます。

デフォルトでは、p4 クライアント・プログラムに組み込まれた差分プログラムが使用されます。この差分プログラムを外部の差分プログラムに切り替えるには、その新しいプログラムを指すように P4DIFF 環境変数またはレジストリ変数を設定します。

オプション

-f	クライアント・ファイルがコマンド edit で作業状態にされていない場合でも、差分比較を強制します。
-dflags	オプションを基本差分ルーチンに渡します。(詳しくは後の「使用上の留意点」のセクションをご覧ください。)
-m max	max までの差分（またはステータス）のみを一覧表示します。
-sa	ディポに保存されているリビジョンと異なる作業状態のファイルまたは行方不明の作業状態のファイルの名前だけを表示します。
-sb	衝突解決済みであり、反映目的で作業状態にされているが、衝突解決後に変更されているファイルの名前だけを表示します。
-sd	クライアント・ワークスペース内にはないがディポ内にはある、作業状態にないファイルの名前だけを表示します。
-se	ディポに保存されているリビジョンと異なる、クライアント・ワークスペース内の作業状態にないファイルの名前だけを表示します。
-sr	ディポに保存されているリビジョンと一致する、クライアント・ワークスペース内の作業状態にあるファイルの名前だけを表示します。
-sl file...	作業状態でないすべての file がディポと比較され、same、diff、または missing のステータスと共に一覧表示されます。 -f と -sl の両方を使用すると、編集目的で作業状態にされたファイルも比較され、そのステータスが一覧表示されます。

`-t` ファイルが `text` タイプでない場合でもリビジョンの差分比較を行います。
`g_opts` 「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用不可	read

- p4 diff がサポートしている差分オプション

オプション	意味
<code>-dn</code>	RCS 形式。ファイルに対して行われた追加および削除と、関連した行範囲を表示します。
<code>-dc</code>	context 形式。変更された行番号の範囲と、その周辺 3 行を表示します。
<code>-ds</code>	要約 (summary)。追加、削除および変更されたかたまり (チャンク) と行数を表示します。
<code>-du</code>	unified 形式。追加および削除された行を、patch(1) ユーティリティに適した内容で表示します。
<code>-dl</code>	差分を発見したときに行末識別 (CR/LF) を無視します。
<code>-db</code>	空白文字の変更を無視します。このオプションは <code>-dl</code> を含みます。
<code>-dw</code>	空白文字を完全に無視します。このオプションは <code>-dl</code> を含みます。

- 差分プログラムに複数のオプションを渡す場合は、次の例のようにそれらのオプションをまとめます。

```
p4 diff -dub file
```

この例では、空白文字の変更を無視する unified 形式の差分が指定されます。
- patch(1) で使用するために `-du` オプションで生成される UNIFIED 形式の差分のヘッダ行には、ローカル・シンタックスではなく PERFORCE シンタックスでファイル名が表示されます。

使用例

```
p4 diff file#5
```

ファイル `file` のクライアント・ワークスペースのリビジョンとディポに保存されているリビジョン 5 を比較します。

```
p4 diff @1999/05/22
```

クライアント・ワークスペースのすべての作業状態のファイルを、1999 年 5 月 22 日午前 0 時の時点でディポに保存されていたリビジョンと比較します。

<pre>p4 diff -du file</pre>	ファイル <code>file</code> に関する比較を実行し、 <code>patch(1)</code> コーティリティに適した形式で出力します。
<pre>p4 diff -sr p4 -x - revert</pre>	作業状態にあって変更されていないファイルをすべて取り消します。 この例では、 <code>p4 revert -a</code> とは異なり、クライアント・ワークスペースのファイルの内容が、ディポのファイルの内容と一致していれば元に戻ります。したがって、衝突解決済みであっても、ディポの内容と一致している場合を含みます。 最初のコマンドが作業状態にあって変更されていないファイルをすべて表示し、第2のコマンドが (<code>p4 -x</code> を実行し、“-”として省略されている引数を標準入力から1行につき1つずつ受け取り) リストアップされた各ファイルを取り消します。 (これは UNIX の場合のコマンドで、パイプを使用します。ほとんどのオペレーティング・システムは、上記の動作を連続して実行する同等の仕組みを備えています。) <code>p4</code> に付ける <code>-x</code> オプションについてさらに詳しくは、「グローバル・オプション」のセクションをご覧ください。

関連コマンド

ディポのファイルの2つのリビジョンを比較する。	<code>p4 diff2</code>
1つのファイルの全内容を参照する。	<code>p4 print</code>

p4 diff2

概要

ディポのファイルの2つのリビジョンを比較します。

構文

```
p4 [g-opts] diff2 [-dflags -q -t -u] file1[rev] file2[rev]
p4 [g-opts] diff2 [-dflags -q -t -u] -b branch [[fromfile[rev]]
tofile[rev]]
```

解説

p4 diff2 は、PERFORCE サーバに組み込まれた差分ルーチンを使用して、ディポ内のファイルの2つのリビジョンを比較します。これらのリビジョンは、通常は同じファイルの2つのバージョンですが、全く別々のファイルのリビジョンであることも可能です。ファイル・リビジョンがファイル引数として明示されていない場合は、最新リビジョンが使用されます。

p4 diff2 は環境変数 P4DIFF によって指定された差分プログラムを使用しません。p4 diff2 が使用する差分アルゴリズムは、PERFORCE サーバのホスト・マシン上で実行され、常にサーバに組み込まれた差分ルーチンを使用します。

ファイルを特定する代わりに、ファイル・パターンを引数に指定することができ、リビジョン指定子も使えます。このため、PERFORCE は与えられたパターンに適合する個々のファイルの組について、何度も差分比較を実行できます。p4 diff2 にファイル・パターンを付けて実行する場合は、クォーテーション・マークまたはバックスラッシュを用いてファイル・パターンを OS シェルからエスケープし、2つのファイル・パターンの中のワイルドカードを一致させてください。

PERFORCE は、次のような形式のヘッダが前に付加された UNIX の差分フォーマットで差分を提示します。

```
==== file1 (filetype1) - file2 (filetype2) ==== summary
```

summary の取り得る値とその位置は次のとおりです。

- content: ファイル・リビジョンの内容が異なります。
- types: リビジョンの内容は同じですが、ファイルタイプが異なります。
- identical: リビジョンの内容もファイルタイプも同じです。

file1 または file2 のいずれかが指定のリビジョンに存在しない場合は、ヘッダの summary の位置に <none> と表示されます。

オプション

-q	差分比較の出力を最低限に抑えます。ヘッダのみを表示し、ファイル・リビジョンの内容もファイルタイプも同じ場合には、ヘッダさえも表示しません。
-dflags	差分ルーチンを標準UNIX差分オプションのいずれかで実行します。標準差分オプションの種類については、後の「使用上の留意点」のセクションをご覧ください。
-b branchname fromfile[rev] tofile[rev]	ブランチ仕様を用いて2つのブランチされたコードラインのファイルの差分比較を行います。比較するファイルは、fromfile または tofile のファイル・パターンで制限することができます。

-t	ファイルが text タイプでない場合も、リビジョンの差分比較を行います。
-u	追加行と削除行を表示して統一出力フォーマットを生成し、データ内容を patch(1) ユーティリティに対応させます。相違箇所のあるファイルのみが含まれます。ファイル名および日付は PERFORCE 構文のままです。
<code>g_opts</code>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用不可	比較する双方のファイルに対して read アクセス権限が必要

- p4 diff2 がサポートしている差分オプション

オプション	意味
-dn	RCS 形式。ファイルに対して行われた追加および削除と、関連した行範囲を表示します。
-dc	context 形式。変更された行番号の範囲と、その周辺 3 行を表示します。
-ds	要約 (summary)。追加、削除および変更されたかたまり (チャンク) と行数を表示します。
-du	unified 形式。追加および削除された行を、patch(1) ユーティリティに適した内容で表示します。
-db	空白文字の変更を無視します。
-dw	空白文字を完全に無視します。

- 差分プログラムに複数のオプションを渡す場合は、次の例のようにそれらのオプションをまとめます。


```
p4 diff2 -dub file1 file2
```

 この例では、空白文字の変更を無視する unified 形式の差分が指定されます。
- patch(1) を使用するために -du オプションで生成される unified 形式のヘッダ行では、差分ファイルを PERFORCE シンタックスで表示します。ローカル・シンタックスではありません。
- p4 diff2 を binary ファイルの差分比較に使用したときは、一致しなければ、


```
... files differ ...
```

 という行が表示されます。
- オプション `-b branch [fromfile[rev] [tofile[rev]]]` は、一見すると不適切のよう見えます。ブランチ仕様は `fromfiles` から `tofiles` までマッピングしているので、`fromfile` と `tofile` の両方のファイル・パターンを指定する必要はありません。しかし、この構文を使うと、`fromfile` のファイル・パターンと `tofile` のリビジョン、または `fromfile` のリビジョンと `tofile` のファイル・パターンを指定することが可能になります。

使用例

<pre>p4 diff2 -ds file#1 file</pre>	<p>ファイル <code>file</code> のリビジョン1とその最新リビジョンを比較し、どのかたまり (チャンク) が追加され、削除され、変更されたかのサマリを表示します。</p>
<pre>p4 diff2 file@34 file@1998/12/04</pre>	<p>チェンジリスト 34 のサブミット後にディポにあった <code>file</code> のリビジョンと、1998年12月4日午前0時時点のディポのリビジョンの差分比較を行います。</p>
<pre>p4 diff2 //depot/rel1/... //depot/rel2/...#4</pre>	<p>//depot/rel1 のすべてのファイルの最新リビジョンと //depot/rel2 のすべてのファイルのリビジョン 4 を比較します。</p>
<pre>p4 diff2 //depot/rel1/* //depot/rel2/...</pre>	<p>認められていません。両方のファイル・パターンのワイルドカードが一致する必要があります。</p>
<pre>p4 diff2 -b branch2 //depot/rel2/...#2 @50</pre>	<p>//depot/rel2/... のファイルのリビジョン 2 と、そこからブランチ仕様 <code>branch2</code> によってブランチされたファイルのチェンジリスト 50 におけるリビジョンとを比較します。</p>

関連コマンド

<p>クライアント・ワークスペースのファイルとディポに保存されたファイル のリビジョンを比較する。</p>	<pre>p4 diff</pre>
<p>ファイルの全内容を参照する。</p>	<pre>p4 print</pre>

p4 dirs

概要

指定されたディポ・ディレクトリ直下のサブディレクトリをリスト表示します。

構文

```
p4 [g-opts] dirs [-C -D -H] depot_directory[revRange]...
```

解説

p4 dirs は、引数として指定された、ディポ・ディレクトリ直下のサブディレクトリを検索するために使用します。ディレクトリの引数はディポ・シンタックスで指定します。また、末尾をワイルドカード「*」にします。ワイルドカード「...」を使用すると、間違った結果が出ます!

p4 dirs は、引数で指定されたディレクトリ直下のサブディレクトリだけを表示します。1つのディレクトリのサブディレクトリを再帰的にすべて表示するには、p4 dirs を繰り返し実行します。

デフォルトでは、削除されていないファイルを少なくとも1つ含むサブディレクトリだけが返信されます。削除されたファイルしか含まないサブディレクトリを含めるには、-D オプションを使用します。

このコマンドは PERFORCE を呼び出すスクリプトの中で使用されることを意図したものです。コマンドラインから呼び出す必要はありません。

オプション

-C	カレント・クライアント・ワークスペースのビューを通してマッピングされるディレクトリだけを表示します。
-D	削除済ファイルしか含まないサブディレクトリも含めます。デフォルトでは、これらのサブディレクトリは表示されません。
-H	カレント・クライアント・ワークスペースの p4 have リストに表示されるファイルを含むディレクトリだけを表示します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか?	ファイル引数に リビジョン範囲を使えるか?	最低限必要なアクセス・レベル
使用可	使用可	list

- ディレクトリ引数にリビジョン指定子またはリビジョン範囲を含めると、返信されるサブディレクトリは、与えられた指定子に合致するファイル・リビジョンを少なくとも1つ含むものだけになります。
- PERFORCE はデータベースの中でディレクトリを追跡しません。つまり、サブディレクトリの値は検索されるのではなく、計算されます。ワイルドカード「...」がサポートされていないことなど、p4 dirs の実装が変わったところがあるのはそのためです。

使用例

p4 dirs //depot/projects/*	//depot/projects の直下のサブディレクトリすべてのリストを返信します。
p4 dirs //depot/a/* //depot/b/*	//depot/a と //depot/b の直下のサブディレクトリすべてのリストを返信します。
p4 dirs //depot/...	ワイルドカード「...」は p4 dirs によってサポートされていません。

関連コマンド

特定の基準に適合するすべてのファイルをリスト表示する。	p4 files
現在の PERFORCE サーバ上にあるすべてのディポをリスト表示する。	p4 depots

p4 edit

概要

クライアント・ワークスペース内で、編集を目的としてファイルを作業状態にします。

構文

```
p4 [g-opts] edit [-c changelist#] [-n] [-t type] file...
```

解説

p4 edit は、クライアント・ワークスペース内で、編集を目的としてファイルを作業状態にします。指定されたファイルはチェンジリストにリンクされます。ディポのファイルは、そのチェンジリストが p4 submit によってサーバに送られるまでは変更されません。

PERFORCE は、ローカル OS のパーミッションを制御します。p4 edit が実行されると、指定されたファイルについて OS の write パーミッションが付加されます。

p4 edit によって編集用に作業状態にされたファイルがディポにサブミットされても、ディポに存在するそのファイルのリビジョンは置き換えられません。その代わりに、それまでのリビジョンをアクセス可能な状態で残しつつ、新しいリビジョンが生成され、連番で次のリビジョン番号が割り当てられます。デフォルトでは、ファイルを参照するコマンドはすべてこの最も新しいリビジョン (最新リビジョン) を対象とします。

デフォルトでは、指定されたファイルはデフォルトのチェンジリストに追加されます。別のチェンジリストを指定するには、-c オプションを使用します。

すでに編集目的で作業状態にされているファイルをあるチェンジリストから別のチェンジリストに移動するには、p4 reopen を使用してください。

オプション

-c change#	指定されたチェンジリストに対してファイルを編集目的で作業状態にします。このオプションが付いていないと、デフォルトのチェンジリストに対してファイルが作業状態になります。
-t type	新しいファイル・リビジョンを、同じファイルのそれまでのリビジョンのタイプをオーバーライドして、指定のタイプで保存します。ファイルタイプについては「ファイル・タイプ」の項を参照してください。
-n	ファイルやメタデータを実際に変更せずに、編集目的で作業状態にするファイルをプレビューします。
g_opts	「グローバル・オプション」の項を参照してください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	open

指定のファイルに関するローカル OS の write パーミッションは p4 edit によって付加されるため、実際にファイルを編集するのはこのコマンドが発行されてからになります。ファイル編集のプロセスは次のようになります。

1. p4 edit を用いて、クライアント・ワークスペース内でファイルを作業状態にします。
2. エディタでファイルを編集します。
3. p4 submit でディポへファイルをサブミットします。

ファイルの古いリビジョンを編集するには、`p4 sync` を用いて過去に保存されていたファイル・リビジョンをクライアント・ワークスペースに呼び出し、`p4 edit` でそのファイルを編集します。このファイル・リビジョンは最新リビジョンではないので、`p4 resolve` を実行してからでなければ、`p4 submit` でディポに保存することはできません。

PERFORCE は、すでに作業状態になっているファイルを作業状態にする操作を妨げません。デフォルトの構成では、複数のユーザが同じファイルを同時に編集し、`p4 resolve` によってファイル衝突を解決することが認められています。他のユーザがすでに特定のファイルを作業状態にしているかどうかを調べるには、`p4 opened -a file` を使用します。自分が `edit` 用に作業状態にしているファイルを他のユーザがチェックアウトするのを防ぐ方法はありませんが、`p4 lock` を使えば、自分が `edit` 用に作業状態にしているファイルを他のユーザがサブミットするのは防げます。

PERFORCE の以前のバージョンでは、`p4 edit` は `p4 open` と呼ばれていました。

使用例

<code>p4 edit -t text+k doc/*.txt</code>	カレント・ディレクトリの <code>doc</code> サブディレクトリで、ファイル名の最後が <code>.txt</code> になっているすべてのファイルを、 <code>edit</code> 用に作業状態にします。これらのファイルはデフォルトのチェンジリストにリンクされ、キーワード拡張が有効の <code>text</code> タイプとして保存されます。
<code>p4 edit -t +l //depotname/...</code>	ディポ内のすべてのファイルに対し、悲観的ロック（排他オープン）を実施します。このチェンジリストのサブミット後は、 <code>depotname</code> というディポにあるファイルを編集できるのは一度に1ユーザのみとなります。
<code>p4 edit -c 14 ...</code>	現在の作業ディレクトリのファイル・ツリーで、すべてのファイルを <code>edit</code> 用に作業状態にします。これらのファイルは <code>text</code> か <code>binary</code> かを調べられ、ファイルの変更はチェンジリスト 14 にリンクされます。
<code>p4 edit status%40jan1.txt</code>	<code>status@jan1.txt</code> という名前のファイルを編集のための作業状態にします。 PERFORCE ワイルドカードとして使用するために予約されているその他の文字の指定方法の詳細については、213 ページの「ファイル名およびエンティティに関する文字の制限」をご覧ください。

関連コマンド

追加を目的としてファイルを作業状態にする。	<code>p4 add</code>
削除を目的としてファイルを作業状態にする。	<code>p4 delete</code>
作業状態のファイルをすべてディポにコピーする。	<code>p4 submit</code>
ディポのファイルをクライアント・ワークスペースにコピーする。	<code>p4 sync</code>
新しいチェンジリストを生成または編集する。	<code>p4 change</code>
作業状態にある全ファイルのリストを表示する。	<code>p4 opened</code>
ファイルを非作業状態に戻す。	<code>p4 revert</code>
作業状態のファイルを別のチェンジリストへ移動、またはファイルタイプの変更を行う。	<code>p4 reopen</code>

p4 filelog

概要

ファイルのリビジョンに関する詳細情報を表示します。

構文

```
p4 [g-opts] filelog [ -h -i -l -L -t -m maxrev ] file...
```

解説

p4 filelog は、引数として指定されたファイルの各リビジョンの詳細を出力します。少なくとも1つのファイルまたはファイル・パターンが引数として指定されなければなりません。

デフォルトでは、出力は日付の逆順で1リビジョンにつき1行で構成されています。各行のフォーマットは次のようになります。

```
... #rev change chnum action on date by user@client (type)
'description'
```

ただし、

- rev はリビジョン番号、
- chnum はサブミットされたチェンジリストの番号、
- action はファイルが作業状態にされた目的、すなわち add、edit、delete、branch、import、または integrate であり、

アクションが import (リモート・ディポからの反映操作) または integrate の場合、次のようなフォーマットで2行目の記述が表示されます。

```
... #integration-action partner-file
```

反映操作の詳細については、p4 integrated をご覧ください。

- date はサブミットの日付 (デフォルト) か、日時 (-t オプション指定時)、
- user はそのリビジョンをサブミットしたユーザの名前、
- client はそのリビジョンをサブミットしたクライアント・ワークスペースの名前、
- type は与えられたリビジョンにおけるファイルのタイプ、
- description は対応するチェンジリストのコメントの最初の30字です。

-l オプションを使用した場合、description は、チェンジリストをサブミットしたときに入力した完全なチェンジリストのコメントです。-L オプションを使用した場合、チェンジリストのコメントは250文字に切り捨てられます。

オプション

-h	ファイル名の履歴の代わりに、ファイル内容の履歴を表示します。リスト表示されるリビジョンには、特定のリビジョンへと (p4 integrate および p4 resolve -at を使用して) ブランチまたはコピーされた、他のファイルのリビジョンが含まれます。コピーまたはブランチにより置き換えられたリビジョンは、たとえそれらが指定されたリビジョンの履歴にある場合でも表示されません。
-i	ブランチにまたがるファイルの履歴をたどります。p4 integrate による反映によって生成されたファイルであれば、PERFORCE はそのファイルのリビジョンの説明を表示し、そのブランチ元ファイルのリビジョンを (元ファイルのブランチ・ポイントまで) 表示します。

-l	各リビジョンの完全な説明をリスト表示します。
-L	各チェンジリストのコメントを 250 文字に切り捨て、長い出力形式でリスト表示します。
-t	日付と同様に時刻も表示します。
-m maxrev	各ファイルの最新のリビジョンから maxrev 数のリビジョンの変更のみをリスト表示します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用不可	list

- p4 filelog の出力は、指定範囲の広いファイル引数で呼び出されると、かなり大量になることがあるため（例えば、p4 filelog //depot/... ではディポのすべてのファイルのリビジョン履歴がリストされる）、p4 filelog コマンドの出力は p4 group によって設定される maxresults の制限を受けることがあります。
- -i オプションと -m maxrev オプションの両方が付いていて、ファイルの最新リビジョンから maxrev の数のリビジョンの範囲内でブランチ生成が行われている場合は、そのブランチ・ポイントからさらに maxrev の数だけリビジョンを逆上り、その間のリビジョンも表示されます。p4 filelog -i は、最新のリビジョンから 50 階層の深さまでブランチを追跡します。どんなサイトでもこれだけ追跡すれば十分でしょう。
- 一時オブジェクト・ファイル（ファイルタイプが +Sn）の古いリビジョンは、動作の部分に purge と表示されます。

使用例

p4 filelog //depot/proj1/...	ディポのproj1ディレクトリ内の全ファイルのリビジョン履歴を表示します。
p4 filelog file1.c file1.h	ローカルの現在の作業ディレクトリにあるファイル file1.c および file1.h のリビジョン履歴を表示します。

関連コマンド

各ファイルに関する追加の情報を読み取る。	p4 files
スクリプトに適したフォーマットでファイル情報を表示する。	p4 fstat
作業状態のファイルのリストを参照する。	p4 opened
使用中のクライアント・ワークスペースに同期したファイルのリストを参照する。	p4 have

p4 files

概要

ディポ内のファイルに関する情報を、ファイルの内容にアクセスすることなく提供します。

構文

```
p4 [g-opts] files [-a] file[revRange]...
```

解説

p4 files は、引数として指定されたファイル・パターンに適合するファイルをリストアップします。リビジョン指定子が与えられていれば、ファイルのその与えられたリビジョンについての説明が表示されます。出力は1行に1ファイルずつ、次のフォーマットで表示されます。

```
depot-file-location#rev - action change change# (filetype)
```

ただし、

- *depot-file-location* はディポのルート・ディレクトリに対するファイルの相対的位置、
- *rev* はファイルの最新リビジョンの *リビジョン番号*、
- *action* は最新リビジョンにおいて実行されたアクション、つまり、add、edit、delete、branch、または integrate、
- *change#* はそのリビジョンがサブミットされたチェンジリストの番号、
- *filetype* は最新リビジョンにおけるファイルの PERFORCE ファイルタイプです。

通常の PERFORCE コマンドと異なり、p4 files はディポの任意のファイルについてレポートします。その範囲は、クライアント・ビューを通して見ることのできるファイルだけに限られません。もちろん、コマンドラインにファイル・パターンがクライアント・シンタックスで与えられていれば、クライアント・ファイルだけが表示されます。

オプション

-a	指定されたファイルに対して、リビジョン範囲における最高位リビジョンだけを表示するのではなく、範囲内のすべてのリビジョンをリストします。
<i>g_opts</i>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	list

- リビジョンはリビジョン範囲で指定することもできます。この場合、指定された範囲内のリビジョンを持つファイルだけがリストされ、デフォルトではその中の最高位のリビジョンが最新リビジョンとして表示されます。(リビジョン範囲における全ファイルの情報を表示するには、p4 files -a を使用します。)
- p4 files の出力は、指定範囲の広いファイル引数で呼び出されると、かなり大量になることがあるため (例えば、p4 files //depot/... では、ディポの全ファイルに関する情報が表示される)、p4 group によって設定される maxresults の制限を受けることがあります。

使用例

<code>p4 files //depot/...</code>	ディポの全ファイルに関する情報を提供します。
<code>p4 files //clientname/...</code>	指定されたクライアントのビューを通して見ることのできる全ディポ・ファイルに関する情報を提供します。
<code>p4 files @2000/12/10</code>	2000年12月10日に存在した全ディポ・ファイルのリビジョンに関する情報を提供します。
<code>p4 files @2001/03/31:08:00,@2001/03/31:17:00</code>	2001年3月31日の営業時間内に変更された全ファイルおよびリビジョンをリスト表示します。
<code>p4 files //depot/proj2/...@p2lab</code>	ラベル p2lab に含まれるディレクトリ //depot/proj2/... のファイルおよびリビジョンをリスト表示します。
<code>p4 files //depot/file.c</code>	//depot/file.c の最新リビジョンの情報を表示します。(これは、暗黙的な範囲である #1, #head における最高位のリビジョンです。)
<code>p4 files -a //depot/file.c</code>	//depot/file.c の全リビジョンの情報を表示します。(これは、暗黙的な範囲である #1, #head におけるすべてのリビジョンです。)

関連コマンド

ファイルのリビジョン履歴をリスト表示する。	<code>p4 filelog</code>
現在作業状態にある全ファイルのリストを見る。	<code>p4 opened</code>
使用中のクライアント・ワークスペースに同期したファイル・リビジョンのリストを見る。	<code>p4 have</code>
ディポ・ファイルの内容を参照する。	<code>p4 print</code>

p4 fix

概要

ジョブを、その修正を行うチェンジリストにリンクします。

構文

```
p4 [g-opts] fix [ -d ] [ -s status ] -c changelist# jobName ...
```

解説

p4 fix コマンドは、ジョブ（実行されるべき作業の記述）を、チェンジリスト（ファイルに対する変更の集合。ジョブに記述された作業を行う）にリンクします。

チェンジリストがまだサブミットされていない場合、ジョブは、リンク先チェンジリストについて p4 submit または p4 change を実行するときを開くフォーム内に表示されます。通常の場合では、チェンジリストがサブミットされた時点で、ジョブの状態が closed に変わります。p4 fix を実行する前にすでにチェンジリストがサブミット済みである場合も、ジョブの状態はやはりデフォルトの状態（通常は closed）に設定されます。

チェンジリストをサブミットする際に、ジョブの状態をデフォルトの状態（通常は closed）以外に変更するには、p4 fix、p4 submit、または p4 change で -s オプションを使用します。

記述された作業を、複数のチェンジリストで修正完了とすることもできます。すなわち、1つのジョブを、複数のチェンジリストにリンクすることができます。また、1つのチェンジリストで 10 のバグを修正完了にすることも可能です。この場合は、複数のジョブを同一のチェンジリストにリンクします。これは、p4 fix に複数のジョブを引数として与えることにより、コマンドを 1 回実行するだけで達成できます。

オプション

-d	指定されたチェンジリストにおける指定されたジョブに関する修正完了の記録を削除します。ジョブの状態は変わりません。
-s status	チェンジリストのサブミット時に、ジョブの状態をデフォルト値である closed ではなく status (または p4 jobspec フォームのフィールド 102 の Presets: に定義されたその他の値) に変更します。 サブミット済チェンジリストにジョブをリンクする場合、この状態の値はすぐにジョブの状態に反映されます。 作業中チェンジリストの場合、p4 submit を -s オプションを指定して実行すると、ジョブの状態はチェンジリストのサブミット時に変更されます。そして、p4 submit を -s オプションを指定して実行すると、フォームの [Jobs:] フィールドには、ジョブ名に続いてジョブの状態が表示されます。 ジョブを未変更のままにしておくには、特殊な状態である same を使用します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	open

- ジョブのフォーマットはサイトごとに変更されることがあるので、使用しているシステム上のジョブには [Status:] (状態:) フィールドがない場合があります。その場合には、ジョ

ブは p4 fix でチェンジリストにリンクすることができますが、PERFORCE は、チェンジリストがサブミットされても、そのジョブのフィールドの値はいつさえも変えませんが、

- 修正完了または未了のジョブの状態は、p4 job でジョブを編集することにより、いつでも変えることができます。
- ジョブを修正完了（または未了）にするには、サブミットされていないチェンジリストについて p4 submit または p4 change のフォームを表示させ、その [Jobs:] (ジョブ:) フィールドに該当のジョブを追加する（あるいは、[Jobs:] フィールドから削除する）という方法もあります。
- デフォルトのチェンジリストに p4 fix でジョブをリンクすることはできません。リンクしたいときは、デフォルトのチェンジリストをディポにサブミットするとき、p4 submit フォームの [Jobs:] フィールドにそのジョブを追加してください。
- p4 fix -s status をジョブに対して使用し、p4 submit または p4 change に対して -s オプションを使用した場合、チェンジリスト・フォームの [Jobs:] フィールドにはジョブの状態値が必要になります。このデフォルトは、p4 fix -s status で指定された値です。チェンジリストのサブミットが成功した場合、ジョブには指定の status が割り当てられます。フォーム中に状態値が指定されていない場合は、次のエラー・メッセージが表示されません。

```
Wrong number of words for field 'Jobs'.
```

p4 fix -s status、p4 submit -s、および p4 change -s は、欠陥追跡システムと連携して使用することを目的としたオプションです。

通常、ユーザがこれらのコマンドを使用することはなく、p4 submit と p4 change を -s オプションなしで使用します。この場合、[Jobs:] フィールドに必要なのはジョブ番号のみで、各ジョブの状態はサブミット完了時にデフォルト値（通常は closed）に設定されます。

使用例

p4 fix -c 201 job000141 job002034	2つのジョブをチェンジリスト 201 によって修正完了とします。 チェンジリスト 201 がまだ pending の場合には、ジョブの状態はチェンジリストがサブミットされた段階で closed に変わります。
p4 fix -c 201 -s suspended job002433	チェンジリスト 201 がサブミットされたときに、job002433 を closed ではなく suspended とします。 p4 submit を -s オプション付きで実行する必要があります。

関連コマンド

作業中チェンジリストに対してジョブを追加または削除する。	p4 change
デフォルトのチェンジリストに対してジョブを追加または削除する。	p4 submit
ジョブとチェンジリストの関係をリスト表示する。	p4 fixes
ジョブを生成または編集する。	p4 job
ジョブのすべて、または一部をリスト表示する。	p4 jobs
サイトで使用するジョブのフォーマットを変更する。(スーパー・ユーザ専用)	p4 jobspec
サイトで使用しているジョブのフォーマットに関する情報を読み出す。	p4 jobspec -o

p4 fixes

概要

ジョブおよびそれを修正するチェンジリストをリスト表示します。

構文

```
p4 [g-opts] fixes [-i -m max -j job -c changelist#] [file[revRange]...]
```

解説

あるジョブが p4 fix、p4 change、または p4 submit によって特定の番号付チェンジリストにリンクされた後は、そのジョブはそのチェンジリストによって 修正完了 になっています (チェンジリストがまだ作業中であっても)。p4 fixes コマンドは、チェンジリストと、それらが修正するジョブをリスト表示します。

引数を指定せずに実行すると、p4 fixes はすべての修正完了記録を次の形式で表示します。

```
jobname fixed by change changelist# on date by user (status)
```

p4 fixes を下記オプションと組み合わせて実行することにより、リスト表示される修正完了記録を制限することができます。

- `-c changelist` オプションを使用すると、その作業中またはサブミット済みのチェンジリストによって修正されたジョブだけがリスト表示されます。
- `-j job` オプションを使用すると、そのジョブを修正完了にする作業中またはサブミット済みのチェンジリストだけがリスト表示されます。
- 引数に 1 つまたは複数のファイル・パターンを指定することもできます。ファイル引数を指定すると、そのファイル・パターンに合致するファイルに影響するサブミット済チェンジリストだけがリスト表示され、作業中チェンジリストは含まれません。リビジョン指定子またはリビジョン範囲が含まれていれば、指定されたリビジョンのファイルに影響したサブミット済チェンジリストだけがリスト表示されます。`-i` オプションをファイル・パターン引数と共に使用すると、特定のファイルへ反映されたチェンジリストによって行われた修正を含めることができます。
- 修正の出力を `max` までに限定するには、`-m max` オプションを使用します。

オプション

<code>-c changelist#</code>	表示される修正完了記録を、指定のチェンジリストを含むものだけに制限します。
<code>-j jobname</code>	表示される修正完了記録を、指定のジョブを含むものだけに制限します。
<code>-i files...</code>	指定ファイルの反映元のファイルに関連するチェンジリストによる修正完了の記録を含めます。
<code>-m max</code>	<code>max</code> までの修正のみを一覧表示します。
<code>g_opts</code>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	list

使用例

<pre>p4 fixes //depot/proj1/...</pre>	<pre>//depot/proj1</pre> <p>のファイルを1つでも含むサブミット済チェンジリストによって行われたすべての修正完了の記録を表示します。</p>
<pre>p4 fixes -c 414</pre>	<p>作業中またはサブミット済みのチェンジリスト 414 によって修正完了にされたすべてのジョブを表示します。</p>

関連コマンド

<p>ジョブの生成または既存のジョブの編集を行う。</p>	<pre>p4 job</pre>
<p>システムが認識しているすべてのジョブをリスト表示する。</p>	<pre>p4 jobs</pre>
<p>ジョブをチェンジリストにリンクする。そのジョブはリンク先チェンジリストによって修正完了にされる。</p>	<pre>p4 fix</pre>
<p>サイトで使用するジョブのフォーマットを変更する。<u>(スーパー・ユーザ専用)</u></p>	<pre>p4 jobspec</pre>
<p>サイトで使用しているジョブのフォーマットに関する情報を読み出す。</p>	<pre>p4 jobspec -o</pre>

p4 flush

概要

クライアント・ワークスペースの所有リストを、ファイルを実際にコピーすることなく更新します。

構文

```
p4 [g-opts] flush [-n] [file[revRange]...]
```

警告

p4 flush は、使用方法を誤ると **重大な結果をもたらす危険性があります**。p4 flush を不適切に使用すると、サーバのメタデータがクライアント・ワークスペースの実際の状態を反映しなくなり、以後、実行する PERFORCE コマンドが、意図したファイルに対して働かなくなります！ p4 flush は、その目的を十分に理解するまで使用しないでください。

なお、p4 flush はほとんど使用する必要のないコマンドです。

解説

p4 flush は、p4 sync の機能を半分だけ実行します。例えば、p4 sync filespec を実行すると、次の 2 つの処理が行われます。

- filespec にあるファイル・リビジョンを、ディポからクライアント・ワークスペースにコピーします。
- クライアント・ワークスペースの 所有リスト (have list) (PERFORCE サーバに保存されている情報で、どのファイル・リビジョンが同期されているかを追跡するリスト) を、新しいクライアント・ワークスペースの内容を反映するように更新します。

p4 flush は、このうち 第 2 の処理だけを実行します。通常、この動作は望ましいものではありません。その理由は、クライアント・ワークスペースの所有リストは、常にクライアント・ワークスペースの実際の内容を反映すべきだからです。しかし、所有リストがもはやクライアント・ワークスペースの内容と同期していないときに、p4 flush を使用して、所有リストを実際の内容に同期させることができる場合があります。p4 flush は実際にファイルを転送するわけではないので、p4 sync の実行よりもずっと速く処理が済みます。

p4 flush は、所有リストをクライアント・ワークスペースの実際の状態に合わせて更新する必要がある場合にのみ、使用してください。後掲の「使用例」のセクションに、そのような場合の例を 2 つ示してあります。

オプション

-n	flush を実行することなしに flush の結果を表示します。これにより、flush を実行する前に、期待どおりの結果が得られるかどうかを確認することができます。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	read

- p4 flush は、ファイルをコピーすることなくクライアント・ワークスペースの所有リストを更新します。一方、p4 sync -f は、クライアント・ワークスペースの内容をその所有

リストに一致させます。したがって、`p4 flush files`に続いて`p4 sync -f files`を実行すれば、`p4 sync files`を実行したのとほぼ同じこととなります。これは、不適切な flush が行われても、後でその flush が行われたファイル・リビジョンに対して`p4 sync -f`を実行すれば、ほぼ完全に修正ができることを意味しています。

しかし、残念ながら、これでも完全には不適切な flush を防げません。`p4 flush`によって所有リストから削除されたファイル・リビジョンは、`p4 sync -f`実行後もクライアント・ワークスペースに残ります。この場合には、削除されたファイル・リビジョンを、手動操作でクライアント・ワークスペースから削除するしかありません。

使用例

- ◆ 同じサイトで作業をする 10 人のユーザが、遠隔地にある同一のディポから、低速リンクを介して、新たに同じクライアント・ワークスペースをセットアップする必要があるとします。通常は、各ユーザが、`p4 sync` コマンドを同じ構文で実行する方法をとりますが、この場合は伝送速度が遅いのが難点です。そこで、次のようにすると、処理を速く済ませることが出来ます。

- 1 人のユーザ (例えばユーザ A) が、自分のクライアント・ワークスペース `firstworkspace` から `p4 sync files` を実行します。
- 他のユーザは、ローカル OS のファイルコピー・コマンドを使用して、ユーザ A のクライアント・ワークスペースから各自のクライアント・ワークスペースへ、新たに同期されたファイルをコピーします。
- さらに、各ユーザが `p4 flush files @firstworkspace` を実行すると、上記のステップで各自のクライアント・ワークスペースへコピーしたファイルと同期した、クライアント・ワークスペースの所有リストが得られます。

`p4 flush` は低速リンクを介したファイル転送をしないので、この処理は、`p4 sync` コマンドを異なるタイミングで 10 回実行するよりはるかに高速になります。

- ◆ ジョーは `ws` というクライアント・ワークスペースを使用していて、その `[Root:]` は `/usr/joe/project1/subproj`

であり、`[View:]` は

```
//depot/joe/proj1/subproj/... //joe/...
```

であるとして、ジョーは今、`/usr/joe/project1` のすべてのファイルを自分のワークスペースに含める必要があると判断し、`[Root:]` を

```
/usr/joe/project1
```

`[View:]` を

```
//depot/joe/proj1/... //joe/...
```

に指定して `p4 client` を実行しました。その結果、カレント・クライアント・ワークスペースのファイルは同じ場所にとどまり、ワークスペースの範囲は他のファイルも含むように拡張されました。ところが、ジョーは次に `p4 sync` を実行したとき、PERFORCE がクライアント・ワークスペースの非作業状態のファイルを残らず削除し、もう一度同じファイルを同じ場所にコピーしていることを知って驚きます！

PERFORCE では、所有リストが各ファイルのクライアント・ルートに対する相対的な位置を記述し、各ファイルの物理的位置は PERFORCE コマンドが実行されるたびに計算されるだけなので、このようなことが起こります。つまり、PERFORCE は各ファイルが再配置されたと考え、`p4 sync` が実行されると、ファイルをそれまでの位置から削除して、新しい位置へコピーするのです。

ジョーが PERFORCE の働きを理解していれば、`p4 flush #have` を実行していたでしょう。そうすれば、実際にはどのファイルもコピーされることなく、クライアント・ワークスペースの所有リストは、ファイルの“新しい”位置を反映するように更新されたはずで

関連コマンド

p4 flush は p4 sync -k と同等に機能します。	p4 sync -k
ディポのファイルをクライアント・ワークスペースにコピーする。	p4 sync
不適切に p4 flush が実行された後の所有リストをクライアント・ワークスペースに同期させる。	p4 sync -f

p4 fstat

概要

ファイル情報をスクリプトによる構文解析に適したフォーマットにダンプします。

構文

```
p4 [g-opts] fstat [-m max -r] [-c|-e change] [-Ox -Rx -Sx] file[rev]...
```

解説

p4 fstat コマンドは、個々のファイルに関する情報を項目ごとに別々の行に分けてダンプします。

ファイルの出力を *max* までに限定するには、`-m max` オプションを使用します。

出力のソート基準となるフィールドを変更するには `-sx` オプションのうち 1 つを使用し、ソート順序を逆にするには `-R` オプションを使用します。

この出力は、各項目が変数としてアクセス可能な PERFORCE API アプリケーションでの使用に最適ですが、スクリプトによる構文解析にも適しています。

フォーム・フィールド

フィールド名	説明	例
clientFile	ファイルのローカル・パス。(デフォルトはローカル・シンタックス。-OP オプションを付けると PERFORCE シンタックス。)	/staff/userid/src/file.c (または、PERFORCE シンタックスでは //workspace/src/file.c)
depotFile	ファイルのディポ・パス。	//depot/src/file.c
path	ファイルのローカル・パス	//workspace/src/file.c
isMapped	マップされたクライアント・ファイルが同期されている場合に設定します。	設定 (... isMapped) または表示されない
headAction	(ディポ内の) 最新リビジョンで実行されたアクション。	add、edit、delete、branch、または integrate のいずれか
headChange	(ディポ内の) 最新リビジョンを含むチェンジリストの番号。	1、2、3 …… n
headRev	(ディポ内の) 最新リビジョンの番号。	1、2、3 …… n
headTime	(ディポ内の) 最新リビジョンのチェンジリスト日付。時刻は 1970 年 1 月 1 日の 00:00:00 UTC からの秒数で測定されます。	919283152 は 1999 年前半の日付
headRevModTime	(ディポ内の) 最新リビジョンの更新時刻 (クライアントにおいてサブミット前にファイルが最後に更新された時刻)。	919280483 1999 年前半の日付
headType	(ディポ内の) 最新リビジョンのタイプ。	text、binary、text+k など (「ファイルタイプ」のセクション参照)
haveRev	最後にワークスペースに同期した (ワークスペース上にある) リビジョン。	1、2、3 …… n

フィールド名	説明	例
desc	チェンジリストのコメント (-e <i>changelist</i> が使用され、ファイルが <i>changelist</i> に含まれていた場合)	PERFORCE のチェンジリスト
digest	ファイルの MD5 ダイジェスト (-O1 オプションが必要)	32 桁の 16 進数
filesize	ファイルのバイト数 (-O1 オプションが必要)	63488
action	(クライアント上で) ファイルを作業状態にしたアクション。	add、edit、delete、branch、または integrate のいずれか
type	(ワークスペース上で) 作業状態になっているファイルのタイプ。	PERFORCE ファイル・タイプ
actionOwner	ファイルを作業状態にしたユーザ (ファイルが作業状態になっている場合)。	PERFORCE ユーザ名
change	(クライアント上で) 作業状態になっているチェンジリストの番号。	1、2、3 …… <i>n</i>
resolved	(もしあれば) 衝突解決済みの反映記録の数。	1、2、3 …… <i>n</i>
unresolved	(もしあれば) 衝突未解決の反映記録の数。	1、2、3 …… <i>n</i>
otherOpen	同じファイルを作業状態にしている他のユーザの数。該当するユーザがない場合はブランクになります。	1、2、3 …… <i>n</i> の前にその <i>n</i> 人のユーザ (0 ~ <i>n</i> -1) のリスト。otherOpen <i>n</i> 、otherAction <i>n</i> 、otherLock <i>n</i> フィールド適用。 ... otherOpen 3 otherOpen0 user1@cws1 otherOpen1 user2@cws2 otherOpen2 user3@cws3
otherOpen <i>n</i>	同じファイルを作業状態にしている個々のユーザに対し、そのファイルを作業状態にしているワークスペースとユーザを表示。	user123@workstation9
otherLock	他のユーザがファイルをロックしている場合に表示されて空文字に設定され、それ以外の場合は表示されません。	未設定 (... otherLock) または表示されない
otherLock <i>n</i>	ロックされたファイルを作業状態にしている個々のユーザに対し、そのロックを保持しているクライアントとユーザを表示。	user123@workstation9 一度に 1 ユーザのみファイルのロックが可能であるため、 <i>n</i> が設定されていれば <i>n</i> は常に 0 となる
otherAction <i>n</i>	同じファイルを作業状態にしている個々のユーザに対し、行われたアクションを表示。	add、edit、delete、branch、または integrate のいずれか
otherChange <i>n</i>	ファイルを作業状態にしているすべてのチェンジリストに対し、そのチェンジリストを表示	チェンジリスト番号

