# Web Content Management with Perforce

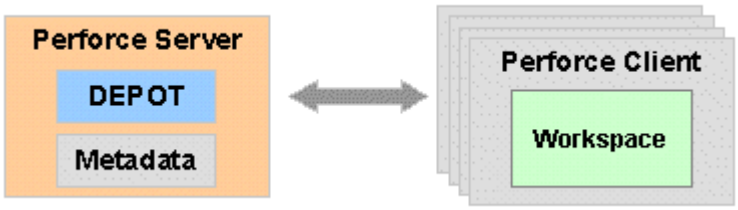
**By Laura Wingerd, Perforce Software**

Presented at the
**1999 Perforce User Conference**

*Perforce is recognized as the fast, low-overhead, high-throughput solution in software configuration management (SCM). What's not as obvious is how Perforce solves the problem of web content management (WCM). Perforce is used in a wide range of WCM applications by: organizations using an intranet for internal documentation; companies whose product is web content, not software; and individuals, companies, and organizations with external web sites. This paper surveys the Perforce deployment models currently in use for web content management, and identifies the features that make Perforce a suitable WCM solution..*

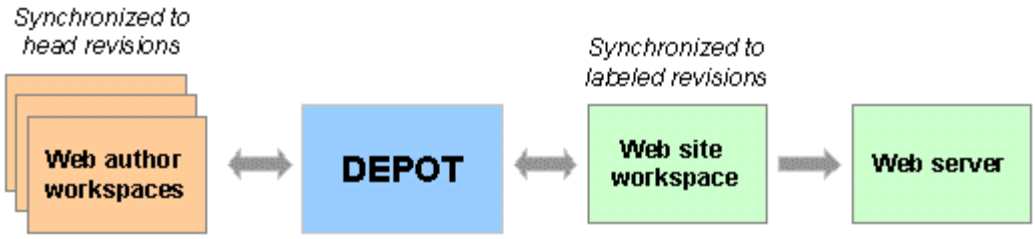
## 1. How Perforce Works

Perforce uses a client/server architecture. The Perforce *server* maintains a repository of versioned files (the *depot*) and a database of SCM information (the *metadata*). A Perforce *client* is any user or application that communicates with the server. A single Perforce server can support a large number of local and geographically distributed clients. Perforce clients get files from the depot, open files to work on them, and submit changed files back to the depot. The files are stored locally in the client *workspaces*.




## 2. A Simple WCM Approach

Web servers read files from a filesystem, and Perforce client workspaces can be mapped to any filesystem. The simplest implementation of Perforce as a web content manager is to establish a dedicated client workspace that *is* the web site. Web authors work in their own workspaces and submit web pages to the depot. A service or daemon can be set up to synchronize the dedicated web site workspace from the depot every few minutes.




This allows web authors concurrent yet controlled access to web pages, while insulating the web site itself from any agent other than either the service that gets files from the depot or the web server that reads and/or executes the files. Because Perforce is managing the web site files, these benefits automatically accrue:

- Files submitted by a web author as one unit of work (a changelist) appear on the web site all at once. There is no possibility that only half of a web author's intended changes make it to the web site.
- Web authors can use Perforce reporting features to tell exactly which file versions are on the web site, which versions are in their own workspaces, and which versions are in each others' workspaces. Complete file histories are also readily available.
- Perforce provides a straightforward merge tool that is invoked if the same file has been modified by more than one web author. This obviates the need to lock files, because it's faster and easier for Author B to merge in Author A's submitted changes in the course of his or her work than it is for Author B to wait until Author A is finished before starting work on the file.
- Perforce passwords and depot protections can be used to fine-tune access permissions authors have on files.
- Web authors can work entirely inside a firewall. Aside from the web server, the only agent that needs extramural access to the web site filesystem is the service that synchronizes the files.

