
Perforce 2013.1

Perforce 入门

2013年4月

This manual copyright 2005-2013 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.

This product is subject to U.S. export control laws and regulations including, but not limited to, the U.S. Export Administration Regulations, the International Traffic in Arms Regulation requirements, and all applicable end-use, end-user and destination restrictions. Licensee shall not permit, directly or indirectly, use of any Perforce technology in or by any U.S. embargoed country or otherwise in violation of any U.S. export control laws and regulations.

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. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

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

本手册的中文版本由上海龙智数码科技有限公司 (<http://www.shdsd.com>) 翻译

本手册版权归Perforce公司 (Perforce Software) 所有

保留一切权利.

Perforce 软件和文档可以从 <http://www.perforce.com> 上获得。你可以下载和使用 Perforce 程序，但你不能转售或发布。你可以下载、打印、复制、编辑和分发这些文档，但你不得销售它们，或销售任何采自它们的文档。不得修改或试图反向工程这些程序。

本产品受制于美国出口管制法律和条例，包括但不限于：美国出口行政法规、国际贸易中的军火出口管制条例的要求和所有适用的最终用途、最终用户及目的地的限制。被许可者不得直接或间接地在任何美国禁运国家或违反任何美国出口管制法律和条例使用任何 Perforce 技术。

对于从我们网站下载的 Perforce 程序和文档，不提供质保或支持。

高性能服务器下的质保和支持是由 Perforce 公司 (Perforce Software) 销售的。

Perforce Software 对这本书中可能出现的任何错误或疏漏不承担任何责任或义务。

下载和使用我们的程序和文档，视同同意上述这些条款。

Perforce 和 Inter-File Branching 是 Perforce Software 的商标。Perforce software 包括由 California 大学，Berkeley 大学 和 他们的 捐赠者 所 开发的 软件。本产品包括由 OpenSSL 项目开发的用于 OpenSSL Toolkit (<http://www.openssl.org/>) 的软件。

所有其他商标或产品名称是他们各自公司或组织的商标或注册商标。

目 录

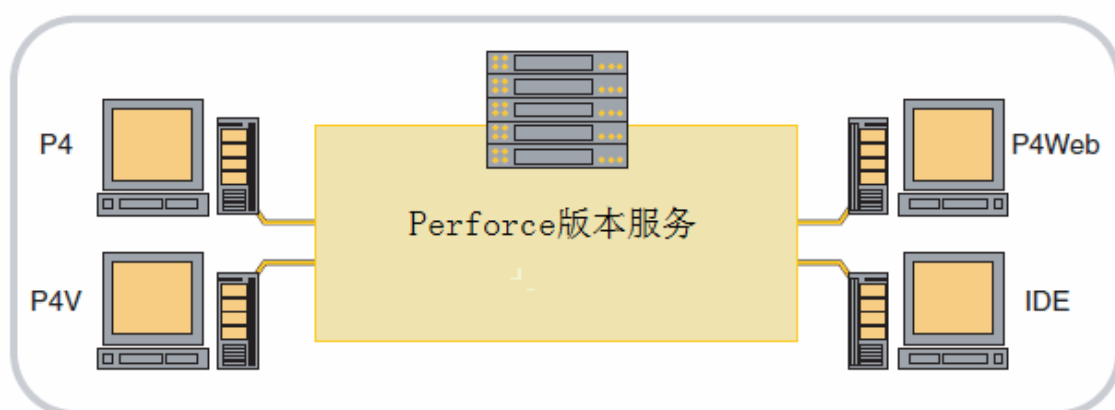
Perforce 入门	5
Perforce 工作原理	5
Perforce 版本管理服务	5
Perforce 应用程序	6
连接到 Perforce 服务	6
把 depot 中的文件映射到工作区	7
其他配置选项	7
使用 Perforce	9
从服务中读取文件	9
同步你的工作区	9
Perforce 文件指定	9
Perforce 语法	10
在视图中使用通配符	10
指定文件的特定版本	11
Perforce 语法以及状态栏	11
支持哪些文件类型?	11
文件操作	11
利用 changelist	12
Changelist 编号是如何工作的	12
编辑文件	12
添加新文件	13
删除文件	13
废弃不需要的变更	13
签入文件	13
解决冲突	13
并行工作	14
比较文件	15
审查单个文件的变更历史	15
审查文件组的变更历史	15
Perforce 语法以及状态栏	16
codeline 管理	17
分支基础知识	17

创建 codeline.....	17
在 codeline 之间复制变更.....	18
解决 codeline 之间的差别.....	19
复制复杂分支结构	19
记录 codeline 之间的变更历史.....	20
使用 Streams	21
设置结构	22
Stream 的类型.....	22
Stream 视图	22
填充 streams	23
复制变更	23
学习更多分支知识	23
后继工作	24
任务并缺陷追踪.....	24
给文件加标签	24
编辑器和合并工具	24
保护和权限	24
用户和许可证	24
从哪里可以更多了解 Perforce	25

Perforce 入门

Perforce 工作原理

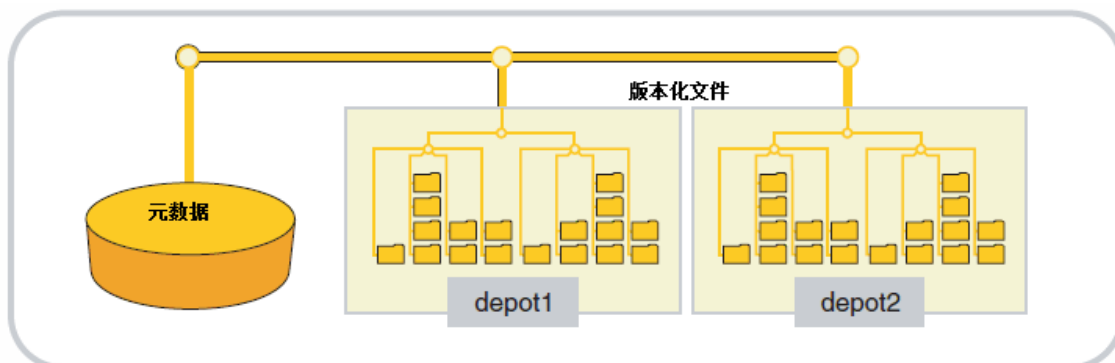
Perforce是一个企业版版本管理系统，用户通过它连接到公用文件信息库。Perforce应用程序用来在文件信息库与个人用户工作站之间传输文件。



本手册假定Perforce版本管理服务已被安装、配置并已运行。有关配置Perforce方面的内容，请参阅《系统管理员手册》。

Perforce 版本管理服务

Perforce版本管理服务管理公用文件信息库（或称之为**depots**）。Depot储存了Perforce控制下的所有文件的所有版本。Perforce把depot中的文件组织成目录树，就像一个大的硬盘。depot中的文件被称之为**depot文件**或**版本化文件**。该服务维护一个数据库以记录变更日志、用户权限以及某个用户在某个时点签出某个文件。存储在该数据库中的信息被称为**元数据**。



Perforce服务通过本地操作系统来管理数据库和版本化文件，因而不需要专门的文件系统或者卷。

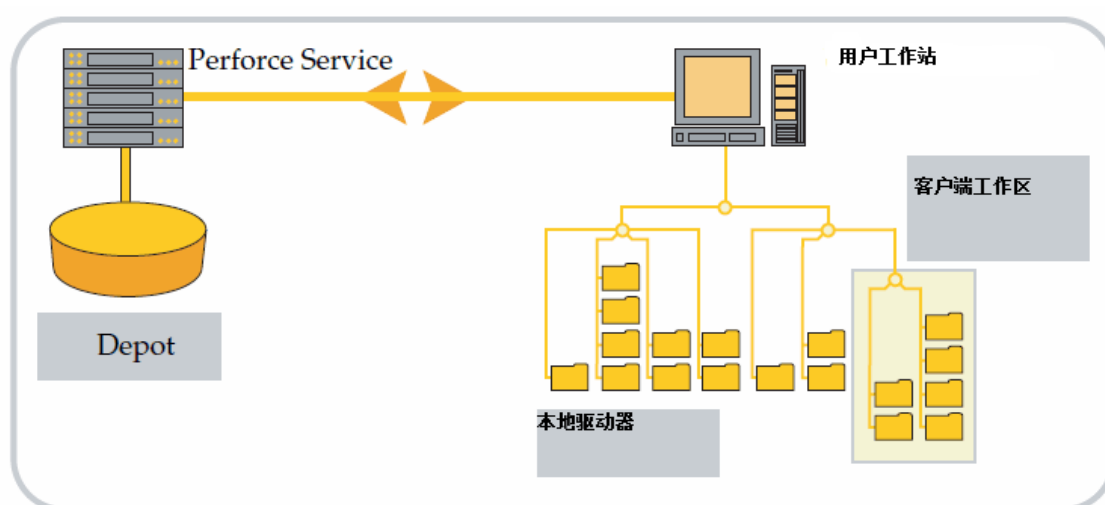
Perforce 应用程序

你运用Perforce应用程序与版本管理服务进行通讯。Perforce应用程序允许你签入签出文件、管理冲突、生成开发分支、错误追踪以及变更请求，等等。

Perforce应用程序包括：

- P4：所有平台上的Perforce命令行客户端。
- P4V：Mac OS X, UNIX, Linux, 以及Windows上的Perforce可视化客户端。
- P4Web：Perforce Web客户端，一个基于浏览器的Perforce界面。
- 集成工具或插件：用于与商业IDE和第三方软件的集成。