フィールド名	説明	例
ourLock	現在のユーザがファイルをロックしている場合に表示されて空文字に設定され、それ以外の場合は表示されません。	未設定 (... ourLock) または表示されない
resolveActionn resolveBaseFilen resolveBaseRevn resolveFromFilen resolveStartFromRevn resolveEndFromRevn	それぞれ、作業中の反映操作、ベース・ファイル、ベース・リビジョン番号、反映元ファイル、開始リビジョン、終了リビジョン。	作業中の反映記録情報の場合は -Or オプションを使用
totalFileCount	調べたファイルの数	-m max オプションをソート・オプションの -sx または -r のうち1つと共に使用した場合に、最初のファイルの出力に表示される

オプション

-c <i>change</i>	指定番号以降のチェンジリストに関連するファイルのみを表示します。リビジョン範囲を用いるより速く、影響を受けたファイルを表示できます。
-e <i>change</i>	指定の番号のチェンジリストの影響を受けたファイルのみを表示します。リビジョン範囲を用いるより速く、影響を受けたファイルを表示できます。
-m <i>max</i>	fstat の出力を <i>max</i> ファイルまでに限定します。
-r	出力を逆の順序でソートします。
-Of	指定されたファイルのすべてのリビジョンについて、other[...] および resolve[...] フィールドを除外して出力します。
-Ol	fileSize フィールドとして、ファイルの長さとしリビジョンごとの要約フィールドを表示します。 リリース 2005.1 より前のサーバにおいて、特に多数のリビジョンがあるテキスト・ファイルの場合、このフィールドを含めた表示には多くの時間を要する可能性があります。ご注意ください。
-Op	clientFile を、ローカル・シンタックスではなく、PERFORCE シンタックスで表示します。
-Or	現在ワークスペースで作業状態になっているファイルの作業中の反映記録データを表示します。
-Os	クライアント・ワークスペース・データ（すなわち clientFile フィールド）を除外することにより、出力を短縮します。
-Rc	コマンドの出力を、現在のワークスペース上でマッピングされているファイルに制限します。
-Rh	コマンドの出力を、have リスト（現在のワークスペースに同期されているファイル）に制限します。
-Rn	コマンドの出力を、最新リビジョンではないリビジョンで作業状態になっているファイルに制限します。
-Ro	コマンドの出力を、現在のワークスペースで作業状態のファイルに制限します。
-Rr	コマンドの出力を、衝突解決済みの作業状態のファイルに制限します。
-Ru	コマンドの出力を、衝突未解決の作業状態のファイルに制限します。
-St	ファイルタイプによりソートします。
-Sd	日付によりソートします。
-Sr	最新リビジョンによりソートします。
-Sh	現在のリビジョンによりソートします。

<code>-Ss</code>	ファイルサイズによりソートします。
<code>g_opts</code>	「グローバル・オプション」のセクションをご覧ください。 <code>-s</code> グローバル・オプション (出力のタイプを <code>error</code> 、 <code>warning</code> 、 <code>info</code> 、 <code>text</code> 、または <code>exit</code> として記述するタグを、出力の各行の前に付ける) は、 <code>p4 fstat</code> では特に有用です。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	<code>list</code>

- `-Ro` オプションを指定して `-e changelist#` を使用すると、作業中の変更のみが考慮されるため、追加のための作業状態になっているファイルが出力に含まれます。
- `p4 fstat` の構文は、リリース 2004.2 で変更されました。従来のオプションである `-C`、`-H`、`-W`、`-P`、`-l`、および `-s` は、互換性のためにサポートされています。
- 特殊文字 `@`、`#`、`*`、および `%` を含んでいるファイルの場合、`clientFile` では特殊文字が表示され、`depotFile` ではその文字の 16 進値の ASCII 表現が含まれたファイル名が表示されます。
- `size` フィールドと `digest` フィールドは正規化 (UNIX の改行規則) された圧縮されていないバージョンのディポ・ファイルに基づいており、クライアント・ワークスペースと同期させた際のファイルの表現には関係しません。

使用例

<code>p4 fstat file.c</code>	<code>file.c</code> に関する情報を表示します。
<code>p4 fstat -Rc 20 *.c</code>	チェンジリスト 20 でのファイルのチェックイン後に影響を受けた、すべての <code>.c</code> ファイルに関する情報を表示します。
<code>p4 fstat -Os file.c</code>	クライアント・ワークスペース情報行 (<code>clientFile</code>) は表示しません。
<code>p4 fstat -Osl file.c</code>	クライアント・ワークスペース情報行は表示ませんが、 <code>fileSize</code> および <code>digest</code> の行は表示します。
<code>p4 fstat -Os -Ol file.c</code>	<code>p4 fstat -Osl</code> と同義。

関連コマンド

各ファイルに関する付加的情報を読み出す。	<code>p4 files</code>
チェンジの説明を含むファイル情報を表示する。	<code>p4 filelog</code>

p4 group

概要

ユーザをグループに追加、またはグループから削除します。グループのメンバーに対する maxresults、maxscanrows、maxlocktime および timeout の各制限を設定します。

構文

```
p4 [g-opts] group [ -a ] groupname
p4 [g-opts] group -d [ -a ] groupname
p4 [g-opts] group -o groupname
p4 [g-opts] group -i [ -a ]
```

解説

グループは、PERFORCE ユーザのリストです。グループを使用して、p4 protect フォームでのアクセス・レベルの設定、特定のユーザが 1 つのコマンドでサーバからアクセスできるデータの最大件数の制限、p4 login チケットのタイムアウト期間の設定を行います。

グループを削除するには、p4 group -d groupname を実行するか、p4 group groupname を起動して、表示されたフォームからユーザ全員を消去します。

フォーム・フィールド

フィールド名	タイプ	説明
Group:	読み取り専用	コマンドライン上で入力されたグループ名です。
MaxResults:	書き込み可能	グループのメンバーが 1 つのコマンドでサーバからアクセスできるデータの最大件数です。デフォルト値は unset です。詳しくは下記「使用上の留意点」を参照してください。
MaxScanRows:	書き込み可能	グループのメンバーが 1 つのコマンドからサーバからスキャンできる最大の列数です。デフォルト値は unset です。詳しくは下記「使用上の留意点」を参照してください。
MaxLockTime	書き込み可能	データ・スキャンの際、ある 1 つの操作で任意のデータベース・テーブルをロックすることのできる最長時間（ミリ秒単位）。デフォルト値は unset です。詳しくは下記「使用上の留意点」を参照してください。
Timeout:	書き込み可能	p4 login によって作成されるセッション・チケットの有効期間（秒単位）です。デフォルト値は 43200 秒（12 時間）です。無期限のチケットを作成するには、[Timeout:]をunlimitedに設定します。
Users:	書き込み可能、複数行	グループのメンバーの PERFORCE ユーザ名です。1 行に 1 ユーザずつを記述し、インデントされていなければなりません。

フィールド名	タイプ	説明
Subgroups:	書き込み可能、 複数行	他の PERFORCE グループの名前です。 あらかじめ定義されているグループのユーザ全員を現在自分が属しているグループに追加するには、p4 group のフォームの [Subgroups:] フィールドにそのグループの名前を入力します。ユーザ名とグループ名は別々のネームスペースを占有します。このため、グループとユーザは同じ名前をとることができます。 [Subgroups:] フィールドに入力される、あらかじめ定義されたグループのメンバーが、全員、これから定義しようとしているグループのメンバーになります。
Owners:	書き込み可能、 複数行	他の PERFORCE ユーザの名前です。 super 権限のないグループ所有者であっても、-a オプションを使用すればこのグループの管理が許可されます。

オプション

-a	(スーパーユーザではない) グループ所有者によるグループの管理を可能にします。そのユーザは、グループの Owner: フィールドにリストされていなければなりません。
-d <i>groupname</i>	グループ <i>groupname</i> を削除します。グループのメンバーは、このグループ削除の結果としてアクセス・レベルまたは <code>maxresults</code> 値が変化する場合にのみ影響を受けます。
-i	ユーザのエディタを立ち上げずに標準入力からフォームを読み込みます。新しいグループ仕様がそれまでの仕様に置き換わります。
-o	ユーザのエディタを立ち上げずに標準出力へフォームを書き出します。
<i>g_opts</i>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか?	ファイル引数に リビジョン範囲を使えるか?	最低限必要なアクセス・レベル
適用外	適用外	super (p4 group -o または -a のリスト)

- 複数のグループに属するユーザのチケットの Timeout 値は、`maxresults` 値と同様の方法で計算されます。つまり、そのユーザが属するすべてのグループの `timeout` 値のうち (unlimited は含めるが unset は無視して) 最大の値が設定されます。どのグループにも属さないユーザには、デフォルトの `timeout` 値である 43200 が設定されます。無期限のチケットを作成するには、[Timeout:] を unlimited に設定します。

- ディポに保存されているファイルの数が増えてくると、ある種のコマンドをパラメータなしで、あるいは制限のない引数を付けて実行したときに、サーバの動作が大幅に遅くなる場合があります。例えば、`p4 print //depot/...` は、ディポに保存されているあらゆるファイルの内容をユーザの画面上に表示し、`p4 filelog //depot/...` は、ディポに保存されているあらゆるファイルのあらゆるリビジョンにおいて、データを取り込もうとします。

PERFORCE スーパー・ユーザは、ユーザのグループについて `MaxResults` 値を設定することにより、サーバがクライアントに返すデータの量を制限することができます。同様に、`MaxScanRows` 値を設定することにより、サーバによってスキャンされるデータの量 (クライアントに返されるか返されないかにかかわらず) を制限することができます。また、`MaxLockTime` 値により、ある 1 つの操作で任意のデータベース・テーブルをロックすることができる時間の長さを設定することができます。

この MaxResults 値、MaxScanRows 値または MaxLockTime 値を超えた場合、サーバへの要求は拒否され、ユーザは問合せを制限するように求められます。

ユーザが複数のグループに属している場合、サーバは、ユーザが属しているすべてのグループに対する MaxResults 値の最大値をそのユーザの MaxResults 値として計算します (設定値 unlimited に遭遇すると制限は削除されますが、デフォルト値 unset のままである設定は無視されます)。どのグループにも属さないユーザがいれば、その MaxResults 値は unset になります。(ユーザの MaxScanRows 値および MaxLockTime 値の制限も同様の方法で計算されます。)

通常のサーバ・ハードウェアの処理速度では、MaxResults を 10,000 以下に設定したり、MaxScanRows を 50,000 以下に設定したり、MaxLockTime を 1000 以下に設定したりする必要はありません。

- あるグループについて maxresults、maxscanrows、maxlocktime、および timeout の制限を表示するには、`p4 groups -v groupname` を使用します。
- `p4 help maxresults` を使用すると、3 つの制限値のどれかに影響を受けるコマンドのリストを取得することができます。

関連コマンド

ユーザのアクセス・レベルを変更する。	<code>p4 protect</code>
既存グループのリストを参照する。	<code>p4 groups</code>

p4 groups

概要

ユーザのグループをリスト表示します。

構文

```
p4 [g-opts] groups [ -m max ] [ [ [ -i ] user | group ] | [ -v group ] ]
```

解説

p4 groups は、p4 group によって生成された現時点のすべてのユーザ・グループをリスト表示します。表示されるのはグループ名だけです。

オプションの引数 *user* が与えられていれば、そのユーザを含むグループだけがリスト表示されます。オプションの引数 *group* が与えられていれば、そのグループを含むグループだけがリスト表示されます。

-i オプションを使用すると、ユーザ (またはグループ) がサブグループのメンバとして属しているグループを含めることができます。グループ引数が指定された場合、指定されたグループを含むグループのみが表示されます。

-v オプションを使用すると、指定されたグループ (グループが未指定の場合はすべてのグループ) の MaxResults、MaxScanRows、MaxLockTime、Timeout の値が表示されます。

グループの出力を max までに限定するには、-m max オプションを使用します。

オプション

-m max	max までのグループのみを一覧表示します。
-i [user group]	user または group が間接的なメンバーとなっている (つまりサブグループに含まれている) グループを表示します。
-v [group]	冗長な出力を表示し、MaxResults、MaxScanRows、MaxLockTime、Timeout の値を含めます。
g_opts	「グローバル・オプション」の項を参照してください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか?	ファイル引数に リビジョン範囲を使えるか?	最低限必要なアクセス・レベル
適用外	適用外	list

- 1つのグループのメンバー全員を見るには、p4 group -o groupname を使います。このオプションが付いた p4 group コマンドは、list のアクセス権限さえあれば使えます。
- -v オプションと -i オプションは、一緒に使用できません。

使用例

```
p4 groups bob      ユーザ bob を含むすべてのグループの名前を表示します。
```

関連コマンド

ユーザ・グループの生成、または既存ユーザ・グループの編集を行う。 p4 group

特定グループのすべてのメンバーおよび仕様のリストを参照する。	<code>p4 group -o groupname</code>
特定グループのメンバーについて PERFORCE アクセス・レベルを設定する。	<code>p4 protect</code>

p4 have

概要

クライアント・ワークスペースに同期しているファイルおよびリビジョンをリスト表示します。

構文

```
p4 [g-opts] have [file...]
```

解説

p4 have は、p4 sync でクライアント・ワークスペースにコピーされているファイルおよびリビジョンをリスト表示します。ファイル・パターンが与えられていれば、表示されるのは、そのパターンに合致するファイル、また、クライアント・ビューに対してマッピングされているファイルに限られます。

p4 have はファイルを 1 行に 1 つずつ、次のフォーマットでリスト表示します。

```
depot-file#revision-number - local-path
```

- depot-file は、ディポ・シンタックスで記述されたファイルへのパスです。
- revision-number は、所有リビジョン、すなわち現在カレント・クライアント・ワークスペースにあるリビジョンです。
- local-path は、ローカル・ファイルシステムの記述（すなわち、ローカル・シンタックス）で表現されたパスです。

オプション

`g_opts` 「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	list

- PERFORCE 関連ドキュメントには、クライアント・ワークスペースの所有リストに触れているものがいくつかあります。所有リストは、p4 have によってレポートされるファイルのリストであり、現在、ディポから同期しているファイル・リビジョンのリストです。このリストには、クライアント・ワークスペースに存在してもディポには存在しないファイルは含まれません。

例えば、p4 add を使って追加するファイルを新たにクライアント・ワークスペースで生成したり、p4 integrate を使ってクライアント・ワークスペースで 1 つのファイルのグループを生成したりしても、まだそれをサブミットしていなければ、それらの新しいファイルは p4 have の出力には表示されません。

クライアント・ワークスペースに存在するすべてのファイルの集合（ファイル・セット）は、p4 have によってリストされるファイル・セットと p4 opened によってリストされるファイル・セットを組み合わせたものに相当します。

- 特殊文字 @、#、*、および % を含んでいるファイルの場合、次の例のように、depot-file フィールドにその文字の 16 進値の ASCII 表現が表示され、local-path フィールドに特殊文字が表示されます。

```
//depot/status/100%25.txt#1 - /staff/status/100%.txt
```

使用例

<pre>p4 sync //depot/name... p4 have //depot/name</pre>	左に示す 2 組のコマンドの場合、第 1 の組の p4 have は、クライアント・ワークスペースにコピーされているファイルの最新リビジョンを表示します。
<pre>p4 sync //depot/name/...#4 p4 have //depot/name</pre>	第 2 の組の p4 have は、現在クライアント・ワークスペースにあるのがリビジョン 4であることを示します。

関連コマンド

ディポからクライアント・ワークスペースへ、ファイル・リビジョンをコピーする。 p4 sync

p4 help

概要

PERFORCE のオンライン・ヘルプ情報を表示します。

構文

```
p4 [g-opts] help
p4 [g-opts] help keyword
p4 [g-opts] help command
```

解説

p4 help は、名前を指定した コマンド または キーワード を説明するヘルプ画面を表示します。説明内容はこのマニュアルとよく似ていますが、テキストは開発者が書いています。

引数の付いていない p4 help は、利用可能な p4 help オプションをすべてリスト表示します。p4 help command は、名前を指定した コマンド に関するヘルプ情報を表示します。p4 help keyword は次のキーワードを引数としてとります。

コマンドとキーワード	意味	このマニュアル中の 関連箇所
p4 help simple	最も基本的な 8 つの PERFORCE コマンドに関する短い説明を表示します。	(特になし)
p4 help commands	すべての PERFORCE コマンドをリスト表示します。	「目次」
p4 help environment	PERFORCE 環境変数とその意味をリスト表示します。	「環境変数とレジストリ変数」
p4 help filetypes	PERFORCE のファイルタイプとその意味をリスト表示します。	「ファイルタイプ」
p4 help jobview	PERFORCE ジョブ・ビューを説明します。	p4 jobs の「解説」の項
p4 help revisions	PERFORCE リビジョン指定子を説明します。	「ファイル指定」
p4 help usage	すべての PERFORCE コマンドで利用できる 6 つのオプションをリスト表示します。	「グローバル・オプション」
p4 help views	PERFORCE のビューの意味を説明します。	「ビュー」

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	なし

関連コマンド

PERFORCE の現在の構成に関する情報を参照する。 p4 info

p4 info

概要

現在のクライアントおよびサーバに関する情報を表示します。

構文

```
p4 [g-opts] info
```

解説

p4 info コマンドは、PERFORCE クライアントおよび PERFORCE サーバに関する情報を表示します。

p4 info による出力の例を次に示します。:

```
User name: joe
Client name: joes_client
Client host: joes_workstation
Client root: /usr/joe/projects
Current directory: /usr/joe/projects/source
Client address: 192.168.0.123:1818
Server address: p4server:1666
Server root: /usr/depot/p4d
Server date: 2008/06/28 12:11:47 -0700 PDT
Server uptime: 752:41:33
Server version: P4D/FREEBSD/2008.1/156375 (2008/05/25)
Server license: P4Admin <p4adm> 20 users (expires 2009/01/01)
```

PERFORCE クライアント・プログラム (p4) のバージョンを確認するには、p4 -v を使用します。

オプション

`g_opts` 「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか?	ファイル引数に リビジョン範囲を使えるか?	最低限必要なアクセス・レベル
適用外	適用外	なし

関連コマンド

PERFORCE のヘルプ情報を表示する。 `p4 help`
 現在使用している PERFORCE クライアント・プログラムのバージョン情報を参照する。 `p4 -v`

p4 integrate

概要

ブランチまたはマージするファイルを作業状態にします。

構文

```
p4 [g-opts] integrate [options] fromFile[revRange] toFile
p4 [g-opts] integrate [options] -b branch [toFile[fromRevRange]...]
p4 [g-opts] integrate [options] -b branch -s fromFile[revRange]
[toFile...]

options: -c changelist# -d -Dflag -f -h -i -I -o -n -r -t -v
```

解説

1 つのファイルに対して行った変更を他のファイルにも伝達する必要がある場合には、p4 integrate でそのプロセスを開始します。このコマンドの最も単純な形式は p4 integrate fromFile toFile という形式で、これは PERFORCE サーバに fromFile の変更を toFile に伝達する必要があることを知らせ、次のような動作を行います。

- toFile がまだ存在していなければ、fromFile を toFile にコピーした上で、クライアント・ワークスペースで toFile をブランチのための作業状態にします。
- toFile が存在し、上記のように fromfile と共通の祖先を持つ場合、toFile を反映のための作業状態にします。この場合には、p4 resolve を用いて fromFile の変更のすべて、または一部を toFile に伝達するか、あるいは伝達しないかの選択をします。p4 resolve コマンドは、theirs として fromFile を、yours として toFile を、base として fromFile の共通の祖先をそれぞれ使用します。
- toFile と fromFile の両方が存在するが、toFile が fromFile と共通の祖先を持たない場合は、反映操作は拒否されます。-i オプションを使用すると、ベースレス・マージが強制されます。
- fromFile が最新リビジョンで削除されている場合、(そして fromFile と toFile 間の前回までの変更がすべて反映されていれば)、toFile をクライアント・ワークスペースで削除のための作業状態にします。

(利用可能なオプションの中には、この動作を変更するものもあります。詳しくは、次のセクション「オプション」をご覧ください。)

このプロセスは、ディポに対して p4 submit toFile を実行した時点で完了します。

複数のファイルを指定するには、fromFile の場合も toFile の場合もワイルドカードを使用します。fromFile で使用するワイルドカードと toFile で使用するワイルドカードは一致していなければなりません。PERFORCE は fromFile のパターンと toFile のパターンを比較し、fromFile/toFile のペアのリストを生成し、各ペアごとに反映操作を実行します。

p4 integrate fromFiles toFiles の形式では、fromFiles の変更を toFiles に伝達する必要が生じるたびに、fromFiles と toFiles の間のマッピングを指定する必要があります。これに代わる方法は、p4 branch で fromFiles と toFiles の間のマッピングを ブランチ・ビュー に保存し、fromFiles と toFiles の間で変更を伝達する必要が生じるたびに p4 integrate -b branchview を使用する、というものです。

オプション

比較的新しい反映オプションの中には、反映プロセスをより複雑にするものがあるので、反映用オプションは 基本オプション と 応用オプション に分けています。

反映用基本オプション

<p><code>-b branchname</code> <code>[toFiles...]</code></p>	<p><code>branchname</code> のブランチ・ビューにある <code>sourceFile/targetFile</code> のマッピングを利用してファイルの反映操作を行います。<code>toFiles</code> の引数が与えられていれば、<code>toFiles</code> によって指定されたパターンに合致するブランチ・ビューの反映先ファイルだけが含まれます。</p>
<p><code>-n</code></p>	<p>実際の反映操作は行わずに、このコマンドが実行することになる反映操作を表示します。</p>
<p><code>-v</code></p>	<p><code>toFiles</code> をクライアント・ワークスペースにコピーせずにファイルをブランチのための作業状態にします。</p> <p>このオプションが付いていなければ、<code>p4 integrate</code> は新たにブランチされた <code>toFiles</code> を <code>fromFiles</code> からクライアント・ワークスペースにコピーします。この <code>-v</code> (<i>virtual</i>) オプションが付いていると、PERFORCE は <code>toFiles</code> をクライアント・ワークスペースにコピーしません。ただし、それが必要なときには、<code>p4 sync</code> で取り出すことができます。</p>
<p><code>-c changelist#</code></p>	<p>指定した作業中チェンジリスト内の <code>toFiles</code> を、ブランチ作成、反映、または削除のための作業状態にします。</p> <p>このオプションが付いていなければ、ファイルはデフォルトのチェンジリスト内で作業状態にされます。</p>
<p><code>g_opts</code></p>	<p>「グローバル・オプション」のセクションをご覧ください。</p>

反映用応用オプション

<p><code>-b branchname -s fromFile[RevRange]</code> <code>[ToFiles...]</code></p>	<p>最も単純な形式 <code>p4 integrate -b branchname -s fromFile</code> では、<code>branchname</code> のブランチ・ビューに含まれる反映元/反映先のマッピングを用いてファイルの反映を行うことができます。ただし、反映元ファイルは <code>fromFile</code> で指定されたパターンに合致するものに限られます。</p> <p>もっと複雑な形式で <code>fromFile</code> も <code>toFile</code> も指定すると、反映は両方向に実行されます。まず、反映は <code>fromFile</code> から <code>toFile</code> へ行われ、次に <code>toFile</code> から <code>fromFile</code> へ行われます。<code>p4 integrate</code> のこのバリエーションは、PERFORCE の Windows クライアント P4WIN に必要な機能を与えるために作成されたものですが、形式が複雑なので使用することはまれです。</p>
<p><code>-b branchname -r [toFiles...]</code></p>	<p>ブランチ・ビューのマッピングを逆にし、反映先ファイルから反映元ファイルへの反映を行います。</p>
<p><code>-d</code></p>	<p><code>-d</code> オプションを付けると、削除済みのリビジョンに関する反映操作を行うことができます。<code>-d</code> は、<code>-Dt</code>、<code>-Ds</code> および <code>-Di</code> の意味を含みます。</p> <p>反映先のファイルが削除された状態で反映元のファイルが変更された場合 (または反映元のファイルが変更された後に削除された場合)、<code>-d</code> を使用すると反映元のファイルの変更は反映先のファイルの最新リビジョンに再ブランチされます。</p> <p>反映元のファイルが削除された状態で反映先のファイルが変更された場合、<code>-d</code> を使用すると反映先のファイルが削除されます。</p> <p>ソース・ファイルが削除後に再度追加された場合、<code>-d</code> を使用するとファイルを削除する前のリビジョンも含め、ファイルの未処理のリビジョンがすべて反映されます。</p> <p><code>-d</code> オプションを付けなければ、反映されていない変更は削除済みファイルにマージされない可能性があります。</p>

-Dt	-Dt オプションは削除された反映先ファイルに関する反映を行います。反映元ファイルは削除された反映先にブランチされます。
-Ds	-Ds オプションは削除された反映元ファイルに関する反映を行います。反映元ファイルが削除されている場合には、更新された反映先ファイルもすべて削除されます。
-Di	-Di オプションは、反映操作のベースの検索時に、ソース・ファイルが削除され再追加されたという事実を無視します。
-f	<i>fromFile</i> および <i>toFile</i> のすべてのリビジョンに対して、たとえその一部のリビジョンが過去に反映操作をされていても、反映を強制します。リビジョン範囲を指定して使用するのが最適です。
-h	反映操作の前に、反映先ファイルを最新リビジョンに同期する処理を、自動的には行いません。現在のクライアント・ワークスペースにあるリビジョンを使用します。
-i	<i>toFile</i> と <i>fromFile</i> が共通の祖先を持たない場合でも、最初のリビジョンを <i>base</i> として使用して反映操作を実行します。
-I	-iと同等です。-Iオプションは、互換性のために存在します。
-o	-oオプションは、後続の衝突解決で 사용되는ベース・ファイルの名前とリビジョンを出力します(衝突解決が必要な場合)。
-t	反映元ファイルのファイルタイプを反映先ファイルに伝達します。 (新たにブランチされるファイルは常に反映元ファイルのファイルタイプを使用しますが、-tを付けていないと、反映先ファイルはそれまでの反映先ファイルのファイルタイプを保持します。)

使用上の留意点

ファイル引数に リビジョン指定子を使えるか?	ファイル引数に リビジョン範囲を使えるか?	最低限必要なアクセス・レベル
使用可	使用可	open

- *FromFiles* は通常、反映元ファイル、*toFiles* は通常、反映先ファイルと呼ばれます。
- p4 integrate の動作対象となる *toFiles* は、必ず p4 client のワークスペース・ビューに含まれていなければなりません。
- デフォルトでは、p4 integrate でブランチまたは反映のための作業状態にされたファイルは、クライアント・ワークスペースでは読み取り専用です。これらのファイルは、サブミット前に、p4 edit であらためて編集のための作業状態にすることができます。
- p4 integrate を用いれば、ファイルをリネームすることができます。その方法は p4 rename の「解説」の項で説明しています。
- p4 integrate は p4 integ と略記することができます(次の「使用例」ではこの省略形が使用されています)。
- PERFORCE は、*fromFile* から *toFile* への反映が行われるたびに、その反映の結果を記述する 反映記録 をデータベースに生成します。この反映記録には、*fromFile* および *toFile* の名前、*toFile* に反映された *fromFile* のリビジョン、*toFile* の新しいリビジョン番号、反映時にとられたアクション(操作)が含まれます。反映操作について詳しくは、「p4 integrated」のセクションをご覧ください。

使用例

```

p4 integ //depot/dev/... //depot/rel2/... //depot/dev/... のすべてのファイルを //depot/rel2/... の対応するファイルにブランチまたはマージします。
//depot/rel2/... 内に対応するファイルが存在しない場合は、このコマンドで生成されます。

p4 integ -b rel2br ブランチ・ビュー rel2br に含まれるすべての fromFiles をそのブランチ・ビューのマッピングに従い、対応する toFiles にブランチまたはマージします。

p4 integ -b rel2br //depot/rel2/headers/... ブランチ・ビュー rel2br の fromFiles を //depot/rel2/headers/... にマッピングし、toFiles にブランチまたはマージします。

p4 integ -b rel2br -r //depot/rel2/README ブランチ・ビュー rel2br のマッピングに従って //depot/rel2/README を toFile から fromFile へブランチまたはマージします。

```

関連コマンド

ブランチ仕様を生成または編集する。	p4 branch
既存のブランチ仕様のリストを参照する。	p4 branches
すでに実行されサブミットされた反映のリストを参照する。	p4 integrated
p4 integrate でファイルを作業状態にした後、1つのファイルから別のファイルへ変更を伝達する。	p4 resolve
特定のファイルに対して実行されたあらゆる反映の履歴を参照する。	p4 filelog

p4 integrated

概要

サブミット済みの反映を表示します。

構文

```
p4 [g-opts] integrated [ -r ] [ -b branch ] file...
```

解説

p4 integrated コマンドは、選択されたファイル反映履歴を次の形式で表示します。

```
file#revision-range - integrate-action partner-file#revision-range
```

ここで

- *file* は、p4 integrated に与えられたファイル引数です。
- *partner-file* は、前記ファイルの反映元または反映先となったファイルです。
- *integrate-action* は、p4 resolve のプロセス中にユーザが行ったアクションの記述で、次のいずれかになります。

反映操作	p4 Resolve のプロセス中にユーザが行ったアクション
branch from	<i>file</i> が事前に存在せず、 <i>partner-file</i> のコピーとして生成された。
branch into	<i>partner-file</i> が事前に存在せず、 <i>file</i> のコピーとして生成された。
merge from	<i>file</i> が <i>partner-file</i> から反映され、 <i>merge</i> が受け入れられた。
merge into	<i>file</i> が <i>partner-file</i> に反映され、 <i>merge</i> が受け入れられた。
copy from	<i>file</i> が <i>partner-file</i> から反映され、 <i>theirs</i> が受け入れられた。
copy into	<i>file</i> が <i>partner-file</i> に反映され、 <i>theirs</i> が受け入れられた。
ignored	<i>file</i> が <i>partner-file</i> から反映され、 <i>yours</i> が受け入れられた。
ignored by	<i>file</i> が <i>partner-file</i> に反映され、 <i>yours</i> が受け入れられた。
delete from	<i>file</i> が <i>partner-file</i> から反映された。 <i>partner-file</i> は事前に削除されていた。
delete into	<i>file</i> が <i>partner-file</i> に反映された。 <i>file</i> は事前に削除されていた。
edit from	<i>file</i> が <i>partner-file</i> から反映され、 <i>file</i> は p4 resolve のプロセス中に編集された。これによってユーザは、この変更を反映し直すべきか判断できる。例えば、自動的な変更 (merge from) は反映し直す必要はないが、resolve 中に行われたユーザの編集 (edit from) は反映させる必要がある、といった判断が可能になる。(PERFORCE 2001.1 以降)
edit into	<i>file</i> が <i>partner-file</i> に反映され、 <i>partner-file</i> はサブミット前に編集のために再度作業状態にされた。(PERFORCE 99.2 以降)
add into	<i>file</i> が事前に存在しなかった <i>partner-file</i> に反映され、 <i>partner-file</i> はサブミット前に追加のために再度作業状態にされた。(PERFORCE 99.2 以降)

ファイル *toFile* がファイル *fromFile* から反映され、*toFile* と *fromFile* がともに p4 integrated の *filepattern* 引数に合致していれば、個々の反映操作は p4 integrated の出力中に 2 回リストされます。すなわち、前記の *from* フォーム内、および *into* フォーム内にそれぞれ 1 回ずつリストされます。

任意指定の *-b branch* オプションが使用されると、ブランチ・ビューにおいて反映元から反映先に反映されたファイルのみが表示されます。

任意指定の `-r` オプションが指定されると、ブランチ・ビューのマッピングが逆にされます。このオプションは `-b branch` オプションと一緒に使用する必要があります。

オプション

`g_opts` 「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	list

関連コマンド

ファイル衝突未解決の反映のリストを見る。	<code>p4 resolve -n</code>
ファイル衝突は解決されたが、まだサブミットされていない反映のリストを見る。	<code>p4 resolved</code>
反映を実行する。	<code>p4 integrate</code>
特定ファイルのすべてのリビジョン（その特定ファイルの反映元となったすべてのファイルを含む）についてとられたアクションを参照する。	<code>p4 filelog [-i] file</code>

p4 job

概要

欠陥、改善要求、またはその他のジョブ仕様の生成または編集を行います。

構文

```
p4 [g-opts] job [ -f ] [ jobName ]
p4 [g-opts] job -d jobName
p4 [g-opts] job -o [ jobName ]
p4 [g-opts] job -i [ -f ]
```

解説

ジョブとは、ディポのファイルに対して実行する必要がある作業を記述したものです。場合によっては、バグの記述（例えば、「スクロール機構が正常に動作していない」）、あるいは改善要求（例えば、「……の動作の発生を強制するオプションの追加を望む」）などになることもあります。

ジョブは、システムに対する変更を任意のテキストとして記述するという点でチェンジリストの記述と似ていますが、チェンジリストは完了した作業を記述するのに対し、ジョブは行う必要がある作業を開発者に伝えるものです。

ジョブは、p4 job によって表示されるフォームで生成・編集します。ユーザはそのフォームに、ジョブを記述するテキストを、バグの重要度やバグ修正を担当する開発者といった情報とともに入力します。PERFORCE スーパー・ユーザは、p4 jobspec でジョブ・フォームのフィールドを変更できるので、1つのジョブを構成するフィールドは個々の PERFORCE サーバによって異なってきます。

p4 job が引数なしで実行されると、jobNNNNNN という新しいジョブが生成されます。NNNNNN は6桁の連番です。このジョブの名前は、エディタを終了する前にフォーム上で変更することができます。p4 job を jobname の引数付きで実行すると、その名前のジョブが生成され、すでに同じ名前のジョブが存在している場合には、そのジョブの編集を行うことができます。

ジョブが生成されたら、そのジョブを p4 fix、p4 change、または p4 submit で修正するチェンジリストにリンクさせることができます。ジョブがチェンジリストにリンクされている場合には、通常、ジョブの状態は closed になります。（詳しくは下記「使用上の留意点」をご覧ください）

フォーム・フィールド

以下に紹介するのは、デフォルトのジョブ・フォームに表示されるフィールドです。ジョブを記述するフィールドは、PERFORCE スーパー・ユーザによって変更されることがあるので、実際に表示されるフォームのフィールドは、これとは大幅に異なっている可能性があります。

フィールド名	タイプ	説明
Job:	書き込み可能	ジョブの名前です。新しいジョブの場合、このフィールドの値は new です。フォームを閉じると、new がジョブ発番シーケンスの次の6桁の連番 NNNNNN を含む jobNNNNNN という名前に置き換わります。 このフィールドの文字を入力し直すことにより、ジョブに任意の名前を付けることもできます。

フィールド名	タイプ	説明
Status:	書き込み可能、既定値	このフィールドの値は open、closed、または suspended でなければなりません。ジョブがチェンジリストにリンクされていれば、このフィールドの値は、チェンジリストがサブミットされたときに closed に設定されます。
User:	書き込み可能	ジョブを生成したユーザの名前です。
Date:	書き込み可能	ジョブが生成された日付です。
Description:	書き込み可能	ジョブに関するコメントを記述することができます。

オプション

-d jobname	関連する作業中またはサブミット済みの修正がない場合に限り、ジョブ <i>jobname</i> を削除します。
-f	強制オプションです。PERFORCE 管理者に読み取り専用フィールドの編集を認めます。
-i	エディタを立ち上げずに標準入力からジョブ・フォームを読み込みます。
-o	エディタを立ち上げずに標準出力へジョブ・フォームを書き出します。
<i>g_opts</i>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	open

- PERFORCE スーパー・ユーザが p4 jobspec でフィールド ID# 102 ([Status:]フィールド) を削除した場合、PERFORCE は、ジョブがリンクされているチェンジリストがサブミットされてもそのジョブを closed にすることができなくなります。詳しくは、p4 jobspec の項および『システム管理者ガイド』をご覧ください。
- ジョブが生成または変更されると、PERFORCE はジョブにインデックスを付け、p4 jobs -e が速やかにそのジョブを見つけられるようにします。インデックス・キーは *word*、*fieldname* です。*word* は大文字 / 小文字を区別せず、句読法 (区切り文字やスペース) も使用できます。日付のフィールドの値は 1970 年 1 月 1 日 00:00:00 からの秒数として保存されます。

使用例

p4 job	新しいジョブを生成します。デフォルトでは、そのジョブの名前は jobNNNNNN になります。
p4 job job000135	ジョブ job000135 を編集します。

関連コマンド

ジョブのすべて、または一部をリスト表示する。	p4 jobs
ジョブを既存のチェンジリストにリンクする。	p4 fix
ジョブとそれを修正するチェンジリストのリストを参照する。	p4 fixes
作業中チェンジリストに対するジョブの追加または削除を行う。	p4 change
サイトにおけるジョブの形式を変更する (スーパー・ユーザのみ使用可能)。	p4 jobspec
サイトにおけるジョブの形式に関する情報を読み出す。	p4 jobspec -o

p4 jobs

概要

PERFORCE サーバに既知のジョブをリスト表示します。

構文

```
p4 [g-opts] jobs [-e jobview] [-i] [-l] [-r] [-m max] [file[rev] ...]
p4 jobs -R
```

解説

p4 jobs は、引数なしで実行されると、サーバに保存されているすべてのジョブをリスト表示します。このコマンドの出力は、オプションや引数で様々な基準を指定することにより、制限することができます。ファイル・パターンを指定すれば、リスト表示されるジョブは特定のファイルに影響を及ぼすチェンジリストにリンクされたものだけに制限されます。-e オプションを使用すれば、リスト表示されたジョブをさらに特定の語を含むものだけに制限することができます。

ジョブは、その名前をキーとして、英数字の昇順（あるいは、-r オプションを使えば、降順）に、1 行に 1 つずつ表示されます。各行の形式は次のとおりです。

```
jobname on date by user *status* description
```

description は、-l (long) オプションが使用されていない限り、先頭から 31 字までに制限されます。

date、user、status、または description のフィールドのいずれかが、PERFORCE スーパー・ユーザにより p4 jobspec で削除されている場合、そのフィールドの値は各ジョブの出力に表示されません。

ジョブのリストを、特定ファイルに関連するチェンジリストによって修正されたものだけに制限するには、p4 jobs filespec を使用します。ファイルまたはファイル・パターンの指定には、リビジョン指定子またはリビジョン範囲を含めることができます。

オプション

-e jobview	jobview によって指定された基準に合致するジョブだけをリスト表示します。ジョブ・ビューについては、下記「使用上の留意点」をご覧ください。
-i files...	指定ファイルに反映されたファイルに関連するチェンジリストによって修正されたジョブも含めます。
-l	各ジョブの完全な記述を出力します。
-m max	英数字の昇順にソートし、先頭から max の数だけのジョブを表示します。-r オプションを付ければ、英数字の降順にソートし、先頭から max の数だけのジョブが表示されます。
-r	ジョブを、その名前をキーとして、英数字の降順で表示します。
-R	ジョブ・テーブルを再構築し、各ジョブにあらためてインデックスを付けます。この操作は、バージョン 98.2 より前のバージョンからのアップグレード、および 99.1 から 2001.1 より新しいバージョンへのアップグレードを行う際に必要となります。また、既存のジョブを検索する際に、句読法（区切り文字やスペース）を使用したいときにも必要となります。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	list

ジョブ・ビュー

p4 jobs -e jobview を使用すれば、リスト表示するジョブを特定の語を含むジョブだけに制限することができます。検索する語は特定のフィールドについて指定することもできますし、ジョブのテキストについて指定することもできます。また、通常のテキストについて検索語を指定する場合ほど多くのオプションは使えませんが、日付のフィールドの値を検索語に指定することもできます。

検索語の適合は大文字 / 小文字の区別なく調べられます。スペースによって区切られた文字列は、(句読法が埋め込まれていても) すべて単語としてインデックスされます。

検索は単語単位で行われ、フレーズ単位の検索は行われません。対象となるのは英数字のテキストだけで、区切り文字は含まれません。(ジョブ・ビューでは、区切り文字は無視されます。)

ジョブ・ビューを 'word1 word2 ... wordN' と記述すれば、word1 から wordN までのすべてをいずれかのフィールドに含んでいるジョブが検索されます。

ジョブ・ビューの検索語の間のスペースは、論理演算子 AND として機能します。前記の検索語のいずれかを含むジョブを検索するには、検索語を “|” で区切ります。

アンパーサンド (&) も論理演算子 AND として使えます。論理演算子は (優先権の高い順に) &、|、スペースの順に結合します。結合の優先度を変えたい場合には、() を使用します。

検索結果は、ジョブ・ビュー・シンタックス “fieldname=value” を用いて特定のフィールドの値を指定することにより、絞り込むことができます。value は英数字 (および日本語文字) および句読法を含んだトークンでなければなりません。

ワイルドカード [*] を使えば、特定の文字列を含む単語を指定することもできます。ジョブ・ビューに “fieldname=string*” と記述すれば、指定したフィールドに “string”、 “stringy”、 “stringlike” などの文字列を含むジョブを検索することができます。

日付のフィールドは、yyyy/mm/dd あるいは yyyy/mm/dd:hh:mm:ss の形式で記述することができます。特定の時刻が与えられていない場合には、同値演算子 (=) で日付の適合するジョブが検索されます。

通常の比較演算子 (=、>、<、>=、<=) も利用できます。

さらに、NOT 演算子 (^) も否定の比較演算に利用できます。(詳しくは、下記「制約」をご覧ください。)

検索で使用する演算子を文字として含む単語を検索するときは、バックスラッシュ (\) でその文字をエスケープしてください。

これらの演算子の働きは、比較するジョブ・フィールドのタイプによって異なります。

フィールドタイプ	比較演算子の働き
word	等号 (=) は、word フィールドの値が厳密に一致することを求めます。比較演算子は ASCII コード表に従って比較を行います。
text	等号 (=) は、与えられた値が指定フィールドのどこかに見つかることを求めます。比較演算子は、指定フィールドの <u>いずれかの</u> 語が与えられた条件に適合すればよいので、ここではあまり役に立ちません。 例えば、あるジョブに gui bug というフレーズだけを含む ShortDescription という text フィールドがあり、ジョブ・ビューを “ShortDesc<filter” としたとすると、bug<filter なので、このジョブも指定に適合することになります。
line	上記のフィールドタイプ text の場合と同じです。
select	等号 (=) は、指定フィールドの値が指定の語のときに適合と見なします。比較演算子は ASCII コード表に従って比較を行います。
date	日付は年代順に比較されます。特定の時刻が与えられていない場合は、演算子 =、<=、>= で日付単位の比較が行われます。

フィールドタイプがよくわからない場合は、サイトで使用中のジョブ仕様を出力する `p4 jobspec -o` を実行します。p4 jobspec の [Fields:] フィールドには、ジョブ・フィールドの名前とデータタイプが含まれています。個々のタイプについては、「p4 jobspec」の項をご覧ください。