使用Perforce，你不必直接在depot中操作文件。而是利用Perforce应用程序来管理工作站上的一个称之为**客户端工作区**的特殊指定区域。工作区存储了depot中部分文件的本地拷贝。



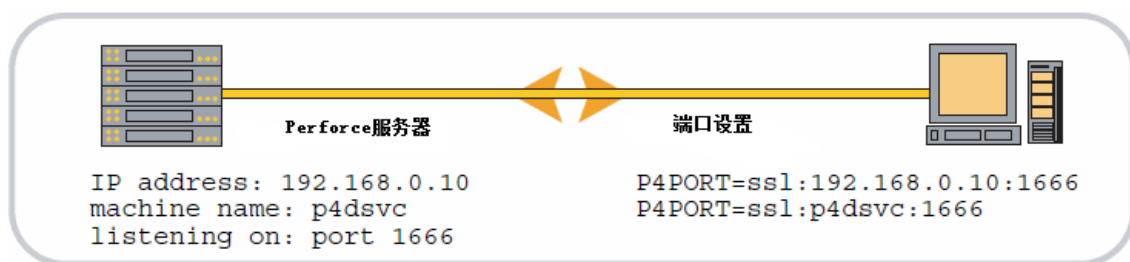
当你把文件读入工作区时，Perforce应用程序从公用文件信息库请求文件。为了把网络流量控制到最低，该服务记录你（和其他用户）读取了哪些文件。Perforce应用程序并不需要保持与版本管理服务的永久连接。

要使用Perforce，你必须配置你的Perforce应用程序使它连接到Perforce服务，在本地硬盘上指定一个区域作为工作区，同时，选择你工作所需的depot文件的子集。

连接到 Perforce 服务

要使用 Perforce，你必须连接到公用的版本管理服务。Perforce 应用程序通过 TCP/IP 与服务通讯，也可以选择使用 SSL 来提高安全性。Perforce 应用程序需要知道与之通讯的 Perforce 信息库的通讯协议、地址和端口。

地址和端口被保存在P4PORT环境变量中。根据你所用的应用程序，配置该变量的过程被称为“**设置端口**”或“**连接到服务**”。



Perforce应用程序的文档和在线帮助包含了如何设置端口的信息。如果你不懂得如何设置端口来连接Perforce服务器，可以咨询你的Perforce管理员。

把 depot 中的文件映射到工作区

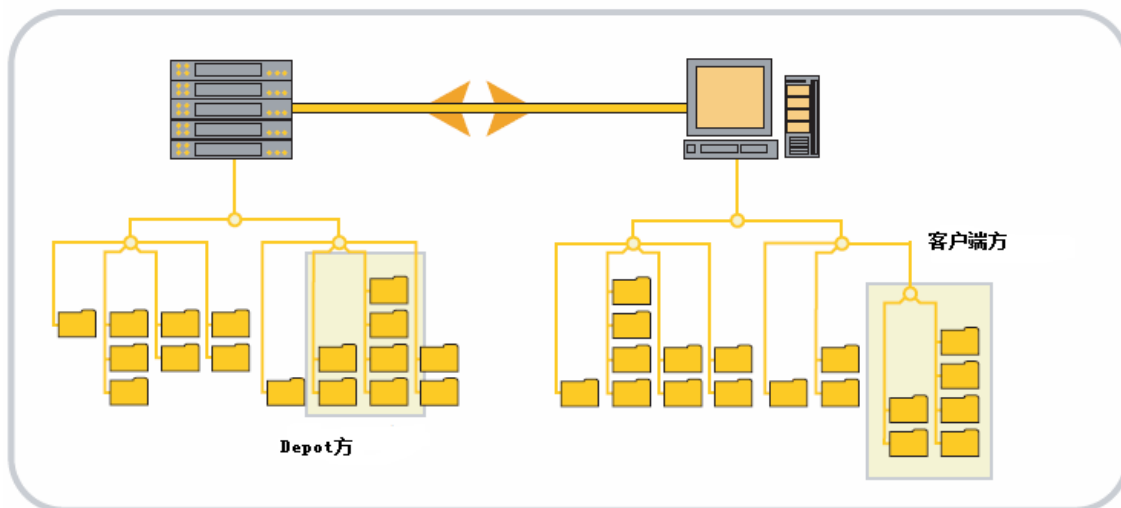
Perforce应用程序在你的本地硬盘的所谓工作区的指定区域管理文件。正如其名，你的工作区就是你进行大部分工作的地方。你可以拥有一个以上的客户端工作区，即使在同一台工作站上。所有工作区的顶层目录被称为**工作区根目录**。

为了控制depot文件在工作区根目录下显示的位置，你必须把公用版本管理服务中的文件和文件夹映射到本地硬盘的相应区域中。该映射构成你的**工作区视图**。

工作区视图：

- 决定depot中的哪些文件可以出现在工作区中。
- 把depot中的文件映射到工作区文件上。

客户端工作区视图包含一条或多条线（**映射**）。工作区视图中的每条线有两个方面：“**depot方**”在depot中指定一个文件子集，“**客户端方**”控制depot方所指定文件在工作区根目录下的存放位置。



创建工作区并没有把任何文件从depot传到你的电脑上。工作区及其视图仅仅定义映射以控制文件传输时depot和工作区的关系。

如果想了解更多关于如何配置映射来定义工作区的话，你可以参照Perforce应用程序的文档和在线帮助。

其他配置选项

工作区的其他选项使你可以控制Perforce各种操作的默认行为。例如，你可以控制跨平台开发上的回车与换行的字符，也可以根据自己的喜好，设置在Perforce上使用的文本编辑器或合并工具。

如果想了解更多关于这些及其他选项的话，你可以参照专门的Perforce应用程序文档。

使用 Perforce

从服务中读取文件

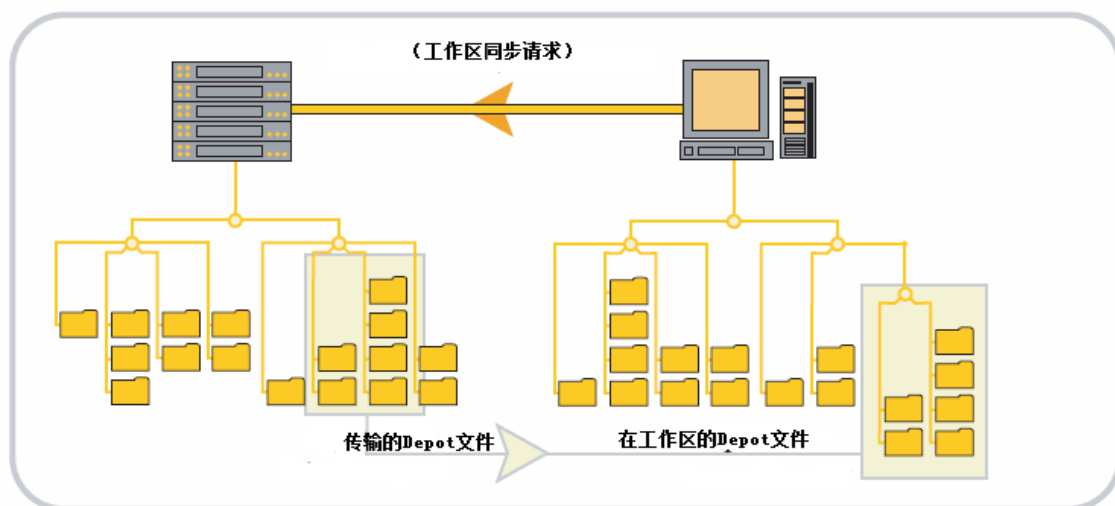
Perforce版本管理服务管理**depot**。Depot储存了系统内的所有文件的所有版本。工作区包含了把depot文件子集映射到工作站某一区域的视图。

为了将depot文件下载到工作区，你必须从版本管理服务读取。在Perforce上，用depot文件更新工作区通常被称为“**同步你的工作区**”。在其他系统上，称之为“更新”、“获取最新版本”或仅称“获取文件”。

同步你的工作区

当你同步你的工作区时，你的Perforce应用程序使用工作区视图来把depot文件映射到你的工作区文件上，根据工作区内容比较结果，增删改工作区文件以使你的工作区保持与depot同步。

同步你的工作区时读取每个文件的最新（“head”）版本的拷贝。（在其他版本管理系统上，也被称之为“tip”版本。）



Perforce应用程序管理工作区上的文件读取权限。在默认状态下，同步到工作区的文件是只读的，在你签出后才变为可写的。

Perforce应用程序也支持各种选项，使你可以获取文件的较早版本，或在特定时点保存的版本，或其他用户已加了用户定义标签的版本集。

Perforce 文件指定

Perforce把depot中的文件组织成目录树，就像一个大的硬盘。Perforce应用程序使用一套规则来定义depot文件与工作区文件的位置关系。无论你设置depot与工作区之间的映射（工作区视图），或用depot文件加载你的工作区，或签入签出文件，这些文件指定规则适合于任何操作系统。

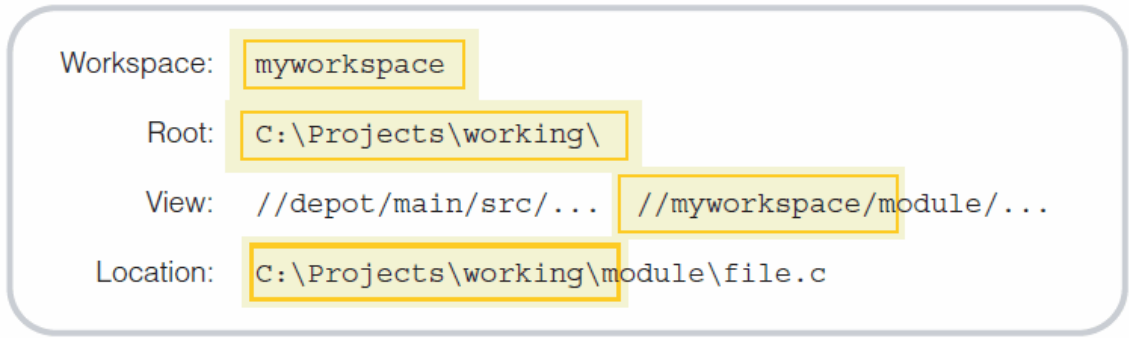
Perforce 语法

当你指定Perforce文件时，你可以指定到客户端工作区根目录的相对路径（“客户端语法”），或指定到depot目录树顶端的相对路径（“depot语法”），也可以指定你本地文件系统上的绝对或相对路径（“本地语法”）。

文件通过客户端语法或depot语法指定时总是以双斜杠(//)开头，后面跟着客户端工作区名或depot名，以及到客户端工作区根目录或depot目录树顶端的相对路径的文件路径名全称。不管本地操作系统用的是什么分隔符，客户端语法以及depot语法上的路径成分都是以单斜杠(/)分隔。

语法	例子
Depot语法	//depot/main/src/file.c
客户端语法	//myworkspace/module/file.c
本地语法	C:\Projects\working\module\file.c

当把depot文件映射到本地硬盘时，工作区名是工作区根目录的别名。



例如，当工作区名为myworkspace，工作区根目录为C:\Projects\working时，通过视图//depot/main/src/... //myworkspace/module/... 将把depot文件//depot/main/src/file.c映射到工作区的C:\Projects\working\module\file.c上。

在视图中使用通配符

你可以在配置Perforce工作区视图时使用这些通配符。

通配符	意义	例子
*	匹配所有字符但不包括目录内的斜杠。	/src/*.c 匹配 /src/file.c 和/src/file2.c, 但不匹配 /src/lib/file.c
...	匹配当前工作目录下的所有文件和子文件夹。	/src/... 匹配 /src下的所有文件和子文件夹
%%1 - %%9	位置指定符，用来替换视图中的部分文件名。	映射 /%%1/%%2/... 到 /%%2/%%1/... 将把 /web/images/file.gif 映射到 /images/web/file.gif

这些通配符也用在命令行客户端上的文件指定。如果想了解更多关于Perforce语法及通配符的话，你可以参照《Perforce命令参考》。

指定文件的特定版本

Perforce用#字符来表示文件版本。Perforce中的文件版本按顺序递增，#1表示第一个版本，依此类推。文件的最近的版本是服务器上该文件的最高版本，被称为**head版本**。你最后同步到工作区的版本被称为**have版本**。文件的零版本被称为**空版本**，不含数据。

当你在Perforce上工作时，开发分支（或称**codeline**）被表示成目录路径。不同分支中的文件有其各自的版本号，从#1开始依次递增。不同分支中的文件祖先被保存在**反映记录**中。如果想了解更多关于Perforce分支的话，你可以参照本手册的[codeline管理]章节。

标识符**file.c#3/4**表示你当前已经把**file.c**的#3版本同步到了工作区，**file.c**的最近版本是#4。与其他版本管理系统不同的是，Perforce没有用难懂的版本号（如**file.c**的**1.2.3**版本）来表示不同开发分支中的文件版本。

语法	指定	备注
file.c#3	file.c的第三个版本	“同步到file.c的第三个版本”
file.c#head	Depot中file.c的最近版本，这是head版本	为了从depot获取文件的最新版本，你把工作区同步成head版本。
file.c#have	最后同步到工作区的file.c的版本，这是have版本。	当你废弃一个文件的变更时，Perforce应用程序把你工作区中的文件恢复到have版本。
file.c#none file.c#0	文件file.c的不存在的或空的版本。	当你用Perforce应用程序从工作区删除文件时，你实际上在工作区中把该文件的版本同步到空版本。

Perforce 语法以及状态栏

head, have和null版本（**#head**, **#have**和**#none**）的语法被用在命令行客户端以及图形应用程序的状态窗口中。详细请参照《P4用户指南》。

支持哪些文件类型？

Perforce文件类型包含6种基本文件类型。

- 文本文件，
- 二进制文件，
- Macintosh上的本地apple文件，
- Mac资源派生，
- 符号链接(symlinks)，
- unicode（以及utf16）文件。

在默认情况下，当某人把文件追加到depot中时，Perforce试图自动判别文件的类型。你通过把文件打开编辑为新的文件类型从而可以改变文件的类型。如果该文件已经打开为编辑状态，你可以重新打开为不同的文件类型。

6种基本文件类型可以附加修饰符（如+w, +x, +k及其它）用来控制诸如锁定行为、工作区内的文件权限或者版本在服务中如何存储。《Perforce命令参考》中包含了文件类型以及可用修饰符的完整列表。

文件操作

Changelist是Perforce中的基本工作单位。所有版本管理系统共通的文件编辑基本操作（如编辑文件、添加文件、删除文件、恢复变更以及签入文件）都在Changelist中执行。

利用 changelist

在你配置完工作区视图并把工作区同步到depot之后，你就可以开始使用Perforce了。在你对工作区中的文件进行操作之前，你必须先在changelist中打开该文件。Changelist包含了一系列文件、它们的版本号、你对这些文件的变更以及描述你所进行工作的注释。