その他の使用上の留意点

- p4 user のフォームには [JobView:] フィールドがあり、ジョブ・ビューを特定のユーザにリンクさせることができます。ユーザがこのフィールドにジョブ・ビューを入力すると、そのユーザが生成するチェンジリストにおいて、このフィールドのジョブ・ビューと適合するジョブが自動的に表示されます。そのチェンジリストによって修正するジョブをこのフォームに残しておき、そうでないジョブは削除する必要があります。
- p4 jobs は、その出力をジョブ名で英数字の昇順に並べ替え、その結果、ジョブの並び順は入力された日付の順序と一致します。ただし、PERFORCE で使用される標準的なジョブ名と異なるジョブ名を使用している場合には、この限りではありません。
- `-m max -r` は、英数字の昇順で最後から `max` の数のジョブを表示するのであって、新しいものから `max` の数のジョブを表示するわけではありません。しかし、(例えば、job001394 のような名前が付く) PERFORCE におけるデフォルトのジョブ命名法を使用していれば、英数字の昇順と入力日付の古い順が同じになります。
- テキストタイプのフィールドが値を含んでいるかどうかは、ワイルドカード「*」を使ってジョブ・ビューを “field=*” と記述することによって確認できます。field が空でないジョブがすべて検索されます。
- `-e jobview` オプションを用いてジョブを検索する場合、特にワイルドカード、論理演算子、() を使用する場合には、オペレーション・システムおよびコマンド・シェルの構文解析、引用、特殊文字のエスケープの動作に注意してください。

制約

- ジョブ・ビューは、空のフィールドを含むジョブの検索には利用できません。言い換えれば、既存のジョブからあるフィールドが削除されたとすると、該当フィールドはインデックスされていないことになり、その“削除済フィールド”の値に適合するジョブ・ビューは存在しません。

- ジョブ・ビュー内では、NOT 演算子 (^) は AND 演算子の後にしか使えません。つまり、“gui ^name=joe”、“gui&^name=joe”というジョブ・ビューは有効ですが、“gui|^name=joe”、“^name=joe”というジョブ・ビューは有効ではありません。
- 上記の制約を回避するには、ワイルドカード「*」が有効です。

例えば、文字列 “unwanted” を含まないすべてのジョブを検索するには、ジョブ・ビューを “job=* ^unwanted” と記述します。スペースより前の部分ですべてのジョブが選択され、文字列 “unwanted” を含まないすべてのジョブとの論理積 AND がとられます。

同様に、ジョブ・ビュー “field=*” は、*field* が空でない（つまり、job フィールドは空ではないと考えられる）あらゆるジョブと適合するので、空のフィールドを持つジョブは “job=* ^field=*” で検索することができます。

使用例

p4 jobs //depot/proj/file#1	//depot/proj/file のリビジョン 1 を含むチェンジリストに添付されたすべてのジョブをリスト表示します。
p4 jobs -i //depot/proj/file	//depot/proj/file のリビジョン、あるいは //depot/proj/file に反映されたファイルのリビジョンを含むチェンジリストに添付されたすべてのジョブをリスト表示します。
p4 jobs -e gui	いずれかのフィールドに単語 gui を含むすべてのジョブをリスト表示します。
p4 jobs -e "gui Submitted-By=joe"	いずれかのフィールドに単語 gui を含み、[Submitted-By:] フィールドに単語 joe を含むすべてのジョブをリスト表示します。
p4 jobs -e "gui ^Submitted-By=joe"	いずれかのフィールドに単語 gui を含み、[Submitted-By:] フィールドが joe <u>以外</u> の値をとっているすべてのジョブをリスト表示します。
p4 jobs -e "window*"	いずれかのフィールドに単語 “window”、“window.c”、“windows” を含むすべてのジョブをリスト表示します。引用符はローカル・シェルがコマンドライン上で “*” を拡張するのを防ぐために使用します。
p4 jobs -e window.c	いずれかのフィールドで window.c を参照しているすべてのジョブをリスト表示します。
p4 jobs -e "job=* ^unwanted"	どのフィールドにも単語 unwanted を含まないすべてのジョブをリスト表示します。
p4 jobs -e "(fast quick) &date>1998/03/14"	いずれかのフィールドに単語 fast または quick を含み、[date:] フィールドの値が 3/14/98 以降を示しているすべてのジョブをリスト表示します。
p4 jobs -e "fast quick" //depot/proj/...	いずれかのフィールドに単語 fast または quick を含み、//depot/proj のファイルに影響を及ぼしたチェンジリストにリンクされているすべてのジョブをリスト表示します。

関連コマンド

ジョブの生成、または既存のジョブの編集を行う。	p4 job
ジョブを特定のチェンジリストにリンクし、ジョブがそのチェンジリストによって修正されることを指示する。	p4 fix
すべてのジョブとそれがリンクされているチェンジリストをリスト表示する。	p4 fixes

特定のチェンジリストについて、それにリンクされているジョブも含め、すべての情報を参照する。 p4 describe

サーバで使用されているジョブの形式を変更する（スーパー・ユーザのみ）。 p4 jobspec

サイトで使用されているジョブの形式に関する情報を読み出す（全ユーザ）。 p4 jobspec -o

デフォルトのジョブ・ビューを設定することにより、そのジョブ・ビューに適合するジョブがすべての新しいチェンジリストに含まれるようにする。 p4 user

p4 jobspec

概要

ジョブ・テンプレートを編集します。

構文

```
p4 [g-opts] jobspec
p4 [g-opts] jobspec [-i]
p4 [g-opts] jobspec -o
```

解説

p4 jobspec コマンドによりフォームが呈示され、PERFORCE 管理者はジョブ・フィールドの編集、生成、削除、改良を実行できます。

p4 jobspec フォームのフィールド名とジョブのフィールド名を混同しないでください。p4 jobspec フォームのフィールドは、p4 jobs フォームのフィールドに関する情報の保存に利用されます。

フォーム・フィールド

フィールド名	説明
Fields:	<p>サイトのジョブにおける各フィールドの定義のリストで、1行に1フィールドずつ定義が表示されます。各行は、<i>code</i>、<i>name</i>、<i>datatype</i>、<i>length</i>、<i>persistence</i> という形式で表示されます。</p> <ul style="list-style-type: none"> • <i>code</i>: PERFORCE 内部でフィールドを識別する固有の整数です。106 ~ 199の整数でなければなりません。101~105のコードはPERFORCE専用です。詳しくは下記の「使用上の留意点」をご覧ください。 • <i>name</i>: フィールド名です。フィールド名はいつでも変更できますが、いったんジョブが生成されたら、コードは変更しないでください。 • <i>datatype</i>: フィールドのデータタイプです。設定値は次のとおりです。 <ul style="list-style-type: none"> • <i>word</i>: 任意の単語です。 • <i>date</i>: 日付 / 時刻です。 • <i>select</i>: 決められた単語の組の1つです。 • <i>line</i>: テキストの1行です。 • <i>text</i>: フィールド名の下に行から始まるテキストブロックです。 • <i>length</i>: このフィールドにアクセスした GUI クライアントにおける表示ボックスの推奨の長さです。PERFORCE クライアント・プログラムが自由に値を選択できるようにするためには0に設定します。 • <i>persistence</i>: 記述しているフィールドにデフォルト値があるか、入力必須か、読み取り専用か、などを設定します。設定値は次のとおりです。 <ul style="list-style-type: none"> • <i>optional</i>: 任意の値を取ることができ、消去も可能です。 • <i>default</i>: デフォルト値が与えられており、変更または消去が可能です。 • <i>required</i>: デフォルト値が与えられており、変更可能ですが、ユーザが値を入力しなければなりません。 • <i>once</i>: 読み取り専用です。このフィールドの値は、一度デフォルト値に設定されると変更されることはありません。 • <i>always</i>: 読み取り専用です。このフィールドの値は、ジョブが編集されたときに新しいデフォルト値に設定されます。変数 <i>\$now</i>、<i>\$user</i> を用いた場合にのみ有効で、ジョブが更新された日付と更新したユーザの名前を変更することができます。

フィールド名	説明
Values:	<p>フィールド名およびフィールド [select] の有効値のリストで構成されます。</p> <p>データタイプが select に設定されたフィールドをそれぞれ 1 行で表示します。各行には、フィールド名の後にスペースをはさんで（例えば下記のように）、スラッシュで区切った有効値のリストが表示されます。</p> <p>JobType bug/request/problem</p>
Presets:	<p>フィールド名および default、required、once、または always の属性を持つ各フィールドのデフォルト値のリストで構成されます。</p> <p>各行には、例えば下記のように、フィールド名とデフォルト値がスペースで区切られて表示されていなければなりません。</p> <p>JobType bug</p> <p>任意の 1 行の文字列または次の 3 つの組込変数のいずれかを使用することができます。</p> <ul style="list-style-type: none"> • \$user: ジョブを生成したユーザです。 • \$now: 現在の日付です。 • \$blank: フレーズ <enter description here> です。 <p>ジョブを登録するとき、そのジョブがシステムに追加される前にユーザは \$blank をデフォルト値とするジョブ仕様のフィールドを、入力しておかなければなりません。</p> <p>フィールド 102 に関して特に考慮すべき点については、下記の「使用上の留意点」をご覧ください。</p>
Comments:	<p>p4 job フォームの先頭に表示されるテキストのコメントです。すべての行がコメント・キャラクタ # で始まっているなければなりません。</p> <p>ユーザが PERFORCE の Windows クライアント P4WIN でジョブを入力する必要がある場合には、このコメントに関して特に考慮すべき点があります。下記の「使用上の留意点」をご覧ください。</p>

オプション

-o	ジョブ仕様のフォームを標準出力へ書き出します。
-i	ジョブ仕様のフォームを標準入力から読み取ります。
<i>g_opts</i>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	admin、あるいは -o オプションを使うためには list

- 101 ~ 105 のフィールドは、変更、リネーム、再定義しようとししないでください。これらのフィールドは PERFORCE によって使用され、削除も変更もしてはいけません。p4 jobspec はジョブに新しいフィールド（すなわち、106 以上）を追加する目的のみに使用してください。

フィールド 101 は PERFORCE に必要で、リネームすることも削除することもできません。

フィールド 102 ~ 105 は、PERFORCE のクライアント・プログラム専用です。これらのフィールドをリネームまたは削除することは可能ですが、望ましくありません。PERFORCE クライアント・プログラムは、チェンジリストがサブミットされるたびに、必ずフィールド 102 (Status: フィールド) の値を closed (またはフィールド 102 の Presets: に定義されたその他の値) に設定します。これは、システム管理者がフィールド 102 を、closed を許容

値として含まないフィールドとして使用するように再定義しても変わりません。むしろ、このような再定義は予測不可能な混乱を生じる危険性があります。

- [Comments:] フィールドの情報は、ユーザにとってジョブ・フォームの記入方法を説明する唯一の参考にできる情報です。コメントは、必ず完全かつ理解可能なものにしてください。
- 各フィールドのコメントの1行目は、PERFORCEのWindowsクライアントP4WINにおいて、フィールド上でポップアップされる説明にも利用されます。各フィールドのコメントの1行目は、そのためにも意味のわかるものしておく必要があります。
- ジョブ状態のフィールド（フィールド102）のPresets:には、p4 fix、p4 change -s、p4 submit -sの各コマンドに対してデフォルトの修正ステータスを示すための特殊な構文があります。

デフォルトでは、p4 fix、p4 change、またはp4 submitを使用すると、ジョブの状態はclosedに設定されます。デフォルトの修正ステータスをclosedから他のfixStatusに変更するには（Values:フィールドに有効なselect設定値としてfixStatusを定義済みであると想定して）、フィールド102（ジョブの状態）のPresets:フィールドでjobStatus,fix/fixStatusという構文を使用します。p4 fix、p4 change、p4 submitの動作を変更し、ジョブの状態を未変更のままにするには、特殊なfixStatusであるsameを使用します。

- 『システム管理者ガイド』の「ジョブ仕様」の章には、ジョブ仕様のサンプルが紹介されています。

関連コマンド

ジョブの生成、編集、またはビューを行う。	p4 job
ジョブをチェンジリストに添付する。	p4 fix
ジョブをリスト表示する。	p4 jobs
特定のチェンジリストに添付されたジョブ、または特定のジョブに添付されたチェンジリストをリスト表示する。	p4 fixes

p4 label

概要

ラベル仕様およびそのビューの生成または編集を行います。

構文

```
p4 [g-opts] label [ -f -t template ] labelname
p4 [g-opts] label -o [ -t template ] labelname
p4 [g-opts] label -d [ -f ] labelname
p4 [g-opts] label -i [ -f ]
```

解説

p4 label を使用して新しいラベル仕様の生成または既存のラベル仕様の編集を行います。ラベル名が必要です。

p4 label を実行すると、ラベルに含めることが認められている一連のファイルを管理するマッピングを構成することができます。ラベルを構成した後、p4 labelsync または p4 tag を使用して、そのラベルでファイルにタグ付けします。

ラベルは自動または静的に設定することができます。自動ラベルは、[View:] フィールドおよび [Revision:] フィールドで指定されたリビジョンを参照します。静的ラベルは、p4 labelsync または p4 tag によりラベルにタグ付けされた特定のリビジョンのみを参照します。

unlocked のラベルの Owner: だけが、p4 labelsync または p4 tag を使用して、そのラベルでファイルにタグ付けすることができます。

フォーム・フィールド

フィールド名	タイプ	説明
Label:	読み取り専用	コマンドで指定されたラベル名です。
Owner:	書き込み可能、任意選択	ラベルの所有者です。デフォルトでは、ラベルを生成したユーザです。ラベルの所有者だけが、どのファイルにそのラベルでタグ付けするかを変更することができます。
Update:	読み取り専用	ラベル仕様が最後に更新された日付です。
Access:	読み取り専用	ラベル上の p4 labelsync の実行によって、またはラベルのリビジョン指定子 @label を持つファイルに対してその他の参照を行うことによって、ラベルが最後にアクセスされた日付と時刻です。
Description:	書き込み可能、任意選択	ラベルの目的を記述します。任意入力です。
Options:	書き込み可能	locked または unlocked です。ラベルが locked の場合、そのラベルでタグ付けされたファイルのリストを、p4 labelsync で変更することができません。

フィールド名	タイプ	説明
Revision:	書き込み可能	自動ラベル用の、任意指定のリビジョン仕様。 リビジョン番号の指定に # 文字を使用する場合、その文字を引用符で囲み、# がフォーム内のコメント・フィールドではなくリビジョン指定子として解析されるようにしなければなりません。
View:	書き込み可能	ラベルでタグ付けすることができるディポ・ファイルのリストです。p4 labelsync が実行されるまでは、実際にはどのファイルもタグ付けされません。 1 つのファイルの集合を別のファイルの集合にマッピングするクライアント・ビューやブランチ・ビューとは異なり、ラベル・ビューは単なるディポ・ファイルのリストです。詳しくは、「ビュー」のセクションをご覧ください。

オプション

-d [-f]	名前の付いた unlocked のラベルを削除します。-f オプションは locked のラベルも強制的に削除します。(ロックされたラベルを削除するには、admin もしくは super の権限が必要です。)
-i	エディタを起動しないで、標準入力からラベル定義を読み込みます。
-o	エディタを起動しないで、標準出力へラベル定義を書き出します。
-f	フィールド [Update:] (更新) の日付の設定を可能にします。-i オプションまたは -t オプションを付けて同じ目的に使用することもできます。
-t <i>template</i>	ラベル <i>template</i> のビューおよびオプションを、そのラベルのフィールド [View:] (ビュー) および [Option:] (オプション) にコピーします。
<i>g_opts</i>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか?	ファイル引数に リビジョン範囲を使えるか?	最低限必要なアクセス・レベル
適用外	適用外	open

関連コマンド

クライアント・ワークスペースのリビジョンにラベルでタグ付けする。	p4 labelsync
システムが認知しているラベルをすべてリスト表示する。	p4 labels
ラベルを作成して、そのラベルでファイルにタグ付けする。	p4 tag

p4 labels

概要

定義されたラベルのリストを表示します。

構文

```
p4 [g-opts] labels [ -u user ] [ -e namefilter -m max ] [ file[revrange] ]
```

解説

p4 labels は、PERFORCE サーバが認識している全ラベルのリストを次のフォーマットで表示します。

```
Label labelname date description
```

特定のファイルにタグ付けされた静的ラベルのリストを見るには、ファイル・パターンを指定し、必要に応じてリビジョン範囲を指定します。(自動ラベルは特定のリビジョン範囲のラベル・ビューにあるすべてのファイルを参照するため、p4 labels と共にファイル・パターンを使用すると、自動ラベルは表示されません。)

ラベルの出力を max までに限定するには、-m max オプションを使用します。

namefilter というパターンに一致する名前を持つラベルに出力を限定するには、-e namefilter オプションを使用します。

指定されたユーザが所有しているラベルに出力を限定するには、-u user オプションを使用します。

オプション

```
-m max          max までのラベルのみを一覧表示します。
-e namefilter   namefilter に一致するラベルのみを一覧表示します。
-u user         user が所有しているラベルのみを一覧表示します。
g_opts         「グローバル・オプション」のセクションをご覧ください。
```

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	list

- 特定のラベルにタグ付けされているファイルのリストを見るには、p4 files @labelname を使用します。

使用例

```
システムの全ラベルをリスト表示します。          p4 labels
file.c のいずれかのリビジョンを含む全ラベルをリスト表示します。  p4 labels file.c
file.c のリビジョン #3～#5 を含むラベルだけをリスト表示します。  p4 labels file.c#3,5
```

関連コマンド

ラベルを作成して、そのラベルでファイルにタグ付けする。	<code>p4 tag</code>
ラベル仕様の生成または編集を行う。	<code>p4 label</code>
1つのラベルに含まれるファイルの追加、削除、変更を行う。	<code>p4 labelsync</code>
1つのラベルに含まれるファイルのリストを参照する。	<code>p4 files @labelname</code>

p4 labelsync

概要

ラベルを現在のクライアント・ワークスペースの内容と同期させます。

構文

```
p4 [g-opts] labelsync [-a -d -n] -l labelname [file[revRange]...]
```

解説

p4 labelsyncは、クライアントに最後に同期した各ファイルのリビジョンをそのラベル名でタグ付けすることによって、指定したラベルにクライアント・ワークスペースの現在の内容を反映させます。その結果、ラベル名を @label としてリビジョン指定内で使用し、ラベルでタグ付けされたファイルのリビジョンを参照することができます。

ファイル引数を付けない場合、p4 labelsync によってラベルにクライアント・ワークスペースの内容が反映され、ラベルでタグ付けされたファイルの集合が追加、削除、更新されます。

ファイルが指定されている場合、p4 labelsync は指定されたファイルに対してのみタグを更新します。ファイル引数にリビジョン指定子が含まれていれば、そのリビジョンがクライアント・ワークスペースに存在するリビジョンの代わりに使用されます。ファイル引数にリビジョン範囲が含まれていれば、その範囲内の最新リビジョンだけが使用されます。

unlocked のラベルの Owner: だけが、p4 labelsync を使用して、そのラベルでファイルにタグ付けすることができます。

フィールド [Options:] (オプション:) が locked に設定されているラベルを、p4 labelsync で更新することはできません。

オプション

-a	ラベル付けされているファイルの一部が最新リビジョンにおいて削除されていても、ファイル・パターン引数に適合するファイルにラベルを追加します。
-d	指定されたファイルからラベルによるタグを削除します。
-l labelname	ファイル・リビジョンに適用されるラベルを指定します。
-n	p4 labelsync が実行する予定の動作を、実際にその動作を実行することなく表示します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	list

- デフォルトでは、p4 labelsync は、クライアント・ワークスペースに最後に同期されたファイルのリビジョンに対して動作します。ファイルの最新リビジョン（または指定した範囲で最も大きい番号のリビジョン）にタグ付けするには、p4 tag を使用します。

関連コマンド

ラベルの生成または編集を行う。	p4 label
システムが認知しているラベルをすべてリスト表示する。	p4 labels
ラベルを作成して、そのラベルでファイルにタグ付ける。	p4 tag

p4 license

概要

ライセンス・ファイルを更新または表示します。

構文

```
p4 [g-opts] license [ -o ]
p4 [g-opts] license [ -i ]
```

解説

p4 license コマンドにより、PERFORCE 管理者は PERFORCE ライセンス・ファイルを更新または表示することができます。このコマンドを実行するには、PERFORCE サーバのルート・ディレクトリ内に既に有効なライセンス・ファイルが存在する必要があります。

p4 license を使用するとライセンス許可ユーザは PERFORCE サーバに追加することができ、サーバを停止してライセンス・ファイルを手動でサーバ・ルートにコピーする必要はありません。

PERFORCE から取得した新しいライセンス・ファイルのほとんどは、サーバの IP アドレスが変更されていない限り、p4 license を使用してインストールが可能です。サーバの IP アドレスが変更されている場合は、PERFORCE サーバを停止し、ライセンス・ファイルを手動で所定の場所にコピーし、サーバを再起動する必要があります。

オプション

-o	現在のライセンス・ファイルを標準出力に表示します。
-i	新しいライセンス・ファイルを標準入力から読み込みます。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	super

使用例

p4 license -o	現在のライセンス・ファイルを標準出力に表示します。
p4 license -i	新しいライセンス・ファイルを標準入力から読み込みます。

p4 lock

概要

作業状態のファイルをロックして、チェンジリストがサブミットされないようにします。

構文

```
p4 [g-opts] lock [-c changelist#] [file ...]
```

解説

ファイルをロックすると、他のユーザがそのファイルへの変更をサブミットするのを防げます。ファイルがすでに他のユーザによってロックされていると、p4 lock は失敗します。ファイルをロックしたユーザがそのファイルをサブミットすれば、ロックは解除されます。

このコマンドは、通常特定のファイル引数を付けて実行されます。ファイル引数が与えられていない場合は、デフォルト・チェンジリスト内で作業状態になっているファイルがすべてロックされます。-c changelist# オプションが付いていると、チェンジリスト changelist# で作業状態になっているファイルのうち、指定されたファイル・パターンに適合するものがすべてロックされます。

オプション

-c changelist#	チェンジリスト changelist# に含まれるファイルだけをロックします。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	write

関連コマンド

ロックされているファイルをアンロックする。	p4 unlock
作業状態でロック中のファイルを表示する。(UNIX)	p4 opened grep "*locked*"

p4 logger

概要

変更されたジョブおよびチェンジリストをレポートします。

構文

```
p4 [g-opts] logger [-c sequence#] [-t countername]
```

解説

p4 logger コマンドは PERFORCE を呼び出す外部プログラムで使用するものです。

PERFORCE と欠陥追跡ツールとの連携 (P4DTI) では p4 logger を使用します。

オプション

-c <i>sequence#</i>	指定のシーケンス番号以後に生じたすべてのイベントを一覧表示します。
-t <i>countername</i>	指定のカウンタ番号以後のすべてのイベントを一覧表示します。
-c <i>sequence#</i> -t <i>countername</i>	指定のカウンタを指定のシーケンス番号で更新し、ログをクリアします。このオプションは、カウンタ名の指定に関係なくログをクリアするので、1 人のユーザしか使用できません。
<i>g_opts</i>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか?	ファイル引数にリビジョン範囲を使えるか?	最低限必要なアクセスレベル
適用外	適用外	review

- p4 logger コマンドはエンドユーザ用ではありません。外部の欠陥追跡システムへの情報伝達をサポートするためのものです。

関連コマンド

特定のファイルをレビューするためにサブスクライブしているユーザを一覧表示する。	p4 reviews
PERFORCE カウンタの値を設定する、または読み取る。	p4 counter
特定のチェンジリストに関する全情報を表示する。	p4 describe
特定の条件に適合するすべてのチェンジリストを一覧表示する。	p4 changes

p4 login

概要

チケットを取得して、PERFORCE サーバにログインします。

構文

```
p4 [g-opts] login [ -a -p ] [ user ]
p4 [g-opts] login [ -s ]
```

解説

p4 login コマンドは、ユーザを認証し、PERFORCE サーバとのセッションを表すチケットを作成します。認証されたユーザは、チケットが期限切れになるか、そのユーザが p4 logout コマンドを発行するまで、PERFORCE サーバにアクセスすることができます。

デフォルトでは、チケットは 12 時間の間有効です。

すべての IP アドレスに対して有効なチケットを取得する (PERFORCE を複数のマシン上で同時に使用する場合など) には、p4 login -a を使用します。ユーザは、すべての IP アドレスに対して有効なチケットを持っていても、1 つの PERFORCE ライセンスしか消費しません。

オプション

-a	すべての IP アドレスに対して有効なチケットを取得します。
-p	ローカル・チケット・ファイルにチケットを保存しないで、チケットを表示します。
-s	現在のチケットのステータスを表示します (ステータスがある場合)。
g-opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか?	ファイル引数にリビジョン範囲を使えるか?	最低限必要なアクセスレベル
適用外	適用外	リスト

- グループごとに、デフォルトのタイムアウト値である 43200 秒 (12 時間) が p4 group フォームで定義されます。
- 無期限のチケットを作成するには、p4 group のフォームにおいて [Timeout:] を unlimited に設定します。
- チケットの有効期間を延長するには、ログインしている間に p4 login を使用します。チケットの有効期間は、初期タイムアウト設定値の 1/3 だけ延長されます (最大でチケットの初期タイムアウト設定値まで)。
- PERFORCE スーパー・ユーザは、パスワードを入力しなくても自分以外のユーザのログイン・チケットを取得できます。スーパー・ユーザ以外のユーザが他のユーザのチケットを取得できるのは、そのユーザのパスワードを正確に入力した場合に限られます。
- チケットは環境変数 P4TICKETS で指定したファイルに保存されます。環境変数が設定されていない場合、チケットは Windows では %USERPROFILE%\p4tickets.txt、その他の OS では \$HOME/.p4tickets に保存されます。

使用例

<pre>p4 login</pre>	ユーザにパスワードの入力を求め、パスワードが正しく入力されると、そのユーザのマシンで有効なチケットを発行します。
<pre>p4 -u builder login -a</pre>	ユーザ <code>builder</code> としてログインします。パスワードが正しく入力されると、すべてのマシンで有効なチケットを発行します。

関連コマンド

ログイン・セッションを終了する。	<code>p4 logout</code>
チケットを表示する。	<code>p4 tickets</code>

p4 logout

概要

チケットを削除または無効にして、PERFORCE からログアウトします。

構文

```
p4 [g-opts] logout [ -a ]
```

解説

ユーザのワークステーション上のチケットを削除するか、サーバ上のチケットを無効にして、PERFORCE からユーザをログアウトさせます。

p4 logout -a を使用すると、チケットはチケット・ファイルに残りますが、サーバ上では無効にされます。チケットのすべてのユーザが同時にログアウトされます。

オプション

-a	サーバ上のチケットを無効にして、チケットのすべてのユーザをログアウトします。
g-opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか？	最低限必要なアクセスレベル
使用不可	使用不可	リスト

- チケットは環境変数 P4TICKETS で指定したファイルに保存されます。環境変数が設定されていない場合、チケットは Windows では %USERPROFILE%\p4tickets.txt、その他の OS では \$HOME/.p4tickets に保存されます。

使用例

p4 logout	ローカル・セッション・チケットを削除して、PERFORCE からログアウトします。
p4 logout -a	チケットを無効にして、ユーザがログインしているあらゆるワークステーションから、そのユーザをログアウトするよう PERFORCE サーバに指示して、PERFORCE からログアウトします。

関連コマンド

ログイン・セッションを開始する (チケットを取得するため)。	p4 login
チケットを表示する。	p4 tickets

p4 monitor

概要

PERFORCE のプロセス情報を表示します。

構文

```
p4 [g-opts] monitor show[ -a -l -e ]
p4 [g-opts] monitor terminate [ id ]
p4 [g-opts] monitor clear [ id | all ]
```

解説

p4 monitor を動作させるには、PERFORCE サーバ上で監視ができるように設定を行う必要があります。この設定は、p4 counter でカウンタ monitor を指定し、サーバを再起動することによって実行できます。サーバプロセスの監視を制御するには、カウンタ monitor を 0 (監視を無効にする)、1 (アクティブなプロセスの監視を有効にする)、または 2 (アクティブなプロセスと休止プロセスの監視を有効にする) に設定します。カウンタの変更を有効にするには、PERFORCE サーバを停止後再起動する必要があります。

p4 monitor によってシステム管理者は、PERFORCE サーバ・マシン上でどのような PERFORCE 関連プロセスが実行中かを監視することができます。出力行には、次のフィールドが含まれます。

```
pid status owner hh:mm:ss command [args]
```

ここで、pid は UNIX 環境におけるプロセス ID (または Windows 環境におけるスレッド ID) であり、status はそのプロセスが実行中か終了のためにマークされているかによって、R または T を示します。owner は、そのコマンドを実行したユーザの PERFORCE ユーザ名です。hh:mm:ss はコマンドを実行してから経過した時間を示し、command と args は PERFORCE サーバが受信したコマンドと引数です。

現在のプロセス情報を表示するには、p4 monitor show を使用します。すべてのプロセスがリストされますが、引数なしでコマンド (例えば、sync、edit、submit) だけが表示されます。この形式で p4 monitor を実行するには、list 権限が必要です。

それぞれのコマンドに関連する引数のリストを表示するには、-a (arguments) オプションまたは -l (long) オプションを使用します。ユーザ環境からの追加情報が必要な場合は、-e (environment) オプションを使用します。これらのオプションを指定するには、admin 権限が必要です。

プロセスを終了させるためにマークするには、p4 monitor terminate id を使用します。このコマンドの実行には、super 権限が必要です。

監視テーブルからエントリを消去するには、p4 monitor clear id を使用します。p4 monitor clear all を実行すれば、テーブル全体を消去することができます。これらのコマンドの実行には、super 権限が必要です。

オプション

g_opts	「グローバル・オプション」のセクションをご覧ください。
-a	プロセスに指定されたすべての引数 (例えば、edit file.c や sync -f //depot/src/... など) を表示します。 PERFORCE ユーザ名は 10 文字に切り捨てられ、出力行の文字数は全部で 80 文字に制限されます。

-e	PERFORCE クライアント・アプリケーション (既知の場合)、ホスト IP アドレス、およびクライアント・ワークスペース名を含む環境情報を表示します。
-l	長い形式で、すべての引数が表示されます。ユーザ名やコマンドライン引数のリストは、切り捨てられません。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	list , admin, super

- 実行中と示されたプロセスは、たとえ `p4 monitor clear` で監視テーブルから消去されたとしても、終了するまで実行し続けます。
- サーバにおいて早いタイミングでコマンドが終了すると、誤って実行中とリストされる場合があります。このようなプロセスは、スーパーユーザが `p4 monitor clear` で消去することができます。
- プロセスが少なくとも 10 秒間実行中でない限り、`p4 monitor terminate` コマンドはプロセスに終了のためのマークを付けません。
- 複数のプロセスを実行するコマンド (例えば、`p4 submit` など) があります。例えば、`dm_CommitSubmit` または `dm_SubmitChange` が、`p4 submit` コマンドの 2 つの独立したフェーズとして `p4 monitor` の出力に表示されることがあります。
- `p4 obliterate` などの一部のコマンドは終了することができません。
- 使用中のサーバに対し休止プロセスの監視を有効にしている場合 (カウンタ `monitor` を 2 に設定)、休止プロセスは `status` が R、`command` が IDLE で表示されます。

使用例

<code>p4 monitor show</code>	PERFORCE のプロセス情報 (コマンドのみ) を表示します。list 権限のみが必要です。
<code>p4 monitor show -l</code>	出力行の長さを制限せずに、コマンドと引数を表示します。admin 権限が必要です。
<code>p4 monitor show -a</code>	出力行ごとに 80 文字に制限し、コマンドと引数を表示します。admin 権限が必要です。
<code>p4 monitor terminate 123</code>	プロセス 123 を終了のためにマークするよう、PERFORCE サーバに通知します。super 権限が必要です。
<code>p4 monitor clear all</code>	監視テーブルのすべてのエントリを消去します。super 権限が必要です。

関連コマンド

監視を行うようサーバを設定する (サーバの再起動が必要)。	<code>p4 counter -f monitor 1</code>
監視を行わないようサーバを設定する (サーバの再起動が必要)。	<code>p4 counter -f monitor 0</code>

p4 obliterate

概要

ファイルおよびその履歴をディポから消去します。

構文

```
p4 [g-opts] obliterate [ -y ] file[revRange] ...
```

警告

p4 delete コマンドは、最新リビジョンを削除扱いにしますが、ファイル情報はディポにそのまま残します。このため、サーバ・データからのリカバリは常に可能です。

これに対して、p4 obliterate はファイル・データそのものを削除し、リカバリの可能性を完全に排除します。

p4 obliterate は **慎重に使ってください**。これは PERFORCE で唯一の、実際にファイル・データを消去するコマンドです。

解説

p4 obliterate は、PERFORCE 管理者が使用し、ディポからのファイルを完全に消去します。ファイルのリビジョン、ファイルのメタデータ、ラベルやクライアント・ワークスペース内にある直接それらのファイルを参照する記録など、当該ファイルに関するあらゆる情報が消去されます。p4 obliterate が実行されると、その影響を受けたファイルは、サーバにとって最初から存在していなかったような状態になります。クライアント・ワークスペース内のファイルはそのままですが、もはや PERFORCE によるソース管理下のファイルとは認識されません。

p4 obliterate は少なくとも 1 つのファイル・パターンを引数として必要とします。実際にファイルの完全消去を実行するには、-y オプションが必要です。これが付いていなければ、p4 obliterate は実際にファイルの完全消去は実行せずに、単にそれを実行したらどうなるかを知らせるだけです。

特定のリビジョンを指定すれば（例えば p4 obliterate file#3）、そのリビジョンのファイルだけが完全消去されます。リビジョン範囲を指定すれば（例えば p4 obliterate file#3,5）、その範囲のリビジョンだけが完全消去されます。

オプション

-y filespec	ファイルの完全消去を実行します。このオプションが付いていなければ、p4 obliterate は単に、実行したらどうなるかを知らせるだけです。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	admin

- p4 obliterate が最もよく使われるのは、必要なくなったファイルを消去してディスク容量を再生するとき、またはユーザが自分の犯したミス、例えば間違った場所にファイル階層を生成するようなミスを帳消しにするときです。
- ファイルを完全消去すると、ユーザ・コマンドの動作が変わってることがあります。完全消去されたリビジョンに同期することはそのファイルをクライアント・ワークスペースから

取り除くことになり、最新リビジョンに同期すると、そのファイルがクライアント・ワークスペースから取り除かれるか（すべてのリビジョンが完全消去されていた場合）、またはそのファイルの完全消去されていないリビジョンのうちの最新リビジョンが呼び出されます。

- ファイルをリビジョン範囲で完全消去すると、完全消去された範囲のリビジョン番号が“スキップ”され、スクリプトの動作が変わってくる場合があります。例えば、リビジョン #2 および #3 を完全消去した後の p4 filelog の出力が次のようになります。

```
... #4 change 1276 edit on 2001/04/18 by user1@dev1 (binary)
'Fixed'
... #1 change 1231 add on 2001/04/12 by user1@dev1 (binary) 'First
try'
```

この場合には、出力の 1 行目にある #4 を元に、p4 filelog の出力には 4 つのチェンジの記述があると推量するようなスクリプトでは問題が生じます。

使用例

p4 obliterate dir/...	ファイルの完全消去は行わず、-y オプションを付けたときに完全消去されるファイルをリスト表示します。 この場合には、dir 以下のディレクトリのすべてのファイルが -y オプションを付けたときの消去の対象になります。
p4 obliterate -y file	file をディポから完全消去します。file のあらゆるリビジョンに関する履歴およびメタデータのすべてが消去されます。
p4 obliterate -y file#3	file のリビジョン #3 だけを完全消去します。 #3 が最新リビジョンであれば、最新リビジョンは #2 に変わり、次に生成されるリビジョンは #3 になります。 #3 が最新リビジョン <u>でなければ</u> 、最新のリビジョン番号は変わりません。
p4 obliterate -y file#3,5	file のリビジョン #3、4、5 を完全消去します。 #5 が最新リビジョンであれば、最新リビジョンは #2 に変わり、次に生成されるリビジョンは #3 になります。 #5 が最新リビジョン <u>でなければ</u> 、最新のリビジョン番号は変わりません。

関連コマンド

最新リビジョンのファイルを削除扱いにするが、そのままディポに残す。 p4 delete
(通常のファイル削除の形態)

p4 opened

概要

作業中のチェンジリスト内にある作業状態のファイルをリスト表示します。

構文

```
p4 [g-opts] opened [-a -c changelist# -c workspace -m max] [file ...]
```

解説

p4 opened は、p4 add、p4 edit、p4 delete、または p4 integrate によって現在作業状態になっているファイルをリスト表示するために使います。デフォルトでは、現在のクライアント・ワークスペースにある作業状態のファイルがすべてリスト表示されます。コマンドライン引数を使えば、特定の作業中チェンジリストに含まれるファイルのみをリスト表示したり、すべての作業中チェンジリストにある作業状態のファイルを表示したり、表示されるファイルの数を制限することができます。

p4 opened に引数としてファイル指定が与えられていれば、リストにはそのファイル指定に適合するファイルだけが含まれます。

作業状態の各ファイルについて表示される情報には、ファイル名、ディポ内での保存場所、ファイルが最後に同期したリビジョンの番号、ファイルが作業状態にされたチェンジリストの番号、ファイルを作業状態にした目的の操作 (add、edit、delete、integrate)、ファイルタイプが含まれます。各ファイルの出力は次のフォーマットで表示されます。

```
depot-file#rev - action chnum change (type) [lock-status]
```

ただし、

- *depot-file* はディポ・シンタックスで記述されたパスです。
- *rev* はリビジョン番号です。
- *action* はファイルが作業状態にされた目的の操作、すなわち add、edit、delete、branch、または integrate です。
- *chnum* はサブミットされるチェンジリストの番号です。
- *type* は与えられたリビジョンのファイルのタイプです。
- ファイルがロックされている場合は (p4 lock 参照)、行末に *locked* の警告が表示されます。

オプション

-a	すべてのクライアント・ワークスペース内にある作業状態のファイルをリスト表示します。
-c <i>changelist#</i>	作業中チェンジリスト <i>changelist#</i> のファイルをリスト表示します。デフォルト・チェンジリストのファイルをリスト表示するには、p4 opened -c default を使用します。
-C <i>workspace</i>	指定されたクライアント・ワークスペース内で作業状態にあるファイルのみをリスト表示します。
-m <i>max</i>	<i>max</i> までの作業状態のファイルのみを一覧表示します。
<i>g_opts</i>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	list

- PERFORCEは、ユーザがすでに作業状態になっているファイルを作業状態にすることを妨げません。デフォルトの構成では、複数のユーザが同じファイルを同時に編集し、p4 resolveでファイル衝突を解決することが認められています。あるファイルを別のユーザがすでに作業状態にしているかどうかを調べるには、`p4 opened -a file`を使います。
- ロックされたファイルは、p4 openedの出力において*locked*と表示されます。UNIXでは、次のコマンドを実行することによって、作業状態でかつロックされているファイルをすべて検索することができます。

```
p4 opened | grep "*locked*"
```

これにより、p4 lockでロックされたロックされた作業状態のファイルをリストします。

使用例

<code>p4 opened -c 35 //depot/main/...</code>	ディポのサブディレクトリ mainにある作業中チェンジリスト 35のファイルをすべてリスト表示します。
<code>p4 opened -a -c default</code>	全クライアントのデフォルト・チェンジリストについて作業状態にあるファイルをすべてリスト表示します。

関連コマンド

ファイルをクライアント・ワークスペース内で作業状態にし、チェンジリストにリストアップする。	p4 add
	p4 edit
	p4 delete
	p4 integrate
1つのチェンジリストから別のチェンジリストへファイルを移動する。	p4 reopen
すべてのチェンジリストから特定のファイルを削除し、以前の状態に戻す。	p4 revert
新しい番号付チェンジリストを生成する。	p4 change
特定の基準に適合するチェンジリストのリストを参照する。	p4 changes

p4 passwd

概要

サーバ上でユーザの PERFORCE パスワードを変更します。

構文

```
p4 [g-opts] passwd [-O oldpassword] [-P newpassword] [user]
```

解説

デフォルトでは、ユーザ・レコードはパスワードなしに生成され、いかなる PERFORCE ユーザも P4USER を設定するか、または グローバルに利用可能な `-u` オプションを用いることによって他のユーザになり代わることができます。他のユーザになり代わられることを防ぐには `p4 passwd` を用いて、コメント・キャラクタ `#` を含まない任意の文字列でパスワードを設定します。

パスワードを設定した後は、PERFORCE コマンドを実行するたびに、PERFORCE サーバ・プログラムにそのパスワードを提示して、パスワードによって認証することができます。次の 3 つの方法のいずれかで、パスワードを PERFORCE サーバに提示することができます。

- 環境変数またはレジストリ変数 `P4PASSWD` をパスワードの値に設定する。
- `P4CONFIG` ファイル内で `P4PASSWD` 用の設定を生成する。
- 例えば下記のように、PERFORCE クライアント・コマンドライン上で `-P password` オプションを使用する。

```
p4 -u ida -P idaspassword sync
```

上記の 3 つの方法の優先度は、最も高いのが c)、次に高いのが b) です。サーバのセキュリティ・レベルによっては、これらの方法のうちのいずれかが許されないことがあります。

セキュリティ・レベル 0 および 1 のサーバに接続する Windows クライアントでは、`p4 passwd` は `p4 set` を使用してローカルのレジストリ変数を変更することによりパスワードを保存します（このレジストリ変数は、暗号化された MD5 ハッシュを保持するだけで、パスワードそのものは保持しません）。セキュリティ・レベル 2 および 3 のサーバに接続する Windows クライアントでは、パスワードのハッシュがレジストリに保存されることも、レジストリから読み取られることもありません。

パスワード・ベースの認証の代わりにチケット・ベースの認証を使用して、セキュリティを強化することができます。パスワードの代わりにチケットで認証するには、最初に `p4 passwd` でパスワードを設定し、次に `p4 login` コマンドと `p4 logout` コマンドを使用して認証を管理します。チケット・ベースの認証のしくみについては、『システム管理者ガイド』をご覧ください。

サーバのセキュリティ・レベルと PERFORCE クライアント・ソフトウェア・リリースの特定の組み合わせでは、ユーザは「強い」パスワードを設定することを求められます。パスワードが 8 文字以上の長さで、以下の記述のうちの少なくとも 2 つが真である場合に、そのパスワードは強いと見なされます。

- パスワードに大文字が含まれている
- パスワードに小文字が含まれている
- パスワードにアルファベット以外の文字が含まれている

例えば、`a1b2c3d4`、`A1B2C3D4`、`aBcDeFgH` などのパスワードは強いと見なされます。チケット・ベースの認証のしくみについては、『システム管理者ガイド』をご覧ください。

オプション

<code>-O oldpassword</code>	コマンドライン上で古いパスワードを指定することにより、プロンプト入力を避けます。サーバがセキュリティ・レベル 3 を使用している場合は、このオプションはサポートされません。
<code>-P newpassword</code>	コマンドライン上で新しいパスワードを指定することにより、プロンプト入力を避けます。サーバがセキュリティ・レベル 3 を使用している場合は、このオプションはサポートされません。
<code>user</code>	スーパー・ユーザはこの引数を付けて他のユーザのパスワードを変更することができます。
<code>g_opts</code>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	list