Changelists有两个用途：

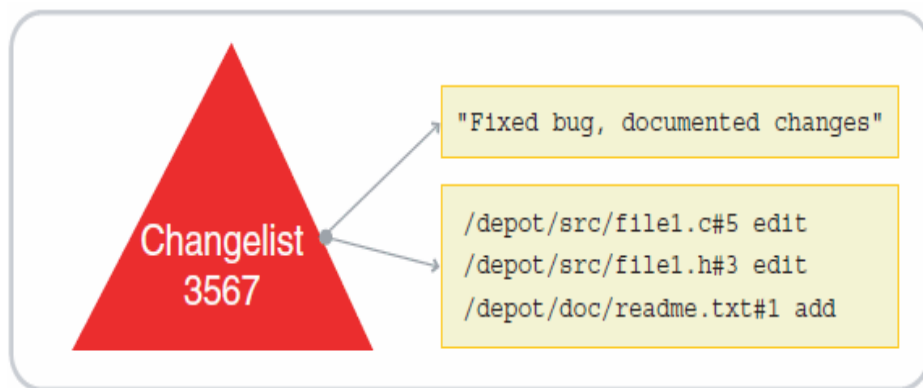
- 通过汇总相关联的文件变更从而把工作组织成逻辑单元。
- 通过把相关联的文件变更一并签入来保证工作的完整性。

如果你在修改某个软件时需要修改三个文件，那你就在一个changelist中打开这三个文件。当你把该changelist签入到depot中时，别的用户可以看到该changelist涉及到三个文件的变更。

Perforce changelist是不可分割的事务。如果一个changelist涉及三个文件，要么这三个文件的变更都被提交到depot，要么都没有被提交。即使在changelist提交期间Perforce客户端与服务端端的连接被中断时，整个提交失败。

Changelist 编号是如何工作的

未提交到depot中去的changelist是**作业中的changelist**。已提交到depot中的包含变更的changelist是**已提交changelist**。每个changelist都有一个**changelist编号**（由Perforce生成）和**changelist注释**（由执行该变更的用户提供）。



当你在Perforce中打开一个文件时，该文件在**默认的changelist**中被打开。当你把文件签入depot时默认changelist被赋予一个changelist编号。你可以中途把你的工作分成若干个待定changelist。当你生成非默认的作业中的changelist时，它被赋予changelist编号（当你把该changelist提交到depot时，该待定changelist将被赋予一个新编号）。

编辑文件

为了编辑文件，你把该文件签出一个changelist。Perforce应用程序使客户端工作区的该文件拷贝变为可写，并通知公用服务你已经把该文件打开为编辑。

为了把你的变更提供给其他用户，你必须把该changelist提交回depot。在你的changelist被提交后，别的用户就能同步他们的工作区并获得你所做变更的副本。

添加新文件

为了添加文件，你在工作区创建文件并在一个changelist中把它标记为追加目的。Perforce应用程序决定文件类型（你可以重设该文件类型），并通知公用服务你准备添加文件。

为了把你的新文件提供给其他用户，你必须把该包含新文件的changelist提交回depot。在你的changelist被提交到depot之后，别的用户就能同步他们的工作区并获得新文件的副本。

删除文件

为了删除文件，你在一个changelist中把它标记为删除目的。文件立即被从工作区中删除。Perforce应用程序通知版本管理服务你准备删除文件，但直到你提交了该changelist之后，该文件才在depot中被标记为已删除。

在你的changelist被提交到depot之后，别的用户看到你的文件被标记为已删除。已删除文件的本地拷贝保留在其他用户的工作区中，直到他们把工作区同步到depot。

被删除的文件实际上并没有从depot中移除。你可以通过同步删除之前的版本到工作区来恢复已删除文件的较旧版本。

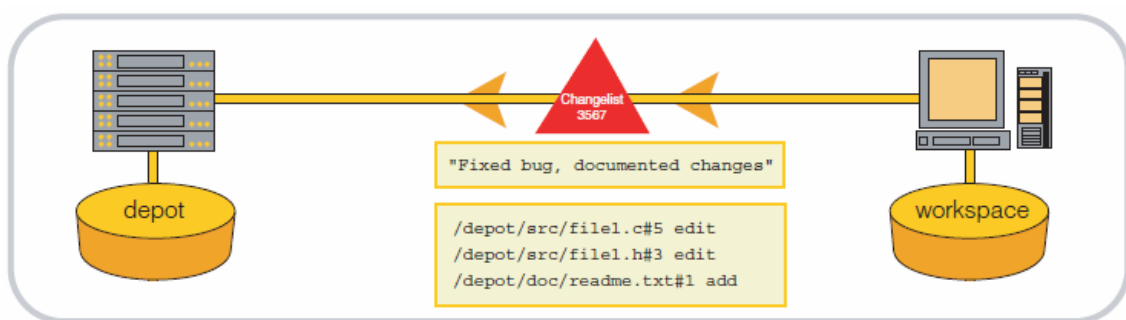
废弃不需要的变更

你可以通过**恢复**文件来废弃你对文件所做的任何变更。恢复文件把该文件从changelist中删除并把你工作区中的文件拷贝恢复到你最后同步到工作区的版本。

如果你恢复一个打开为编辑或标记为删除的文件，你最后同步的文件版本将被恢复到工作区。如果你恢复一个标记为添加的文件，该文件从changelist中删除，但文件的本地拷贝将保留在工作区中。

签入文件

当你对你所做的文件变更感到满意并希望与别人共享你的工作时，你可以通过**提交**该changelist把你的工作签入到depot中。



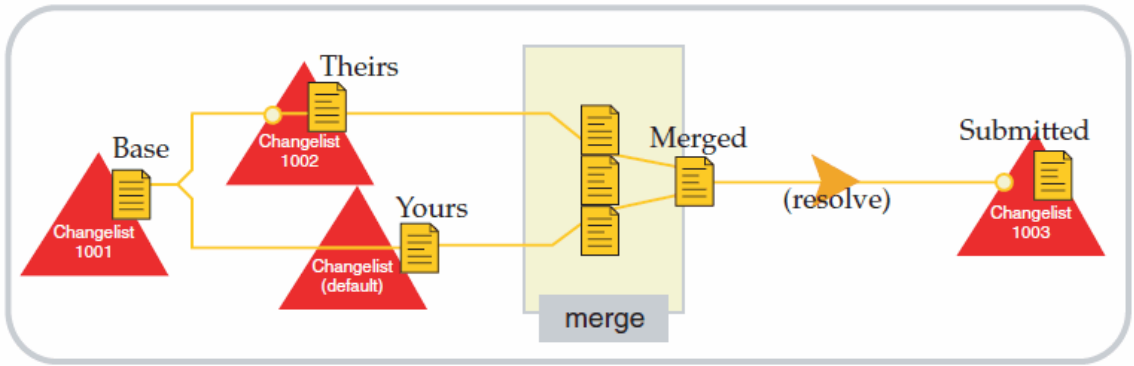
不存在部分提交的changelist。Changelist提交是不可分割的事务，要么changelist中的所有文件都被成功提交，要么都没有被成功提交。

解决冲突

当两个用户同时编辑同一个文件时，他们的变更将发生冲突。如果你的变更与别的用户较早提交的变更产生冲突，Perforce将要求你**解决冲突后重新提交该changelist**。因为Changelist是不

可分割的事务，在你解决冲突之前，changelist中的如何文件的如何变更都不会出现在depot中。

冲突解决的过程允许你选择需要做的事情：是用你的文件覆盖别的用户的文件？或是抛弃你的文件转而选择别的用户的变更？还是把这两个冲突的文件合二为一？根据你的请求，Perforce将对这两个冲突的文本文件以及产生这两个冲突文件的文件进行三路合并。



并行工作

Perforce可以有效的协调团队的并行工作。冲突解决以及三路合并过程使多个用户可以同时在一个文件上工作而互不影响。

三路合并过程有助于解决文本文件的变更冲突，但对图像或编译代码之类的二进制文件不一定有效。如果你所处理文件的合并无效，你可以锁定这些文件以免别的用户的修改影响你的工作。Perforce支持两种类型的文件锁定。你可以用文件锁定来防止文件被签入，也可以以独占方式打开来防止文件被签出：

- 为防止别的用户对你正在处理的文件签入变更，**锁定该文件**。别的用户仍可以签出该文件，但必须等到你提交完变更后他们才能提交影响被锁定文件的changelist。（要允许别的用户在你提交之前提交影响被锁定文件的changelist，必须解除该文件的锁定。）
- 为防止同一文件同时被多个用户签出，使用以**独占方式打开文件类型修饰符+1**。具有+1文件类型修饰符的文件每次只能被一个用户打开。Perforce管理员可以用一个名叫typemap的特殊的表来自动指定某些文件类型为以独占方式打开。

例如，IDE用户可能希望锁定他们正在处理的文件，这样他们就不必转到Perforce来提交他们的作业。处理图像或视频内容的用户可能希望能自动把.gif或.mpg文件列为以独占方式打开。

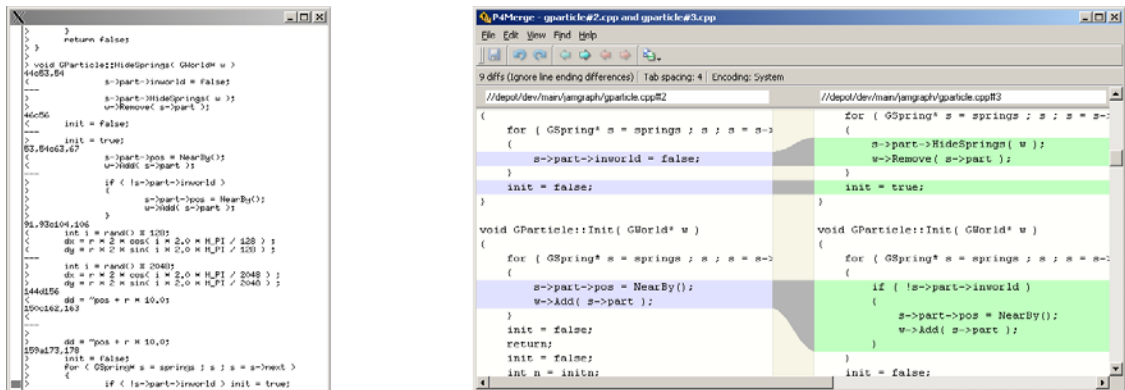
你正在编辑的文件	是否锁定？	意义
File(type)	不锁定	任何人都可以签出文件并提交变更。 如果一个用户提交了对你已打开的文件的变更，在你提交changelist时你必须解决与他们的冲突。
File(type)	锁定	任何人都可以签出文件，但必须等到你提交完变更或解除该文件的锁定后他们才能提交文件的变更。
File(type+1)	不锁定或锁定	每次只能被一个用户打开在changelist中。与锁定的状态无关。别的用户无法提交包含该文件的changelist，因为他们无法签出该文件。

如果你想了解更多关于文件锁定、以独占方式打开文件类型修饰符以及typemap表单的话，你可以参照《Perforce命令参考》以及《系统管理员手册》。

比较文件

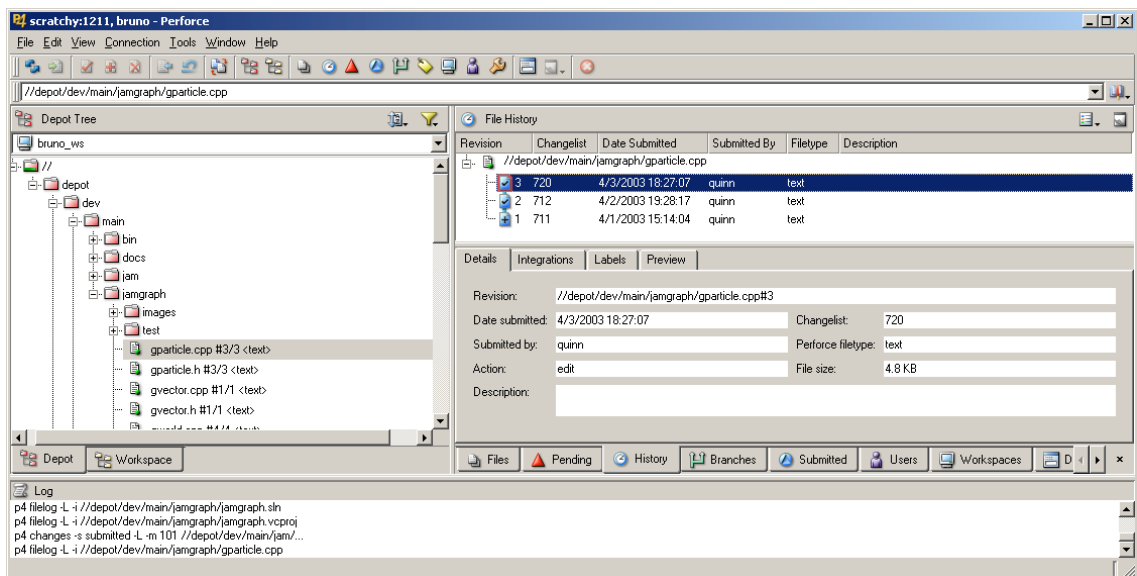
你可以用Perforce来比较同一文件的任何两个版本或depot中的任何两个文件或depot中的文件与工作区中的相应拷贝。

p4 diff和**p4 diffs**命令产生与UNIX和Linux系统中的标准diff命令类似的输出。其它的Perforce客户端程序（包括P4V）含有P4Merge，能提供文件差分的图形化视图。例如：



审查单个文件的变更历史

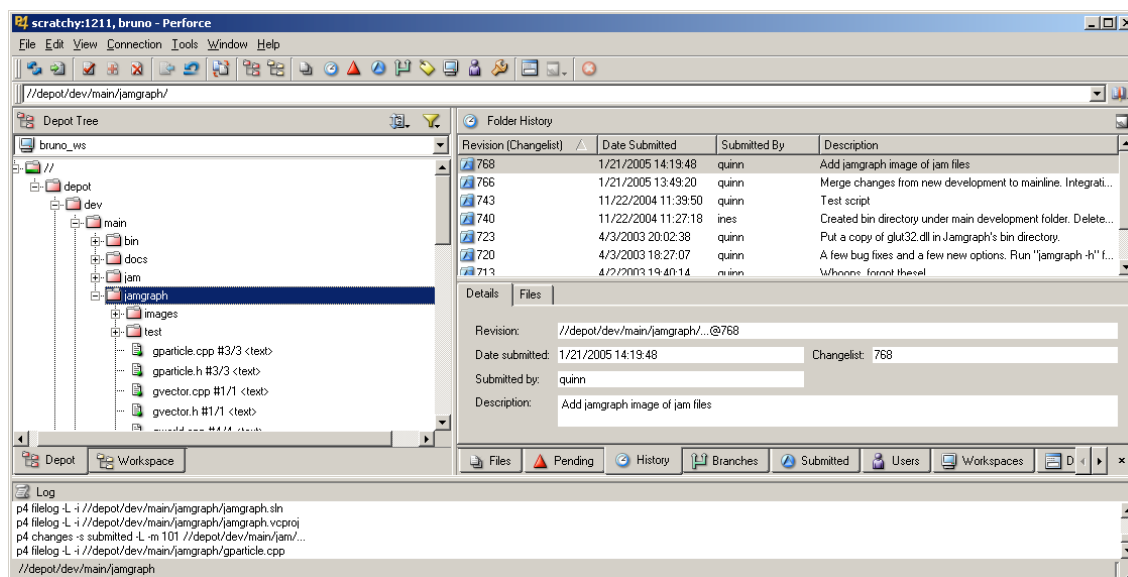
文件的历史用一系列文件版本表示，每个文件一个版本。每个文件版本与一个changelist关联。你可以通过工作区中的版本来比较文件，也可以通过任何存储在depot中的版本来比较。



P4V截图显示depot中保存了//depot/dev/main/jamgraph/gparticle.cpp文件的三个版本。最近的版本是#3，是在changelist 720中被提交的。

审查文件组的变更历史

文件夹的历史用一系列changelist表示。文件夹没有单个的版本号。相反的，每个包含至少一个文件的changelist都被看做文件夹历史的一部分。



P4V截图显示影响`//depot/dev/main/jamgraph`中至少一个文件的最近的changelist是 changelist #768.