- p4 passwd コマンドは、プレーンテキストのパスワードをネットワークを介して送ることはありません。暗号化されたパスワードをサーバに送るには、チャレンジ/レスポンスのメカニズムが利用されます。
- パスワードはスペースを含んでもかまいません。コマンドラインでそのようなパスワードを使用するときには、引用符を付ける必要があります。例えば、my passw というパスワードを PERFORCE に渡すには、`p4 -P "my passw" command` と実行します。
- ユーザが自分のパスワードを忘れた場合には、PERFORCE スーパー・ユーザはコマンドライン上で `p4 passwd username` とユーザ名を指定して、それをリセットすることができます。
- パスワードの最大長はどのプラットフォーム上でも 1024 文字です。
- パスワードを削除するには、パスワードの値として空文字列を設定します。サーバのセキュリティ・レベルによっては、空文字列のパスワードの設定が許可されないこともあります。
- チケット・ベースの認証を使用している場合、ユーザがパスワードを変更すると、自動的にそのユーザのすべてのチケットが無効にされ、ユーザはログアウトされます。つまり、パスワードを変更することは、`p4 logout -a` を実行することと同じです。

関連コマンド

他のユーザの仕様を変更する。	p4 user
ユーザのアクセス・レベルを変更する。	p4 protect
パスワードの代わりにチケットを使用してログインする。	p4 login

p4 print

概要

ディポ・ファイルのリビジョンの内容を表示します。

構文

```
p4 [g-opts] print [ -a ] [ -o outfile ] [ -q ] file[revRange] ...
```

解説

p4 print は、ディポ・ファイルの内容を標準出力へ書き出します。リビジョン範囲を指定した場合、指定された範囲内のリビジョンを持つファイルだけを書き出します。デフォルトでは、範囲内の最高位リビジョンだけを書き出します。(指定されたリビジョン範囲における全ファイルを出力するには、p4 print -a を使用します。)

ディポのどのファイルでも、p4 protect によって認められたアクセス権限に従って表示させることができます。ファイル引数がクライアント・ビューによってマッピングされていない場合は、ディポ・シンタックスで指定しなければなりません。

デフォルトでは、ファイルの内容とともに、そのファイルのディポにおける位置を記述するヘッダ、表示されたファイルのリビジョン番号、そのリビジョンがサブミットされるチェンジリストの番号も表示されます。ヘッダを抑制するには、-q (quiet) オプションを使用します。

複数のファイル・パターンを含めることもできます。その場合には、それらのファイル・パターンのいずれかに適合するすべてのファイルが表示されます。

オプション

-a	すべてのファイルに対して、指定されたリビジョン範囲内の最新リビジョンだけでなく、全リビジョンを書き出します。
-q	通常はPERFORCEによって追加される1行のファイル・ヘッダの表示を抑制します。
-o outfile	ディポのオリジナル・ファイルと同じファイルタイプ、属性、パーミッション設定を保持して、出力をローカル・ディスク上の指定された出力ファイルにリダイレクトします。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	read

- p4 print のファイル引数としてリビジョン範囲を指定することができます。適合するファイルの最新のリビジョンだけが表示されます。(これは、指定されないときの暗黙の範囲が #1, #head であり、その最高位が #head であることを意味しています。) 指定された (もしくは暗黙の) 反映内の全ファイルを書き出すには、-a オプションを指定します。
- ファイル引数を詳細に定義せずに p4 print を実行すると、結果が大量に出力されることがあります。(例えば、p4 print //depot/... と指定すると、ディポのすべてのファイルの内容が表示されます。) 出力は、p4 group で設定された maxresults の制限に準拠します。
- 多くの場合、p4 print の出力は OS のシェルを介してファイルにリダイレクトすれば十分です。

-o オプションは、ユーザがファイルタイプやパーミッションの自動設定を必要とする状況に対処するために使用します。これは UNIX シンボリック・リンク (タイプ `symlink` として保存) のようなファイル、タイプ `apple` のファイル、またはタイプ `text+x` などとして保存されている UNIX シェル・スクリプトの実行ビットの自動設定には便利です。

関連コマンド

ディボ・ファイルの2つのリビジョンの内容を比較する。 `p4 diff2`
クライアント・ワークスペースで作業状態になっているファイルの内容を、`p4 diff`
ディボ・ファイルのリビジョンと比較する。

p4 protect

概要

ユーザによるファイル、ディレクトリ、コマンドへのアクセスを管理します。

構文

```
p4 [g-opts] protect
p4 [g-opts] protect -o
p4 [g-opts] protect -i
```

解説

p4 protect は、PERFORCE パーミッションを制御します。以下のように使用できます。

- どのユーザがどのファイルにアクセスできるかを制御します。
- どのユーザがどのコマンドを使用できるかを制御します。
- 上記の2点を組み合わせると、あるユーザに対して、あるファイル・セットへは書き込み可能で他のファイルは読み取りのみ、という制御ができます。
- ユーザに対して特定の権限を付与または拒否するには、=read、=open、=write、=branch の各権限を使用します。それより下位の権限を再度付与する必要はありません。
- p4 group で定義したユーザ・グループにパーミッションを与えます。
- 特定のIPアドレス上のユーザのみがPERFORCEを実行できるよう、アクセスをこのIPアドレスに限定します。

以下に、PERFORCE のパーミッション・レベルとアクセス権限を示します。

パーミッション・レベル / 権限	ユーザが実行できること
list	すべての PERFORCE メタデータにアクセスできますが、ファイルの内容にはアクセスできません。p4 files、p4 client、p4 job、p4 describe、p4 branch などの PERFORCE オブジェクトを記述するコマンドをすべて実行できます。
read	list 権限で許可されていることをすべて実行できます。さらに、ファイルのデータの読み取りを伴うコマンドもすべて実行できます (p4 print、p4 diff、p4 sync など)。
=read	この権限が拒否されると、ユーザはファイルに対して p4 print、p4 diff、p4 sync を実行できません。
open	read 権限で実行できることをすべて許可します。また、ファイルを追加 (p4 add)、編集 (p4 edit)、削除 (p4 delete)、反映 (p4 integrate) するパーミッションを与えます。ただし、ファイルのロックやディポへのサブミットは許可されていません。
=open	この権限が拒否されると、ユーザは p4 add、p4 edit、p4 delete、p4 integrate によってファイルを作業状態にすることができません。
write	上記のすべてを実行できます。さらに、p4 submit によるファイルへの書き込み、p4 lock によるファイルのロックを実行できます。
=write	この権限が拒否されると、ユーザは作業状態のファイルをサブミットできません。
=branch	この権限が拒否されると、ユーザはファイルを p4 integrate の反映元として使用できません。

パーミッション・レベル / 権限	ユーザが実行できること
review	PERFORCE にアクセスする外部プログラム専用のパーミッションです。list 権限とread権限で実行できることすべてのパーミッションを外部プログラムに与えます。また、p4 review と p4 counter を実行するパーミッションを与えます。ただし、open および write 権限は含まれません。
admin	上記のすべてに加え、メタデータへの変更をオーバーライドするための管理者用コマンドを実行するパーミッションを与えます。ただし、サーバの動作に影響するコマンドを実行するパーミッションは与えません。 これに該当するのは、p4 branch -f、p4 change -f、p4 client -f、p4 job -f、p4 jobspec、p4 label -f、p4 obliterate、p4 typemap、p4 unlock -f および p4 verify です。
super	上記のすべてを含み、さらに p4 admin、p4 counter、p4 triggers、p4 protect などのスーパー・ユーザ・コマンドにもアクセスできます。

フォーム内のフィールド

p4 protect を実行すると、PERFORCE はフィールド [Protections:] だけのフォームを表示します。各パーミッションは、[Protections:] の見出しに対してインデントされた、専用の行に指定されます。各行には次の 5 つの設定値があります。

列	説明
アクセス・レベル	前項で定義された、list、read、open、write、review、super または =read、=open、=write、=branch の各権限のうちの 1 つ。
ユーザまたはグループ	プロテクションが適用されるのは、ユーザかグループかの指定。 user か group のいずれかでなければなりません。
グループ名またはユーザ名	ユーザ名または p4 group で定義したグループ名。 すべてのユーザにこのパーミッションを与えるときは、ワイルドカード「*」を使用します。
ホスト	IP アドレス。CIDR 表記がサポートされています。また、CIDR 表記を使用していない場合にのみ、すべての IP アドレスを参照するためにワイルドカード「*」を使用できます。
ディポ・ファイルのパス	PERFORCE のディポ・シンタックスで、このパーミッションが与えられているディポ・ファイルのパス。ファイルの指定は、PERFORCE のワイルドカードを含むことができます。 このマッピングをパーミッション・セットから除外するには、設定値の頭にダッシュ (-) を付けます。

除外マッピングが設定されていない場合、各ユーザにはユーザ、ユーザの IP アドレス、およびユーザがアクセスしようとしているファイルと適合するすべてのマッピングにリストされたパーミッション・レベルの中で最高のレベルが与えられます。この場合、マッピングの順序は関係ありません。

除外マッピングが設定されている場合、マッピングの順序が関係します。除外マッピングは、それより上の行にリストされているプロテクションのうち適合するものはすべて無効にします。除外プロテクションでどのアクセス・レベルが拒否されていても、適合するユーザ、ファイル、IP アドレスに対するすべてのアクセス・レベルが拒否されます。

除外マッピングを使用して、group1 のメンバーに対してディポのある領域へのアクセスを拒否し、group2 のメンバーに対してはディポの同じ領域へのアクセスを許可した場合、group1 と group2 の両方のメンバーであるユーザは、プロテクション・テーブル内でどちらの行が後に記述されているかによってアクセスの可否が決まります。

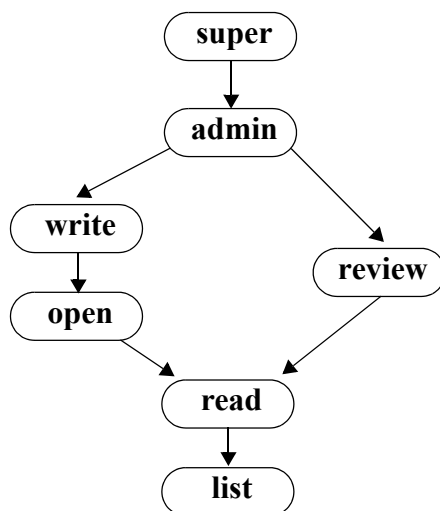
オプション

-i	エディタを起動せずに、標準出力からフォームを読み取ります
-o	エディタを起動せずに、フォームを標準出力に書き出します
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル変数はリビジョン指定子を使用できるか	ファイル変数はリビジョン範囲を使用できるか	要求される最低アクセス・レベル
使用不可	使用不可	super

- 下図のように、各パーミッションのレベルはその下にあるすべてのアクセス・レベルを含みます。



- 特殊な権限である =read、=open、=write、=branch を使用して、下位のアクセス・レベルを自動的に包含しないようにすることができます。これにより、個々の権限を無効にすることができ、後で下位の権限を再度付与する必要はありません。
- 使用できるコマンドは、アクセス・レベルによって決まります。下表は、各コマンドで最低限必要なアクセス・レベルを示しています。例えば、p4 add では少なくとも open アクセス権限が必要なので、ユーザは open、write、admin、または super のアクセス権限を持っている場合に p4 add を実行できます。

コマンド	アクセス・レベル	備考
add	open	
admin	super	
annotate	read	
branch	open	既存のメタデータや他のユーザのデータをオーバーライドするための -f オプションには、admin 権限が必要です。
branches	list	
change	open	既存のメタデータや他のユーザのデータをオーバーライドするための -f オプションには、admin 権限が必要です。
changes	list	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも 1 つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。
client	list	既存のメタデータや他のユーザのデータをオーバーライドするための -f オプションには、admin 権限が必要です。
clients	list	
counter	review	既存のカウンタの値を参照するには、list 権限が必要です。カウンタの値を変更したり、新しいカウンタを生成したりするには、review 権限が必要です。
counters	list	
delete	open	

コマンド	アクセス・レベル	備考
depot	super	このコマンドの <code>-o</code> オプションは、フォームの編集ではなく読み取りを可能にするもので、 <code>list</code> 権限さえあれば十分です。
depots	list	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも 1 つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。
describe	read	このコマンドに <code>-s</code> オプションを付けるとファイルの内容は表示されないため、 <code>list</code> 権限さえあれば十分です。
diff	read	
diff2	read	
dirs	list	
edit	open	
filelog	list	
files	list	
fix	open	
fixes	list	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも 1 つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。
fstat	list	
group	super	このコマンドの <code>-o</code> オプションは、フォームの編集ではなく読み取りを可能にするもので、 <code>list</code> 権限さえあれば十分です。
groups	list	このコマンドの <code>-a</code> オプションには、ユーザがグループ所有者としてもリストされている場合に限り、 <code>list</code> 権限のみが必要です。
have	list	
help	none	
info	none	
integrate	open	ユーザは反映先ファイルに対する <code>open</code> 権限と反映元ファイルに対する <code>read</code> 権限が必要です。
integrated	list	
job	open	このコマンドの <code>-o</code> オプションは、フォームの編集ではなく読み取りを可能にするもので、 <code>list</code> 権限さえあれば十分です。
jobs	list	既存のメタデータや他のユーザのデータをオーバライドするための <code>-f</code> オプションには、 <code>admin</code> 権限が必要です。
jobspec	admin	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも 1 つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。
jobspec	admin	このコマンドの <code>-o</code> オプションは、フォームの編集ではなく読み取りを可能にするもので、 <code>list</code> 権限さえあれば十分です。

コマンド	アクセス・レベル	備考
label	open	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも1つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。 既存のメタデータや他のユーザのデータをオーバーライドするための <code>-f</code> オプションには、 <code>admin</code> 権限が必要です。
labels	list	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも1つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。
labelsync	list	
license	super	
lock	write	
login	list	
logout	list	
monitor	list	プロセスを終了または消去するには <code>super</code> 権限が必要です。引数を参照するには <code>admin</code> 権限が必要です。
obliterate	admin	
opened	list	
passwd	list	
print	read	
protect	super	
protects	list	<code>-a</code> 、 <code>-g</code> 、 <code>-u</code> オプションを使用するには、 <code>super</code> 権限が必要です。
reopen	open	
resolve	open	
resolved	open	
revert	open	
review	review	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも1つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。
reviews	list	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも1つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。
set	list	
sizes	list	
submit	write	
sync	read	
tag	open	
tickets	none	
triggers	super	
typemap	admin	このコマンドの <code>-o</code> オプションは、フォームの編集ではなく読み取りを可能にするもので、 <code>list</code> 権限さえあれば十分です。
unlock	open	既存のメタデータや他のユーザのデータをオーバーライドするための <code>-f</code> オプションには、 <code>admin</code> 権限が必要です。

コマンド	アクセス・レベル	備考
user	list	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも1つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。
users	list	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも1つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。
verify	admin	
where	none	このコマンドは特定のファイルに作用するものではありません。つまり、ユーザがディポの少なくとも1つのファイルに対して権限を指定されていれば、コマンド実行のパーミッションは認められています。

- 新しいPERFORCEサーバをインストールしたときは、使用したい人すべてがPERFORCEを使用でき、すべてのPERFORCEユーザがスーパー・ユーザとなります。初めてp4 protectが実行されると、その実行者がスーパー・ユーザとなり、他のユーザにすべてのファイルに対するwriteパーミッションが与えられます。インストール後すぐにp4 protectを実行してください。
- 通常のコマンドで与えられるアクセス・レベルは、主にlist、read、writeおよびsuperです。openとreviewはあまり使用されません。
- p4 describeのようなファイルを一覧表示するコマンドは、ユーザが最低list権限を持っているファイルのみを一覧表示します。
- -f オプションを指定できるコマンド（例えば、以前にサブミットされたチェンジリストを編集するp4 change）がありますが、これにはadminもしくはsuper権限が必要です。
- アクセス・レベルopenで、ファイルを変更することはできますがディポにサブミットすることはできません。一時的にコードラインを凍結させたいが開発者の作業を止めたくない場合や、試験担当者に対し、個人使用のためのコード変更は認めるがコードラインに永久的な変更を加えることは認めないような場合などに、open権限を使用してください。
- アクセス・レベルreviewは、カウンタ値にアクセスする必要のあるレビュー・デーモンを意図しています。
- review権限とwrite権限の両方を必要とするがsuper権限は必要としないレビュー・デーモンを記述する場合、プロテクション・テーブルには、別々の行でreview権限とwrite権限の両方をデーモンに与えてください。
- あるサーバ上のファイルを別のサーバからのリモート・ディポとして使用する（p4 depotで定義）のを制限したり排除するには、remoteユーザに対するプロテクションを作成してください。リモート・ディポは、常にremoteという名前の仮想ユーザによってアクセスされます。
- 詳細情報については、『システム管理者ガイド』の「プロテクション」章をご覧ください。

使用例

ユーザ joe が、p4 group による設定に基づいて devgroup グループと buggroup グループのメンバーであると仮定すると、プロテクション・テーブルは次のようになります。

super	user	bill	*	//...
write	group	devgroup	*	//depot/...
write	group	buggroup	*	-//depot/proj/...
write	user	joe	192.168.100.0/24	//...

joe はいくつかの操作を試みます。各操作が成功するか失敗するかは次のとおりです。

操作元 IP アドレス	試みる操作	結果
10.14.10.1	p4 print //depot/misc/...	成功します。2行目で、これらのファイルへの write 権限が joe に与えられています。write 権限は read 権限を含み、以降のどの行でもこのプロテクションは除外されていません。
10.14.10.1	p4 print //depot/proj/README	失敗します。3行目で、このディレクトリ中のあらゆるファイルに対する joe のすべてのパーミッションが除去されています。(2番目と3番目のプロテクションが入れ替わっていた場合、以降のプロテクションが除去を無効にするので、joe はこの操作に成功します。)
192.168.100.123	p4 print //depot/proj/README	成功します。joe は、4行目で操作元としてこのパーミッションを与えられた IP アドレスから操作しています。
192.168.100.123	p4 verify //depot/misc/...	失敗します。p4 verify は super 権限を必要とし、どの IP アドレスからアクセスしても joe はこのアクセス・レベルを持っていません。

関連コマンド

ユーザ・グループを作成または編集する。	p4 group
すべてのユーザ・グループを一覧表示する。	p4 groups

p4 protects

概要

指定されたユーザ、グループまたはパスに設定されたプロテクションを表示します。

構文

```
p4 [g-opts] protects [ -a | -u user | -g group | -h host ] [-m] [
file... ]
```

解説

ユーザ、グループまたはファイル・セットに対して適用されるプロテクション・テーブルの行を表示するには、p4 protects コマンドを使用してください。

オプションを指定しない場合、p4 protects は現在のユーザに適用されるプロテクション・テーブルの行を表示します。file 引数を指定すると、指定されたファイルに適用されるプロテクション・テーブルの行のみを表示します。

-a オプションを使用するとすべてのユーザに適用される行が表示され、-u user、-g group、または -h host を使用すると特定のユーザ、グループ、またはホスト IP アドレスに適用される行が表示されます。

オプション

-a	すべてのユーザに適用されるプロテクション行を表示します。このオプションには super アクセスが必要です。
-g group	指定されたグループに適用されるプロテクション行を表示します。このオプションには super アクセスが必要です。
-h host	指定されたホスト IP アドレスに適用されるプロテクション行を表示します。このオプションには super アクセスが必要です。
-u user	指定されたユーザに適用されるプロテクション行を表示します。このオプションには super アクセスが必要です。
-m	適用可能な最大アクセス・レベルについて一語で要約を表示します。(ここでは排他的なマッピングは考慮されません。)
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	list, -a , -h, -g, -u オプションには super

関連コマンド

プロテクション・テーブルを編集する。	p4 protect
--------------------	------------

p4 rename

概要

PERFORCE にあるファイルの名前を変更します。

構文

```
p4 [g-opts] integrate fromFile toFile
p4 [g-opts] delete fromFile
p4 [g-opts] submit fromFile
```

解説

PERFORCE に `rename` コマンドはありませんが、`p4 integrate` を使用して `fromFile` を新しい `toFile` にコピーし、`p4 delete` を使用して `fromFile` を削除し、次に `p4 submit` を使用してこれらのファイル変更をディポに格納することにより、ファイルの名前を変更します。

`fromFile` と `toFile` の名前に適合するワイルドカードを使うと、複数ファイルの名前を変更することもできます。

使用上の留意点

ファイル変数はリビジョン指定子を使用できるか	ファイル変数はリビジョン範囲を使用できるか	要求される最低アクセス・レベル
<code>fromFile</code> : 使用可 <code>toFile</code> : 使用不可	使用不可	<code>fromFile</code> には read アクセス <code>toFile</code> には write アクセス

使用例

<pre>p4 integrate -c 413 //depot/p2/... //depot/guiProj/... p4 delete -c 413 //depot/p2/... p4 submit -c 413</pre>	<p>ファイル・セットの名前変更には3つの手順があります。</p> <ul style="list-style-type: none"> • <code>p4 integrate</code> により、<code>p2</code> ディレクトリの全ファイルを <code>guiProj</code> ディレクトリにコピーします • <code>p4 delete</code> により、<code>p2</code> ディレクトリの全ファイルを削除します。 • <code>p4 submit</code> により、これらの変更が1つのチェンジリストにまとめられて、ディポに送られます。
--	--

関連コマンド

ファイルをコピーし、それを PERFORCE の管理下に置き続ける。	<code>p4 integrate</code>
ディポからファイルを削除する。	<code>p4 delete</code>
チェンジをディポにサブミットする。	<code>p4 submit</code>

p4 reopen

概要

作業状態のファイルをチェンジリスト間で移動するか、ファイルタイプを変更します。

構文

```
p4 [g-opts] reopen [-c changelist#] [-t filetype] file...
```

解説

p4 reopen には、内容は異なりますが関連する 2 つの用途があります。

- p4 reopen -c changelist# file を使用して、作業状態のファイルを現在の作業中チェンジリストから別の作業中チェンジリスト changelist# に移動します。
- p4 reopen -c default を使用して、ファイルをデフォルトのチェンジリストに移動します。
- p4 reopen -t filetype を使用して、ファイルのタイプを変更します。

ファイル・パターンが指定されている場合、そのパターンに適合するすべての作業中ファイルは、移動させられるかタイプを再設定されます。ファイル移動とタイプ変更を 1 回の操作で行うには、2 つのオプションを組み合わせます。

オプション

-c changelist# file	ファイル・パターン file に適合するすべての作業中ファイルを、作業中チェンジリスト changelist# に移動します。デフォルト・チェンジリストにファイルを移動するには、チェンジリスト番号として default を使用します。
-t filetype file	サブミットされたときに、ファイルを指定したタイプ (filetype) で格納します。再度タイプが変更されない限り、以降のすべてのリビジョンはこのファイルタイプとなります。 ファイルタイプの一覧については、「ファイルタイプ」のセクションをご覧ください。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の注意点

ファイル変数はリビジョン指定子を使用できるか	ファイル変数はリビジョン範囲を使用できるか	要求される最低アクセス・レベル
使用不可	使用不可	open

使用例

p4 reopen -t text+k //...	すべての作業中ファイルを、キーワード拡張を使用できる、テキスト・ファイルとして再度作業状態にします。
---------------------------	--

```
p4 reopen -c 410 //depot/proj1/... //.../README
```

//depot/proj1 ディレクトリにあるすべての作業中ファイル、および「README」と名前を付けられたすべての作業中ファイルを作業中チェンジリスト 410 に移動します。

```
p4 reopen -c default -t binary+S //....exe
```

すべての作業中 .exe ファイルを、デフォルトのチェンジリストに移動します。その際、ディポ内の対応するファイルの古いリビジョンを上書きします。

関連コマンド

チェンジリストをディポにサブミットする。	p4 submit
新しいチェンジリストを作成する。	p4 change
すべての作業中チェンジリストから、ファイルを除去する。	p4 revert
作業中ファイルを一覧表示する。	p4 opened
あるチェンジリストに含まれるすべてのファイルを一覧表示する。	p4 opened -c changelist#
すべての作業中チェンジリストを一覧表示する。	p4 changes -p pending
編集を目的として、あるチェンジリスト内のファイルを任意のタイプで作業状態にします	p4 edit -c changelist# -t type
追加を目的として、あるチェンジリスト内のファイルを任意のタイプで作業状態にします。	p4 add -c changelist# -t type
ディポ内のすべてのファイルに対し、悲観的ロック（排他オープン）を実施します。このチェンジリストのサブミット後は、depotname というディポにあるファイルを編集できるのは一度に 1 ユーザのみとなります。	p4 edit -t +l //depotname/...

p4 resolve

概要

ファイル・リビジョン間の衝突を解決します。

構文

```
p4 [g-opts] resolve [-af -am -as -at -ay -dflag -f -n -o -t -v] [file ...]
```

解説

p4 resolve は、2つのファイルまたはファイル・リビジョンの内容を、1つのファイル・リビジョンにまとめるのに使用します。ファイルをサブミットする前に p4 resolve を使用しなければならない状況は、以下の2つです。

- 単純な衝突が存在しているとき。すなわち、クライアント・ワークスペースに対して最後に同期させたファイルのリビジョンが、サブミット時の最新リビジョンではないときです。

例えば、アリスが p4 sync を行い、その後ファイル file.c の p4 edit を実行します。ボブも同じことをします。アリスが p4 submit で file.c をサブミットし、次にボブが file.c をサブミットしようとしても失敗します。というのは、ボブの file.c がディポに受け入れられるとアリスが file.c に加えた変更がもはや参照できなくなってしまうためです。

- p4 integrate を使用して、1つのファイルから他のファイルへの変更の反映をスケジュールしているとき。

この2つの状況の主な違いは、単純なファイル衝突を解決するには1つのファイルの複数リビジョンを必要としますが、変更の反映のために衝突を解決するには、異なる2つのファイルを組み合わせる必要があるということです。いずれの場合も下記ようになります。

- ファイル・タイプがテキストである場合、p4 resolve により、ディポ内のファイル・リビジョンにクライアント・ワークスペース内のファイルを上書きするか、クライアント・ワークスペース内のファイルにディポ内のファイルを上書きするか、またはディポのリビジョンとクライアント・ワークスペースのリビジョン両方からの変更を1つのファイルにマージするかを選択できます。
- ファイル・タイプがバイナリである場合、通常は次ページの表に示す最初の2つのオプション（ディポ内のファイルをクライアント・ワークスペース内のファイルで上書きする、またはクライアント・ワークスペース内のファイルをディポ内のファイルで上書きする）しか利用できません。というのは、バイナリ・ファイルに対して通常マージは機能しないからです。

p4 resolve ダイアログは4つのファイル・リビジョンを参照しますが、単純に解決できるファイル衝突を修正するのか、変更の反映のために使うのかによって、リビジョンの意味が次のように変わります。

用語	単純に衝突を解決する場合	反映のために衝突を解決する場合
<i>yours</i>	クライアント・ワークスペース内のファイルのリビジョン	変更が伝達されようとしている先のファイル。(反映の用語では、これを <u>反映先</u> ファイルと言います。) 変更はクライアント・ワークスペース内のこのファイルのバージョンに対して加えられ、このファイルは後でディポにサブミットされます。
<i>theirs</i>	ディポ内のファイルの最新リビジョン。	変更が伝達される元である、ディポ内のファイル・リビジョン。(反映の用語では、これを <u>反映元</u> ファイルと言います。) このファイルは、ディポやクライアント・ワークスペースでは変更されません。
<i>base</i>	編集目的で作業状態になる前にクライアント・ワークスペースに対して同期したファイル・リビジョン。	<i>theirs</i> の過去の反映済リビジョン。 <i>yours</i> と <i>theirs</i> の最新の共通の祖先。
<i>merge</i>	<i>yours</i> 、 <i>theirs</i> 、および <i>base</i> から PERFORCE が生成したファイル・リビジョン。ファイルがテキスト・ファイルである場合、解決プロセス中にこのリビジョンを編集できます。	左の意味と同じです。

対話式の p4 resolve ダイアログでは以下のオプションが表示されます。ダイアログ・オプションはコマンドライン・フラグと同じではないことに注意してください。

ダイアログオプション	簡潔な意味	内容	デフォルトでバイナリ・ファイルに利用できるか
e	edit merged	PERFORCE が生成したマージ済ファイルを編集します。	利用できない
ey	edit yours	現在クライアント内にあるファイルのリビジョンを編集します。	利用できる
et	edit theirs	クライアントのリビジョンと衝突しているディポ内のリビジョン (通常、最新リビジョン) を編集します。この編集は読み取り専用です。	利用できる
dy	diff yours	<i>yours</i> と <i>base</i> 間の差分を表示します。	利用できない
dt	diff theirs	<i>theirs</i> と <i>base</i> 間の差分を表示します。	利用できない
dm	diff merge	<i>merge</i> と <i>base</i> 間の差分を表示します。	利用できない
d	diff	<i>merge</i> と <i>yours</i> 間の差分を表示します。	利用できる
m	merge	次のコマンドを起動します。 P4MERGE base theirs yours merge このオプションを使用するには、最初の 3 つのファイルをマージし、その結果を 4 番目に書き込むサード・パーティ製のプログラムの名前に、環境変数 P4MERGE を設定する必要があります。P4MERGE が設定されていない場合、このコマンドは無効です。	利用できない
?	help	p4 resolve のヘルプを表示します	利用できる

ダイアログオプション	簡潔な意味	内容	デフォルトでバイナリ・ファイルに利用できるか
s	skip	すぐに解決を実行しません。	利用できる
ay	accept yours	<i>theirs</i> で加えられた可能性のある変更を無視して、 <i>yours</i> を受け入れます。	利用できる
at	accept theirs	<i>theirs</i> を解決済のリビジョンとしてクライアント・ワークスペースに受け入れます。クライアント・ワークスペースにあったリビジョン (<i>yours</i>) は、上書きされます。 単純な衝突を解決している場合、このオプションは、クライアント・ワークスペースのファイルに対して <code>p4 revert</code> を実行するのと同じです。反映のために衝突を解決している場合、このオプションは反映元ファイルを反映先ファイルにコピーします。	利用できる
am	accept merge	<i>merged</i> を解決済のリビジョンとしてクライアント・ワークスペースに受け入れます。クライアント・ワークスペースにあったリビジョン (<i>yours</i>) は、上書きされます。	利用できない
ae	accept edit	ファイル編集 (P4 resolve ダイアログで “e” を選択して) した場合に、クライアント・ワークスペースにその編集済のバージョンを採用します。クライアントの元々のリビジョン (<i>yours</i>) は上書きされます。	利用できない
a	accept	PERFORCE が推奨する結果を保持します <ul style="list-style-type: none"> • <i>theirs</i> が <i>base</i> と等しい場合、<i>yours</i> を受け入れます • <i>yours</i> が <i>base</i> と等しい場合、<i>theirs</i> を受け入れます • <i>yours</i> と <i>theirs</i> が <i>base</i> と異なり、<i>yours</i> と <i>theirs</i> の間に衝突がない場合、<i>merge</i> を受け入れます • 上記以外の場合、<i>yours</i> と <i>theirs</i> の間に衝突があるため、このファイルをスキップします 	利用できない

ファイルの衝突解決は、accept ダイアログ・オプションのいずれかが選択されたとき、または skip オプションを選択してファイルをスキップしたときに完了します。

p4 resolve により、ファイル・リビジョンに対して加えられた 4 種類の変更のカウン트가表示されるため、どのオプションを選択するかを決めるのに役立ちます。

```
Diff Chunks: 2 yours + 3 theirs + 5 both + 7 conflicting
```

これらの値の意味は次のとおりです。

カウント	意味
<i>n</i> yours	<i>yours</i> の <i>n</i> 個の非衝突セグメントが <i>base</i> と異なっています。
<i>n</i> theirs	<i>theirs</i> の <i>n</i> 個の非衝突セグメントが <i>base</i> と異なっています。
<i>n</i> both	<i>theirs</i> と <i>yours</i> の双方に <i>n</i> 個の等しい非衝突セグメントがありますが、 <i>base</i> とは異なっています。
<i>n</i> conflicting	<i>theirs</i> と <i>yours</i> の <i>n</i> セグメントが互いに異なり、 <i>base</i> とも異なります。

衝突しているチャンクがない場合、PERFORCE が生成したマージ・ファイルを受け入れてもほぼ安全です。PERFORCE が、*yours* と *theirs* からのすべての変更を *base* に代入するからです。

衝突しているチャンクがある場合、*merge* ファイルを編集する必要があります。この場合、PERFORCE は衝突している *yours*、*theirs*、および *base* テキストを *merge* ファイルに含めます。どのバージョンのチャンクを保持するかは、任意に選択してください。

異なるテキストは、ファイル・マーカにより明示されています。

```
>>>> ORIGINAL VERSION file#n
<text>
==== THEIR VERSION file#m
<text>
==== YOUR VERSION file
<text>
<<<<
```

保持したいテキストを選択し、衝突しているチャンクとすべての識別マーカを削除してください。

オプション

-am	解決ダイアログをスキップします。次のようにして自動的にファイル衝突を解決します。
-af	
-as	<ul style="list-style-type: none"> -am: 自動モード。PERFORCE 推奨のファイル・リビジョンを自動的に受け入れます。
-at	<ul style="list-style-type: none"> <i>theirs</i> が <i>base</i> と等しい場合、<i>yours</i> を受け入れます。<i>yours</i> が <i>base</i> と等しい場合、<i>theirs</i> を受け入れます。<i>yours</i> と <i>theirs</i> が <i>base</i> と異なり、<i>yours</i> と <i>theirs</i> の間に衝突がない場合、<i>merge</i> を受け入れます。上記以外の場合、<i>yours</i> と <i>theirs</i> の間に衝突があるため、このファイルをスキップします。
-ay	<ul style="list-style-type: none"> -ay: <i>Yours</i> を受け入れ、<i>theirs</i> を無視します。 -at: <i>Theirs</i> を受け入れます。クライアント・ワークスペース内のファイルが上書きされますので、このオプションは注意して使用してください！ -as: 安全受け入れ。<i>yours</i> と <i>theirs</i> の片方が <i>base</i> と異なる場合（そして変更が共通である場合）、そのリビジョンを受け入れます。両方が <i>base</i> と異なる場合、このファイルをスキップします。 -af: 強制受け入れ。どんなファイルであれ <i>merge</i> ファイルを受け入れます。<i>merge</i> ファイルに衝突マーカがある場合、マーカはファイルに残されるため、ファイルを編集してマーカを削除する必要があります。
-dflag	<p>ファイルをマージする際、空白文字または行末規則に関して指定された差分を無視します。（これらのオプションを指定し、かつ差分が空白文字だけだった場合、p4 resolve はクライアントのファイル内にあるテキストを使用します。）</p> <ul style="list-style-type: none"> -db: 空白文字のみの変更を無視します。（例えば、タブを 8 個の空白文字に置換した場合。） -dw: 空白文字を完全に無視します。（例えば、タブやその他の空白文字を削除した場合。） -dl: 行末規則の相違を無視します。
-f	すでに衝突解決済みであるがまだサブミットされていないファイルの衝突再解決を可能にします。
-n	衝突解決が必要なファイルを、実際に衝突解決を実行せずに一覧表示します。
-o	衝突解決時に使用されるベース・ファイルの名前とリビジョンを出力します。
-t	バイナリ（非テキスト）ファイルに対しても、3 ウェイマージを強制します。これにより、あらゆるタイプのファイル間の差分を検査できます。P4MERGE が、非テキスト・ファイルをマージできるユーティリティに設定されている場合、ユーザに非テキスト・ファイルをマージさせます。
-v	<i>yours</i> と <i>base</i> 間、および <i>theirs</i> と <i>base</i> 間のすべての変更を示す衝突マーカを、ファイルに含めます。通常、 <i>yours</i> と <i>theirs</i> が衝突している場合にのみ、衝突マーカが含まれます。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか	最低限必要なアクセス・レベル
使用不可	使用不可	open

◆ p4 resolve は、すでに衝突解決がスケジュールされているファイルにのみ作用します。ファイルの衝突解決をスケジュールする操作は3つあります。

- p4 integrate によりファイルの反映を行います。
- 現在の最新リビジョン以外のリビジョンと同期した作業中ファイルをサブミットします。サブミットは失敗し、ファイルの衝突解決がスケジュールされます。
- 作業中ファイルに対してp4 submitを実行する代わりに、p4 syncを実行します。クライアント・ワークスペースには何もコピーされず、代わりにファイルの衝突解決がスケジュールされます。(p4 sync 経由でスケジュールすることの利点は、サブミットが失敗しないことです。)

p4 resolve をファイル引数なしで実行すると、すでに衝突解決がスケジュールされているクライアント・ワークスペース内のすべてのファイルに対して作用します。

関連コマンド

衝突解決済みであるがサブミットされていないファイルの一覧を表示する。	p4 resolved
2つの異なるファイル間の変更の伝達をスケジュールする。	p4 integrate
変更済ファイル・セットをディポにサブミットする。	p4 submit
クライアント・ワークスペースにファイルをコピーするか、作業中ファイルの衝突解決をスケジュールする。	p4 sync

p4 resolved

概要

ファイルの衝突を解決済みでまだサブミットされていないファイルの一覧を表示します。

構文

```
p4 [g-opts] resolved [-o][file...]
```

解説

p4 resolved は、ファイルの衝突については解決済みであるが、まだサブミットされていないファイルの一覧を表示します。ファイルは、次の形式で 1 行に 1 つずつ表示されます。

```
localFilePath - action from depotFilePath#revisionRange
```

localFilePath は、ローカル・ホスト上の解決済ファイルの完全なパス名です。*depotFilePath* は、ディポの最上階層を基準としたディポ・ファイルの相対パスです。*revisionRange* は反映されたリビジョン範囲で、*action* は merge、branch、delete のいずれかです。

ファイル・パターンを引数に指定した場合、そのファイル・パターンに適合し、ファイルの衝突を解決済みの、サブミットされていないファイルのみがリストに入ります。

p4 resolved という名前から、p4 resolve プロセスを経由したファイルのみが表示されるかのように見えますが、そうとは限りません。ブランチ作成や削除を目的として p4 integrate で作業状態にしたファイルも、p4 resolve で衝突を解決したファイルも、「衝突が解決された」とみなされます。

オプション

-o	衝突解決時に使用されたベース・ファイルの名前とリビジョンを出力します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか	ファイル引数にリビジョン範囲を使えるか	最低限必要なアクセス・レベル
適用外	適用外	open

関連コマンド

サブミットされた反映の一覧を参照する。	p4 integrated
衝突解決されていない反映の一覧を参照する。	p4 resolve -n
あるファイルから別のファイルへのチェンジの伝達をスケジュールする。	p4 integrate
ファイル衝突を解決するか、p4 integrate でスケジュールしたとおりにチェンジを伝達する。	p4 resolve

p4 revert

摘要

作業中のファイルに加えられた変更を廃棄します。

構文

```
p4 [g-opts] revert [ -a -n -k -c changelist# ] file...
```

解説

p4 revert は、作業中のファイルに加えられた変更を廃棄し、p4 sync でディポから最後に同期させたリビジョンに戻します。また、変更済ファイルに関連する作業中チェンジリストから削除します。

p4 delete により作業状態にしたファイルを元に戻すと、そのファイルはクライアント・ワークスペース内に復元されます。p4 add により作業状態にしたファイルを元に戻すと、PERFORCE はクライアント・ワークスペースのファイルには手を付けません。p4 integrate により作業状態にしたファイルを元に戻すと、PERFORCE はクライアント・ワークスペースからそのファイルを削除します。

オプション

-a	作業状態になってから（内容またはファイルタイプが）変更されていないファイルのみ元に戻します。 元に戻されるのは、次のクライアント・リビジョンを持っているファイルのみです。 <ul style="list-style-type: none">編集目的の作業状態にあるが、内容が未変更でファイルタイプが未変更であるファイル。p4 integrate により反映の目的で作業状態になったが、p4 resolve による衝突解決がまだ行われていない。
-n	実際に元に戻す操作は実行せず、元に戻した結果をリストします。 このオプションにより、事前に元に戻したときの結果を確認できます。
-k	ワークスペース・ファイルを保存します。ファイルはチェンジリストから削除され、もはや作業状態でないことがサーバに記録されますが、クライアント・ワークスペース内では変更されません。
-c changelist#	指定されたチェンジリスト内のファイルのみ元に戻します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル変数にリビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	open

- p4 revert は、一般的にファイル引数を **必要とする** という点で他のほとんどの PERFORCE コマンドと異なります。コマンド・ラインのファイル引数とクライアント・ワークスペースのビューの共通部分にあるファイルが元に戻されます。

-a オプションを使用している場合、ファイル引数を指定する必要はありません。

- 編集目的で作業状態にしたファイルを元に戻すと、ファイルが作業状態にされてからユーザが加えたすべてのチェンジが上書きされます。p4 revert を実行する前に、p4 revert -n によって結果を確認する方が安全だと言えます。

使用例

p4 revert //...	自分の作業中チェンジリストにあるすべての作業中ファイルを、作業前の状態に戻します。
p4 revert -c default //...	デフォルト・チェンジリスト内にある作業中のファイルを、すべて作業前の状態に戻します。
p4 revert -n *.txt	現在のディレクトリにあるすべての作業状態の .txt ファイルを元に戻した結果をプレビューします。実際には元に戻す操作は実行されません。
p4 revert -c 31 *.txt	現在のディレクトリでチェンジリスト 31 内にあり、拡張子 .txt を持つファイルを、すべて作業前の状態に戻します。
p4 revert -a	変更されていないすべてのファイルを元に戻します。このコマンドは多くの場合チェンジリストのサブミット前に使用されます。

関連コマンド

追加を目的としてファイルを作業状態にする。	p4 add
削除を目的としてファイルを作業状態にする。	p4 delete
作業中ファイルをすべてディポにコピーする。	p4 submit
ディポからクライアント・ワークスペースへファイルを読み込む。	p4 sync
すべての作業中ファイルを一覧表示する。	p4 opened
PERFORCE がユーザのファイルであると判断したファイルを、強制的にクライアント・ワークスペースと同期させる。その際、作業状態にない書き込み可能なファイルはすべて上書きされる。	p4 sync -f

p4 review

概要

指定されたチェンジリスト番号より大きい番号を持つサブミット済チェンジリストを一覧表示します。

構文

```
p4 [g-opts] review [-c changelist#] [-t countername]
```

解説

p4 review -c changelist# は、changelist# と、それより大きい番号を持つサブミット済チェンジリストを、最大番号のサブミット済チェンジリストまですべて一覧表示します。リストの各行は次のような形式となります。

```
Change changelist# username <email-addr> (realname)
```

username、email-addr、およびrealnameは、p4 reviewが実行される時、必ずusername に対する p4 user フォームから取り込まれます。

p4 review -t counternameとして使用した場合、PERFORCEのカウンタ変数counternameの値より大きい値のサブミット済チェンジリストがすべて一覧表示されます。(カウンタは p4 counter で設定します。)引数なしで使用した場合、p4 review はすべてのサブミット済チェンジリストを一覧表示します。

p4 review コマンドは、PERFORCE にアクセスする外部プログラムで使用します。PERFORCE のチェンジ・レビュー・デーモンは (『システム管理者ガイド』の解説をご覧ください)、当社の Web サイトから入手でき、p4 review を使用します。

オプション

-c changelist#	changelist# とそれより大きい番号のサブミット済チェンジリストをすべて一覧表示します。
-t countername	PERFORCE のカウンタ countername より大きい値のサブミット済チェンジリストをすべて一覧表示します。
-c changelist# -t countername	カウンタ変数 countername の値を changelist# に設定します。このコマンドは p4 counter に変わりましたが、下位互換性を保つために維持されています。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか?	ファイル引数にリビジョン範囲を使えるか?	最低限必要なアクセス・レベル
適用外	適用外	review