Perforce 语法以及状态栏

Perforce拥有语法格式，用来指定某个文件通过提交带编号的changelist而在depot中存在，或是被打上某个易记的标签，或是某个日期和时间点的。这些语法格式(`@changelist`, `@labelname`, 或 `@date`, 或 `#start, end`)通常只用在命令行客户端上，但它们也出现在其他Perforce应用程序的状态窗口中。详情请参照《P4用户指南》。

codeline 管理

codeline是一起演变的关联文件集。为了把相关文件按某个目的归类，比如作为新产品或发布版本，你可以创建分支。为了在分支间复制变更，你可以反映changelist。为了产生某个特定状态的文件的快照，你可以创建标签，或通过指定日期或changelist编号来指定文件集合。

注意：不要混淆Perforce分支与分支映射，后者是定义分支之间关系的式样。

分支基础知识

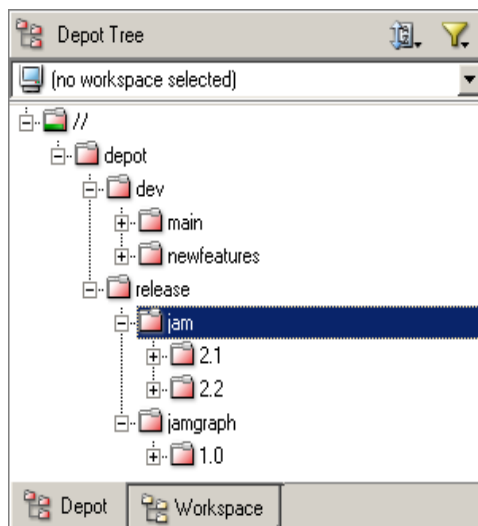
分支是管理两个或更多相关文件集合的变更的方法。Perforce **Inter-File Branching**机制使你在把任何文件集拷贝到depot新位置时，一个文件集的变更可以被复制或**反映**到另一个文件集。新的文件集（或**codeline**）与原有的文件单独开发，但通过**反映**的办法，一个codeline中的变更可以复制到另一个codeline。

大多数版本管理系统都支持某种形式的分支。Perforce的机制是独特的，因为它模拟了用户在没有分支机制可用的情况下创建文件拷贝的模式。

假设你正在没有版本管理的环境中编写程序。你可以发布程序了，你将如何处理代码？很有可能你必须把所有文件拷贝到新的位置。某个文件集成为发布codeline，但该文件集产生了缺陷修正。其他文件是开发文件集，代码的新功能被追加到这些文件上。

Perforce把depot中的文件组织成目录树，就像一个大的硬盘。当你创建了一个新codeline时，它在depot中显示为子文件夹。

例如//depot/dev/main/jam表示正在进行的开发工作，//depot/release/jam/2.1表示发布版本2.1，//depot/release/jam/2.2表示发布版本2.2。



创建 codeline

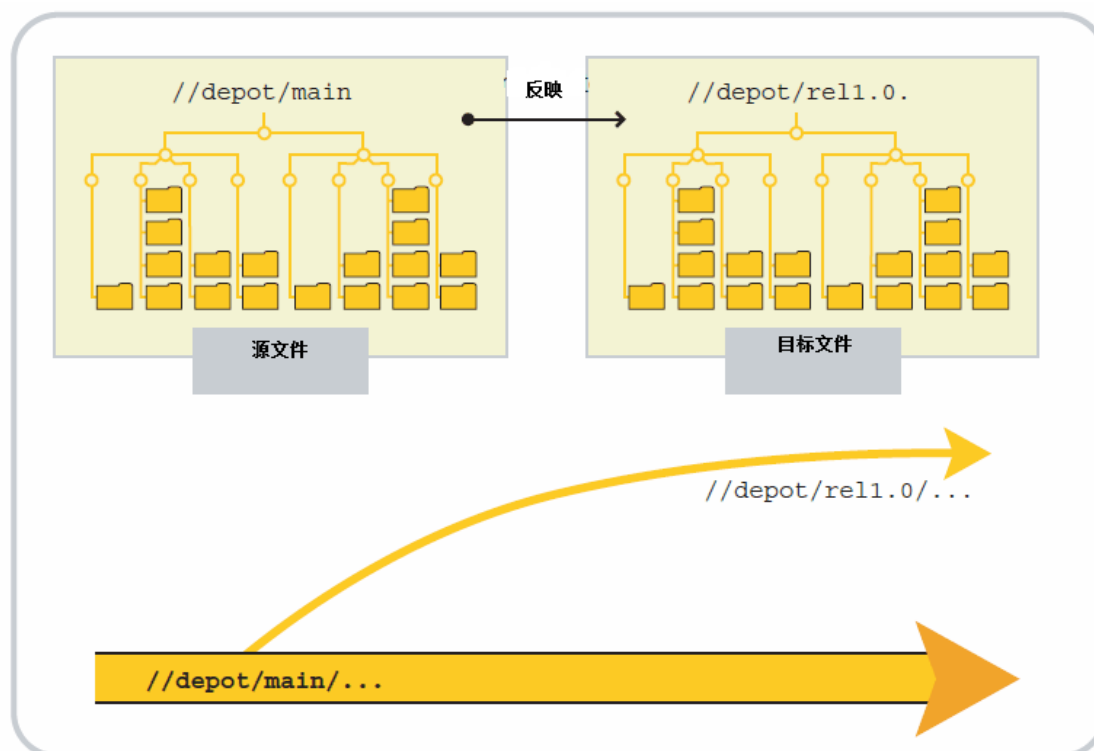
为了创建codeline或开发分支，决定哪些文件属于该分支（**源文件**）并把那些文件**反映**到新的codeline以创建**目标文件**。Perforce服务器在changelist中“**以分支/同步为目的打开目标文件**”。

以**分支/同步**为目的打开文件正象以**追加、编辑或删除**为目的打开文件一样，文件被打开在一个changelist中，你的客户端工作区视图必须包含该目标文件。类似地，在你提交changelist之前，depot没有任何变化。Changelist的原子特性确保当你创建codeline时，它包含了你分支的所有文件。

如果没有版本管理系统，你可以从一个目录拷贝文件到另一个目录来创建分支。相对于拷贝文件并把拷贝追加到depot的新目录，反映的优势是当你把文件从一个codeline反映到另一codeline时，Perforce能把相关文件的联系记录在反映记录中，便于在两个**文件集**之间跟踪和复制变更。

反映还能让Perforce执行文件的“**偷懒拷贝**”。当你分支文件时，版本管理服务实际上并没有保留文件的两个拷贝——它保留源文件以及记录源文件和目标文件关系的指针。偷懒拷贝使分支成

为低负荷操作，版本管理服务不需要记录文件的两个拷贝。



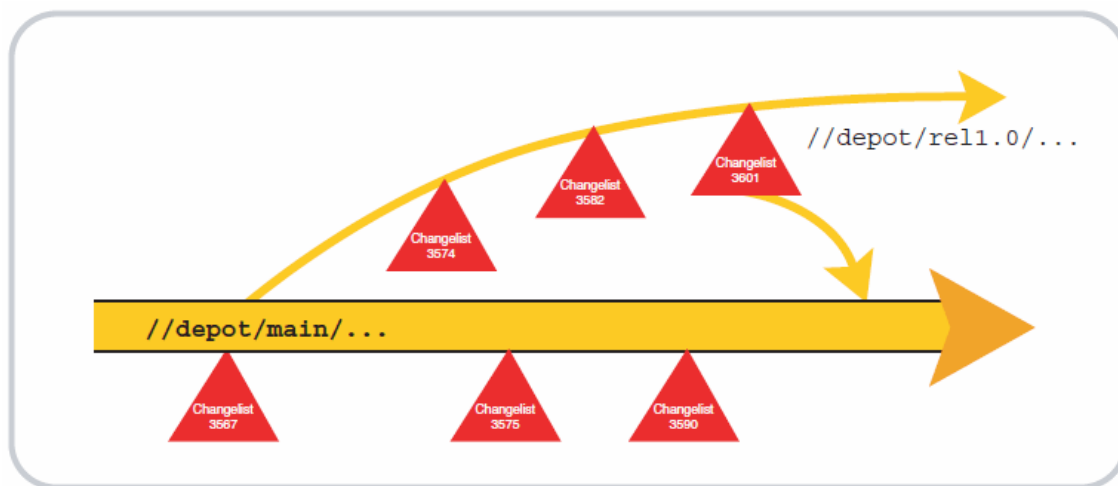
把文件从源codeline反映到目标codeline：

- 目标必须在工作区视图中。
- 源codeline不需要在工作区视图中(尽管你要有读源文件的权限)。
- 通过反映操作，你可以在新的changelist 中把文件打开为分支。
- 通过提交该changelist，你在新codeline中创建文件。
- 当你提交目标文件的changelist时，新codeline中的目标文件的版本号是#1。
- 反映记录使你能审查新codeline中文件的历史，包括它们是通过从源文件反映而创建的。