- p4 review、p4 reviews、および p4 counter コマンドは、すべて、PERFORCE にアクセスする外部プログラムで使用されることを想定しています。
- p4 counter と同じ警告が適用されます。

関連コマンド

あるファイルを参照するために登録したユーザを一覧表示する。	p4 reviews
PERFORCE カウンタの設定や読み取りを行う。	p4 counter
あるチェンジリストに関する完全な情報を参照する。	p4 describe
ある基準で制限したすべてのチェンジリストの一覧を参照する。	p4 changes

p4 reviews

摘要

あるファイルをレビューできるように登録しているすべてのユーザを一覧表示します。

構文

```
p4 [g-opts] reviews [-c changelist#] [file...]
```

解説

p4 reviews は、PERFORCE にアクセスする外部プログラムでの使用を想定しています。

ファイルをレビューできるように登録するには、レビューしたいユーザの p4 user フォームのフィールド [Reviews:] (レビュー) にファイル・パターンを入力します。

p4 reviews -c changelist# は、サブミット済のチェンジリスト changelist# に含まれている任意のファイルをレビューできるように登録しているユーザをリストします。このコマンドの代わりに p4 reviews file... のフォームを使うと、引数として指定したファイル・パターンに適合する任意のファイルをレビューできるように登録しているユーザがリストされます。p4 reviews に引数を指定しない場合、ファイルをレビューできるように登録しているユーザがすべてリストされます。

オプション

-c changelist#	サブミット済のチェンジリスト changelist# に含まれている任意のファイルをレビューできるように登録しているユーザをすべて一覧表示します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定 子を使えるか?	ファイル引数にリビジョン範囲 を使えるか?	最低限必要なアクセス・レベル
使用不可	使用不可	list

- 構文 p4 reviews -c changelist# file... は、ファイル引数を完全に無視します。
- p4 reviews は変わったコマンドです。外部デーモンをサポートするよう作成されましたが、きわめて特別な意味を持つ、p4 users フォームのフィールド [Reviews:] なしでは何も行いません。

もっと一般化されたデーモンを作成するため、PERFORCE が本来意図していない値を、フィールド [Reviews:] に入力することができます。例えば、PERFORCE では、新しいジョブがサブミットされたときに、ジョブをレビューできるように登録している全ユーザに電子メールを送るジョブ・デーモンを実行しています。しかし、ユーザがジョブをレビューできるように登録する手段が PERFORCE には何も組み込まれていないため、フィールド [Reviews:] の 1 行をむりやり利用します。PERFORCE は、実在していないパス //depot/jobs/... をレビューするべく登録した全ユーザにジョブ電子メールを送ります。

関連コマンド

ファイルをレビューできるように登録する。	p4 user
指定されたチェンジリスト番号より大きい番号のサブミット済チェンジリストをすべてリストする。	p4 review
PERFORCE のカウンタ値の設定や読み取りを行う。	p4 counter
特定のチェンジリストに関する完全な情報を読み取る。	p4 describe

p4 set

概要

PERFORCE の変数を、Windows のレジストリで設定します。

構文

```
p4 [g-opts] set [ -s ] [ -S svcname ] [ var=[value] ]
```

解説

PERFORCE のクライアントとサーバは、ある一定のシステム変数を使用します。

Windows では、これらの変数の値を p4 set によりレジストリに設定できます。同様のことを他の OS で実行する場合、PERFORCE は環境変数を使用します。

現在のユーザに対してレジストリ変数の値を設定するには、p4 set var=value を使用します。Windows 管理者は、p4 set -s var=value を使用して、ローカル・マシン上のすべてのユーザに対して、レジストリ変数のデフォルト値を設定できます。

PERFORCE サーバをサービスとして実行している Windows 管理者は、p4 set -S svcname var=value によりサービスが使用している変数 (例えば、P4JOURNAL など) を設定できます。

ある変数に対して値を未設定にするときは、value を空白のままにします。

すべての PERFORCE 変数の値の一覧を参照するには、p4 set を引数なしで使用します。UNIX では、関連のある環境変数の値が表示されます。Windows では、MS-DOS 環境変数 (設定されている場合) またはレジストリの値が表示され、さらに p4 set (現在のユーザ用) と p4 set -s (ローカル・マシン用) のどちらを使用して定義されたかが表示されます。

Windows 以外の OS では、p4 set を使用して変数の値を参照できますが、Windows 以外の OS 上で p4 set を使用して変数を設定しようとする、PERFORCE はエラー・メッセージを表示します。

オプション

-s	ローカル・マシンのレジストリ変数の値を設定します。 このオプションがないと、p4 set は HKEY_CURRENT_USER ハイブに変数を設定します。-s オプションを使用すると、変数は HKEY_LOCAL_MACHINE ハイブに設定されます。 これらの場所の違いは、Windows 上での p4 set の出力で確認できます。
-S svcname	svcname という名前のサービスが使用するレジストリ変数の値を設定します。これを実行するには、管理者特権を持っている必要があります。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使用できるか?	ファイル引数にリビジョン範囲を使用できるか?	最低限必要なアクセス・レベル
適用外	適用外	list

- 本書の「環境変数」のセクションに、PERFORCE の変数の一覧と説明があります。
- Windows でレジストリの値を変更しても、影響を受けるのはローカル・マシンだけです。したがって、PERFORCE の Windows サービスに対する P4JOURNAL の設定は、サービスが起動されているマシン上で行う必要があります。

- Windowsでは、以下の3つの方法のどれかでレジストリ・キーの値をオーバーライドできます。
 - 同じ名前の環境変数の優先順位が高い
 - P4CONFIG ファイル内の値の方がさらに優先される
 - グローバル・オプションのオプションが最も優先される
- もし、Windows マシンにおいて UNIX 系の環境をご利用の場合(例えば Cygwin)、p4 set ではなく環境変数をお使いください。(この場合、PERFORCE コマンドライン・クライアントは、単純にそれが UNIX 環境であるかのように振舞います。)

使用例

p4 set	すべてのプラットフォームにおいて、PERFORCE 変数の一覧を、値を変更しないで表示します。
p4 set P4MERGE=	Windows において、P4MERGE の値を未設定にします。
p4 set P4PORT=tea:1666	Windows においてレジストリ変数を設定し、ホスト teaにあるPERFORCEサーバのポート1666と通信するよう PERFORCE クライアントに通知します。
p4 set -s P4PORT=tea:1666	上と同じように P4PORT を設定しますが、あるユーザに対してではなく、そのシステム上のすべてのユーザに対して設定します。
p4 set -S p4svc P4PORT=1666	NT サービスの p4svc に対して、PERFORCE クライアントからの接続要求をポート番号 1666 で接続待ちするよう指示します。 この操作には、Administrator の権限が必要です。
p4 set P4EDITOR="C:\File Editor\editor.exe"	Windows 上で、現在のローカル・ユーザに対して、デフォルトのエディタへのパスを設定します。 エディタへのパスが空白を含んでいる場合は、パスを引用符で囲む必要があります。

p4 sizes

概要

ディポ内にあるファイルの大きさの情報を表示します。

構文

```
p4 [g-opts] sizes [ -a -s -b blocksize ] file[revRange] ...
```

解説

p4 sizes コマンドは、ディポ内に格納されているファイルの大きさを表示します。オプションを付けずに実行すると、単一または複数のファイルの最新リビジョンの大きさだけが表示されます。1ファイルにつき1行ずつ出力されます。

-a オプションを使用すると、指定したリビジョン範囲にある最新リビジョンのみでなく、その範囲の各リビジョンによってどれだけのスペースが占有されているかが確認できます。1ファイル、1リビジョンにつき1行ずつ出力されます。

-s オプションを使用すると、指定されたすべてのファイルの合計が表示されます。出力は1行のみで、ファイル指定、総ファイル数、必要とされる総バイト数、および (-b オプションが指定されている場合) 必要とされる総ブロック数が表示されます。

オプション

-a	範囲内の最新リビジョンのみではなく、範囲内のすべてのリビジョンを含めます。
-b blocksize	結果を blocksize バイトのブロック単位で表示します。合計のファイル・サイズは直近の blocksize バイトに切り上げられます。
-s	指定されたファイル引数のファイル・サイズの合計を計算します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数に リビジョン指定子を使えるか？	ファイル引数に リビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	list

- p4 sizes コマンドの機能は、UNIX の du コマンドに類似しています。
- リビジョン範囲が指定されていない場合、暗黙的なリビジョン範囲の #1 から #head が想定されます。
- ファイル・サイズはクライアント・ワークスペースに同期されたときのファイルの状態に関係なく、標準 (UNIX の行送り規則) かつ圧縮されていないバージョンのディポ・ファイルに基づきます。

使用例

p4 sizes file.c	ディポ内の file.c の最新リビジョンの大きさを示します。
p4 sizes -a file.c	ディポに格納されている file.c の各リビジョンの大きさを示します。
p4 sizes -s -a file.c	ディポに格納されている file.c のすべてのリビジョンの合計サイズを示します。

```
p4 sizes -s -a -b 512 //depot/... //depot/...で稼動しているPERFORCEサーバが現在  
使用しているファイル数およびディスク容量の合計(バ  
イト単位で512バイトのブロック)を示します。  
  
p4 sizes -s //workspace/... ファイル数および workspace というクライアント・  
ワークスペースにマップされたファイルの最新リビ  
ジョンを同期するために必要なローカルのディスク容  
量の合計 (バイト単位)
```

p4 submit

概要

作業状態のファイルに加えられた変更をディポに送ります。

構文

```
p4 [g-opts] submit [-r] [-s] [-f submitoption]
p4 [g-opts] submit [-r] [-s] [-f submitoption] files
p4 [g-opts] submit [-r] [-f submitoption] -d description
p4 [g-opts] submit [-r] [-f submitoption] -d description files
p4 [g-opts] submit [-r] [-f submitoption] -c changelist#
p4 [g-opts] submit -i [-r] [-s] [-f submitoption]
```

解説

p4 add、p4 edit、p4 delete、および p4 integrate を使用してファイルを作業状態にすると、そのファイルは チェンジリスト に入ります。p4 submit を使用してチェンジリストをディポに送らない限り、ユーザがファイルに加える変更は、クライアント・ワークスペース・コピー内だけで実行されます。

デフォルトでは、デフォルト・チェンジリスト内でファイルが作業状態になりますが、新しい番号付チェンジリストを p4 change で作成できます。デフォルト・チェンジリストをサブミットするには p4 submit を使用し、番号付チェンジリストをサブミットするには p4 submit -c changelist# を使用します。

デフォルトでは、チェンジリスト内のすべてのファイルがディポにサブミットされ、編集、追加、およびブランチを目的としたファイルの作業状態は、ファイルへの変更の有無にかかわらず、サブミットによって解除されます。このデフォルト動作を変更するには、自分のワークスペースの p4 client フォームで [SubmitOptions:] フィールドを設定します。自分のワークスペースの [SubmitOptions:] の設定をコマンドラインから上書きするには、p4 submit -f submitoption を使用します。

p4 submit をデフォルト・チェンジリストとともに使用すると、EDITOR (または P4EDITOR) の環境変数またはレジストリ変数で定義したエディタ内に編集用のフォームが表示されます。フォームからファイルを削除することによりチェンジリストからファイルを削除できますが、これらのファイルは次のデフォルト・チェンジリスト内で作業状態のままとなります。ファイルを作業状態から解除し、すべてのチェンジリストから除去するには、p4 revert を使用します。

すべてのチェンジリストにはフィールド [Status:] (ステータス:) があり、このフィールドの値は [pending] (作業中) または [submitted] (サブミット済) です。サブミット済チェンジリストは、p4 submit で正常にサブミットされたもので、作業中チェンジリストはユーザが作成したがまだ正常にサブミットされていないものです。

p4 submit はチェンジリスト単位で機能します。このコマンドにより、チェンジリスト内のすべてのファイルがディポ内に保存されるか、1 つも保存されないかのいずれかになります。p4 submit は、中断されたり、チェンジリスト内のファイルが 1 つでもクライアント・ワークスペース内に見つからない場合、ファイルが 1 つでも他のクライアント・ワークスペース内でロックされている場合、衝突解決が必要でまだ解決されていない状態の場合、失敗します。

デフォルト・チェンジリストの処理中に p4 submit が失敗すると、デフォルト・チェンジリストにはチェンジリストの次の続き番号が割り当てられ、デフォルト・チェンジリストは空になります。サブミットに失敗したチェンジリストは、問題を解決した後、割り当てられた番号で再度サブミットする必要があります。

チェンジリストの説明をコマンドラインから指定するには、`-d` オプションを使用します。説明変更ダイアログは表示されません。`-d` オプションはデフォルト・チェンジリストにのみ作用し、番号付チェンジリストでは機能しません。

フォーム・フィールド

フィールド名	タイプ	説明
Change:	読み取り専用	チェンジ番号またはデフォルト・チェンジをサブミットする場合は <code>new</code> 。
Client:	読み取り専用	現在のクライアント・ワークスペースの名前。
User:	読み取り専用	現在の PERFORCE ユーザの名前。
Status:	読み取り専用値	<code>pending</code> 、 <code>submitted</code> 、または <code>new</code> のいずれか。ユーザは編集できません。 チェンジリストを作成したときは <code>new</code> 、チェンジリストを作成したがまだ <code>p4 submit</code> でディポにサブミットしていないときは <code>pending</code> 、 <code>p4 submit</code> でディポ内にチェンジリストの内容を保存したときは <code>submitted</code> です。
Description:	書き込み可能	チェンジリストのテキストによる説明。この値は <u>変更されなければなりません</u> 。
Jobs:	リスト	このチェンジリストで修正されたジョブの一覧。 デフォルトでは、 <code>p4 user</code> フォームで指定したジョブ・ビュー基準に適合するジョブが一覧表示されますが、一覧から削除することもできます。適切なジョブがない場合、このフィールドは表示されません。
Files:	リスト	このチェンジリストにサブミットしようとしているファイルの一覧。この一覧からファイルを削除してもかまいませんが、ファイルの変更や追加はできません。

オプション

<code>-c changelist#</code>	番号が <code>changelist#</code> であるチェンジリストをサブミットします。 チェンジリストへの番号の割り当ては、 <code>p4 change</code> によりユーザが手動で行うか、デフォルト・チェンジリストのサブミットに失敗したときに PERFORCE が自動で行うかのいずれかです。
<code>-d description</code>	コマンドラインから <code>description</code> を指定してデフォルト・チェンジリストをただちにサブミットし、対話式のフォームを表示させません。このオプションはスクリプト実行時に有用ですが、ジョブの追加やデフォルト・チェンジリストの修正はできません。

<code>-f submitoption</code>	<p>p4 client フォームでの [SubmitOptions:] の設定を上書きします。有効な <code>submitoption</code> の値は以下のとおりです。</p> <ul style="list-style-type: none"> • <code>submitunchanged</code> すべての作業状態のファイル（変更の有無にかかわらず）がディポにサブミットされます。これは PERFORCE のデフォルトの動作です。 • <code>submitunchanged+reopen</code> すべての作業状態のファイル（変更の有無にかかわらず）がディポにサブミットされ、すべてのファイルが自動的にデフォルト・チェンジリストで再度作業状態にされます。 • <code>revertunchanged</code> 内容またはファイルタイプが変更されたファイルのみがディポにサブミットされます。未変更のファイルは元に戻します。 • <code>revertunchanged+reopen</code> 内容またはファイルタイプが変更されたファイルのみがディポにサブミットされ、デフォルト・チェンジリストで再度作業状態にされます。未変更のファイルは元に戻しますが、デフォルト・チェンジリストでは作業状態にされません。 • <code>leaveunchanged</code> 内容またはファイルタイプが変更されたファイルのみがディポにサブミットされます。未変更のファイルはデフォルト・チェンジリストに移動されます。 • <code>leaveunchanged+reopen</code> 内容またはファイルタイプが変更されたファイルのみがディポにサブミットされます。未変更のファイルはデフォルト・チェンジリストに移動され、変更済みのファイルはデフォルト・チェンジリストで再度作業状態にされます。このオプションは <code>submitunchanged+reopen</code> と似ていますが、未変更のファイルはディポにサブミットされません。
<code>-i</code>	標準入力からチェンジリスト仕様を読み取ります。入力は、p4 submit フォームで使用されているものと同じ形式でなければなりません。
<code>-r</code>	編集目的または追加目的の作業状態が、サブミット後もデフォルト・チェンジリストにおいて、引き続き作業状態になります。
<code>-s</code>	<p>このチェンジリストのサブミット時に、ジョブにデフォルトの状態 <code>closed</code> ではなく、任意の状態を割り当てます。ジョブを未変更のままにしておくには、特殊な状態である <code>same</code> を使用します。</p> <p>新規のチェンジリストでは、修正完了の状態として <code>ignore</code> という特殊な状態が表示されます（状態が変更されなければ、ジョブはチェンジリストをサブミットしても修正完了になりません）。</p> <p>このオプションは、<code>p4 fix -s</code> と合わせて動作します。これは欠陥追跡システムと連携して使用することを目的としたオプションです。</p>
<code>g_opts</code>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	write

- ディポ内でのファイルの位置は、p4 client フォーム内で設定されるクライアント・ビューに応じて、クライアント・ワークスペース内のファイル位置との共通部分により決まります。
- p4 submit はチェンジリスト単位で実行されるため、ファイルを目的別にチェンジリストにグループ化することができます。例えば、1つのチェンジリストに、1つのバグを修正する3つのファイルへの変更を含ませることができます。

- 番号付きチェンジリストとともに p4 submit を使用すると、フォームが表示されません。番号付きチェンジリストに関するコメントの情報を変更するには、`p4 change -c changelist#` を使用します。
 - デフォルト・チェンジリストの p4 submit に対するパラメータとして、単一のファイル・パターンを指定できます。このファイル・パターンにより、デフォルト・チェンジリスト内のどのファイルをサブミットに含めるかを限定します。このファイル・パターンに適合しないファイルは、次のデフォルト・チェンジリストに移されます。
- p4 submit に対するファイル・パターンのパラメータは、デフォルト・チェンジリストをサブミットするときのみ使用できます。

使用例

<code>p4 submit</code>	デフォルト・チェンジリストをサブミットします。このチェンジリスト内のファイルのユーザ・リビジョンは、ディポ内に保存されます。
<code>p4 submit -c 41</code>	チェンジリスト 41 をサブミットします。
<code>p4 submit *.txt</code>	デフォルト・チェンジリスト内で拡張子 <code>.txt</code> を持つファイルのみをサブミットします。デフォルト・チェンジリスト内の他のファイルをすべて、次のデフォルト・チェンジリストに移します。
<code>p4 submit -d "header files" *.h</code>	デフォルト・チェンジリスト内で拡張子 <code>.h</code> を持つファイルのみを "header files" というコメントを付けてサブミットします。チェンジリスト・フォームは表示されません。デフォルト・チェンジリスト内の他のファイルをすべて、次のデフォルト・チェンジリストに移します。

関連コマンド

新しい番号付きチェンジリストを作成する。	p4 change
クライアント・ワークスペース内のファイルを作業状態にし、それをチェンジリストにリストする。	p4 add p4 edit p4 delete p4 integrate
ファイルのあるチェンジリストから別のチェンジリストに移す。	p4 reopen
あるファイルをすべてのチェンジリストから除去し、そのファイルを以前の状態に戻す。	p4 revert
ある基準に適合するチェンジリストの一覧を参照する。	p4 changes
あるチェンジリストの完全な記述を読み込む。	p4 describe
ディポ内のファイルをクライアント・ワークスペースに読み込む。	p4 sync
クライアント・ワークスペース内のファイルと、ディポ内のファイル間のマッピングを編集する。	p4 client

p4 sync

概要

ファイルをディポからワークスペースにコピーします。

構文

```
p4 [g-opts] sync [-f] [-n] [-k] [file[revRange]...]
p4 [g-opts] sync [-n] [-p] [file[revRange]...]
```

解説

p4 sync は、ファイル・パターン引数に適合するファイルを、ディポからクライアント・ワークスペースにコピーすることにより、クライアント・ワークスペースをディポと同期させます。コマンドラインにファイル・パターンを指定しない場合、次の基準のすべてを満たしている特定のディポ・ファイルをコピーします。

- クライアント・ワークスペース・ビューを通して見えるファイル
- p4 edit、p4 delete、p4 add、または p4 integrate により作業状態になっていないファイル
- すでにクライアント・ワークスペース内に最後のリビジョン（最新リビジョン）として存在していないファイル

新しい空のワークスペースでは、すべてのディポ・ファイルが後の2つの基準に適合するため、ユーザのビューを通して見ることのできるすべてのファイルが、ワークスペースにコピーされます。

コマンドラインでファイル・パターンを指定すると、そのファイル・パターンに適合し、かつ上記の基準に適合するファイルのみがコピーされます。

ファイル・パターンにリビジョン指定子が含まれている場合、指定したリビジョンがクライアント・ワークスペースにコピーされます。

ファイル引数にリビジョン範囲が含まれている場合、そのリビジョン範囲によって選択されたファイルのみが更新され、範囲中で最も大きい番号のリビジョンが使用されます。ファイル引数にリビジョン範囲が含まれている場合、もうワークスペース・ビュー内に存在しないファイルには影響しません。

新しく同期したファイルは、p4 edit か p4 delete により作業状態にしない限り、編集目的に使用できません。新しく同期したファイルは読み取り専用で、p4 edit か p4 delete によりファイルが書き込み可能になります。ファイルを書き込み可能にするときは、通常 PERFORCE の機能を使用し、OS のコマンドを使用しないでください。

オプション

-f	同期を強制実行します。すでにクライアント・ワークスペースに特定のリビジョンのファイルがあったとしても、PERFORCE は同期を実行します。ファイルが書き込み可能であっても、そのファイルは上書きされます。 このオプションは、作業状態のファイルには作用しませんが、クライアント・オプション noclobber をオーバーライド <u>しません</u> 。
-n	実際に同期を実行せずに、同期の結果を表示します。 このオプションにより、同期を実行する前に同期の結果を確認できます。

-k	既存のワークスペース・ファイルを保存します。クライアント・ワークスペースを更新せずに所有リストを更新します。所有リストを更新してクライアント・ワークスペースの実状に一致させる必要がある場合のみ p4 sync -k を使用してください。
-p	クライアント・ワークスペースにデータを書き込みますが、所有リストを更新しません。既に同期または作業状態にされたファイルは警告メッセージを出力し、処理しません。 このオプションは通常、初回の同期後にワークスペースの状態を追跡する必要のない（ビルド環境や公開用環境などの）プロセスで使用されるワークスペースに使用されます。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	read

- クライアント・ビューが最後の同期から変更されている場合は、次の同期のときに、クライアント・ビューを通して見えなくなっている（リビジョン範囲が使用されていない場合）ファイルがクライアント・ワークスペースから削除され、以前は見えなかったディポ・ファイルがクライアント・ワークスペースにコピーされます。

デフォルトでは、クライアント・ビュー内の空のディレクトリからファイルが削除されますが、ディレクトリ自体は削除されません。同期するときに空のディレクトリを削除するには、p4 client フォーム内の rmdir オプションを有効にします。

- ユーザが、PERFORCE の管理外で、OS のコマンドによりいずれかのファイルを書き込み可能にすると、通常 p4 sync はこれらのファイルを上書きしません。ただし、p4 client フォームの clobber オプションが有効になっている場合は、これらのファイルも上書きされます。
- p4 flush は p4 sync -k と同義です。p4 flush に適用されるすべての警告が p4 sync -k にも当てはまります。

使用例

p4 sync	クライアント・ビューを通してマッピングされたとおりに、すべてのファイルの最新リビジョンを、ディポからクライアント・ワークスペースにコピーします。そのファイルがすでにクライアント・ワークスペースで作業状態になっている場合や、そのファイルの最新リビジョンがクライアント・ワークスペースにある場合は、コピーされません。
p4 sync file.c#4	file.c の 4 番目のリビジョンをクライアント・ワークスペースにコピーします。例外は上記と同じです。
p4 sync //depot/proj1/...@21	クライアント・ビューを通してマッピングされたとおりに、//depot/proj1 ディレクトリ内の全ファイルを、ディポからクライアント・ワークスペースにコピーします。最新リビジョンはコピーしないで、チェンジリスト 21 がサブミットされた後、ディポ内にあるファイルのリビジョンを使用してください。

p4 sync @labelname	<p>labelnameがp4 labelによって作成されたラベルで、さらに p4 labelsync によってファイルが割り当てられている場合、labelname によって特定されたファイルおよびそのリビジョンが、ワークスペースにコピーされます。</p> <p>ただし、labelname にリストされているファイルであっても、クライアント・ビューで選択されていないファイルはコピーされません。</p> <p>また、labelname にリストされていないファイルは、ワークスペースから削除されます。(@labelname は、labelname で指定されたリビジョンを含め、それまでのすべてのリビジョンを指します。これには、リストにないファイルの存在しないリビジョンも含まれます。)</p>
p4 sync @labelname,@labelname	<p>p4 sync @labelnameと同様にラベルの内容をワークスペースにコピーしますが、labelname にリストされていないファイルは現状のままです。</p> <p>(リビジョン範囲 @labelname,@labelname は、ラベル名そのもので指定されたリビジョンだけを適用します。リストにないファイルの存在しないリビジョンは、含みません。)</p>
p4 sync @2001/06/24	<p>2001年6月24日 0:00 現在のディポの内容を、ワークスペースにコピーします。(6月23日に実施されたすべての変更を含みます。)</p>
p4 sync status%40junelst.txt	<p>文字の 16 進値の ASCII 表現を使用して、PERFORCE ワイルドカードが含まれたファイル名を同期させます。この例では、クライアント・ワークスペース内のファイルは status@junelst.txt です。</p> <p>詳細については、213 ページの「ファイル名およびエンティティに関する文字の制限」をご覧ください。</p>
p4 sync file.c#none	<p>file.c の存在しないリビジョンに同期します。ファイルはワークスペースから削除されます。</p>
p4 sync ...#none	<p>すべてのファイルの存在しないリビジョンに同期します。ワークスペース内のすべてのファイル (PERFORCE が制御しているファイル) は削除されます。</p>

関連コマンド

クライアント・ワークスペースにあるファイルを作業状態にし、それをチェンジリストにリストする。	<p>p4 add</p> <p>p4 edit</p> <p>p4 delete</p> <p>p4 integrate</p>
クライアント・ワークスペースにあるファイルに対する変更を、ディポにコピーする。	<p>p4 submit</p>
クライアント・ワークスペースに同期したファイルとリビジョンの一覧を参照する。	<p>p4 have</p>

p4 tag

概要

ファイルにラベルでタグ付けします。

構文

```
p4 [g-opts] tag [ -d -n ] -l labelname file[revRange]...
```

解説

p4 tag を使用して、指定したファイル・リビジョンにラベルでタグ付けします。labelname が必要です。labelname という名前のラベルがない場合は、その名前のラベルが自動的に作成されます。このラベルがすでに存在する場合、そのラベルでファイルのタグ付けまたはタグ解除を行うには、ユーザはそのラベルの Owner: でなければならず、ラベルは unlocked でなければなりません（ラベルの所有者やロック・ステータスを変更するには、p4 label を使用します）。

file 引数にリビジョン指定が含まれていない場合は、最新リビジョンがラベルでタグ付けされます。file 引数にリビジョン範囲指定が含まれている場合は、その範囲のリビジョンを持つファイルだけがタグ付けされます（指定した範囲に 1 つのファイルの複数のリビジョンが存在する場合は、指定した範囲で最も大きい番号のリビジョンがタグ付けされます）。

オプション

-d	指定したファイルからラベル・タグを削除します。
-n	実際の操作を実行せずに、p4 tag の動作内容を表示します。
-l labelname	ファイル・リビジョンに適用されるラベルを指定します。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	open

- デフォルトでは、p4 tag は、ディポ内のファイルの最新リビジョンに対して動作します。クライアント・ワークスペースの状態を保存するには、p4 labelsync を使用します。このコマンドはクライアント・ワークスペースに最後に同期されたファイルのリビジョンに対して動作します。

使用例

p4 tag -l rel1 //depot/1.0/...	//depot/1.0/... 内のファイルの最新リビジョンにラベル rel1 でタグ付けします。 ラベル rel1 が存在しない場合は、それを作成します。
p4 tag -l build //depot/1.0/...@1234	//depot/1.0/... 内のファイルでチェンジリスト 1234 のサブミット時点で最も新しいリビジョンにラベル build でタグ付けします。 ラベル build が存在しない場合は、それを作成します。
p4 files @labelname	labelname でタグ付けされたファイル・リビジョンをリストします。

関連コマンド

ラベルを作成または編集する。	p4 label
システムに認識されているすべてのラベルをリストする。	p4 labels
クライアント・ワークスペースのリビジョンにラベルでタグ付けする。	p4 labelsync
ラベルを作成して、そのラベルでファイルにタグ付けする。	p4 tag

p4 tickets

概要

p4 login によってユーザに許可されたすべてのチケットを表示します。

構文

```
p4 [g-opts] tickets
```

解説

p4 tickets コマンドは、ユーザのチケット・ファイルに保存されているすべてのチケットをリストします。

オプション

`g_opts` 「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定 子を使えるか？	ファイル引数にリビジョン範囲 を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	なし

- チケットは環境変数 P4TICKETS で指定したファイルに保存されます。環境変数が設定されていない場合、チケットは Windows では %USERPROFILE%\p4tickets.txt、その他の OS では \$HOME/.p4tickets に保存されます。

使用例

```
p4 tickets
```

ユーザのローカル・チケット・ファイルに保存されたすべてのチケットを表示します。

関連コマンド

ログイン・セッションを開始する (チケットを取得するため)	p4 login
ログイン・セッションを終了する (チケットを削除するため)	p4 logout

p4 triggers

概要

p4 triggers は、チェンジリストのサブミットやフォーム更新の都度、または PERFORCE を外部の認証メカニズムと統合するときに条件に応じて実行されるスクリプトのリストを編集します。

構文

```
p4 [g-opts] triggers
p4 [g-opts] triggers -i
p4 [g-opts] triggers -o
```

解説

PERFORCE トリガは、ユーザが記述するスクリプトであり、特定の操作（チェンジリストのサブミットやフォームの変更など）が実行されるたびに PERFORCE サーバによって呼び出されます。スクリプトが値 0 を返すと操作は続行し、他の値を返すと操作は失敗します。失敗したとき、スクリプトの標準出力（エラー出力ではありません）は、失敗した操作のエラー・メッセージのテキストとして使用されます。

PERFORCE は 12 種類のトリガ・タイプをサポートしており、それらは 4 つのカテゴリに分類されます。チェンジリスト・サブミット・トリガ (change-submit、change-content、および change-commit) はユーザがチェンジリストをサブミットしたときに起動されます。修正トリガ (fix-add および fix-delete) は、p4 fix、p4 submit、または p4 change コマンドによりチェンジリストに修正を追加または削除したときに起動されます。フォーム・トリガ (form-save、form-out、form-in、form-commit、form-delete) は、ユーザがフォーム仕様を生成または変更したときに起動されます。認証トリガ (auth-check および auth-set) は管理者が PERFORCE を LDAP や Active Directory などの外部認証メカニズムと統合しようとするときに起動されます。

change-submit トリガ・タイプを使用して、チェンジリスト作成の後、ファイルがサーバに転送される前に起動するトリガを作成します。change-submit トリガは、ファイルがサーバに転送される前に起動されるので、ファイルの内容にアクセスできません。サブミット・トリガは、システムファイルの内容にアクセスする必要のないレポート・ツールやシステムによる反映操作で役に立ちます。

content トリガ・タイプを使用して、チェンジリスト作成とファイル転送の後、データベースにサブミットをコミットする前に、起動されるトリガを作成します。

commit トリガ・タイプを使用して、チェンジリスト作成、ファイル転送、およびデータベースへのチェンジリストのコミットの後に起動されるトリガを作成します。チェンジリストのサブミットの成功を前提とする（または必要とする）プロセスの場合は、コミット・トリガを使用します。

change-submit トリガ・スクリプトや change-content トリガ・スクリプトが成功した場合でも、後続のトリガの失敗などが原因となってサブミットが失敗することがあります。change-submit トリガおよび change-content トリガは検証目的でのみ使用し、サブミットの正常完了を条件とする処理には change-commit トリガまたはデーモンを使用してください。

ユーザがジョブを修正したときにトリガ・スクリプトを実行するよう PERFORCE を設定するには、修正トリガを使用します。修正トリガのタイプは fix-add および fix-delete です。修正トリガを使用して、ユーザが修正をチェンジリストに追加または削除したときにコマンドを実行することができます。

ユーザが仕様フォームを編集したときにトリガ・スクリプトを実行するよう *PERFORCE* を設定するには、フォーム・トリガを使用します。フォーム・トリガのタイプは *form-save*、*form-in*、*form-out*、*form-commit*、*form-delete* です。フォーム・トリガを使用して、ユーザ用にカスタマイズされた仕様の生成、仕様フォームが変更される際の他ユーザへの通知、プロセス制御および管理ツールとのやり取りを行うことができます。

外部パスワード認証管理システム (LDAP や Active Directory など) を *PERFORCE* と共に使用するには、*認証トリガ* (*auth-check* および *auth-set*) を使用します。%user% 変数を使ってユーザ名をスクリプトのコマンドに渡します。認証プロセスの一部としてユーザに入力されたパスワードは、コマンドライン上ではなく標準入力として認証スクリプトに渡されます。詳しくは、『システム管理者ガイド』を参照してください。

トリガはテーブルにリストされた順序で起動します。トリガ・スクリプトが特定のタイプに対して失敗すると、そのタイプに関連している後続のトリガ・スクリプトも起動しません。

同じトリガ・スクリプトを複数のファイル・パターンで使用するには、トリガ・テーブルに同じトリガ・タイプを複数リストします。除外マッピングを使用して、ファイルがトリガ・スクリプトを有効にするのを防ぎます。この場合も、除外マッピングをビュー内で使用するときと同様に、トリガ・エントリの順序は重要です。特定のトリガの名前とタイプが複数リストされている場合、そのトリガの名前とタイプが最初に使用されているスクリプトだけがアクティブになります。

フォーム・フィールド

p4 triggers のフォームには、[Triggers:] (トリガ:) フィールドだけがあります。他の *PERFORCE* のフォームと同様、[Triggers:] フィールドの各行はタブでインデントします。各行には値が 4 つあり、以下のようになっています。

フィールド	意味
name	ユーザ定義のトリガ名。 連続する行に同一のトリガ名がある処理は、複数の <i>path</i> が指定できるように単一のトリガとして扱われます。この場合、そのようなトリガ行の最初の <i>command</i> のみが使用されます。
type	トリガ・タイプは 12 種類あり、4 つのサブタイプ (チェンジリスト・サブミット・トリガ、修正トリガ、フォーム・トリガ、認証トリガ) に分類されます。 チェンジリスト・サブミット・トリガ <ul style="list-style-type: none"> • <i>change-submit</i> : チェンジリストの作成後、ファイル転送の前にチェンジリスト・トリガを実行します。トリガはファイルの内容にアクセスできません。 • <i>change-content</i> : チェンジリストの作成およびファイルの転送後、ファイルのコミット前にチェンジリスト・トリガを実行します。 ファイルの内容を取得するには、リビジョン指定子 <i>@=change</i> を指定して <i>p4diff2</i>、<i>p4 files</i>、<i>p4 fstat</i>、<i>p4 print</i> などのコマンドを使用します。この場合、<i>change</i> は、%changelist% 変数でスクリプトに渡される作業中チェンジリストのチェンジリスト番号です。 • <i>change-commit</i> : チェンジリスト作成、ファイル転送、およびファイル・コミットの後、チェンジリスト・トリガを実行します。 修正トリガ <ul style="list-style-type: none"> 特殊変数 %jobs% を使用して拡張が可能です。この変数は <i>p4 fix</i> コマンドライン (あるいは <i>p4 change</i> フォームや <i>p4 submit</i> フォームの [Jobs:] フィールド) にリスト表示されるすべてのジョブに対して 1 つずつ引数を拡張します。そのため、トリガ・スクリプトに指定される最後の引数でなければなりません。 • <i>fix-add</i> : 修正を追加する前に修正トリガを実行します。 • <i>fix-delete</i> : 削除を追加する前に修正トリガを実行します。

フィールド	意味
	<p>フォーム・トリガ</p> <ul style="list-style-type: none"> • <code>form-save</code>: フォームの内容が解析された後、その内容が PERFORCE データベースに保存される前に、フォーム・トリガを実行します。トリガは、<code>%formfile%</code> 変数で指定されたフォームを変更できません。 • <code>form-out</code>: エンド・ユーザへのフォームの生成時にフォーム・トリガを実行します。トリガはフォームを変更できます。 • <code>form-in</code>: PERFORCE サーバによって内容が解析され、検証される前に、編集済みフォームに対してフォーム・トリガを実行します。トリガはフォームを変更できます。 • <code>form-delete</code>: フォームの内容が解析された後、その内容が PERFORCE データベースから削除される前に、フォーム・トリガを実行します。トリガはフォームを変更できません。 • <code>form-commit</code>: フォーム・トリガがコミットされた後それを実行し、ジョブ名、日付などの自動生成フィールドへのアクセスを有効にします。ジョブ・フォームの場合、このトリガは <code>p4 fix</code> コマンドと同様 <code>p4 job</code> コマンドでも実行されます (ステータス更新後)。<code>form-commit</code> トリガは <code>p4 job</code> によって作成された新しいジョブ名へのアクセス権を持っています。<code>form-in</code> トリガと <code>form-save</code> トリガはジョブ名が作成される前に実行されます。 <p>ジョブ・フォームの場合、このトリガは <code>p4 change</code> コマンド (ジョブがチェンジリストの <code>[Jobs:]</code> フィールドの編集によって追加または削除されている場合)、および <code>p4 submit</code> コマンド (チェンジリストの <code>[Jobs:]</code> フィールドに存在するすべてのジョブに対して) によっても実行されます。これらの場合は、ジョブの <code>form-commit</code> トリガ・コマンドラインで特殊変数 <code>%action%</code> を使用した拡張が可能です。このトリガはフォームを変更することはできません。</p> <p>認証トリガ:</p> <ul style="list-style-type: none"> • <code>auth-check</code>: 認証チェック・トリガを実行し、ログイン時または新規パスワードの設定時にパスワードを外部パスワード管理機構に対して検証します。<code>auth-check</code> トリガが存在する場合、認証はトリガ・スクリプトにより制御されるため、PERFORCEsecurity カウンタ (および関連するパスワードの長さについての必要条件) は無視されます。 • <code>auth-set</code>: 認証セット・トリガを実行し、新規パスワードを外部パスワード管理機構に送信します。 <p><code>auth-check</code> トリガを追加したら、PERFORCE サーバを再起動しなければなりません。</p> <p>チェンジリスト・サブミット・トリガ (<code>change-submit</code>、<code>change-content</code>、および <code>change-commit</code>) の場合は、ディポ・シンタックスのファイル・パターンです。このファイル・パターンと一致するファイルが含まれているチェンジリストをユーザがサブミットしたときに、このトリガにリンクされたスクリプトが実行されます。除外マッピングを使用すると、指定したファイルに対してトリガが実行されるのを防ぐことができます。</p> <p>修正トリガ (<code>fix-add</code> または <code>fix-delete</code>) の場合は、修正をパスの値として使用します。</p> <p>フォーム・トリガ (<code>form-save</code>、<code>form-out</code>、<code>form-in</code>、または <code>form-delete</code>) の場合は、フォームのタイプの名前 (<code>branch</code>、<code>change</code>、<code>client</code>、<code>depot</code>、<code>group</code>、<code>job</code>、<code>label</code>、<code>protect</code>、<code>spec</code>、<code>triggers</code>、<code>typemap</code>、または <code>user</code> のうちどれか) です。</p> <p>認証トリガ (<code>auth-check</code> または <code>auth-set</code>) の場合は、<code>path</code> の値に <code>auth</code> を指定してください。</p>
path	

フィールド	意味
command	<p>一致する <i>path</i> がトリガ・タイプに適用されたときに、PERFORCE サーバが実行するコマンドです。PERFORCE サーバのアカウントがコマンドを見つけて実行できるように、コマンドを指定します。コマンドは、引用符で囲む必要があります。また、『システム管理者ガイド』の「トリガ・スクリプトの変数」に示した変数を引数に取ることができます。</p> <p>change-submit トリガと change-content トリガの場合、トリガ・スクリプトが 0 で終了したときはチェンジリストのサブミットが実行され、スクリプトがゼロ以外の値で終了したときはチェンジリストのサブミットは失敗します。change-commit トリガの場合、トリガ・スクリプトの終了コードに関係なくチェンジリストのサブミットは成功します。しかし、スクリプトがゼロ以外の値で終了したときは、後続の change-commit トリガは起動しません。</p> <p>form-in トリガ、form-out トリガ、form-save トリガ、form-delete トリガの場合、スクリプトが 0 で終了したときに仕様内のデータが PERFORCE データベースに組み込まれます。ゼロ以外の値で終了したときは、データベースは更新されません。</p> <p>form-commit トリガ・タイプは絶対に変更を拒否しません。これは最初から存在するため、ジョブ作成処理中にスクリプトがジョブ番号に (%formname% 値から) アクセスできます。</p> <p>fix-add トリガと fix-delete トリガでは、トリガ・スクリプトが 0 で終了した場合は修正の追加や削除が実行し、スクリプトがゼロ以外の値で終了したときは失敗します。</p> <p>auth-check トリガ (p4 login により起動) の場合、ユーザが入力したパスワードは標準入力からトリガ・コマンドに渡されます。トリガの実行に成功すると、PERFORCE チケットが発行されます。ユーザ名はコマンドラインに %user% を渡すことにより入手できます。</p> <p>auth-set トリガ (p4 passwd により auth-check トリガによる検証成功後に起動) の場合、ユーザの古いパスワードと新しいパスワードは標準入力からトリガに渡されます。ユーザ名はコマンドラインに %user% を渡すことにより入手できます。</p>

オプション

-i	エディタを起動せずに、標準入力からトリガ・テーブルを読み取ります。
-o	エディタを起動せずに、トリガ・テーブルを標準出力に記述します。
<i>g_opts</i>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	super

警告 out トリガでは、同じ out トリガを起動する PERFORCE コマンドを使用しないでください。そのようなコマンドを使用すると、無限に反復することになります。例えば、job 仕様に対して起動する out トリガ・スクリプト内から p4 job -o を実行しないでください。

警告 トリガ・フォームから起動するトリガが作成されており、そのトリガが失敗して p4 triggers コマンドが動作しなくなった場合、回復させるにはサーバ・ルート・ディレクトリから db.triggers ファイルを削除するしかありません。

- ◆ トリガ・スクリプトに引数を渡すには、以下の変数を使用します。

引数	解説	使用可能なタイプ
%action%	チェンジリストまたはジョブに実行されるアクションを反映している、ヌルまたは文字列。 例えば、“pending change 123 added” や “submitted change 124 deleted” は change フォームで想定される %action% の値であり、“job000123 created” や “job000123 edited” はフォームで想定される %action% の値です。	form-commit
%changelist% %change%	サブミットするチェンジリストの番号（短縮形の %change% は %changelist% と同等です）。	change-submit、 change-content、 change-commit fix-add、 fix-delete、 form-commit
%client%	トリガを行うユーザのクライアント・ワークスペース名	すべて
%clienthost%	クライアントのホスト名	すべて
%clientip%	クライアントの IP アドレス	すべて
%jobs%	p4 fix コマンドで指定されたジョブ番号ごとに、または p4 submit フォームか p4 change フォームの [Jobs:] フィールドに追加（削除）されたジョブ番号ごとに、引数を1つずつ拡張したジョブ番号の文字列。	fix-add、 fix-delete
%oldchangelist%	サブミット時にチェンジリストの番号が付け替えられた場合、この値に古いチェンジリスト番号が収められます。	change-commit
%serverhost%	PERFORCE サーバのホスト名	すべて
%serverip%	サーバの IP アドレス	すべて
%serverport%	PERFORCE サーバの IP アドレスとポート。形式は <i>ip_address:port</i>	すべて
%serverroot%	PERFORCE サーバの P4ROOT ディレクトリ	すべて
%user%	トリガを実行するユーザの PERFORCE ユーザ名	すべて
%formfile%	一時フォーム仕様ファイルのパス。in トリガまたは out トリガからフォームを変更するには、このファイルを上書きします。タイプ save のトリガおよび delete のトリガの場合、このファイルは読み取り専用です。	form-commit、 form-save、 form-out、 form-in、 form-delete
%formname%	フォームの名前（ブランチ名やチェンジリスト番号など）	form-commit、 form-save、 form-out、 form-delete
%formtype%	フォームのタイプ (branch や change など)	form-commit、 form-save、 form-out、 form-in、 form-delete

- ◆ トリガ・スクリプトがチェンジリスト内のサブミットされた（サブミットされる）ファイルを認識する必要がある場合は、p4 opened -ac changelist を使用します。

- ◆ サブミット前トリガ・スクリプトは、サブミット対象のファイルの内容にサーバからアクセスすることはできません。これは、サブミット前トリガの実行の時点では、ファイルの内容はまだサーバに転送されていないためです。
- ◆ トリガ・スクリプト内の PERFORCE コマンドは、常に特定の PERFORCE ユーザが実行します。ユーザが指定されていないと、SYSTEM という名前のユーザ（または、UNIX 環境では、p4d プロセスを所有するユーザ）のための PERFORCE ライセンスが余分に使用されます。これを防ぐには以下の操作を行います。
 - 各 PERFORCE コマンドを呼び出すスクリプトに %user% 引数を渡して、各コマンドがそのユーザによって呼び出されるようにする。例えば、Joe が、トリガ・スクリプト trigger.pl を実行するチェンジリストをサブミットし、trigger.pl が p4 changes コマンドを呼び出す場合、このスクリプトはコマンドを p4 -u %user% changes として実行できます。
 - トリガ・スクリプトを実行するアカウントの P4USER を、既存ユーザの名前に設定する（PERFORCE サーバが Windows 上にサービスとしてインストールされている場合、Windows サービスは P4USER 値を持つことができないので、Windows 上で前述のようにユーザ値を各コマンドに渡す必要があります）。
- ◆ 4 つのフォーム・トリガ・タイプ（form-in、form-out、form-save および form-delete）の場合、ジョブ作成時に %formname% 変数は設定されません。この制約があるのは、ジョブ生成後までジョブの名前がサーバに認識されないためです。

ジョブ生成中にそのジョブ名にアクセスするには、form-commit トリガを使用します。これはジョブ作成処理の間にその（%formname% 変数内の）ジョブ名にアクセスできる唯一のトリガ・タイプです。

ジョブ生成後に、そのジョブに対する後続のユーザ変更で %formname% が正しく設定され、フォーム・トリガ・スクリプトによって使用されます。
- ◆ トリガ・タイプの名前はリリース 2005.2 において変更されました。以下に示す従来のトリガ・タイプ名も依然として機能しますが、推奨されません。

従来のトリガ・タイプ	新しいトリガ・タイプ (2005.2 時点)
submit	change-submit
content	change-content
commit	change-commit
out	form-out
in	form-in
save	form-save
delete	form-delete

- リリース 2007.3 の時点では、トリガの標準出力はトリガ・スクリプトの成功時と失敗時にクライアント・プログラムに送られます。それ以前は、標準出力がトリガの失敗時にのみクライアント・プログラムに送られていました。

使用例

トリガ・テーブルが、以下のエンタリで構成されていると仮定します。

```

trig1 change-submit //depot/dir/... "/usr/bin/s1.pl %changelist%"
trig2 change-submit //depot/dir/file "/usr/bin/s2.pl %user%"
trig1 change-submit -//depot/dir/z* "/usr/bin/s1.pl %user%"
trig1 change-submit //depot/dir/zed "/usr/bin/s3.pl %client%"

```

1行目と3行目は、両方ともスクリプト `/bin/s1.pl %changelist%` を呼び出します。最初に出現した特定のトリガ名が、それ以降にそのトリガ名が使用されたときに実行されるスクリプトを決定するからです。

ファイル `//depot/dir/zebra` がサブミットされた場合、どのトリガもアクティブになりません。3行目でこのファイルが除外されているからです。`//depot/dir/zed` がサブミットされた場合は、`trig1` のスクリプト `/usr/bin/s1.pl %changelist%` が実行されます。4行目は3行目をオーバーライドしますが、名前 `trig1` に関連付けられている最初のスクリプトだけが呼び出されます。

より詳細な例については、『システム管理者ガイド』をご覧ください。

関連コマンド

サブミットしようとしているチェンジリストの情報を取得する。	<code>p4 describe</code>
	<code>p4 opened</code>
デーモンの作成を支援する。	<code>p4 review</code>
	<code>p4 reviews</code>
	<code>p4 counter</code>
	<code>p4 counters</code>
	<code>p4 user</code>

p4 typemap

概要

p4 typemap は、ファイル名をファイルタイプにマッピングしているテーブルを修正します。

構文

```
p4 [g-opts] typemap
p4 [g-opts] typemap -i
p4 [g-opts] typemap -o
```

解説

p4 typemap コマンドにより、PERFORCE 管理者は、PERFORCE ファイルタイプをファイル指定にリンクするテーブルを設定できます。ファイル名がタイプマップ・テーブル中のエントリに適合する場合、このコマンドは、ファイル名に適合するファイルタイプがないときに PERFORCE クライアントが割り当てるファイルタイプをオーバーライドします。

デフォルトでは、PERFORCE はファイルの先頭の 8192 バイトの分析に基づいて、ファイルのタイプがテキストなのかバイナリなのかを自動的に判別します。先頭の 8192 バイトの各バイトで最上位ビットが 0 の場合、PERFORCE はこのファイルタイプをテキストとみなし、そうでなければバイナリとみなします。

このデフォルトの動作は、`-t filetype` オプションを使用すればオーバーライドできますが、この事実はよく見落とされます。ファイルタイプが通常は（常にではありません）正確に検出される場合において、特にそうです。典型的な例は、PDF ファイル（8192 バイト以上の ASCII コメントで始まるものがあります）や、通常は書式コードが埋め込まれている RTF ファイルの場合に見られます。

p4 typemap コマンドは、より完全な解決策を提供します。このコマンドで管理者は、特定のファイル（例えば名前が `.pdf` や `.rtf` で終わるファイル）がディポに追加される時、デフォルトのタイプ検出機構が働かないようにし、常に希望する PERFORCE ファイルタイプが確実に割り当てられるようにすることができます。

ユーザは、PERFORCE のコマンドラインでファイルタイプを明確に指定することにより、タイプマップ・テーブルで定義されたあらゆるファイルタイプ・マッピングをオーバーライドできます。

フォーム・フィールド

p4 typemap のフォームには [TypeMap:] フィールドだけがあり、このフィールドには、ディポ・シンタックスで指定されたファイルにファイルタイプをリンクする値が含まれます。

列	説明
<code>filetype</code>	有効な PERFORCE ファイルタイプ。 有効なファイルタイプのリストについては、「ファイルタイプ」のセクションをご覧ください。
<code>pattern</code>	ディポ・シンタックスのファイルパターン。 ユーザがこのパターンに適合するファイルを追加する場合、そのデフォルトのファイルタイプはテーブルで指定されたファイルタイプとなります。

オプション

`-i` ユーザのエディタを起動せずに、標準入力からタイプマップ・テーブルを読み取ります。

`-o` ユーザのエディタを起動せずに、標準出力にタイプマップ・テーブルを記述します。
`g_opts` 「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	admin、もしくは <code>-o</code> オプションを使用する場合は list

- 希望するサブディレクトリまたはその配下にあつて、特定の拡張子を持つすべてのファイルを指定するには、ディレクトリ名に続けてピリオドを4個使用し、その後に拡張子を付けます（例えば、`//path/....ext`）。最初の3つのピリオドは「このレベル以下にあるすべてのファイル」を指定します。4番目のピリオドと付属の拡張子は、「これらの文字で終わる」と解析されます。
- タイプマップ・テーブルでは、ファイルタイプ修飾子を使用できます。有効な用途としては、ディレクトリ・ツリー全体にわたるキーワード拡張を強制的にオン／オフするとか、元のファイルの更新時刻（`+m` ファイルタイプ修飾子）をサードパーティ製DLLのディレクトリに強制的に保存する、またはペシミスティック・ロッキングのポリシーを採用する場合、などがあります。
- `-t` オプションとファイルタイプ修飾子を使用してコマンドラインでファイルタイプを指定する場合、参照しているファイルが `p4 typemap` マッピングの対象になっていれば、コマンドラインで指定したファイルタイプは、タイプマップ・テーブルで指定したファイルタイプをオーバーライドします。

使用例

PERFORCE サーバに、すべての PDF および RTF ファイルをバイナリとみなすよう通知するには、`p4 typemap` を実行してタイプマップ・テーブルを以下のように変更します。

```
Typemap:
    binary //....pdf
    binary //....rtf
```

指定子中の最初の3個のピリオド（「`....`」）は、ルート・ディレクトリ配下のすべてのファイルがマッピングの一部に含まれることを規定する PERFORCE ワイルドカードです。4個目のピリオドとファイルの拡張子は、この指定子が「`.pdf`」（または「`.rtf`」）で終わるファイルに適用されることを規定しています。

サイトによっては、より複雑な状況が発生することもあります。例えば、ディポ内のあるエリアにいるユーザが、文書を収録しているプレーンな ASCII テキスト・ファイルに拡張子 `.doc` を使用する一方で、別のエリアで作業しているユーザが、人気ワープロ・ソフトで使用されているバイナリ・ファイル形式でファイルを参照するために `.doc` を使用する、といった状況です。このような状況で有効なタイプマップ・テーブルは、次のようになります。

```
Typemap:
    text //depot/dev_projects/....doc
    binary //depot/corporate/annual_reports/....doc
```

すべての .c および .h ファイルについてキーワード拡張を有効にし、一方、自分の .txt ファイルについてはキーワード拡張を無効にするには、次のようにします。

```
Typemap:
    text+k //depot/dev_projects/main/src/....c
    text+k //depot/dev_projects/main/src/....h
    text //depot/dev_projects/main/src/....txt
```

指定したディレクトリ内のファイルに、確実に元のファイルの修正時刻を（サブミットの日付に関係なく）保存させるには、次のようにします。

```
Typemap:
    binary //depot/dev_projects/main/bin/...
    binary+m //depot/dev_projects/main/bin/thirdpartydll/...
```

binディレクトリやその配下にあるすべてのファイルに、タイプbinaryが割り当てられます。後のマッピングが前のマッピングをオーバーライドするので、サブディレクトリbin/thirdpartydll内のファイルには、タイプbinaryではなくタイプbinary+mが割り当てられます。+m（修正時刻）ファイルタイプ修飾子について詳しくは、「ファイルタイプ」のセクションをご覧ください。

デフォルトでは、PERFORCE は同時並行開発をサポートしていますが、ファイルを編集のために作業状態にするのは一度に1人のユーザだけであることが想定される環境では、部分ファイルタイプで修飾子+1（排他的作業状態）を使用してペシミスティック・ロックを実行することができます。次のように [typemap:] を定義すると、+1 修飾子がディポ内にあるすべての新規追加されたファイルに対し自動的に適用されます。

関連コマンド

タイプマップ・テーブルをオーバーライドして、新しいファイルを `p4 add -t type file` 特定のタイプで追加する。

タイプマップ・テーブルのあらゆる設定をオーバーライドして、作業状態にあるファイルのファイルタイプを変更する。

p4 unlock

概要

p4 unlock は、ファイルのロックを解除します。

構文

```
p4 [g-opts] unlock [-c changelist#] [-f] file...
```

解説

p4 unlock コマンドは、p4 lock で生成したロックを解除します。

ファイルがデフォルト・チェンジリスト以外の作業中チェンジリストで作業状態になっている場合、その作業中チェンジリストを指定するには `-c` オプションが必要です。チェンジリストを指定しないと、p4 unlock はデフォルト・チェンジリストにあるファイルのロックを解除します。

管理者は、`-f` オプションを使用することによって、別のユーザが作業状態にしたファイルのロックを強制的に解除することができます。

ファイル名を入力しない場合は、指定したチェンジリスト中のすべてのファイルのロックが解除されます。

オプション

<code>-c changelist#</code>	作業中チェンジリスト <code>changelist#</code> にあるファイルのロックを解除します。
<code>-f</code>	スーパー・ユーザのみが使用できる強制オプションで、これによりスーパー・ユーザは他のユーザによって作業状態にされたファイルのロックを解除できます。
<code>g_opts</code>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	write

関連コマンド

他のユーザがサブミットできないようにファイルをロックする。	p4 lock
作業状態でロック中のファイルを表示する。(UNIX)	p4 opened grep "*locked*"

p4 user

概要

p4 user は、PERFORCE ユーザの仕様および基本設定の作成または編集を行います。

構文

```
p4 [g-opts] user [-f] [username]
p4 [g-opts] user -d [-f] username
p4 [g-opts] user -o [username]
p4 [g-opts] user -i [-f]
```

解説

デフォルトでは、システム・ユーザであれば誰でも、最初にディポまたはそのメタデータを更新する可能性がある PERFORCE コマンドを使用した時点で有効な PERFORCE ユーザになります。PERFORCE は、コマンドを実行したユーザのために、デフォルト設定のユーザ仕様を自動的に作成します。p4 user コマンドは、これらの設定の編集や新しいユーザ記録の作成に使用できます。(PERFORCE をインストールした後、新しいユーザの自動作成を防止するには、PERFORCE スーパー・ユーザとして p4 protect を使用します。)

p4 user を *username* なしで実行すると、現在のユーザの仕様を編集します。*username* を指定して実行した場合は、ユーザ仕様は表示されますが変更はできません。P4EDITOR 環境変数またはレジストリ変数で定義したエディタにフォームが表示されます。

PERFORCE スーパー・ユーザであれば、新規のユーザを作成したり、-f (強制) オプションを使用して他のユーザの仕様を編集したりすることができます。その場合の構文は p4 user -f *username* です。

PERFORCE コマンドを発行するユーザと、コマンド実行の名義者となるユーザとは、必ずしも同じではありません。特定のコマンドを実行するユーザは、以下のように決定されます。

- コマンドを実行するユーザが PERFORCE スーパー・ユーザであり、かつ構文 p4 user -f *username* を使用する場合は、ユーザ *username* が編集されます。
- コマンドラインで -u *username* オプションが使用される場合 (例えば、p4 -u joe submit)、コマンドは該当ユーザ名で実行されます (パスワードが必要になる場合もあります)。
- コマンドラインで -u *username* オプションが使用されない場合でも、環境変数またはレジストリ変数 P4CONFIG によって指定されるファイル中に P4USER の設定があれば、コマンドは該当ユーザ名で実行されます。
- -u *username* オプションも使用されず、ファイルも指定されていない場合でも、環境変数またはレジストリ変数 P4USER が設定されていれば、コマンドは該当ユーザ名で実行します。
- 上記のどれも当てはまらない場合、ユーザ名は OS レベルの環境変数 USER または USERNAME から取り込まれます。

フォーム・フィールド

フィールド名	タイプ	説明
User:	読み取り専用	p4 user を起動した PERFORCE ユーザ名。デフォルトは、システムで使用されているユーザ名です。
Email:	書き込み可能	ユーザの電子メールアドレス。デフォルトは <i>user@client</i> です。

フィールド名	タイプ	説明
Update:	読み取り専用	この仕様が最後に更新された日時。
Access:	読み取り専用	このユーザが最後に PERFORCE コマンドを実行した日時。
FullName:	書き込み可能	ユーザのフルネーム。
JobView:	書き込み可能	すべての新しいチェンジリストに自動的に表示されるジョブの説明。(詳しくは下記「使用上の留意点」をご覧ください。)
Password:	書き込み可能	ユーザのパスワード。(詳しくは下記「使用上の留意点」をご覧ください。)
Reviews:	書き込み可能リスト	ユーザが参照したいファイルのリスト。(詳しくは下記「使用上の留意点」をご覧ください。)

オプション

-d <i>username</i>	指定したユーザを削除します。ユーザ <i>username</i> または PERFORCE スーパー・ユーザのみが、このコマンドを実行できます。
-f	スーパー・ユーザのみが使用できる強制オプションで、これによりスーパー・ユーザは指定したユーザの修正や削除、または最終更新日の変更が行えます。
-i	標準入力からユーザ仕様を読み取ります。入力は p4 user フォームの形式に従っている必要があります。
-o	ユーザ仕様を標準出力に記述します。
<i>g_opts</i>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか?	ファイル引数にリビジョン範囲を使えるか?	最低限必要なアクセス・レベル
適用外	適用外	list

- スーパー・ユーザ以外のユーザは、-d オプションを使用して、p4 user コマンドを起動したユーザの仕様のみを削除することができます。PERFORCE スーパー・ユーザは、あらゆる PERFORCE ユーザを削除できます。
- 指定したユーザがファイルを作業状態にしている場合、そのユーザの削除は失敗します。作業状態にあるファイルをサブミットしてから、または元に戻してから、ユーザを削除してください。
- デフォルトでは、ユーザ記録はパスワードなしで作成されるので、どの PERFORCE ユーザも p4 user を設定するか、グローバル・オプション -u を使用することにより、他のユーザとして操作を行うことができます。自分の名前を他のユーザが使用するのを防ぐために、p4 passwd コマンドを使用してパスワードを設定します。

パスワードの作成、編集、および変更は、p4 user フォームで行うことも、p4 passwd コマンドを使用して行うこともできます。p4 user フォームでのパスワードの設定は、セキュリティ・レベル 0 または 1 のみでサポートされます。p4 passwd では任意のセキュリティ・レベルでパスワードを設定することができます。より高いセキュリティ・レベルでパスワードを設定するには、p4 passwd を使用する 必要があります。各種のセキュリティ・レベルの詳細については、『システム管理者ガイド』をご覧ください。

p4 user のフォームでパスワードを編集する場合、パスワード内にコメント文字 # を使用しないでください。PERFORCE は、同じ行にある、この文字に続くすべてをコメントとみなし、パスワードの一部としては保存しません。

- パスワードは、その長さに関係なく、p4 user フォームに 6 つのアスタリスクとして表示されます。

- チケット・ベースの認証（詳細は「p4 login」を参照）を使用している場合は、ユーザがパスワードを変更すると、そのユーザに対して発行されているすべてのチケットが自動的に無効になります。
- 各ユーザの [Email:]（電子メール:）フィールドの値は p4 users コマンドで一覧表示でき、あらゆる目的に使用できます。
- PERFORCEチェンジ・レビュー・デーモンで使用される p4 reviews コマンドは、[Reviews:]（レビュー:）フィールドの値を使用します。このデーモンは、ユーザが [Reviews:] フィールドで指定したファイルが変更されるたびに、該当ユーザに電子メールを送ります。このフィールドにリストするファイルは、ディポ・シンタックスで指定される必要があります。例えば、ユーザであるジョーの [Reviews:] フィールドの値が次のようになっていますとします。

```
//depot/main/...
//depot/.../README
```

この場合、チェンジ・レビュー・デーモンは、README ファイルがサブミットされるたびに、また、//depot/main 配下のファイルがサブミットされるたびに、ジョーに電子メールを送ります。

- PERFORCE チェンジ・レビュー・デーモンで使用される、ジョブ・レビューの特別な設定があります。[Reviews:] フィールドの値が次のようになっている場合、ジョブが変更されるたびに、該当ユーザに電子メールを送ります。

```
//depot/jobs
```

- [Jobview:] フィールドに有効なジョブ・ビューを設定すると、該当ユーザが作成するどのチェンジリストにも、そのジョブ・ビューに適合するジョブが表示されます。そのチェンジリストで修正されるジョブは、p4 submit でチェンジリストがサブミットされるときにチェンジリストに残っていなければならない、他のジョブはサブミットの前にフォームから削除されなければなりません。

例えば、サイトのジョブに [Owned-By:]（所有者:）というフィールドがあるとします。この場合、ユーザは p4 user の [Jobview:] フィールドを Owned-By=*yourname*&status=open と設定します。これで、該当ユーザが所有しているすべての作業中ジョブが、ユーザが作成するすべてのチェンジリストに表示されるようになります。ジョブ・ビューの使用法や構文について、詳しくは p4 jobs のセクションをご覧ください。

使用例

p4 user joe	PERFORCE ユーザ joe のユーザ仕様を参照します。
p4 user	現在の PERFORCE ユーザのユーザ仕様を編集します。
p4 user -d sammy	PERFORCE ユーザ sammy のユーザ仕様を削除します。
p4 -u joe -P hey submit	hey というパスワードを持つユーザ joe として、p4 submit を実行します。 このコマンドは、高いセキュリティ・レベルでは機能しません。
p4 user -f joe2	もし、これが PERFORCE スーパー・ユーザによる実行で、かつ joe2 が PERFORCE ユーザとして存在していなければ、joe2 という名前のユーザを新規に作成します。joe2 が存在していれば、スーパー・ユーザはこのユーザの設定を変更することができます。

関連コマンド

すべての PERFORCE ユーザのリストを参照する。	p4 users
ユーザのパスワードを変更する。	p4 passwd
特定ファイルのレビューを予約しているユーザのリストを参照する。	p4 reviews

p4 users

概要

p4 users は、現在のサーバが認識しているすべてのユーザを一覧表示します。

構文

```
p4 [g-opts] users [ -m max ] [user...]
```

解説

p4 users は、現在の PERFORCE サーバが認識しているすべてのユーザを一覧表示します。各ユーザについて表示される情報には、ユーザの PERFORCE ユーザ名、電子メール・アドレス、本名、およびユーザが最後にサーバにアクセスした日時が含まれています。

引数 *user* を指定した場合は、そのユーザのみが表示されます。引数 *user* は、ワイルドカード「*」を含むことができます。この場合、指定したパターンに適合するすべてのユーザについてレポートされます。(ワイルドカードを使用する場合は、ワイルドカードがカレント・ディレクトリ内のファイル名に合致するよう、OS がワイルドカードを展開しようとするので、必ず引数 *user* に引用符を付けてください。)

ユーザの出力を *max* までに限定するには、`-m max` オプションを使用します。

オプション

<code>-m max</code>	<i>max</i> までのユーザのみを一覧表示します。
<code>g_opts</code>	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定 子を使えるか？	ファイル引数にリビジョン範囲 を使えるか？	最低限必要なアクセス・レベル
適用外	適用外	list

関連コマンド

特定ユーザに関する情報を追加または編集する。	p4 user
現在のクライアント・ワークスペースに関する情報を編集する。	p4 client

p4 verify

概要

p4 verify は、サーバ・アーカイブの正当性を検証します。

構文

```
p4 [g-opts] verify [ -m maxRevs -q -u -v ] file[revRange]...
```

解説

p4 verify は、リビジョン指定情報およびリビジョン内容の MD5 ダイジェスト（指紋）をレポートします。

デフォルトでは、p4 verify は各リビジョンの MD5 ダイジェストを計算して表示します。あるリビジョンがアーカイブから失われており、したがって再現できない場合、そのリビジョンの出力行は MISSING! で終わります。また、ダイジェストに差異がある場合、破損しているファイルの出力行は BAD! で終わります。

オプション

-q	出力を抑制して実行します。以前に生成された MD5 ダイジェストに対してファイルの正当性を検証し、エラーがある場合にのみ出力を表示します。
-u	以前にダイジェストが保存されたことがない場合にのみ、各ファイルの MD5 ダイジェストを PERFORCE データベースに保存します。以後 p4 verify を使用すると、p4 verify で計算した値と保存済みの値とを比較します。
-v	各ファイルの MD5 ダイジェストを PERFORCE データベースに保存します。該当ファイルの保存済ダイジェストがある場合は、その既存ダイジェストに上書きします。
g_opts	「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用可	使用可	admin

- p4 verify がエラーを返したときは、PERFORCE の技術サポートにご連絡ください。
- p4 verify -q //... を実行して定期的にディポの正当性を検証することは、よい管理方法です。

詳しくは、『システム管理者ガイド』をご覧ください。

- Release 2005.1 では PERFORCE サーバはファイル長のメタデータをリビジョン毎に追跡します。新規にファイルがサブミットされると、ファイル長のメタデータが自動的にデータベースに追加されます。(2005.1 にアップグレード後も、少なくとも 1 度は p4 verify -u を実行し、ファイル長が保存されていない 2005.1 より前のファイルについてファイル長のメタデータを更新しなければなりません)
大規模なサイト（リビジョン数が一千万を超える規模）では、2005.1 にアップグレードした直後、管理者が一度にレポジトリ全体のファイル長メタデータを更新しようとするメモリの制限をオーバーしてしまうことがあります。このような場合は -m maxRevs オプションを使用し、一回のコマンドで更新するリビジョンの数を制限してください。例えば p4 verify -u -m 1000000 //... とすると、ファイル長メタデータを一度に再計算するファイルの数を 100 万個に制限することができ、管理者は p4 verify を何度かに分けて実行してファイル長のメタデータの再計算を行うことができます。

p4 where

概要

p4 where は、クライアント・ビューで決定された特定ファイルの位置を表示します。

構文

```
p4 [g-opts] where [file...]
```

解説

p4 where は、p4 client で設定したクライアント・ビューとクライアント・ルートを使用して、ディポの最上層、クライアント・ワークスペースの最上層、およびローカル OS のディレクトリ・ツリーの最上層との相対関係におけるファイルの位置を表示します。このコマンドはファイルが存在するかどうかは確認しません。単に、ファイルが 存在するとしたらどこにあるか をレポートするだけです。

パラメータとして指定された各ファイルについて、1 セットのマッピングが出力されます。マッピングの各セットは、3 つの部分からなる行で構成されています。最初の部分はディポ・シンタックスで表されたファイル名、2 番目の部分はクライアント・シンタックスで表されたファイル名、3 番目の部分はローカル OS 上でのファイルのパスです。

オプション

`g_opts` 「グローバル・オプション」のセクションをご覧ください。

使用上の留意点

ファイル引数にリビジョン指定子を使えるか？	ファイル引数にリビジョン範囲を使えるか？	最低限必要なアクセス・レベル
使用不可	使用不可	なし

- マッピングはクライアント・ビューから導出されます。ディポをクライアント・ワークスペース内の 1 つのディレクトリにマッピングしている単純なクライアント・ビューの場合、出力は 1 行です。

より複雑なクライアント・ビューは複数行の出力を生成しますが、これには除外マッピングが含まれていることがあります。例えば、次のようなクライアント・ビューがあるとします。

```
View: //a/... //client/a/...
      //a/b/... //client/b/...
```

この場合、p4 where //a/b/file.txt を実行すると、次のような出力が生成されます。

```
-//a/b/file.txt //client/a/b/file.txt
//home/user/root/a/b/file.txt
//a/b/file.txt //client/b/file.txt /home/user/root/b/file.txt
```

この出力は、クライアント・ビューの最初のマッピングによってファイルは /home/user/root/a/b/file.txt として現れるはずのところ、最初のマッピングは 2 番目のマッピングによってオーバーライドされている、と解釈できます。このオーバーライドを実行するために除外マッピングが使用され、2 番目のマッピングが適用されてファイルは /home/user/root/b/file.txt に送られます。

- 最も単純なケース（ディポ・シンタックス、クライアント・シンタックス、およびローカル・シンタックスでの各ファイル名を表示する、ファイル 1 つにつき 1 行の出力）の方が、はるかに一般的です。

使用例

<code>p4 where file.c</code>	<code>file.c</code> のディポ、クライアント・ワークスペース、およびローカル・ファイルシステムの位置（つまり、 <code>file.c</code> がディポ内に存在する場合に表示される場所）を示します。
<code>p4 where 100%25.txt</code>	ファイル <code>100%.txt</code> の位置を示すために “%” 文字の ASCII 拡張を使用します。 ASCII 拡張は、@ (%40)、# (%23)、* (%2a)、および % (%25) の 4 つの特殊文字に対してサポートされています。

関連コマンド

ディポから同期されたファイルのリビジョンを一覧表示する。	<code>p4 have</code>
------------------------------	----------------------

p4 workspace

概要

クライアント・ワークスペース仕様およびそのビューの生成または編集を行います。

構文

```
p4 [g-opts] workspace [-f -t template] [workspacename]
```

```
p4 [g-opts] workspace -o [-t template] [workspacename]
```

```
p4 [g-opts] workspace -d [-f] workspacename
```

```
p4 [g-opts] workspace -i [-f]
```

解説

p4 workspace は p4 client と同等に機能します。

p4 workspaces

概要

現在システムが認識しているすべてのクライアント・ワークスペースのリストを表示します。

構文

```
p4 [g-opts] workspaces [ -u user ] [ -e namefilter -m max ]
```

解説

p4 workspaces は p4 clients と同等に機能します。

環境変数とレジストリ変数

各オペレーティング・システム（OS）とシェルは、環境変数を設定するための固有の構文を持っています。次の表に、各 OS やシェルでの環境変数 P4CLIENT の設定方法を示します。

OS またはシェル	環境変数の例
UNIX: ksh, sh, bash	P4CLIENT=value ; export P4CLIENT
UNIX: csh	setenv P4CLIENT value
VMS	def/j P4CLIENT "value"
Mac MPW	set -e P4CLIENT value
Windows	p4 set P4CLIENT=value PERFORCE をサービスとして実行している Windows の管理者は、p4 set -S svcname var=value により、特定のサービスについて変数を設定したり、p4 set -s var=value により、ローカル・マシン上のすべてのユーザについて変数を設定したりできます。 (Windows における PERFORCE レジストリ変数の設定について、詳しくは本書の p4 set のセクションをご覧ください。)

PERFORCE の環境変数は、次の 4 つのカテゴリに大別できます。

- **重要**: このカテゴリに属する変数は、ほとんどの場合、クライアント上で設定する必要があり、デフォルト値で済ませるケースはまずありません。これらの変数を理解することは、ユーザにとっても管理者にとっても非常に重要です。
- **有用**: このカテゴリに属する変数は、ユーザに追加的機能を提供しますが、ほとんどの PERFORCE 操作には必要ありません。
- **高度**: このカテゴリに属する変数は、デフォルト値で使用すれば十分であり、値を変更する必要はほとんどありません。
- **サーバ**: このカテゴリに属する変数は、PERFORCE のシステム管理者が、PERFORCE サーバを動作させているマシン上で設定します。変数のいくつかは PERFORCE クライアントも使用しますが、そのような変数は 2 つのカテゴリに属することになります。

重要な変数	有用な変数	高度な変数	サーバの変数
P4CLIENT	P4CONFIG	P4PAGER	P4AUDIT
P4PORT	P4DIFF	PWD	P4JOURNAL
P4PASSWD	P4EDITOR	TMP, TEMP	P4LOG
P4USER	P4MERGE	P4LANGUAGE	P4PORT
	P4CHARSET	P4TICKETS	P4ROOT
		P4COMMANDCHARSET	P4DEBUG
		P4DIFFUNICODE	
		P4MERGEUNICODE	

P4AUDIT

解説

サーバ監査ログファイルの場所を示します。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	<code>p4d -A auditlog</code>	適用外

明示的に設定されない場合の値

OS	値
すべて	なし。ログファイルが指定されていない場合、監査は無効にされます。

注

P4AUDIT により、監査ログファイルの場所が指定されます。

監査が有効にされていると、サーバからクライアントへファイル・コンテンツが転送されるたびに、サーバが監査用ログファイルに一行追加します。アクティブなサーバ上では、監査用ログファイルは非常に急速に増大します。

監査用ログ内の行は次の形式で出力されます。

```
日付 時間 ユーザ@クライアント クライアント IP アドレス コマンド ファイル# リビジョン
```

例：

```
2006/05/09 09:52:45 karl@nail 192.168.0.12 diff //depot/src/x.c#1
2006/05/09 09:54:13 jim@stone 127.0.0.1 sync //depot/inc/file.h#1
```

コマンドが PERFORCE サーバを起動しているマシン上で実行されると、クライアント IP アドレスは 127.0.0.1 と表示されます。

詳細情報については、『システム管理者ガイド』をご覧ください。

P4CHARSET

解説

Unicode とのコード変換に使用される文字セットです。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	<code>p4 -C charset cmd</code>	設定できる

明示的に設定されない場合の値

OS	値
すべて	なし。PERFORCE サーバが unicode モードで起動されていても、P4 CHARSET が設定されていなければ、PERFORCE のクライアント・プログラムはエラー・メッセージを返します。

注

P4CHARSET の設定が有効となるのは、Unicode および utf16 のファイルのみです。Unicode 以外のタイプのファイルは、コード変換されません。

サーバの設定がデフォルト（非 Unicode モード）のときは、P4 CHARSET を設定してはいけません。P4 CHARSET が設定されているのに、サーバが国際語モードでない場合は、サーバが次のエラー・メッセージを返します。

```
Unicode clients require a unicode enabled server.
```

一方、サーバの設定が Unicode モードのときは、P4 CHARSET を設定しなければなりません。P4 CHARSET が設定されていないのに、サーバが Unicode モードの場合は、クライアント・プログラムが次のエラー・メッセージを返します。

```
Unicode server permits only unicode enabled clients.
```

Unicode モードに関する詳細情報、およびバイト・オーダー・マークを含むまたは含まない UTF-8、UTF-16、UTF-32 の各種文字セットに対する P4CHARSET の設定については、以下の *PERFORCE 国際語モードに関する注意事項 (i18nnotes.txt)* を参照してください。

<http://www.perforce.com/perforce/doc.073/user/i18nnotes.txt>

P4COMMANDCHARSET

解説

コマンド・ライン・クライアントからの UTF-16 および UTF-32 の文字セットをサポートするために使用されます。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	p4 -Q <i>commandcharset cmd</i>	設定できる

明示的に設定されない場合の値

OS	値
すべて	なし。

注

P4CHARSET を UTF-16 または UTF-32 に設定した場合、p4 コマンド・ライン・クライアントを使用するためには P4COMMANDCHARSET を UTF-16、UTF-32 以外の値に設定する必要があります。詳しくは、*PERFORCE 国際語モードに関する注意事項 (i18nnotes.txt)* を参照してください。

<http://www.perforce.com/perforce/doc.073/user/i18nnotes.txt>

P4CLIENT

解説

P4CLIENT は、現在のクライアント・ワークスペースの名前です。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	<code>p4 -c clientname cmd</code>	設定できる

明示的に設定されない場合の値

OS	値
Windows	環境変数 <code>COMPUTERNAME</code> の値
Windows 以外	ホスト・マシン名

使用例

```
cinnamon  
computer1  
WORKSTATION
```

P4CONFIG

解説

P4CONFIG は、パスなしのファイル名です。この変数が指示しているファイル（構成ファイル）は、P4CONFIG 以外の PERFORCE 環境変数およびレジストリ変数を定義するのに使用します。ファイルの検索は、現在の作業ディレクトリ（PWD により返信）およびそれより上位のディレクトリ内で行われます。ファイルが存在していれば、ファイル内の変数設定が使用されます。ファイル内の変数設定は各行に単独で置かれ、`variable=value` 形式を取る必要があります。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	なし	適用外

明示的に設定されない場合の値

OS	値
すべて	設定されない場合は使用されません。

使用例

例えば、P4CONFIG ファイルには次のような行が含まれます。

```
P4CLIENT=joes_client
P4USER=joe
P4PORT=ida:3548
```

注

P4CONFIG の設定により、プロジェクトを切り換えるときの PERFORCE 環境の切り換えがほとんど問題になりません。各クライアント・ワークスペース内に構成ファイルを配置し、そのファイルを指示するように P4CONFIG を設定しておけば、あるワークスペースから別のワークスペースのディレクトリに移動したとき、PERFORCE の設定は自動的に構成ファイル内の設定に変わります。

以下の変数を、P4CONFIG ファイルで設定できます。

- P4CHARSET
- P4CLIENT
- P4DIFF
- P4EDITOR
- P4HOST
- P4LANGUAGE
- P4MERGE
- P4PASSWD
- P4PORT
- P4TICKETS
- P4USER

P4DEBUG

解説

P4DEBUG は、PERFORCE サーバまたはプロキシのトレース・オプションを設定します。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用しない	使用する	なし	設定できない

明示的に設定されない場合の値

OS	値
すべて	設定されない場合は使用されません。

使用例

```
server=1
server=2
server=3
```

注

PERFORCE サーバのトレース・オプションが役に立つのは、多くの場合、PERFORCE の技術サポートや問題の診断、調査にあたっている管理者のためです。

PERFORCE サーバ（またはプロキシ）にトレース・オプションを設定するときは、p4d（または p4p）コマンドライン上に設定する方法がよく使われます。技術的な理由から、この方法は Windows 上で PERFORCE サーバまたはプロキシをサービスとして実行しているサイトには機能しません。このようなサイトの管理者は、p4 set を使用して P4DEBUG 内でトレース・オプションを設定することができ、このオプションが有効になった状態でサービスを実行できます。

サーバのトレース・オプションを設定するには、特定のサーバ・リリース・レベルを必要とする場合があります。

PERFORCE サーバ (p4d) でサーバ・デバッグ・レベルを設定しても、PERFORCE プロキシ (p4p) プロセスのデバッグ・レベルには何の影響もありません。この逆も同様です。

詳細情報については、『システム管理者ガイド』をご覧ください。

P4DIFF

解説

p4 resolve と p4 diff に使用する差分プログラムの名前とパスです。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	なし	設定できる

明示的に設定されない場合の値

OS	値
Windows	環境変数 DIFF が設定されている場合、DIFF の値となります。それ以外で環境変数 SHELL が <i>任意の値</i> に設定されている場合、diff コマンドが使用されます。それ以外では p4diff.exe となります。
Windows 以外	環境変数 DIFF が設定されている場合、DIFF の値となります。それ以外では、PERFORCE の内部差分プログラムが使用されます。

使用例

```
diff
diff -b
windiff.exe
```

注

diff -u のように、実行するプログラムに対するオプションを、P4DIFF の値に含めることができます。

コマンド p4 describe、p4 diff2、および p4 submit はすべて、PERFORCE のサーバ・プログラム p4d に組み込まれている差分プログラムを使用します。これは変更できません。

P4DIFFUNICODE

解説

コマンド・ライン・クライアントからの UTF-16 および UTF-32 の文字セットをサポートするために使用されます。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	なし	設定できる

明示的に設定されない場合の値

OS	値
すべて	なし。

注

この環境変数は、比較対象のファイルが unicode または utf16 タイプである場合に P4DIFF の代わりに使用されます。文字セットはコマンドへの最初の引数として渡されます。詳しくは *PERFORCE 国際語モードに関する注意事項 (i18nnotes.txt)* を参照してください。

<http://www.perforce.com/perforce/doc.073/user/relnotes.txt>

P4EDITOR

解説

P4EDITOR は、フォームを使用する PERFORCE コマンドにより起動されるエディタです。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	なし	設定できる

明示的に設定されない場合の値

OS	値
UNIX	EDITOR が任意の値に設定されている場合は EDITOR の値。それ以外の場合は vi
Windows	SHELL が任意の値に設定されている場合は vi それ以外では notepad
VMS	POSIX\$SHELL が設定されている場合は vi それ以外では edit
Macintosh	EDITOR_SIGNATURE が設定されている場合は、その 4 文字の作成者名が付いているプログラム それ以外では SimpleText

使用例

```
/usr/bin/vi
emacs
SimpleText
```

注

フォームを使用するコマンド、すなわちフォームを使用するためにこの変数にアクセスする PERFORCE の標準コマンドは、p4 branch、p4 change、p4 client、p4 job、p4 label、p4 submit、および p4 user です。

フォームを使用するスーパー・ユーザ・コマンドは、p4 depot、p4 group、p4 jobspec、p4 protect、p4 triggers および p4 typemap です。

P4HOST

解説

P4HOST は、別のホスト・コンピュータからアクセスしているかのように見せかけるときのホスト名です。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	<code>p4 -H hostname command</code>	設定できる

明示的に設定されない場合の値

OS	値
すべて	<code>p4 info</code> で表示されるクライアント・ホスト名。

使用例

```
workstation123.perforce.com
```

注

PERFORCE ユーザは、`p4 client` フォームのフィールド `[Host:]` (ホスト) を使用して、特定のホスト・マシンからのみ特定のクライアント・ワークスペースを使用できるよう規定することができます。このフィールドを設定すると、現在どのマシンを使用していても、P4HOST 変数を使用してサーバに通知し、指定したホスト・マシン上に見せかけることができます。これは非常に高度な要求なので、通常はこの変数を設定する理由はありません。

ホスト名は、`p4 info` をそのホストから実行したときの出力に表示されるとおりに入力する必要があります。

P4JOURNAL

解説

PERFORCE サーバ・データベースのジャーナル・データを保持するファイルです。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用しない	使用する	<code>p4d -J file</code>	適用外

明示的に設定されない場合の値

OS	値
すべて	<code>P4ROOT/journal</code>

使用例

```
journal
off
/disk2/perforce/journal
```

注

相対パスを使うときは、PERFORCE サーバのルートからの相対パスを指定してください。
P4JOURNALを`off`に設定するとジャーナル・データを保持しなくなりますが、お勧めしません。
詳細情報については、『システム管理者ガイド』をご覧ください。

P4LANGUAGE

解説

この環境変数は、システム管理者向けに用意されています。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	<code>p4 -L language cmd</code>	設定できる

明示的に設定されない場合の値

OS	値
すべて	設定されない場合は使用されません。

P4LOG

解説

P4LOG は、PERFORCE サーバのエラーが書き込まれるファイルの名前とパスです。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用しない	使用する	p4d -L <i>file</i> p4p -L <i>file</i>	適用外

明示的に設定されない場合の値

OS	値
すべて	標準エラー出力

使用例

```
log  
/disk2/perforce/log
```

注

相対パスを使用する場合は、PERFORCE サーバのからの相対パスを指定してください。

詳細情報については、『システム管理者ガイド』をご覧ください。

P4MERGE

解説

P4MERGE は、p4 resolve のマージ・オプションが使用する、サード・パーティ製のマージ・プログラムです。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同等に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	なし	設定できる

明示的に設定されない場合の値

OS	値
すべて	環境変数 MERGE (または p4 set で設定した Windows 上でのレジストリ変数) が設定されている場合はその値になります。それ以外の場合は値なし。

使用例

```
c:¥Perforce¥p4winmrg.exe
c:¥progra~1¥Perforce¥p4winmrg.exe
```

注

この変数内に設定されるプログラム名で表示されるプログラムは、p4 resolve のマージ・オプションによってのみ使用されます。p4 resolve によりこのプログラムにアクセスすると、このプログラムは 4 つの引数を獲得します。引数は (順に) *base*、*theirs*、*yours* で、4 つめの引数でマージ結果の *merge* ファイルを保持します。