在 codeline 之间复制变更

你可以用类似于创建codelines的方法通过反映在相关codelines之间复制变更。（创建codeline等同于把整个源文件的一系列变更复制到空的目标文件集。）

当你创建codeline时，目标文件按照定义是空的，你的变更不可能产生冲突。当你在已有的codeline之间复制变更时可能产生冲突，因为源文件和目标文件中的变更可能产生冲突。



如上例所示，**re11.0** codeline是通过changelist 3567把源文件从**//depot/main**分支到目标**//depot/re11.0**而创建的。Changelists 3574, 3582和3601代表发布分支上所做的工作，changelists 3575和3590代表主线上所做的工作。

为了把发布分支上所做的工作复制到主线上，你可以通过解决发布分支和主线上所做变更引起的所有冲突，把**//depot/re11.0**上的源文件反映到**//depot/main**中。

解决 codeline 之间的差别

当你把changelist从源codeline反映到既存的目标codeline时，Perforce将安排一个解决方案，即文件之间的三路合并。

工作区中的目标文件的版本被称为“yours”，depot中的源文件的版本被称为“theirs”。当这些文件的变更没有冲突时，变更能自动被合并。当发生冲突时，你必须选择接受哪个变更。

因为changelist是不可分割的，在提交成功之前你必须解决changelist中的所有文件的冲突。你可以通过三种方法中的一种来解决：

- 自动：在很多场合，你知道你想接受“yours”变更（即工作区中的目标文件的版本）还是“theirs”变更（即depot中的源文件的版本）。不管你接受“yours”或是“theirs”，由于没有合并，这种解决方案被称为**自动解决方案**。
- 接受合并结果：有时，“theirs”中的文件发生变更而不与“yours”产生冲突。在这种情况下，Perforce把该两个文件合并并让你接受合并结果。这样的解决方案被称为安全的**自动解决冲突**（包含合并）。
- 手动合并：最后，有可能“theirs”和“yours”中的相同的行发生变更。这些行将发生冲突。当变更冲突时，Perforce尽可能多地解决差异并产生包含冲突标记的合并文件以待手工解决。你要么在提交之前手工编辑该合并文件，要么接受包含冲突标记的合并文件并在随后的changelist中解决该冲突。

复制复杂分支结构

Perforce提供两种分支机制：**利用文件式样反映**和**利用分支映射反映**。

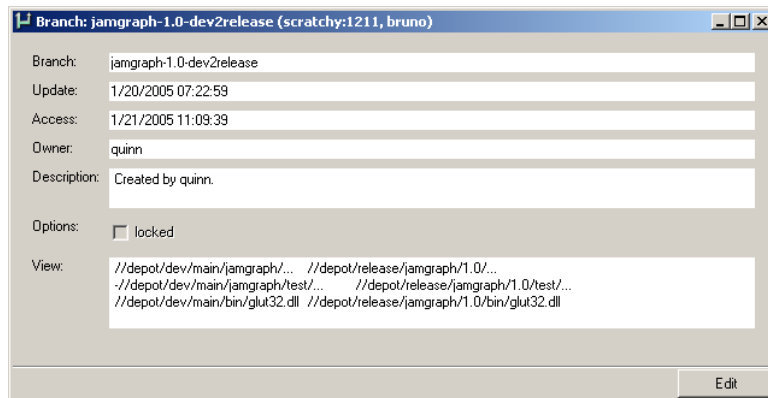
对于简单的分支结构，你可以手动指定源文件和目标文件的路径并利用文件式样把源文件反映

到目标分支上。每次利用文件式样进行分支，你都必须手动指定源和目标codeline。

对于较复杂的分支结构，你可以设置分支映射，使你能可靠地复制最复杂的**分支结构**。分支映射保存一套映射规则（**分支视图**）用来控制源分支中的文件如何反映到目标分支。在你设置了分支映射之后，你就能利用分支映射来执行视图中所指定的所有反映。

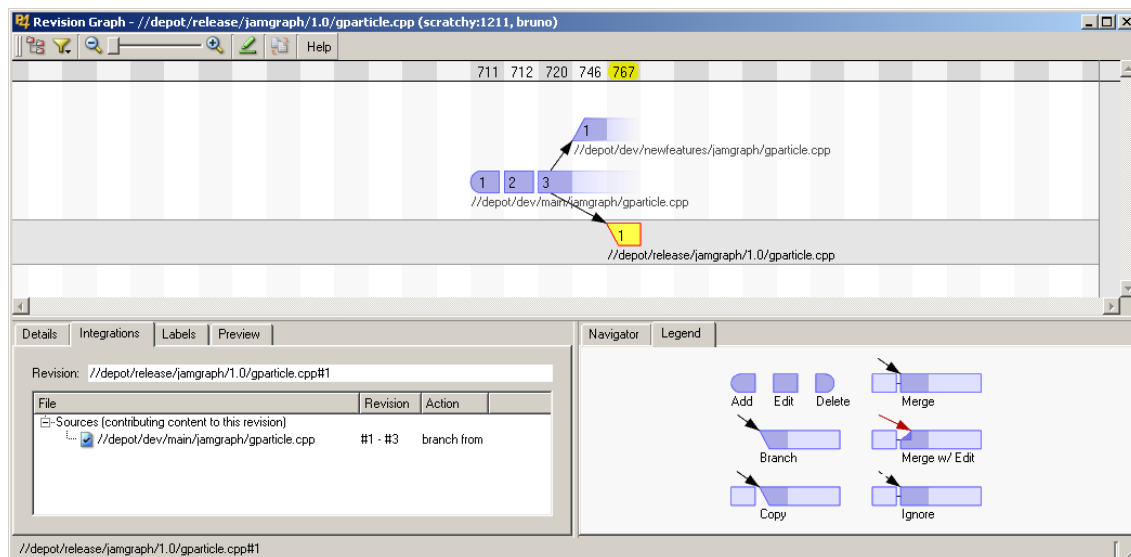
例如，以下截图中的分支映射显示三个映射：

1. 把
//depot/dev/main/jamgraph中的所有文件映射到
//depot/release/jamgraph/1.0目录中。
2. 一个排他映射保证
/jamgraph/test 中的测试工作不从主线拷贝。
3. 一个映射包含共通/bin目录中的DLL (glut32.dll)，与jamgraph项目无关。



记录 codeline 之间的变更历史

P4V的版本图功能是使分支间的文件历史可视化的方便途径。



例子中的截图显示了一个简单的版本图。版本#1到#3所代表的文件的变更被从主线(//depot/dev/main/...) 反映到 发布分支(//depot/release/jamgraph/1.0/...) 和 开发分支(//depot/dev/newfeatures...)。

使用 Streams

Perforce streams是组成项目、codeline以及组件的文件的结构化容器。Perforce命令行客户端以及P4V（Perforce可视化客户端）之类的应用程序为streams提供了广泛的支持。如果想了解使用streams工作的详细信息的话，你可以参照《Perforce命令参考》、`p4 command line help` 和 `P4V online help`。

使用Streams有如下好处：

- 保证对分支的层次化访问。
- 提供一个富有创意的最佳实践分支策略。
- 为Perforce服务提供分支层次的metadata。
- 提供构造代码（稳定性和层次）的标准方法。
- 自动生成客户端工作区视图和分支视图。
- 提供stream结构和状态的引人注目的可视化。
- 使你能组织和观察代码（主体）。
- 提供规则使开发更容易。

Perforce streams为以下四个问题提供解决方案，从而改进了Perforce的默认分支模型。

- **你无法命名分支：**分支是路径的集合。你可以命名分支式样，但分支式样缺少起源和稳定性。创建分支视图难，定义streams容易。
- **没有外部工具，你就无法定义好的分支结构：**Perforce意外地允许你做任何你想做的事。小型安装能免除许多细节，但企业级用户需要符合他们商业和技术流程的准则和最佳实践。Streams确保了层次型分支，使分支之间的非正常关系最小化。
- **复制变更的效率低下：**利用定义明确的streams，用户可以只分支将发生变化的文件并能保证生成有意义的反映历史。
- **变更复制不正确：**默认情况下，你可以无限制地反映文件。Streams保证变更沿着层次流动以及合并和冲突解决正确。

Streams并不**保证**效率和确定性，但一个好的层次结构控制了哪些需要分支并保证变更传递的正确性。

想利用streams工作，你要执行下列步骤：

1. 创建一个stream depot
2. 创建并填充一个mainline stream
3. 分支开发stream和发布stream
4. 合并以及拷贝变更

你（主要）利用下列命令来管理streams：

- `p4 stream`
- `p4 streams`
- `p4 merge`
- `p4 copy`
- `p4 resolve`
- `p4 cstat`