使用するプログラムの引数が異なる順番になる場合、P4MERGE をシェル・スクリプトかバッチ・ファイルに設定して引数を並べ替え、正しい順番の引数を持つ適切なマージ・プログラムにアクセスしてください。

Windows で操作している場合は、使用するサード・パーティ製のマージ・プログラムがすでに PERFORCE の指定する順番で引数を受け入れている場合でも、バッチ・ファイルにアクセスしてください。これは、Windows 内の制約によるものです。例えば、Windows で MERGE.EXE というプログラムを使用したい場合、バッチ・ファイルは次のようになります。

```
SET base=%1
SET theirs=%2
SET yours=%3
SET merge=%4
C:¥FULL¥PATH¥TO¥MERGE.EXE %base %theirs %yours %merge
```

P4MERGEUNICODE

解説

コマンド・ライン・クライアントからの UTF-16 および UTF-32 の文字セットをサポートするために使用されます。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	なし	設定できる

明示的に設定されない場合の値

OS	値
すべて	なし。

注

この環境変数は、比較対象のファイルが unicode または utf16 タイプである場合に P4MERGE の代わりに使用されます。文字セットはコマンドへの最初の引数として渡されます。詳しくは *PERFORCE 国際語モードに関する注意事項 (i18nnotes.txt)* を参照してください。

<http://www.perforce.com/perforce/doc.073/user/relnotes.txt>

P4PAGER

解説

P4PAGERは、`p4 resolve`の差分オプションからの出力をページ表示するために使用されるプログラムです。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	なし	設定できない

明示的に設定されない場合の値

OS	値
すべて	変数 <code>PAGER</code> が設定されていれば <code>PAGER</code> の値になります。それ以外の場合には値なし。

使用例

```
/bin/more (UNIX)
```

注

この変数の値は、`p4 resolve`の差分ルーチンの出力を表示するためにのみ使用されます。この変数が設定されていない場合、出力はページ表示されません。

P4PASSWD

解説

P4PASSWD は、あらゆる PERFORCE クライアント・コマンドに、現在の PERFORCE ユーザのパスワードを与えます。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	<code>p4 -P passwd command</code>	設定できる

明示的に設定されない場合の値

OS	値
すべて	なし

注

PERFORCE のパスワードは、`p4 passwd` を介して設定するか、または `p4 user` によって表示されるフォーム内で設定します。P4PASSWD の設定は、ユーザが本人であるか検証するのに使用されます。パスワードが設定されていない場合、P4PASSWD が設定されていてもその値は使用されません。

環境変数 P4PASSWD は手動でプレーン・テキストのパスワードに設定できますが、より安全な方法は `p4 passwd` コマンドを使用することです。UNIX の場合、`p4 passwd` コマンドは、パスワードを安全確実に PERFORCE サーバに送るチャレンジ/レスポンス機構が起動します。Windows の場合、`p4 passwd` コマンドは、MD5 ハッシュによって暗号化したパスワードを、環境変数 P4PASSWD に設定します。

Windows プラットフォームでは、ユーザが P4WIN (Windows の PERFORCE クライアント) を介してパスワードを設定している場合、レジストリ変数 P4PASSWD の値が該当ユーザのために設定されます。P4WIN でパスワードを設定することは、MS-DOS コマンドラインから `p4 passwd` (または `p4 set PASSWD`) を使用し、レジストリ変数に、MD5 ハッシュによって暗号化されたパスワードが設定することに似ています。暗号化されていないパスワードそのものがレジストリに格納されることはありません。

チケット・ベースの認証を使用している場合に、P4PASSWD の設定に依存するスクリプトがあるときは、`p4 login -p` を使用して、パスワードと同じように PERFORCE コマンドに渡すことのできる (つまり、コマンドラインから渡すか、または P4PASSWD を有効なチケットの値に設定して渡すことのできる) チケットの値を示します。

P4PCACHE

解説

PERFORCE プロキシが使用するディレクトリで、キャッシュするファイルやディレクトリを保持します。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用しない	使用する	<code>p4p -r directory</code>	適用外

明示的に設定されない場合の値

OS	値
すべて	<p>p4p のディレクトリ。</p> <p>PERFORCE プロキシ・プロセスをサービスとして実行している Windows の管理者は、指定したサービスの P4PCACHE の値を設定する場合、<code>p4 set -S svcname P4PCACHE=directory</code> とする必要があります。</p>

注

このディレクトリを作成してから、PERFORCE プロキシ (p4p) を起動してください。

p4p を実行しているアカウントだけが、このディレクトリの読み取り／書き込みパーミッションを持つ必要があります。

PERFORCE プロキシのセットアップに関する詳細情報については、『システム管理者ガイド』をご覧ください。

P4PFSIZE

解説

PERFORCE プロキシにおいて、キャッシュされるファイルの最少サイズ（バイト単位）です。P4PFSIZE のバイト数よりもサイズが大きいファイルはすべてキャッシュされます。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用しない	使用する	<code>p4p -e size</code>	適用外

明示的に設定されない場合の値

OS	値
すべて	0; すなわちすべてのファイルをキャッシュする

注

PERFORCE プロキシのセットアップに関する詳細情報については、『システム管理者ガイド』をご覧ください。

P4POPTIONS

解説

Windows サービスに対して PERFORCE プロキシのオプションを設定します。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用しない	使用する	p4p %P4POPTIONS%	適用外

明示的に設定されない場合の値

OS	値
すべて	Null

注

例えば、通常、以下のコマンドを使ってプロキシを実行するとします。

```
p4p -p 1999 -t mainserver:1666
```

この場合、Windows サービスの proxysvc の起動オプションを P4POPTIONS 変数を使って以下のように設定することができます。

```
p4 set -S "Perforce Proxy" P4POPTIONS="-p 1999 -t mainserver:1666"
```

“PERFORCE プロキシ” サービスのもとで P4P を実行すると、プロキシはポート番号 1999 からの要求を待機し、mainserver:1666 において PERFORCE サーバと通信を行います。

ほとんどのインストールでは P4POPTIONS を使用する必要はありません。大部分の p4p オプションには、それに関連付けられている環境変数があるからです。上記の例の場合、P4PORT と P4TARGET を使用することができます。該当する環境変数のないオプションを付けて p4p を呼び出す必要がある場合、または Windows サービスのコンテキスト内で p4p を呼び出す必要がある場合に P4POPTIONS を使用してください。

PERFORCE プロキシのセットアップに関する詳細情報については、『システム管理者ガイド』をご覧ください。

P4PORT

解説

PERFORCE サーバまたは PERFORCE プロキシにおいては、通信の接続待ちに使用するポートの番号です。

PERFORCE クライアントにおいては、通信相手である PERFORCE サーバまたはプロキシのホスト名とポート番号です。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用する	<code>p4 -p host:port cmd</code>	設定できる

明示的に設定されない場合の値

プログラム	値
PERFORCE サーバ	1666
PERFORCE プロキシ	1666
PERFORCE クライアント	perforce:1666

使用例

PERFORCE クライアントでの使用例	PERFORCE サーバでの使用例
1818	1818
squid:1234	1234
perforce.squid.com:1234	1234
192.168.0.123:1818	1818

注

PERFORCE クライアント上での P4PORT の形式は `host:port` です。または、PERFORCE サーバとクライアントの両方が同じホスト上で動作している場合は、単に `port` です。

P4PORT に IP アドレスとポート番号の両方を指定すると、PERFORCE サーバは P4PORT に指定されている IP アドレス以外の IP アドレスからの要求を無視します。

デフォルト値である `perforce` を PERFORCE サーバで使用するには、UNIX の `/etc/hosts` または Windows の `%SystemRoot%\system32\drivers\etc\hosts` で、サーバが動作しているホストのエイリアスとして `perforce` を定義するか、DNS を使用してください。

ポート番号は、1024 から 32767 の範囲になければなりません。

P4ROOT

解説

PERFORCE サーバが、自身のファイルとサブディレクトリを格納するディレクトリです。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用しない	使用する	<code>p4d -r <i>directory</i></code>	適用外

明示的に設定されない場合の値

OS	値
すべて	p4d のディレクトリ。 PERFORCE のバックエンド・プロセスをサービスとして実行している Windows の管理者は、指定したサービスの P4ROOT の値を設定する場合、 <code>p4 set -S svcname P4ROOT=<i>directory</i></code> とする必要があります。

注

このディレクトリを作成してから、PERFORCE サーバ (p4d) を起動してください。

p4d を実行しているアカウントだけが、このディレクトリの読み取り／書き込みパーミッションを持つ必要があります。

PERFORCE サーバのセットアップに関する詳細情報については、『システム管理者ガイド』をご覧ください。

P4TARGET

解説

PERFORCE プロキシがターゲットとする PERFORCE サーバ (P4P はこのサーバに対してプロキシとして動作します) の、ホスト名とポート番号を設定します。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用しない	使用する	<code>p4p -t host:port</code>	適用外

明示的に設定されない場合の値

プログラム	値
PERFORCE プロキシ	<code>perforce:1666</code>

使用例

PERFORCE クライアントでの使用例	PERFORCE サーバでの使用例
1818	1818
<code>squid:1234</code>	<code>squid:1234</code>
<code>perforce.squid.com:1234</code>	<code>perforce.squid.com:1234</code>
<code>192.168.0.123:1818</code>	<code>192.168.0.123:1818</code>

注

PERFORCE プロキシ上での P4TARGET の形式は `host:port` です。または、PERFORCE サーバと同じホスト上で動作している場合 (ありそうもない構成ですが) は、単に `port` です。

ポート番号は、1024 から 32767 の範囲になければなりません。

詳細情報については、『システム管理者ガイド』をご覧ください。

P4TICKETS

解説

p4 login で使用されるチケット・ファイルの場所を示します。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	N/A	できる

明示的に設定されない場合の値

プログラム	値
Windows	%USERPROFILE%\p4tickets.txt
その他すべて	\$HOME/.p4tickets

使用例

```
/staff/username/p4tickets.txt
```

注

P4TICKETS 環境変数は、p4tickets.txt または .p4tickets が存在すると思われるディレクトリではなく、チケット・ファイルそのものを表す必要があります。P4TICKETS にディレクトリ名を指定すると、ログインできなくなります。

<http://www.perforce.com/perforce/doc.052/user/i18nnotes.txt>

P4USER

解説

現在の PERFORCE ユーザ名です。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	<code>p4 -u username command</code>	設定できる

明示的に設定されない場合の値

OS	値
Windows	環境変数 USERNAME の値。
Windows 以外	環境変数 USER の値。

使用例

```
edk
lisag
```

注

デフォルトでは、PERFORCE のユーザ名は OS のユーザ名と同じです。

ある PERFORCE ユーザがパスワードを設定していない場合、他の PERFORCE ユーザの誰もが、PERFORCE クライアント・コマンドを `-u` オプション付きで実行することにより、パスワード未設定のユーザの権限でコマンドを実行することができます。これを防ぐため、ユーザは `p4 user` コマンドまたは `p4 passwd` コマンドでパスワードを設定する必要があります。

あるユーザが PERFORCE パスワードを設定していても、(そのパスワードを知っている) 他のユーザの誰もが、`p4 -u username -P password command` を使用することにより、そのユーザとしてコマンドを実行できます。

PERFORCE スーパー・ユーザは、あるユーザのパスワードを知らなくても、そのユーザとしてコマンドを実行することができます。詳しくは、『システム管理者ガイド』をご覧ください。

PWD

解説

PERFORCE のクライアント・コマンドを実行する際、引数として与えられたファイル名を相対パスで解決するためのディレクトリです。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用しない	<code>p4 -d directory command</code>	設定できない

明示的に設定されない場合の値

OS	値
UNIX	シェルで設定された PWD の値。シェルで設定されていない場合は、 <code>getcwd()</code> が使用されます。
UNIX 以外	現在の作業ディレクトリ。

注

変数 PWD は、シェル間で適切に継承されないことがあります。例えば、`cs`h に加えて `ksh` または `sh` を実行している場合、PWD は `cs`h 環境から継承されますが、適切に更新されないため、以降の PERFORCE コマンドで混乱を招く可能性があります。

このような問題に直面したときは、`.profile` ファイルまたは `.kshrc` ファイルで PWD を設定解除したかどうかを確認してください。(sh または ksh をログイン・シェルとして実行している場合、スタートアップ・ファイルでどんな設定解除をしても、PWD はシェルによって適切に管理されます。混乱は、変数がサブシェルにエクスポートされるときにのみ起きます。)

TMP, TEMP

解説

PERFORCE クライアントと PERFORCE サーバが一時ファイルを書き込むディレクトリです。

使用上の留意点

クライアントが使用するか？	サーバが使用するか？	同様に機能するコマンドライン	P4CONFIG ファイルで設定できるか？
使用する	使用する	なし	設定できない

明示的に設定されない場合の値

OS	値
UNIX	/tmp
UNIX 以外	PERFORCE クライアントでは 現在の作業ディレクトリ。 PERFORCE サーバでは P4ROOT

注

TEMP が設定されていれば、TEMP が使用されます。そうでない場合、TMP が設定されていれば、TMP が使用されます。TEMP と TMP のどちらも設定されていなければ、一時ファイルは上の表に記述されているディレクトリに書き込まれます。

追加情報

このセクションは、複数のコマンドで使用する PERFORCE の機能を説明しています。以下の項目に関する情報が記載されています。

- あらゆる PERFORCE コマンドで利用できる オプション
- デイポ・シンタックス、クライアント・シンタックス、およびローカル・シンタックスでの PERFORCE ファイル指定の使用方法
- PERFORCE の ファイルタイプ
- クライアント・ワークスペース、ブランチ、およびラベルを記述する ビューの作成方法と使用方法

上記や他の項目の基本的な考え方および詳細な情報については、弊社ウェブサイト <http://www.perforce.com> にある『PERFORCE 概要』をご覧ください。

グローバル・オプション

概要

PERFORCE コマンドのためのグローバル・オプションです。これらのオプションは、PERFORCE コマンドに先行させてコマンドラインに入力します。

構文

```
p4 [-cclient -ddir -Hhost -pport -Ppass -user -xfile -Ccharset] [-G]
[-s] cmd [args ...]
```

```
p4 -V
```

```
p4 -h
```

オプション

-c <i>client</i>	P4CLIENT の設定を、指定したクライアント名でオーバーライドします。
-d <i>dir</i>	PWD の設定 (例えば、現在の作業ディレクトリ) をオーバーライドし、指定したディレクトリに置換します。
-G	すべての出力 (および -i オプション付きコマンドのためのバッチ入力) を Python ディレクトリ・オブジェクトとしてフォーマットします。 これは、スクリプト作成時に最も多く使用されます。
-H <i>host</i>	P4HOST の設定をオーバーライドし、指定したホスト名に置換します。
-p <i>port</i>	P4PORT の設定を、指定したポート番号でオーバーライドします。
-P <i>pass</i>	P4PASSWD の設定を、指定したパスワードでオーバーライドします。
-s	PERFORCE コマンドが生成する出力の各行の先頭に、解説フィールド (例えば、text:、info:、error:、exit:) を付加します。 これは、スクリプト作成時に最も多く使用されます。
-u <i>user</i>	P4USER、USER、および USERNAME の設定を、指定したユーザ名でオーバーライドします。
-x <i>file</i>	指定したファイルから 1 行につき 1 つの引数を読み取るよう、PERFORCE に指示します。ファイルが 1 つのハイフン (-) である場合は、標準入力から読み取ります。
-C <i>charset</i>	P4CHARSET の設定を、指定したキャラクタ・セットでオーバーライドします。
-L <i>language</i>	この機能は、システム管理者向けに用意されています。
-V	p4 クライアント・プログラムのバージョンを表示し、コマンドを終了します。
-h	基本的な使用方法を表示し、コマンドを終了します。

使用上の留意点

- グローバル・オプションは、コマンドラインで PERFORCE コマンドに先行させて指定しなければならないことに注意してください。PERFORCE コマンドの後に続けて指定したオプションは、グローバル・オプションとはみなされず、実行しようとしているコマンドのためのオプションとみなされます。したがって、同じコマンドラインで同じオプションを 2 回指定することが可能で、それぞれ別々のものとみなされます。

例えば、コマンド `p4 -c anotherclient edit -c 140 file.c` は、クライアント・ワークスペース `anotherclient` 内で、作業中チェンジリスト 140 に含まれるファイル `file.c` を編集のための作業状態にします。

- x オプションは、何度も繰り返す必要のある単純なタスクを自動化するのに便利です。例えば、一度に数個のファイルを追加したい場合、これらのファイルの名前を記述したテキスト

ト・ファイル `textfile` を作成し、`p4 -x textfile add` を実行すると、これらのファイルを一度に追加できます。

`-x` オプションは、入力そのものを生成してしまうと言っていいほど、使い方によってはきわめて強力なものになります。例えば、UNIX の開発者が、`file.h` をインクルードしているすべてのファイルを編集しようとしていると仮定します。この場合は、例えば

```
grep -l file.h *.c | cut -f1 -d: | p4 -x - edit
```

を使用することができます。

この例の場合、`grep` コマンドは、すべての `*.c` ファイルにおける `file.h` の出現をリストします。`-l` オプションは、各ファイルを 1 回だけリストするよう `grep` に指示します。そして `cut` コマンドによって、`grep` の出力からファイル名を分離し、その出力を `p4 -x` コマンドに渡します。

- `-s` オプションは、自動化されたスクリプトにおいて使用すると便利です。

例えば、`p4 -s` コマンドを実行し、`info:` で始まるあらゆる出力行を廃棄し、`error:` で始まる出力行がある場合はユーザに警告する、といった内製のビルド・プロセスの一部として、スクリプトを記述することができます。

- Python の開発者がスクリプトを作成する際、`-G` オプションはたいへん有用です。例えば、ID が分かっているジョブのフィールドについてディレクトリを得る場合、次のように実行します。

```
job_dict = marshal.load(os.popen('p4 -G job -o ' + job_id, 'rb'))
```

クライアント・プログラムがどのキーを使用するか、すぐには分からないケースがあります。`p4 -G` の出力をパイプで下記のスクリプトに渡した場合、各レコードはキー / 値の組で表示されます。

```
#!/usr/local/bin/python
import marshal, sys
try:
    num=0
    while 1:
        num=num+1
        print '\n--%d--' % num
        dict = marshal.load(sys.stdin, 'rb')
        for key in dict.keys(): print "%s: %s" % (key,dict[key])
except EOFError: pass
```

Windows 上で Python を使用する開発者は、潜在的な CR/LF 変換に注意する必要があります。上の例において、バイナリ (“rb”) モードでデータを読む場合、`marshal.load()` を呼び出す必要があります。

- グローバル・オプションは、使い方によっては無意味なものとなります。

例えば、`p4 -c anotherclient help` の出力は、`p4 help` の出力と全く同じです。

使用例

```
p4 -p new_server:1234 sync
```

環境変数またはレジストリ変数 `P4PORT` の設定に関わりなく、サーバ `new_server` とポート `1234` を使用して同期を実行します。

<pre>p4 -c new_client submit -c 100</pre>	<p>最初の <code>-c</code> は、クライアント名を指定するグローバル・オプションです。2 番目の <code>-c</code> は、チェンジリスト番号を指定します。</p>
<pre>p4 -s -x filelist.txt edit</pre>	<p><code>filelist.txt</code> がファイルのリストを含む場合、このコマンドはリストにある各ファイルを編集目的で作業状態にし、スクリプトによる解析に適した出力を生成します。</p> <p>この場合、<code>error:</code> で始まる行があるか出力を調べることにより、自動化された <code>p4 edit</code> コマンドの結果としてのエラー（例えば、<code>filelist.txt</code> のリストにあるファイルが見つからない）を簡単に検出できます。</p>

ファイル指定

概要

クライアント・シンタックス、ディポ・シンタックス、またはローカル・シンタックス中のあらゆる PERFORCE コマンドにおいて、どんなファイルでも指定できます。クライアント・ワークスペース名とディポ名は、同じネームスペースを共有しているため、PERFORCE サーバがクライアント名とディポ名を混同することは決してありません。

シンタックス・フォーム

ローカル・シンタックスは、ローカル・シェルか OS で指定されたファイル名を参照します。ローカル・シンタックスで参照されるファイル名は、ファイル名の絶対パス、または現在の作業ディレクトリを起点とする相対パスで指定されます。(相対パスの構成要素は、ファイル指定子の最初の部分にしか現れません。)

PERFORCE には、OS が異なっても変わることのない独自のファイル指定方法があります。ファイルがクライアント・ルートを起点とする相対パスで指定された場合、ファイルは クライアント・シンタックス で示されているということになります。また、ディポの最上層を起点とする相対パスで指定された場合は、ディポ・シンタックス で示されているということになります。このどちらかで指定されたファイルは、PERFORCE シンタックスで指定されたと言えます。

PERFORCE のファイル指定子は常に 2 つのスラッシュ (//) で始まり、その後にクライアントかディポの名前、さらにその後にクライアントかディポのルート・ディレクトリを起点とするファイルのフル・パス名が続きます。

クライアント・シンタックスおよびディポ・シンタックスのパス構成要素は、ローカルの OS やシェルが使用する構成要素区切り文字に関わりなく、常にスラッシュ (/) で区切られます。

各シンタックスの例を以下に示します。

構文	例
ローカル・シンタックス	/staff/user/usercws/file.c
ディポ・シンタックス	//depot/source/module/file.c
クライアント・シンタックス	//usercws/file.c

ワイルドカード

PERFORCE システムでは、3 つのワイルドカードが使用できます。

ワイルドカード	意味
*	1 つのディレクトリ内で、スラッシュを除くすべての文字に合致します。
...	現在の作業ディレクトリおよびすべてのサブディレクトリ内で、すべてのファイルに合致します。(つまり、複数のサブディレクトリにわたって、スラッシュも含めてすべてに合致します。)
%%1 - %%9	ビューの中で使用されるとき、ファイル名中の部分文字列を再配列するための位置指定子です。

使用例を以下に示します。

式	一致するファイル
J*	現在のディレクトリにある、J で始まるファイル
*/help	現在のサブディレクトリにある、help という名前のすべてのファイル
./...	現在のディレクトリおよびそのサブディレクトリにあるすべてのファイル

式	一致するファイル
<code>./....c</code>	現在のディレクトリおよびそのサブディレクトリにある、最後に <code>.c</code> が付くすべてのファイル
<code>/usr/bruno/...</code>	<code>/usr/bruno</code> の下にあるすべてのファイル
<code>//bruno_ws/...</code>	<code>bruno_ws</code> という名前のワークスペースまたはディポの下にある、すべてのファイル
<code>//depot/...</code>	当該ディポにあるすべてのファイル
<code>//...</code>	すべてのディポにあるすべてのファイル

リビジョン指定子の使用

ファイル指定子は、`#` か `@` を付加することにより修正できます。これらのファイル指定子は、ディポに保存されたファイルの特定リビジョンを参照するために使用します。

修飾子	意味
<code>file#n</code>	リビジョン指定子: <code>file</code> の <code>n</code> 番目のリビジョン。
<code>file#none</code>	存在しないリビジョン: <code>file</code> のリビジョンがディポに存在する場合、そのリビジョンは無視されます。
<code>file#0</code>	これは、ディポ内にあるファイルをそのままディポ内に残す一方、クライアント・ワークスペースからは削除したいときに便利で、例えば <code>p4 sync file#none</code> のように使います。 ファイル指定子 <code>#0</code> は、 <code>#none</code> と同様に使用することができます。存在しないリビジョンとは、1 番目のリビジョンがディポにサブミットされる前に「存在していた」リビジョンと考えることができます。
<code>file#head</code>	<code>file</code> の最新リビジョン。これは、明確に言及されている場合を除き、リビジョン指定子なしでファイルを参照することと同じです。
<code>file#have</code>	現在のクライアントにあるリビジョン: <code>p4 sync</code> で最後にクライアント・ワークスペースに同期させたファイルのリビジョン。
<code>file@n</code>	チェンジ番号: チェンジリスト <code>n</code> がサブミットされた直後の <code>file</code> のリビジョン。
<code>file@labelname</code>	ラベル名: ラベル <code>labelname</code> 内の <code>file</code> のリビジョン。
<code>file@clientname</code>	クライアント名: クライアント・ワークスペース <code>clientname</code> に最後に取り込まれた <code>file</code> のリビジョン。
<code>file@datespec</code>	日付と時刻: 指定した日付と時刻における <code>file</code> のリビジョン。時刻を指定しない場合、指定した日の午前 00:00:00 における最新リビジョンが返されます。日付と時刻は、 <code>yyyy/mm/dd:hh:mm:ss</code> または <code>yyyy/mm/dd hh:mm:ss</code> の形式で指定します (日付と時刻の間にスペースかコロンをどちらかを入れます)。 日付指定子 <code>@now</code> は、現在の日時を指定するのと同様に使用することができます。

リビジョン指定子は、一度に多くのファイルを操作するために使用することができます。 `p4 sync //myclient/...#4` は、作業状態にないすべてのファイルの 4 番目のリビジョンを、クライアント・ワークスペースにコピーします。

ファイルを日付と時刻で指定する場合 (つまり `file@datespec` という形式の指定子を使用する場合)、ローカル・シェルによって日付指定子が単一トークンとして解釈される必要があります。日付だけでなく時刻を指定するために日付指定子を使用する場合は、日付指定子の前後に引用符を使用しなければならないことがあります。

PERFORCE のファイル指定文字は、ローカル・シェルによって途中で取得され解釈されてしまうことがあります。そのような文字は、使用する前にエスケープしておく必要があります。例

例えば、# はたいていの UNIX シェルでコメント文字として使用され、/ は (PERFORCE 以外の) DOS コマンドによってオプション指定子として解釈されます。スペースが入っているファイル名は、コマンドラインにおいて引用符で囲まなければならないことがあります。

これらの問題、およびプラットフォームに特有な他の問題については、プラットフォーム別のリリース・ノートをご覧ください。

リビジョン範囲の使用

PERFORCE コマンドのいくつかは、リビジョン範囲を使用してファイル引数を変更することができます。リビジョン範囲は2つの別個のリビジョン指定子で、カンマによって区切られます。例えば、`p4 changes file#3,5` は、ファイル `file` の3番目、4番目、および5番目のリビジョンをサブミットしたチェンジリストをリストします。

リビジョン範囲には、使用するコマンドにより、次のように異なる2つの意味があります。

- 指定した範囲内にあるすべてのリビジョンに対して、コマンドを実行します。例えば、`p4 jobs //...#20,52` は、どのファイルかに関わりなく、ファイルの20番目から52番目までのリビジョンをサブミットしたチェンジリストによって修正された、すべてのジョブをリストします。

リビジョン範囲は、`p4 changes`、`p4 fixes`、`p4 integrate`、`p4 jobs`、および `p4 verify` コマンドとともに使用される場合、上記のように解釈されます。

- 指定した範囲内で最大番号のリビジョンのみに対して、コマンドを実行します。例えば、コマンド `p4 print file@30,50` は、チェンジリスト 30～50 でサブミットされたファイル `file` の、最大番号のリビジョンを出力します。これは、`p4 print file@50` とは次のように異なります。例えば、ファイル `file` のリビジョン 1 がチェンジリスト 20 でサブミットされ、ファイル `file` のリビジョン 2 がチェンジリスト 60 でサブミットされた場合、`p4 print file@30,50` は何も出力しません(チェンジリスト 30～50 でサブミットされたファイル `file` のリビジョンは存在しないので)。これに対し、`p4 print file@50` は、ファイル `file` のリビジョン 1 を出力します(チェンジリスト 50 の時点までにサブミットされたファイル `file` のリビジョンはリビジョン 1 なので)。

コマンド `p4 files`、`p4 print`、および `p4 sync` はすべて、このようにリビジョン範囲を使用します。

リビジョン範囲は、非常に強力なものとなりえます。例えば、`p4 changes file#3,@labelname` は、ファイル `file` の3番目のリビジョンから、ラベル `labelname` に保存されているリビジョンまでをサブミットしたすべてのチェンジリストをリストします。

ファイル名およびエンティティに関する文字の制限

国際語化のサポートにより、PERFORCE では、表示 / 印刷ができない文字 (例えば非 ASCII 文字) であっても、ファイル名、ラベル名、クライアント・ワークスペース名、およびその他の識別子に使用することができます。

パス名構成要素区切り記号 (/) は、ファイル名、ディポ名、クライアント・ワークスペース名に使用することはできませんが、ラベル名、ジョブ名、ユーザ名の中に含めることができます。再帰的サブディレクトリ・ワイルドカード (...) は、ファイル名、ラベル名、およびその他の識別子に使用することはできません。

文字	理由
...	PERFORCE ワイルドカード：すべてに合致し、カレント・ディレクトリのレベルで機能し、カレント・レベル以下のすべてのディレクトリ・レベルにあるファイルを含みます。
/	パス名の構成要素を区切る PERFORCE の区切り記号。

PERFORCE リビジョン指定子ワイルドカード (@ と #)、ファイル一致ワイルドカード (*)、または位置置換ワイルドカード (%) がファイル名または任意のディレクトリ構成要素に含まれた

ファイルを参照するには、文字の 16 進値の ASCII 表現を使用します。ASCII 拡張は、以下の 4 つの文字にのみ適用されます。

文字	ASCII 拡張
@	%40
#	%23
*	%2A
%	%25

status@june.txt などのファイルを追加するには、次のようなコマンドを使用して、特殊文字の文字解釈を強制します。

```
p4 add -f //depot/path/status@june.txt
```

チェンジリストをサブミットすると、文字が自動的に拡張され、次のように変更サブミット・フォームに表示されます。

```
//depot/path/status%40june.txt
```

ファイルが追加されたチェンジリストをサブミットした後で、そのチェンジリストをワークスペースに同期させる場合や、ワークスペース内で編集する場合は、ASCII 拡張を使用する必要があります。

```
p4 sync //depot/path/status%40june.txt
p4 edit //depot/path/status%40june.txt
```

特殊文字の大半は、クロスプラットフォームの環境では、原則としてファイル名には使用できません。UNIX ではパスの構成要素を / で区切りますが、多くの DOS コマンドは / をコマンドラインのオプションとして解釈します。たいていの UNIX シェルは # をコメントの始まりとして解釈します。DOS および UNIX のシェルは両方とも * を拡張して複数のファイルに合致させ、DOS コマンドラインは % を変数の参照に使用します。

同様に、非 ASCII 文字をファイル名と PERFORCE 識別子に使用することは可能ですが、それらの文字をコマンドラインから入力するときに、プラットフォーム特有の解決策が必要になることがあります。GUI ベースのファイル・マネージャのユーザは、このようなファイルを、ドラッグ・アンド・ドロップ操作で取り扱うことができます。

ビュー

概要

ビューには クライアント・ビュー、ブランチ・ビュー、および ラベル・ビュー の 3 タイプがあります。クライアント・ビューは、ディポ内のファイルをクライアント・ワークスペース内のファイルにマッピングし、ブランチ・ビューはディポ内のファイルをディポの他の部分にマッピングし、ラベル・ビューはディポにおけるファイルのグループを単一のラベルに関連付けます。

各タイプのビューを構成する行は、ディポからのファイルを適切なネームスペースにマッピングしています。クライアント・ビューおよびブランチ・ビューのマッピングは、2つのファイルの指定です。マッピングの左側は常にディポのネームスペースを参照し、マッピングの右側はクライアント・ワークスペースまたはディポのネームスペースを参照します。ラベル・ビューでは、マッピングの左側（ディポのネームスペース）のみ入力する必要があります。ファイルは自動的に特定のラベルに関連付けられます。

すべてのビューは、ディポ内のファイルと、クライアント・ワークスペース、ブランチ、またはラベル内のファイルとの間で一対一のマッピングを作成しています。マッピングの1つ以上の行が同じファイルを参照している場合、前のマッピングがオーバーライドされます。ハイフン(-)で始まるマッピングは、そのマッピングに適合するファイルをすべて除外します。クライアント・ビューでは、プラス記号(+)で始まるマッピングが以前のマッピングをオーバーレイします（オーバーレイ・マッピングはブランチ・ビューやラベル・ビューでは適用されません）。

マッピング内の ファイル指定 は通常の PERFORCE シンタックスで入力できます。// で始まりディポ名またはワークスペース名が続き、さらにディポまたはワークスペース内での実際のファイル名が続きます。（リビジョン指定子をビューの中で使用することはできません。）

使用上の留意点

ビューは、p4 client、p4 branch、または p4 label コマンドで、それぞれクライアント・ワークスペース、ブランチ・ビュー、またはラベル・ビューを作成するプロセスの一部として設定します。

クライアント・ビューまたはブランチ・ビュー内でのマッピングの順序は重要です。例えば、以下の2つのマッピングで定義されるビューがあります。

```
//depot/... //cws/...  
//depot/dir1/... //cws/dir2/...
```

このビュー内では、ディポはクライアント・ワークスペースにマッピングされ、ファイル //depot/dir1/file.c は //cws/dir2/file.c にマッピングされます。しかし、ビュー内の行の順序が逆になったとします。

```
//depot/dir1/... //cws/dir2/...  
//depot/... //cws/...
```

すると最初のマッピング（ファイルを //cws/dir2 にマッピング）が 2 番目のマッピング（ディポ全体をクライアント・ワークスペースにマッピング）にオーバーライドされるので、ファイル //depot/dir1/file.c は //cws/dir1/file.c にマッピングされます。ビュー内で後にあるビューが常に、前にあるマッピングをオーバーライドします。

パスおよびファイルの名前に含まれる空白

ワークスペース・ビュー、ブランチ・ビューまたはラベル・ビューにあるパスやファイルの名前に空白が含まれる場合、必ずパスに引用符を付けてください。

```
//depot/dir1/... "//cws/dir one/..."
```

パスおよびファイルの名前に含まれる特殊文字

@、#、*、または%の文字が含まれたファイル名やディレクトリ名をマッピングするには(つまり、それらの文字を PERFORCE ワイルドカードとしてではなくパスやファイル名の構成要素として解釈するには)、次のようにそれらの文字を等価の ASCII 表現に拡張します。

文字	ASCII 拡張
@	%40
#	%23
*	%2A
%	%25

クライアント・ビュー

クライアント・ビューは、ディポ内のファイルをクライアント・ワークスペース内のファイルにマッピングします(またはその逆)。クライアント・ワークスペースとは、ユーザが作業を行う領域で、ファイルはクライアント・ワークスペースに同期され、編集のための作業状態になり、編集され、ディポにチェックインされます。

ファイルが同期される時、ファイルはディポからクライアント・ワークスペース内のファイルがマッピングされている場所にコピーされます。同様に、ファイルがディポにサブMITされるときは、マッピングが逆になり、ファイルはクライアント・ワークスペースからディポ内のファイルの適切な場所にコピーされます。

以下の表は、クライアント・ビューのいくつかの例をリストにしています。

クライアント・ビュー	マッピングの例
ディポ全体にマッピングされたクライアント・ワークスペース全体	//depot/... //cws/...
ディポの一部にマッピングされたクライアント・ワークスペース全体	//depot/dir1/... //cws/...
ディポ内のいくつかのファイルが、クライアント・ワークスペースの別の部分にマッピングされています。	//depot/... //cws/... //depot/rel1/... //cws/release1/...
ディポ内のいくつかのファイルが、クライアント・ワークスペースから除外されています。	//depot/dir1/... //cws/... -//depot/dir1/exclude/... //cws/dir1/exclude/...
クライアント・ワークスペース内のファイルが、自身のディポ名とは異なる名前にマッピングされます。	//depot/dir1/old.* //cws/renamed/new.*
ディポ内でのファイル名の一部が、クライアント・ワークスペース内で並べ替えられます。	//depot/dir1/%1.%2 //cws/dir1/%2.%1
ファイルは両方向で見たときに同じ場所にマッピングされません。2行目が優先され、1行目は無視されます。	//depot/dir1/...//cws/build/... //depot/dir2/...//cws/build/...
オーバーレイ・マッピングを使用して、複数のディポ・ディレクトリをワークスペース内の同じ場所にマッピングします。	//depot/dir1/...//cws/build/... +//depot/dir2/...//cws/build/...

クライアント・ビューを作成するには、p4 client を使用して、ディポ内のファイルをクライアント・ワークスペース内のファイルにどのようにマッピングするかを指定できる画面を表示します。

ブランチ・ビュー

ソース・ツリーのブランチを作成することにより、複数のファイル・セットを別のパス上で変更することができます。ブランチ・ビューを作成することによって、PERFORCE はブランチ作成に関連するファイルのコピーや編集の伝達作業を自動管理できます。

ブランチ・ビューは、ディポの既存の領域（反映元ファイル）をディポの新しい領域（反映先ファイル）にマッピングします。ブランチ・ビューはクライアント・ビューの定義に使用される方法とほぼ同じ方法で定義されますが、ファイルを直接クライアント・ワークスペースにマッピングするというよりは、ディポ内のマッピングをセットアップするだけです。反映はどちらの方向にも実行される可能性があるため、ブランチ・ビュー内の各行ではどちらかの方向を明確に示さなければなりません。したがって、ブランチ・ビューではオーバーレイ・マッピングは使用できません。

ブランチ・ビュー	マッピングの例
メインのコードラインから新しいコードのブランチを作成します。	<code>//depot/main/...</code> <code>//depot/1.1dev/...</code>
新しいリリースでディレクトリを並べ替えます。	<code>//depot/main/...</code> <code>//depot/1.1dev/...</code> <code>//depot/main/*.c</code> <code>//depot/1.1dev/src/*.c</code> <code>//depot/main/*.txt</code> <code>//depot/1.1dev/doc/*.txt</code>

ブランチ・ビューを作成するには、`p4 branch newbranch`を使用します。このコマンドにより画面（`p4 client`で表示する画面と同様）が表示され、メインのソースツリーからの反映元ファイルを新しいブランチの反映先ファイルにマッピングできます。

ブランチ・ビューが最初に作成されたときは、ファイルは1つもコピーされません。ファイルをコピーするには、それらのファイルを使用する目的で、新しく作成したファイルがクライアント・ワークスペース・ビューの中に確実に記述されていなければなりません。新しくマッピングされたディポのブランチを現クライアント・ワークスペース・ビューに追加し、`p4 sync`コマンドを実行すれば、コピーは完了します。

ラベル・ビュー

ラベル・ビューは、ディポ内のファイルのセットにラベルを割り当てます。クライアント・ビューやブランチ・ビューと異なり、ラベル・ビューはファイルをコピーしません。ラベル・ビューを使用して、ラベルによりタグ付け可能なファイルのセットを限定します。

ラベル・ビュー	マッピングの例
新しいリリース	<code>//depot/1.1final/</code>
新しいリリースのためのソースコード	<code>//depot/1.1final/src/...</code>
クライアントに適した配置	<code>//depot/1.1final/bin/...</code> <code>//depot/1.1final/doc/...</code> <code>//depot/1.1final/readme.txt</code>

ラベルを作成するには `p4 label labelname` を使用し、ビューのディポ・サイドを入力します。ラベルはファイルとリビジョン・レベルの単なるリストであるため、指定する必要があるのはビューのディポ・サイド（左側）だけであり、オーバーレイ・マッピングは使用できません。

ファイルタイプ

概要

PERFORCE は、6 つの基本ファイルタイプをサポートしています。

- text ファイル
- 圧縮された binary ファイル
- Macintosh 本来の apple ファイル
- Macintosh のリソース・フォーク (resource)
- シンボリック・リンク (symlink)
- unicode ファイルおよび utf16 ファイル

これらの基本タイプには、RCS キーワード拡張やサーバ上でのファイル圧縮などをサポートするため、ファイルタイプ修飾子が適用されます。

ファイルを追加のための作業状態にすると、PERFORCE はファイルタイプを自動で判断しようとします。ファイルが通常ファイルまたはシンボリック・リンクの場合、ファイルタイプはそれに従って設定されます。PERFORCE は、次にファイルの最初の 8192 バイトを調べて、ファイルがテキストとバイナリのどちらであるかを判断します。非テキスト文字が見つければそのファイルはバイナリとみなされ、見つからなければテキストとみなされます。(Unicode 環境のファイルでは検出方法が異なります。詳細については、「*PERFORCE 国際語モードに関する注意事項 (i18nnotes.txt)*」をご覧ください。)

PERFORCE の管理者は、タイプマップ機能 (p4 typemap) を使用して PERFORCE のデフォルトのファイルタイプ検出機構をオーバーライドすることができます。この機能は、バイナリ・ファイル形式 (例えば、Adobe 社の PDF やリッチ・テキスト形式) に対して有効です。バイナリ・ファイル形式は、8192 バイト以上の ASCII テキストで始めることができますので、この機能がなければバイナリ・ファイルをテキスト・ファイルと間違えられてしまうことがあります。

基本ファイルタイプ

PERFORCE の基本ファイルタイプは、次のとおりです。

キーワード	解説	注釈	サーバ記憶
text	テキスト・ファイル	クライアント上でテキストとして扱われます。Windows と Macintosh のクライアントでは、行末変換が自動的に実行されます。	RCS 形式のデルタ
binary	非テキスト・ファイル	クライアント上でバイナリ・ファイルとしてアクセスされます。ディポ内に圧縮して格納されます。	フル・ファイル 圧縮
symlink	シンボリック・リンク	UNIX クライアント (および BeOS クライアント) は、シンボリック・リンクとしてアクセスします。非 UNIX クライアントは、(小さい) テキスト・ファイルとして扱います。	RCS 形式のデルタ
apple	マルチ・フォーク Macintosh ファイル	Mac のデータ・フォーク、リソース・フォーク、ファイルタイプ、およびファイル・クリエーターの AppleSingle 記憶。 詳細については、Mac クライアントのリリース・ノートをご覧ください。	フル・ファイル 圧縮、AppleSingle 形式

キーワード	解説	注釈	サーバ記憶
resource	Macintosh リソース・フォーク	PERFORCE 99.1 以前における Mac リソース・フォークの唯一のファイルタイプ。サポートは続けていますが、apple ファイルタイプの使用がより適切です。 詳細については、Mac クライアントのリリース・ノートをご覧ください。	フル・ファイル圧縮
unicode	Unicode ファイル	Unicode モードで動作する PERFORCE サーバは、unicode ファイルタイプをサポートします。これらのファイルは、P4CHARSET により指定されたローカルの文字セットにコード変換されます。 Unicode モードではない PERFORCE サーバは、unicode ファイルタイプをサポートしません。 詳細については、「 <i>PERFORCE 国際語モードに関する注意事項 (i18nnotes.txt)</i> 」をご覧ください。	UTF-8
utf16	Unicode ファイル	PERFORCE サーバが Unicode モードで動作している場合、ファイルは P4CHARSET により指定されたローカルの文字セットにコード変換されます。 PERFORCE サーバが Unicode モードで動作していない場合、ファイルは UTF-8 で転送され、クライアント・ワークスペースにおいて UTF-16 (BOM 付き、クライアント・マシンに適したバイト順) にコード変換されます。 詳細については、「 <i>PERFORCE 国際語モードに関する注意事項 (i18nnotes.txt)</i> 」をご覧ください。	UTF-8

ファイルタイプ修飾子

ファイルタイプ修飾子は、次のようになっています。

修飾子	解説	注釈
+w	ファイルは常にクライアントで書き込み可能です。	
+x	クライアントで実行権を付与します。	実行ファイルに使用します。
+ko	旧式のキーワード拡張	\$Id\$ と \$Header\$ キーワードのみを拡張します。 この修飾子は本来、PERFORCE の 2000.1 より前のバージョンとの下位互換性を保つためのもので、PERFORCE の旧バージョンの +k (ktext) 修飾子に対応しています。