- p4 istat

其他接收stream参数的命令如下:

- p4 branch
- p4 client
- p4 clients
- p4 diff2
- p4 dirs
- p4 integrate
- p4 interchanges

如果想了解命令语法和标示的详细信息的话,你可以参照《Perforce命令参考》或使用**p4 help commandname**命令。你可以用**p4 help streamintro**命令来了解大概。下列章节详细介绍streams相关的任务。

设置结构

首先创建一个stream depot。Stream depots除了以下几点外与本地depots相同:

- 流程受stream式样控制(尽管手工反映可以利用streams定义或编辑时生成的分支视图)。
- Streams只接收从与stream关联的工作区的提交。
- 可以同步和编辑导入文件,但不能提交。你可以整合stream和本地depot的文件。

接着,创建分支层次中的每个stream,定义stream式样,指定stream名、stream双亲和stream类型。默认情况下,stream从其双亲继承文件结构和内容。为了满足进一步的要求,你可以通过定义路径映射来指定映射,该映射重定义stream包含的文件路径。下列章节解释stream类型和视图。

Stream 的类型

Stream类型标志着相对固定。目前支持下列的stream类型:

- **mainline**: 不含双亲的stream。期待从较稳定的streams向下合并。期待从较不稳定的streams向上拷贝。作为stream系统的稳定主干。
- **release**: 较双亲稳定的stream。期待从较稳定的子streams向下合并。不期望从双亲stream向上拷贝。有利于持续的稳定、缺陷修改和发布维护。
- **development**: 较双亲不稳定的stream。期待从双亲streams向下合并。期待从较不稳定的子streams向上拷贝。不期望有更稳定的子streams。有助于长期项目,主要是新功能。
- **task**: 象开发stream一样工作。用于轻型的短期分支。是一种不常见的分支。在使用时,只有与task stream关联的工作区能看见stream中的所有文件。其他工作区只能看到你在task stream中变更了的并拷贝到双亲的文件版本。
- **virtual**: 对双亲stream视图的筛选有用。

为了从更稳定的双亲或子stream中更新stream,需要向下合并;在较不稳定stream更新后,需要把更新拷贝到更稳定的stream。简而言之:向下合并,向上拷贝。

Stream 视图

为配置stream所含文件,你定义**stream视图**。视图由一系列路径组成,你对这些路径赋予控

制它们特性的下列类型：

- **share**：（默认）共享路径中的文件是可同步、可提交且可反映的。共享路径用于变更可以在streams间流动的文件。
- **isolate**：文件可编辑但变更无法反映到别的streams。孤立路径用于保存每日构建和其他产生的文件。
- **import**：文件可以从指定位置读取，但无法编辑或反映。导入路径用于在stream中必须存在但并不工作的第三方组件。导入路径从双亲stream继承视图，除非它的depot位置被显式定义。
- **exclude**：防止双亲stream中的文件成为子stream的一部分。excluded路径中的文件是不可同步、不可提交且不可反映的。

当你指定路径时，在视图的右侧省略目标路径的开头的depot和stream名（除非你正在重映射导入路径）。例如：

`//AlphaTest/main/doc/... doc/...`

有疑问时，用“Development”。

例如：构建stream可能具有如下的映射：

```
share ...
import relbin/... //Ace/REL2.1/bin/...
import qt4/... //import/Qt4.0/...
import jam/... //depot/main/jam/src/...
```

因为需要访问源代码和库，但并不会把变更签入那些路径。注意导入路径重新映射被导入文件的位置。

填充 streams

在定义了mainline stream后，你通过追加文件或反映文件，把它们拷贝到较不稳定的子streams或合并到较稳定的双亲从而复制该stream。

复制变更

一般的，你先定义并复制mainline，然后在mainline下方创建开发streams。你通过从mainline合并来保持开发streams为最新。当开发工作稳定时，你把它拷贝到主线。当需要准备发表版本时，你可以创建发布stream并在其中使发表版本稳定，然后作为发布stream的子stream创建开发stream并在其中修改缺陷。

学习更多分支知识

虽然Perforce的分支机制相对简单，但分支的理论可以相对复杂。什么时候创建分支？代码变更在什么时点从一个codeline复制到另一个codeline？谁负责执行合并？这些问题对任何版本管理系统而言都是共通的，但答案并不简单。

后继工作

任务并缺陷追踪

Perforce包含一个叫做jobs的基本缺陷跟踪系统。Perforce job主要负责任务描述，诸如缺陷修改或变更请求之类。Perforce的任务跟踪机制允许你把一个或多个任务链接到changelist上以实现任务所指定的工作。关联job和changelist有助于团队了解任务是否已完成，何时完成的，谁负责该任务，哪些文件版本跟该任务有关。当changelist被提交时，与changelist关联的job被标记为closed。

可以定制job系统所追踪的信息内容和类型；Perforce管理员可以添加、修改或删除Perforce作业所用的字段。详情请参照《系统管理员指南》。

可以容易地把Perforce与第三方缺陷跟踪和工作流系统整合。P4DTG（Perforce缺陷跟踪平台）是包含图形化配置编辑器和复制引擎的综合平台。详情请参照：

http://www.perforce.com/product/components/defect_tracking_gateway

给文件加标签

Perforce标签是加上标签的文件版本的集合，它允许你在工作区内复制特定的文件组。标签与changelist不同的是，changelist编号代表changelist被提交时depot内的所有文件的状态。标签可以用来标识任意的文件组，即使当那些文件版本代表工作在多个changelist中被提交时。

Changelist与标签的另一个不同点是，changelists编号由Perforce指定，而标签由用户命名。例如，你可能想给组成特定发布版本的文件加上**rel12.1**标签。以后，你可以更新带有**rel12.1**标签的版本以反映之后的changelist中所执行的修改，并通过把工作区同步为该标签来把所有加标签的版本读入该工作区。

详情请参照《P4 用户指南》。

编辑器和合并工具

Perforce应用程序带有选项，允许你指定所喜欢的文本编辑器或合并工具。例如，命令行客户端用环境变量**P4EDITOR**和**P4MERGE**来调用所喜欢的编辑器或合并工具。

详情请参照Perforce应用程序的文档。

保护和权限

Perforce提供权限表，以防未经许可的或无意的访问。权限表决定哪些Perforce命令可以执行，在哪些文件上由谁在哪个工作站执行。Perforce管理员可以在Perforce命令行客户端上利用**p4 protect**命令或者用Perforce管理工具**P4Admin**来设置权限。

详情请参照《系统管理员指南》。

用户和许可证

该共享版本管理服务根据它所支持的用户数而获得许可。该许可证信息保存在服务器根目录

中的叫做**license**的文件中。**license**文件是由Perforce软件提供的普通文本文件。如果没有**license**文件, Perforce版本管理服务限制为20个用户/20个客户端工作区(文件数无限制),或用户数和工作区数量无限制(但文件数限制在1000个以内)。

详情请参照《系统管理员指南》或联系技术支持。

从哪里可以更多了解 Perforce

为了从所有Perforce应用程序中获得在线帮助:

- 在Perforce应用程序的图形界面中利用帮助菜单。
- 在命令行客户端的命令行上输入p4 help

可以从以下网址获得Perforce文档:

http://www.perforce.com/documentation/perforce_technical_documentation

Perforce知识库: 可以从以下网址获得论文的完整列表:

<http://kb.perforce.com/Technotes>

Perforce论坛是用户提问并接收其他用户信息的场所:

<http://forums.perforce.com/>

下面的perforce用户邮件列表被反映在论坛上:

<http://maillist.perforce.com/mailman/listinfo/perforce-user>

Perforce支持人员通过email或电话支持。

Perforce还提供培训、咨询和其他专业服务。详情请参照:

<http://www.perforce.com/support-services>

Perforce中国合作伙伴, 上海龙智数码科技有限公司 (<http://www.shdsd.com>) 提供本地化的Perforce咨询、支持、培训等服务。可以通过以下方式取得联系:

电话: 021-6180-6285*8003

邮箱: perforce@shdsd.com

或者关注 [新浪微博](#)、[博客](#):

[@Perforce中国授权合作伙伴](#)

随时获取Perforce相关的最新咨询。