修飾子	解説	注釈
+k	RCS キーワード拡張	RCS (Revision Control System : リビジョン管理システム) キーワードを拡張します。 RCS キーワードは、大文字と小文字を区別します。ファイルの中でキーワードを使用するときは、キーワードの後のコロン (例えば、\$Id:\$) は省略可能です。 サポートしているキーワードは、次のとおりです。 <ul style="list-style-type: none"> • \$Id\$ • \$Header\$ • \$Date\$ • \$DateTime\$ • \$Change\$ • \$File\$ • \$Revision\$ • \$Author\$
+l	排他的な作業状態にする (ロック)。	設定すると、ファイルを編集のための作業状態にできるのは、同時に 1 人だけになります。 複数の開発者による変更をマージしても意味をなさないバイナリ・ファイルタイプ (例えば図表) に対して有効です。
+C	サーバは、各ファイル・リビジョンの完全圧縮バージョンを格納します。	バイナリ・ファイルに対するサーバのデフォルトの格納機構。
+D	サーバは、デルタを RCS 形式で格納します。	テキスト・ファイルに対するサーバのデフォルトの格納機構。
+F	サーバは、リビジョンごとにフル・ファイルを、圧縮せずに格納します。	サイズが大きいバイナリ・ファイルや、PostScript ファイルのようにユーザがテキストと読み取れない長い ASCII ファイルに対して有効です。
+S	最新リビジョンだけがサーバに格納されます。	新しいリビジョンがサブミットされると、古いリビジョンはディポから消去されます。実行ファイルまたは .obj ファイルに対して有効です。
+Sn	最新の n 個のリビジョンだけがサーバに格納されます。 n は 1 から 10 まで、または 16、32、64、128、256、512 のいずれかの数値です。	新しいリビジョンを n 個を超えてサブミットするか、既存の +Sn ファイルの n を現在のリビジョン数より少ない値に変更すると、古いリビジョンは消去されます。
+m	オリジナルの修正時刻を保存します。	ファイルのローカル・ファイルシステム上のタイムスタンプが、サブミットのときに保存、同期のときにリストアされます。Windows 環境におけるサード・パーティ製の DLL に対して有効です。

ファイルタイプは通常リビジョン間で保持されますが、add、edit、または reopen 操作中に -t オプションでオーバーライドや変更することができます。

- `p4 add -t filetype filespec` は、指定したファイルタイプでファイルを追加します。
- `p4 edit -t filetype filespec` は、指定したファイルタイプで、ファイルを編集のための作業状態にします。ファイルタイプは、ディポにサブミットしたときに指定した `filetype` に変わります。
- `p4 reopen -t filetype filespec` は、すでに追加や編集のための作業状態になっているファイルのタイプを変更します。

引数 *filetype* は、*[basetype]+modifiers* のように指定します。例えば、`script.sh` のタイプを実行ファイル・テキストに変更し、RCS キーワード拡張を行うには、`p4 edit -t text+kx script.sh` を使用します。

部分ファイルタイプも使用することができます。例えば既存のテキストファイルを `text+kx` に変更したい場合、`p4 reopen -t +x script.sh` とします。多くの部分ファイルタイプ修飾子は、ファイルタイプに付加するものですが、格納修飾子 (+C、+D、+F) を使うとファイルの格納形式が変更されます。修飾子を消去したい場合はフル・ファイルタイプを指定しなければなりません。

一般的なファイル拡張子に対する PERFORCE のファイルタイプ

以下の表に、一般的なファイル拡張子と、それに対する PERFORCE ファイルタイプおよび修飾子を示します。

ファイルタイプ	PERFORCE ファイルタイプ	説明
.asp	text	アクティブサーバページ・ファイル
.avi	binary+F	Windows ビデオ・ファイル
.bmp	binary	Windows ビットマップ・ファイル
.btr	binary	Btrieve データベース・ファイル
.cnf	text	カンファレンス・リンク・ファイル
.css	text	カスケード・スタイルシート・ファイル
.doc	binary	Microsoft Word ファイル
.dot	binary	Microsoft Word テンプレート
.exp	binary+w	エクスポート・ファイル (Microsoft Visual C++)
.gif	binary+F	GIF 画像ファイル
.htm	text	HTML ファイル
.html	text	HTML ファイル
.ico	binary	アイコン・ファイル
.inc	text	アクティブサーバ・インクルード・ファイル
.ini	text+w	アプリケーション初期設定ファイル
.jpg	binary	JPEG 画像ファイル
.js	text	Java スクリプト言語ソースコード・ファイル
.lib	binary+w	ライブラリファイル (複数のプログラミング言語)
.log	text+w	ログファイル
.mpg	binary+F	MPEG ビデオ・ファイル
.pdf	binary	Adobe PDF ファイル
.pdm	text+w	Sybase Power Designer ファイル
.ppt	binary	Microsoft PowerPoint ファイル
.xls	binary	Microsoft Excel ファイル
.zip	binary+F	ZIP 圧縮アーカイブ・ファイル

PERFORCE のファイルタイプとファイル名とのマッピングについての詳細は、`p4 typemap` コマンドをご覧ください。

キーワード拡張

RCS キーワードは、次のように拡張されます。

キーワード	拡張先	例
\$Id\$	ディポ・シンタックス内のファイル名とリビジョン番号	\$Id: //depot/path/file.txt#3 \$
\$Header\$	\$Id\$ と同じ	\$Header: //depot/path/file.txt#3 \$
\$Date\$	最後のサブミットの日付を、YYYY/MM/DD 形式で表したもの	\$Date: 2000/08/18 \$
\$DateTime\$	最後のサブミットの日付と時刻を、YYYY/MM/DD hh:mm:ss 形式で表したもの 日付と時刻は、サブミット時の PERFORCE サーバにおけるローカル・タイムで記録されます。	\$DateTime: 2000/08/18 23:17:02 \$
\$Change\$	ファイルがサブミットされた PERFORCE チェンジリスト番号	\$Change: 439 \$
\$File\$	ディポ・シンタックスで表記されたファイル名のみ (リビジョン番号なし)	\$File: //depot/path/file.txt \$
\$Revision\$	PERFORCE リビジョン番号	\$Revision: #3 \$
\$Author\$	ファイルをサブミットしている PERFORCE ユーザ	\$Author: edk \$

使用上の留意点

- すでに存在するファイルのタイプは、p4 opened か p4 files で判断できます。
- デルタ記憶 (テキスト・ファイルのデフォルト方式) は、ファイルのリビジョン間の差分 (つまり、デルタ) だけを保存する方法です。フル・ファイル記憶 (バイナリ・ファイルでのデフォルト方式) は、ファイル全体を保存します。ファイルタイプにより、フル・ファイル記憶とデルタ記憶のどちらを使用するかが決まります。PERFORCE は、デルタ記憶に RCS 形式を使用します。
- ファイルタイプによっては、ディポ内に保存するために gzip 形式に圧縮されます。圧縮はサブミットのプロセス中に行われ、解凍は同期中に行われます。プロセスはユーザに対して透過的です。クライアント・ワークスペースには常にサブミットしたときのファイルがあります。
- 非 UNIX クライアント・ワークスペース上のシンボリック・リンクは、リンク先のファイルへの相対パスを含む小さなテキスト・ファイルとして表示されます。非 UNIX クライアントでこれらのファイルを編集するときは注意してください。これらのファイルをディポにサブミットすると、シンボリック・リンクが UNIX クライアント・ワークスペース上に存在しないファイルを指示することになる場合があるからです。
- ファイルタイプを変更しても、ディポに保存された以前のリビジョンには影響しません。

例えば、+Sn (一時オブジェクト) 修飾子を追加することによりファイルタイプを変更することは、ディポ内のファイルの最新 *n* 個のリビジョンのみを保存するよう PERFORCE に通知したことになります。既存のファイルを一時オブジェクトに変更した場合、それ以降 (*n* 番目より後) は古くなった最新リビジョンが消去されるようになりますが、+Sn 修飾子を追加する 前の リビジョンについては影響を受けません。(+Sn 修飾子を追加した後の最新でないリビジョンを同期しようとする、そのファイルはクライアント・ワークスペースから削除されます。このようなリビジョンは、p4 filelog の出力において purge 操作として表示されます。)

- 修正時刻 (+m) 修飾子は特別な事例です。これは、ファイルのオリジナルのタイムスタンプを保持したい開発者が使用する修飾子です。(通常、PERFORCE はファイルを同期させるときにタイムスタンプを更新します。) この修飾子により、p4 sync 後のクライアント・ワークスペースにあるファイルのタイムスタンプを、確実にサブミット時のファイルにあるオリジナルのタイムスタンプに戻すことができます。(つまり、サブミット時の PERFORCE サーバでの時刻でもなく、同期時のクライアントでの時刻でもありません。)

この修飾子が有効になる典型的な事例は、Windows 環境によくあるサード・パーティ製 DLL を含む開発です。こうしたファイルのタイムスタンプは、通常 (開発環境内や OS によって) バージョン情報の代わりとして使用されるため、PERFORCE ユーザのクライアント設定にかかわらず、時々ファイルのオリジナルのタイムスタンプを保持する必要があります。

ファイルの +m 修飾子は次のような状況を引き起こします。+m 修飾子が設定されている場合、PERFORCE はファイルを同期するときにクライアント・ワークスペースの modtime (“サブミット時のファイルのタイムスタンプ”) または nomodtime (“同期時のクライアントでの日付と時刻”) オプション設定を無視し、常にサブミット時のファイルのオリジナルのタイムスタンプをリストアします。

- PERFORCE の 99.1 より前バージョンでは、一連のキーワードを使用してファイルタイプを指定していました。以下の表は、旧キーワードとそれに対応する現在の基本ファイルタイプおよび修飾子とをリストにしたものです。

旧キーワード	解説	基本ファイルタイプ	修飾子
text	テキスト・ファイル	text	なし
xtext	実行可能なテキスト・ファイル	text	+x
ktext	RCS キーワード拡張を行ったテキスト・ファイル	text	+k
kxtext	RCS キーワード拡張を行った実行可能なテキスト・ファイル	text	+kx
binary	非テキスト・ファイル	binary	なし
xbinary	実行可能なバイナリ・ファイル	binary	+x
ctext	圧縮されたテキスト・ファイル	text	+C
cxtext	圧縮された実行可能なテキスト・ファイル	text	+Cx
symlink	シンボリック・リンク	symlink	なし
resource	Macintosh リソース・フォーク	resource	なし
uresource	圧縮されていない Macintosh リソース・フォーク	resource	+F
ltext	長いテキスト・ファイル	text	+F
xltext	実行可能な長いテキスト・ファイル	text	+Fx
ubinary	圧縮されていないバイナリ・ファイル	binary	+F
ubinary	圧縮されていない実行可能なバイナリ・ファイル	binary	+Fx
tempobj	一時オブジェクト	binary	+FSw
ctempobj	圧縮された一時オブジェクト	binary	+Sw
xtempobj	実行可能な一時オブジェクト	binary	+FSwx
xunicode	実行可能な Unicode ファイル	unicode	+x

索引

記号

コメント・キャラクタ 94
パスワードには使用不可 170
ファイル名内の 9, 70, 76, 176, 214, 216
リビジョン指定子 212

%
ファイル名内の 9, 70, 76, 176, 214, 216

%%n
ワイルドカード 211

&
論理演算子 AND 89

*
p4 users で使用するワイルドカード 173
p4 user フォーム中でパスワードを隠す 170
ジョブ検索のワイルドカード 89
ファイル名内の 9, 70, 76, 176, 214, 216
プロテクション・テーブル中のワイルドカード 120
ワイルドカード 211

+m
更新時刻の保持 166

…
p4 add で使用禁止のワイルドカード 10
ワイルドカード 211

/
ジョブ・テンプレートでの値の区切り記号 94

/tmp
および TEMP 206

=, >, /, >=,
比較演算子 89

@
リビジョン指定子 212

@
ファイル名内の 9, 70, 76, 176, 214, 216

^
論理演算子 NOT 89

|
論理演算子 OR 89

A

admin アクセス・レベル 119
allwrite 30
API
PERFORCE、p4 fstat 67
.asp ファイル 221
.avi ファイル 221

B

BeOS
BeOS とシンボリック・リンク 218
.bmp ファイル 221
.btr ファイル 221

C

clobber 30, 153
.cnf ファイル 221
compress 30
COMPUTERNAME
Windows 上のデフォルトのクライアント・ワークスペース 183
CR/LF の変換
LineEnd の設定 31
crlf 30
.css ファイル 221

D

DNS
P4PORT 200
.doc ファイル 221
.dot ファイル 221
-d オプション
チェンジリストの削除 20

E

EDITOR_SIGNATURE
Macintosh 上での P4EDITOR 188
emacs

デフォルト・フォーム・エディタとして設定
する 188
.exp ファイル 221

F

-f オプション
クライアント・ワークスペースの設定のオー
バーライド 29
サブミット済チェンジリストの編集 20
ラベルの強制削除 97

G

getcwd()
PWD の代わり 205
.gif ファイル 221
gzip 222

H

hosts ファイル
P4PORT 200
.htm ファイル 221
.html ファイル 221

I

.ico ファイル 221
.inc ファイル 221
.ini ファイル 221
integrate
ファイルを作業状態にする 80
IP アドレス
アクセス管理 118
-i オプション
チェンジリストと反映済ファイル 23

J

JobView フィールド
利用 90
JobView フィールド
p4 user フォーム 90
Jobview フィールド
p4 user 171
チェンジリスト 20
.jpg ファイル 221
.js ファイル 221
-J オプション
p4d 181, 190, 191

L

labelsync
必要な所有権 96
必要なラベルの所有権 100
.lib ファイル 221
LineEnd 31
CR/LF 29
list アクセスレベル 118

locked 30
.log ファイル 221
-L オプション
p4d 192
p4p 192
チェンジのコメントの長さ 57
-l オプション
ジョブのコメントの長さ 88
チェンジのコメントの長さ 57

M

Macintosh
デフォルト・フォーム・エディタを変更 188
ファイルタイプ 218
リソース・フォーク・ファイルタイプ 219
maxlocktime
影響を受けるコマンド 73
maxresults
p4 filelog 57
p4 files 58
p4 group での設定 71
影響を受けるコマンド 73
maxscanrows
p4 group での設定 71
影響を受けるコマンド 73
MD5
p4 verify 174
パスワード 114, 196
modtime 31
2000.1 以降の変更 31
.mpg ファイル 221

N

noallwrite 30
noclobber 30, 153
nocompress 30
nocrlf 30
nomodtime 31
2000.1 以降の変更 31
normdir 31

O

open アクセスレベル 118

P

P4ROOT
およびディポ・ファイル 42
p4 add 9
p4 admin 12
p4 annotate 14
p4 branch 16
p4 integrate 80
p4 branches 18
p4 change 19
p4 changes 22

p4 client 27
 オプションおよび p4 sync 153

p4 clients 34

p4 counter 35, 146

p4 counters 37

p4 delete 38
 p4 obliterate 110

p4 depot 40

p4 depots 43

p4 describe 44

p4 diff 46
 P4DIFF 186

p4 diff2 49
 ブランチ・ビュー 17

p4 dirs 52

p4 edit 54

p4 filelog 56

p4 files 58

p4 fix 60

p4 fixes
 チェンジリスト 21

p4 flush 64

p4 fstat 67

p4 group 71

p4 groups 74

p4 have 76
 ワークスペース内のファイル 76

p4 help 78

p4 info 79

p4 integ
 p4 integrate の省略形 82

p4 integrate 80

p4 integrated 84

p4 job 86

p4 jobs 88

p4 jobspec 93
 P4WIN 95

p4 labels 98

p4 labelsync 100

p4 license 102

p4 lock 103

p4 logger 104

p4 login 105

p4 logout 107

p4 monitor 108

p4 obliterate 110
 デイポの削除 41

p4 open 55

p4 opened 112
 チェンジリスト 21

p4 passwd 114
 P4PASSWD 196
 パスワードの設定 196

p4 print 116

p4 protect 118
 Protections フィールド 120

新しいデイポ生成時に必要なプロテクション 41

サーバのインストール後に必要なこと 124

p4 protects 127

p4 rename 128

p4 reopen 129

p4 resolve 131
 P4DIFF 186
 P4MERGE 193
 P4PAGER 195

p4 resolved 136

p4 revert 137
 p4 resolve -at 133

p4 review 139

p4 reviews 141

p4 set 143

p4 submit 148

p4 sync 152
 ブランチ・ビュー 217

p4 tag 155

p4 tickets 157

p4 triggers 158

p4 typemap 165, 218
 and p4 add 9

p4 unlock 168

p4 user 169
 JobView フィールド 90
 Reviews フィールド 141
 ジョブ・ビューと p4 submit 149
 パスワードの設定 196

p4 users 173

p4 verify 174

p4 where 175

p4 workspaces 177, 178

P4AUDIT 180

P4CHARSET 181

P4CLIENT 183

P4COMMANDCHARSET 182

P4CONFIG 184

p4d
 ジャーナル・ファイルの指定 181, 190, 191
 ファイルにエラーを記録 192

P4DEBUG 185

P4DIFF 186
 p4 describe とは無関係 44
 p4 diff 46
 p4 diff2 とは無関係 49

P4DIFFUNICODE 187

P4EDITOR 188
 影響を受けるコマンド 188

p4executable
 バージョン 79

P4HOST 189

P4JOURNAL 190

P4LANGUAGE 191

P4LOG 192

P4MERGE 132, 193
Windows で必要なバッチ・ファイル 193
P4MERGEUNICODE 194
p4p
ファイルにエラーを記録 192
P4PAGER 195
P4PASSWD 196
p4 passwd 196
P4PCACHE 197
P4PFSIZE 198
P4POPTIONS 199
P4PORT 200
P4ROOT 201
Windows サーバ上の一時ファイル 206
P4TARGET 202
P4TICKETS 203
P4USER 204
Windows 上のサブミット前トリガ 163
P4WIN
ジョブ・テンプレートのコメント 94
ポップアップとジョブ仕様 95
.pdf ファイル 221
.pdm ファイル 221
PERFORCEAPI
p4 fstat 67
PERFORCE クライアント
P4PORT 200
一時ファイル 206
PERFORCE クライアントと PERFORCE サーバ
バージョン情報の表示 79
PERFORCE サーバ
P4PORT 200
P4ROOT 201
安全にインストールする 124
一時ファイル 206
管理 12
正当性の検証 174
チェックポイントとジャーナル 12
停止 12
PERFORCE シンタックス 211
PERFORCE の管理 12
Perforce ファイルタイプ 221
PERFORCE プロキシ
および P4PCACHE 197, 198, 199
および P4PTARGET 202
POSIX\$SHELL
VMS 上での P4EDITOR の値 188
.ppt ファイル 221
Protections フィールド 120
PWD 205

R
RCS キーワード拡張 220
例 222
RCS ファイル形式 222
read アクセスレベル 118

remote 42
Reviews フィールド
p4 user 141
使用 171
review アクセスレベル 119
rmdir 31

S

Status フィールド
p4 submit 148
subgroups
グループ 72
super アクセスレベル 119
-s オプション
p4 fstat 70

T

tag 155
TMP,TEMP 206
-t オプション
クライアント・ワークスペースのテンプレート 29
ファイルタイプ 220

U

unicode 182, 187, 194
UNIX
デフォルト・フォーム・エディタを変更する 188
行末識別 31
マウントされたドライブの行末 32
unlocked 30
UTF-16 182, 187, 194
UTF-8 182, 187, 194

V

vi
デフォルト・フォーム・エディタの変更 188
VMS
デフォルト・フォーム・エディタの変更 188

W

Windows
P4MERGE に必要なバッチ・ファイル 193
Windows サービス用の変数設定 179
行末 32
行末識別 32
サードパーティの DLL 32, 223
サービスとトリガ 163
デフォルトのクライアント・ワークスペース 183
デフォルトのクライアント・ワークスペースとしての COMPUTERNAME 183
デフォルト・フォーム・エディタ 188
パスワードの設定 196

レジストリ変数 143
レジストリ変数のオーバーライド 144
write アクセスレベル 118

X

.xls ファイル 221
-x オプション
p4 diff の使用例 48

あ

アクセス
管理者 119
IP アドレスで制限 118
スーパー・ユーザ 119
レベル 118
アクセスレベル
p4 group 71
アクセスレベルとコマンドの一覧表示 121
圧縮
ファイルの自動圧縮 222
アップグレード
98.2 以前からのアップグレード 88
アトミック・チェンジ 148

い

位置指定子 211
一時ファイル
格納場所 206
一覧表示
カウンタ 37
クライアント・ワークスペース 34
グループ 74
作業状態のファイル 112
修正 62
衝突解決がスケジュールされたファイル 136
ジョブ 88
チェンジリスト 22
チェンジリストにリンクしたジョブ 62
ディポ 43
ディポ内ファイル 58
ファイル内のラベル 98
ファイル反映 84
ラベル 98

え

エディタ
P4EDITOR でフォームを指定する 188
フォームを使用するコマンド 188
演算子
比較演算子とジョブ・ビュー 89
論理演算子とジョブ・ビュー 89

お

オーバーライド
レジストリ変数設定値 144

オーバーレイ・マッピング 215
オプション
クライアント・ワークスペース用 30
グローバル・オプション 208
オンラインヘルプ
p4 の使用 78

か

カウンタ
p4 review 139
review アクセス権限 124
継続的に使用される変数の一覧表示 37
継続的に使用される変数の設定 35
制限 36
空のディレクトリ
sync 実行時の削除 31
カレント・ディレクトリ 205
非 UNIX クライアントにおける一時ファイル
206
環境変数
P4AUDIT 180
P4CHARSET 181
P4CLIENT 183
P4COMMANDCHARSET 182
P4CONFIG 184
P4CONFIG による自動変更 184
P4DEBUG 185
P4DIFF 186
P4DIFFUNICODE 187
P4EDITOR 188
P4HOST 189
P4JOURNAL 190
P4LANGUAGE 191
P4LOG 192
P4MERGE 193
P4MERGEUNICODE 194
P4PAGER 195
P4PASSWD 196
P4PCACHE 197
P4PFSIZE 198
P4POPTIONS 199
P4PORT 200
P4ROOT 201
P4TARGET 202
P4TICKETS 203
P4USER 204
PWD 205
TMP,TEMP 206
Windows サービス用の設定 179
Windows レジストリ 143
設定方法 179
ヘルプ 78
環境変数 MERGE
P4MERGE 193
環境変数 PAGER
P4PAGER 195

環境変数 SHELL

Windows の場合の P4DIFF 186

Windows の場合の P4EDITOR 188

環境変数の設定 179

Windows サービス上 179

Windows サービス用 143

管理

パスワードのリセット 115

き

キーワード

RCS、拡張 220

RCS、例 222

旧 PERFORCE ファイルタイプを指定する 223

基本ファイルタイプ 218

行末 31

く

クライアント

クライアントと一時ファイル 206

ラベル 100

クライアント・シンタックス 211

p4 files 58

変換 175

クライアント・ビュー 216

p4 client 27

p4 print 116

クライアント・ビューと p4 sync 152

定義 215

クライアント・ワークスペース

p4 have で表示される同期ファイル 76

オプション 30

削除 29

生成および編集 27

設定の自動変更 184

全クライアント・ワークスペースのリスト 34

定義 27

ディポとファイルの比較 46

ディポ・ファイルをコピーして取り込む 152

名前 183

ファイルタイプによってクライアント・ワークスペース内のファイルのパーミッションを設定する 219

別名のルート 28

ラベルとの同期 100

ルート 28

クライアント・ワークスペーステンプレート 29

グループ

subgroups 72

アクセス管理 118

削除 71

生成 71

ユーザの一覧表示 74

グローバル・オプション 208

ヘルプ 78

クロスプラットフォーム開発

行末文字制御 31

け

警告

p4 counters 35

p4 flush 64

p4 obliterate 110

p4 revert 138

カウンタと p4 review 139

サブミット前トリガ 158

スーパーユーザのアクセス権限と p4 protect 124

検索

空のジョブ・フィールド 91

ジョブ・ビューによるジョブの検索 88

こ

更新時刻の保持 166

コードライン

ファイルの比較 49

ブランチ・ビュー 17

コマンド

アクセス管理 118

アクセスレベル別に一覧表示 121

ヘルプ 78

コマンドライン・オプション

グローバルオプション 208

コメント

ジョブ・テンプレート P4WIN 94

さ

サーバ

IP アドレスの変更 102

P4PORT 200

アップグレード 88

一時ファイル 206

インストールと p4 protect 124

エラー・ログ・ファイルを指定 192

管理 12

ジャーナル・ファイルを指定 181, 190, 191

正当性の検証 174

チェックポイントとジャーナル 12

停止 12

ディスク容量の再生 110

ライセンス 102

サーバ変数

設定 35

一覧表示 37

サーバ・ルート

およびディポ 42

201

Windows サーバ上の一時ファイル 206

最新リビジョン

p4 delete 38

p4 edit 54

指定 212

- 作業状態のファイル
 - p4 reopen によるタイプの変更 129
- 作業中チェンジリスト 19, 148
 - 一覧表示 22
 - ファイルの一覧表示 112
- サブディレクトリ
 - 一覧表示 52
- サブディレクトリの一覧表示 52
- サブミット済チェンジリスト 19, 148
 - 一覧表示 22
- サブミット前トリガ 158
 - スクリプト作成のヒント 162
- 差分チャンク
 - ファイル衝突 133
- 差分プログラム
 - p4 describe 44
 - p4 diff 46
 - p4 diff2 49
 - PERFORCE の内部ルーチン 186
 - サード・パーティー製、指定 186
- し**
- シエル
 - ファイル指定の解釈 173, 212
- 指定
 - p4 resolve の出力を表示するプログラム 195
 - サード・パーティー製差分プログラム 186
 - 最新リビジョン 212
 - チェンジ番号でファイルを指定 212
 - デフォルト・フォーム・エディタを P4EDITOR で指定 188
 - 反映操作を行うファイル 80
 - 日付と時刻で指定する 212
 - ファイルタイプ 219
 - リビジョンでファイルを指定 212
 - リビジョン範囲 213
- 指定子
 - 位置指定子 211
 - リビジョン指定子 212
- ジャーナル 12
- ジャーナル・ファイル
 - P4JOURNAL で指定する 190
- 修飾子
 - ファイルタイプ、+m 166
- 修正
 - p4 fix -d による修正完了記録の削除 60
 - 一覧表示 62
 - 複数のチェンジリストによるジョブの修正 60
- 修正ステータス
 - デフォルト 95
- 仕様
 - ジョブ仕様の表示 90
- ジョブ
 - 状態の変更 61
- 状態
 - ジョブの状態を変更する 61
- 仕様ディポ
 - 書き込み 12
- 衝突解決
 - ファイルをスケジュールする 135
- 衝突未解決のファイル
 - 一覧表示 136
- 仕様トリガ 159
- 除外マッピング 215
 - p4 protect 120
 - トリガ 159
- ジョブ
 - * ワイルドカード 89
 - p4 fix によるチェンジリストへのリンク 60
 - p4 submit で作業状態を終了する 149
 - 一覧表示 88
 - 空フィールド 91
 - ジョブ・ビューに関する制約の回避 91
 - チェンジリスト 20
 - チェンジリストへのリンクの一覧表示 62
 - チェンジリストへのリンクの表示 21
 - 複数のチェンジリストによる修正 60
 - ワイルドカード 91
- ジョブ仕様
 - 表示 90
- ジョブの作業状態を終了する
 - p4 submit を使用して 149
- ジョブ・テーブル
 - 再インデックス付け 88
- ジョブ・テンプレート
 - 編集 93
 - 中のコメントと P4WIN 94
- ジョブ・ビュー
 - ジョブ検索 88
 - 制約 90
 - 比較演算子 90
 - フィールドタイプ 90
 - ヘルプ 78
- 所有者
 - ラベル所有者の変更 96
- 所有リスト
 - p4 delete 38
 - p4 have による一覧表示 76
 - 定義 76
 - ワークスペースファイルとの関係 76
- 所有リビジョン 76, 212
- シンタックス・フォーム
 - p4 where によるフォーム間での変換 175
- シンボリック・リンク 218
 - 非 UNIX システム 218, 222
- す**
- スーパー・ユーザ 119
 - 新規ユーザ 124
 - ユーザの作成 169

スクリプト作成
p4 dirs 52
p4 fstat 67
-s と p4 fstat 70
-x オプション、例 48
トリガ 158
トリガ、スクリプト作成のヒント 162
スペース
ファイル名に含まれる 30

せ

生成
ディポ仕様 40
ブランチ・ビュー 16
正当性の検証 174
制約
ジョブ・ビュー 90
セキュリティ
p4 protect 124
セキュリティ・レベル 170

た

タイプ
ファイルタイプの変更 129
タイプマッピング 165
タイプマップ 9
タイムスタンプ
DLL 上に保持 223
DLL 上での保存 32

ち

チェックポイント 21
21
チェンジリスト
p4 fixes による関連ファイルの一覧表示 21
p4 opened による関連ファイルの一覧表示 21
p4 review で一覧表示する 139
一覧表示 22
意味 20
再サブミット時に指定する 148
作業中チェンジリストで作業状態にある
ファイルの一覧表示 112
作業中チェンジリストとサブミット済チェ
ンジリスト 148
削除 20
サブミット 148
ジョブ 20, 60
ジョブ・ビューとユーザ 90
生成および編集 19
チェンジリスト間でファイルを移動する 129
定義 19
デフォルト・チェンジリストと p4 submit
148
発番 19
番号付き 148

番号付きチェンジリストのコメント情報を
変更する 151
ファイル削除時の指定 38
ファイル追加時の指定 9
ファイル編集時の指定 54
明細表示 44
リンクされたジョブの一覧表示 62
目的 150
チェンジリスト・サブミット・トリガ 159
チェンジリスト番号
最大値 36
作業中チェンジリストとサブミット済チェ
ンジリスト 36
チェンジリストをサブミットする 148
チェンジリスト番号
最大値 36
チェンジ・レビュー・デーモン 119, 124, 139, 141,
171
チケット・ファイル
場所 203

て

ディスク容量 146
再生 110
ディポ
およびサーバ・ルート 42
一覧表示 43
空のファイル 10
クライアント・ワークスペースとのファイル
の比較 46
削除 41
生成または編集 40
正当性の検証 174
ファイルの一覧表示 58
ファイルの取得 152
ファイルの追加 10
ファイルの2つのリビジョンの比較 49
ファイルの保存方法 222
変更をサブミットする 148
リモート 40
リモート・ディポ 42
リモート・ディポとプロテクション 124
ディポからのファイル取得 152
ディポ・シンタックス 211
p4 branch 16
p4 print 116
所有リスト 76
プロテクション・テーブル 120
変換 175
ディレクトリ
およびスペース 30
カレント・ディレクトリ 205
ディレクトリ、空
sync による削除 31
デーモン
review アクセス権限 124

作成のヒント 162
チェンジリスト番号 36
チェンジ・レビュー 124, 139, 141, 171
チャレンジ・レビュー 119
テキスト・ファイル 218
デフォルト・チェンジリスト
作業状態のファイルの一覧表示 112
デフォルト・チェンジリスト
p4 submit 148
デルタ記憶
定義 222
テンプレート
クライアント・ワークスペース 29
ジョブ・テンプレートの編集 93

と

同期 152
トラブルシューティング
ローカル・シェルとファイル指定 212
トリガ
タイプ 159
158
Windows サービス 163
引数を渡す 162

ね

ネットワーク
データ圧縮 30

は

バージョン
PERFORCE クライアント・プログラムおよび
PERFORCE サーバプログラム
のバージョン 79
パーミッション
新しいディポにアクセスするために必要な
パーミッション 41
ファイルタイプによってクライアント・ワー
クスペース内のファイルのパーミッ
ションを設定する 219
ファイルのパーミッションと p4 edit 54
付与と拒否 118
バイナリ・ファイル 218
比較 50
パスワード
およびユーザ 170
P4PASSWD 196
コマンドラインでの指定 114
最大長 115
削除 114
設定 114
特殊文字 170
ユーザ 170
リセット 115
パスワード中のスペース
引用符で囲む 115

パスワードの再設定 115
パスワードの削除 114
バッチ・ファイル
P4MERGE 193
反映
一覧表示 84
スケジュール 131
番号付チェンジリスト 148

ひ

比較
バイナリ・ファイル 50
ファイル 46, 49
比較演算子
ジョブ・ビュー 89
非存在リビジョン
指定 212
ビュー
概要 215
クライアント・ビュー 216
クライアント・ビューと p4 sync 152
ブランチ・ビュー 217
ブランチ・ビューと p4 integrate 80
ブランチ・ビューと p4 diff2 17
ブランチ・ビューと p4 sync 217
ブランチ・ビュー、生成および編集 16
ヘルプ 78
ラベル・ビュー 217
標準出力
p4 print 116

ふ

ファイル
トリガ 155
ラベルからの削除 155
#none で除去する 212
.asp 221
.avi 221
.bmp 221
.btr 221
.cnf 221
.css 221
.doc 221
.dot 221
.exp 221
.gif 221
.htm 221
.html 221
.ico 221
.inc 221
.ini 221
.jpg 221
.js 221
.lib 221
.log 221
.mpg 221

-
- p4 integrate で削除するために作業状態にする 80
 - p4 integrate でブランチするために作業状態にする 80
 - .pdf 221
 - .pdm 221
 - PERFORCE ファイルタイプをファイル名に適合させる 165
 - .ppt 221
 - t オプションによるタイプの変更 220
 - .xls 221
 - アクセス管理 118
 - 圧縮して保存 222
 - 位置の表示 175
 - 一覧表示 58
 - 完全消去 110
 - 更新時刻の保持 166
 - コードライン間の比較 49
 - 最新リビジョンの取得 212
 - 作業状態のファイル 112
 - 作業状態のファイルの一覧表示 112
 - 削除するために作業状態にする 38
 - サブミット 148
 - 指定 211
 - 衝突解決 131
 - 衝突解決がスケジュールされているファイルの一覧表示 136
 - 衝突解決時の *yours, theirs, base, merge* の意味 132
 - 衝突解決をスケジュールする 135
 - 衝突、衝突解決 131
 - 衝突未解決ファイルの一覧表示 136
 - スクリプトのための情報表示 67
 - 正当性の検証 174
 - タイプ 218
 - タイプの変更 129
 - タイプを指定する 219
 - チェックポイントとジャーナル 12
 - チェンジ番号で指定する 212
 - チェンジリストからの削除 137
 - チェンジリスト間の移動 129
 - チェンジリストに含まれているファイルの詳細情報 44
 - チェンジリストへのリンクの一覧表示 21
 - 追加するために作業状態にする 9
 - デフォルトタイプの追加、指定 9
 - ディポからコピーする 152
 - ディポからの削除 38
 - ディポからの取得 152
 - ディポへの追加 9
 - デフォルト・タイプへの追加、指定 165, 218
 - デルタ記憶、フル・ファイル記憶 222
 - 同期 152
 - 特定リビジョンの内容の一覧表示 116
 - 名前の変更 128
 - バイナリファイルの比較 50
 - 反映操作のために作業状態にする 80
 - 比較 46, 49
 - 日付と時刻で指定する 212
 - ファイル間の変更の反映 131
 - ファイル反映履歴の表示 84
 - ファイルへの変更をディポに送る 148
 - ファイルへの変更を廃棄する 137
 - 古いリビジョンの編集 55
 - 変更をディポ保存する 148
 - 編集 54
 - 編集するために作業状態にする 54
 - 編集前の状態に戻す 137
 - 他のディポにあるファイルへのアクセス 40
 - 他のユーザによる編集を防止する 103
 - マルチ・フォーク 218
 - ラベルからの削除 100
 - ラベルに含まれているファイルの詳細情報 98
 - ラベルへの追加 100
 - リビジョンで指定する 212
 - リビジョン履歴の表示 56
 - ロック 103
 - ロック解除 168
 - ファイル・サイズ 146
 - ファイル指定
 - p4 revert 137
 - p4 submit 151
 - ヘルプ 78
 - ローカル・シェルによる解釈 212
 - ファイル衝突の解決 131
 - ファイルタイプ 221
 - apple 218
 - binary 218
 - p4 add 10
 - p4 edit 54
 - PERFORCE による判断 218
 - resource 219
 - symlink 218
 - text 218
 - t オプションで指定 220
 - 一覧表示 223
 - 解説 218
 - キーワード 221
 - 旧キーワード 223
 - クライアント・ワークスペースにおけるパーミッション 219
 - 指定 219
 - 修飾子 223
 - ディポへの保存 222
 - 表示 222
 - ファイル名へのマッピング 165
 - ヘルプ 78
 - 変更 129
 - ファイルタイプの変更
 - t オプションを使用する 220
 - ファイルの完全消去 110

- ファイルの削除 38
- ファイルの差分比較 46, 49
- ファイルの消去
 - 完全消去 110
- ファイルの追加 9
 - デフォルトタイプの指定 9
 - デフォルト・ファイルタイプの指定 165
- ファイルのロック 103
- ファイルのロック解除 168
- ファイルの追加
 - デフォルト・ファイルタイプの指定 218
- ファイルへの変更を元に戻す 137
- ファイル名
 - およびスペース 30
 - 特殊文字 9, 70, 76, 176, 214, 216
 - コマンドラインでスペースを使用する 213
 - ファイルタイプへのマッピング 165
- ファイル名中のスペース
 - 引用符で囲む 213
- ファイル名の変更 128
- ファイルを作業状態にする
 - 削除する目的で 38
 - 追加する目的で 9
 - 編集する目的で 54
- ファイルをサブミットする 148
- フィールド
 - 空のフィールドを持つジョブ 91
- フォーム
 - P4EDITORでデフォルトのエディタを指定する 188
 - フォームを使用するコマンド 188
- ブランチ
 - ファイル比較 49
- ブランチ作成 16
- ブランチ仕様
 - 生成および編集 16
- ブランチ・ビュー 217
 - p4 branch 16
 - p4 diff2 17
 - p4 integrate 80
 - コードライン指定例 17
 - 定義 215
 - ブランチ・ビューと p4 sync 217
- フル・ファイル記憶
 - 定義 222
- プロキシ
 - エラー・ログ・ファイルを指定 192
 - および P4PCACHE 197, 198, 199
 - および P4TARGET 202
- プロテクション
 - 表示 127
 - IP アドレス 118
 - 付与および拒否 118
- プロテクション・テーブル 118
 - 例 125

へ

- ヘルプ
 - p4 help の使用 78
- 変更
 - 最小単位 148
 - 衝突、衝突解決 131
- 変更の取消 137
- 変更を廃棄する 137
- 編集
 - ディポ仕様 40
 - ファイル 54
 - ユーザ仕様 169
- 変数
 - 環境変数の設定方法 179
 - サーバ変数の一覧表示 37
 - サーバ変数の設定 35
 - レジストリ変数 143

ほ

- ポート番号
 - クライアントおよびサーバでの設定する 200
- ホスト
 - 別のホストからアクセスしているように見せかける 189
- 保存
 - ディポ内でのファイル保存 222
- ポップアップ 95

ま

- マッピング
 - オーバーレイ 215
 - p4 client 27
 - クライアント・ビューにおけるマッピング 216
 - 除外マッピング 215
 - 除外マッピングとトリガ 159
 - 除外マッピングとプロテクション・テーブル 120
 - スペースを含むディレクトリ 30
 - 反映と p4 branch 80
 - ブランチ・ビュー 16
 - ブランチ・ビュー 217
 - プロテクション・テーブル 120
 - ラベル・ビュー 97
 - ラベル・ビュー 217
 - ローカル・ディポとリモート・ディポ 41
- マッピングの順序
 - トリガ 159
 - プロテクションにおける 120
- マルチ・フォーク・ファイル 218

ゆ

- ユーザ
 - およびパスワード 170
 - p4 reviews で一覧表示 141

- P4CONFIG や P4USER で変更する 169
- P4PASSWD 196
- アクセス管理 118
- 一覧表示 173
- 仮想ユーザ 42
- 仮想ユーザ、remote 124
- グループ 71
- グループへのアクセス許可 118
- グループ、一覧表示 74
- 削除 170
- 作成および編集 169
- パスワード 114, 170
- パスワードを忘れたとき 115
- ファイルのロック解除 168
- 他のユーザによるファイル編集の防止 103
- ユーザ仕様の作成 169
- ユーザ設定
 - 設定 169
- ユーザの一覧表示 173
- ユーザの削除 170

ら

- ライセンス
 - 仮想ユーザ 42
 - サブミット前トリガ 163
- ラベル 155
 - ラベルからの削除 155
 - 一覧表示 98
 - クライアントとの同期 100
 - 所有者の変更 96
 - ファイルの一覧表示 98
 - ファイルの削除 100
 - ファイルの追加 100
 - ロックの解除 96
- ラベルクライアント 100
- ラベル・ビュー 217
 - 定義 215

り

- リソース・フォーク 219
- リビジョン
 - カレント・クライアントにおけるファイルリビジョン 212
 - 最新リビジョンを指定する 212
 - 指定子 212
 - ファイルリビジョンの表示 116
 - ラベル 100
- リビジョン指定子 212
 - p4 changes 22
 - p4 sync 152
 - ヘルプ 78
- リビジョン範囲
 - and p4 files 58
 - p4 changes 22
 - p4 fixes 62
 - p4 integrate 80

- p4 print 116
- p4 sync 152
- 指定 213
- リビジョン履歴
 - 完全消去 110
 - 表示 56
- リビジョン範囲
 - p4 resolved 136
- リモート・ディポ 40, 42
- リモート・ディポ
 - プロテクション 124

れ 例

- p4 typemap 166
- RCS キーワード拡張 222
- クライアント・ワークスペースへの同期 153
- 作業状態のファイルの一覧表示 113
- 作業中チェンジリストで作業状態になるファイルの一覧表示 113
- サブミット前トリガの使用 163
- さまざまな基準によるジョブの一覧表示 91
- 衝突解決のスケジューリング 83
- スクリプトによる構文解析のための出力の生成 70
- チェンジリスト間でのファイル移動 130
- ディポからのファイル取得 153
- パイプと -x 48
- ファイルタイプの変更 129
- ファイルの反映 83
- ファイル名の変更 128
- ファイルを作業前の状態に戻す 138
- ファイルをチェンジリストでサブミットする 151
- ブランチ作成とコードライン 17
- ブランチを介したファイルの比較 51
- プロテクションの影響 125
- プロテクション・テーブル 125
- 変更の伝達 83
- 他のユーザとして作業する 171
- ユーザ情報の参照 171
- ユーザ情報の編集 171
- ユーザの削除 171
- レジストリ
 - ブレーン・テキストのパスワードを保存しない 114
 - ブレーン・テキストのパスワードを保存しない 196
 - 変数の設定 143
- レジストリ変数
 - 設定をオーバーライドする 144

ろ

- ローカル・シンタックス
 - 所有リスト 76
- ローカル・シンタックス 211

変換 175
論理演算子
ジョブ・ビュー 89

わ

ワークスペース
クライアント, 別名のルート 28
クライアント・ワークスペース、一覧表示 34
クライアント・ワークスペース、生成および編
集 27
ワークスペース内のファイルと所有リスト
76

ワイルドカード
p4 add 10
p4 integrate 80
ジョブ・ビュー 89
ファイルを指定 211
ユーザの一覧表示 173