## 3. Reviewing Before Publishing

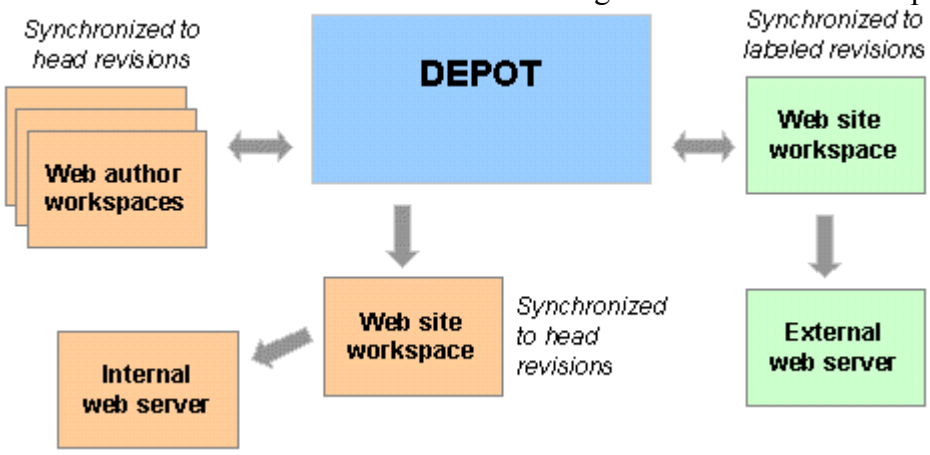
In the simple model described above, an author's changes are visible to web browsers nearly as soon as his files are submitted, depending on how frequently the web site is synchronized from the depot. This leaves little time for anyone else to review his changes before they are "published." While acceptable for an intranet, this model would be unsatisfactory for an external web site. Either of two Perforce mechanisms can be used to assure pages do not get published until they are reviewed.

### 3.1 Publishing with A Label

The first is by labelling the revisions suitable for publication. The web site is synchronized to the labelled revisions instead of to the head revisions. Reviewers

synchronize their workspaces to head revisions, and after they approve files, the web master can apply the web site label to the newly approved files. The next web site synchronization then gets the approved files.

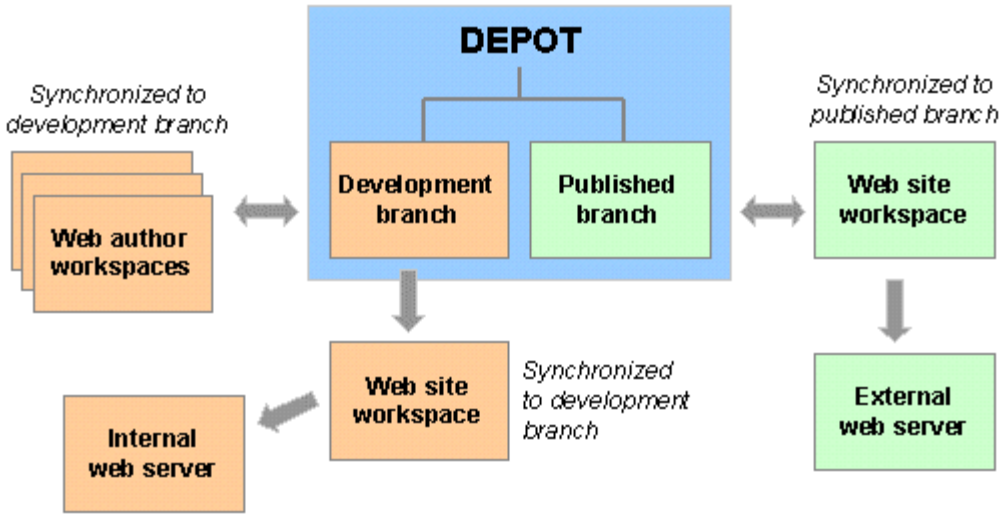
In order to preview how the behavior of web content is handled by a web server, two web site workspaces can be established, one synchronized to head revisions and used by an internal test web server, and one synchronized to published revisions and used by the external web server. Reviewers can direct their browsers to the internal web server to test changes to links and CGI scripts.




The advantage of publishing with a label is that it's easy to understand and practically effortless to set up. However, Perforce does not version labels. As a consequence, there is no automatic way to audit state changes of the external web site (although it would be fairly simple to extend the web site synchronization script to log its activities).

### 3.2 Publishing with A Branch

A second, more sophisticated approach to supporting pre-publication review is to use branches. Authors work on files in a "development" branch. Reviewers proofread and test files in the development branch, and the web master propagates approved changes to a "published" branch. Perforce client view specifications control which branch is synchronized to each web site; the external web site workspace views files from the published branch, and the internal web site workspace views files from the development branch. Because the external web site workspace is continually synchronized to the head revisions of the published branch, the Perforce metadata about the published branch provide a complete inventory *and history* of the web site.



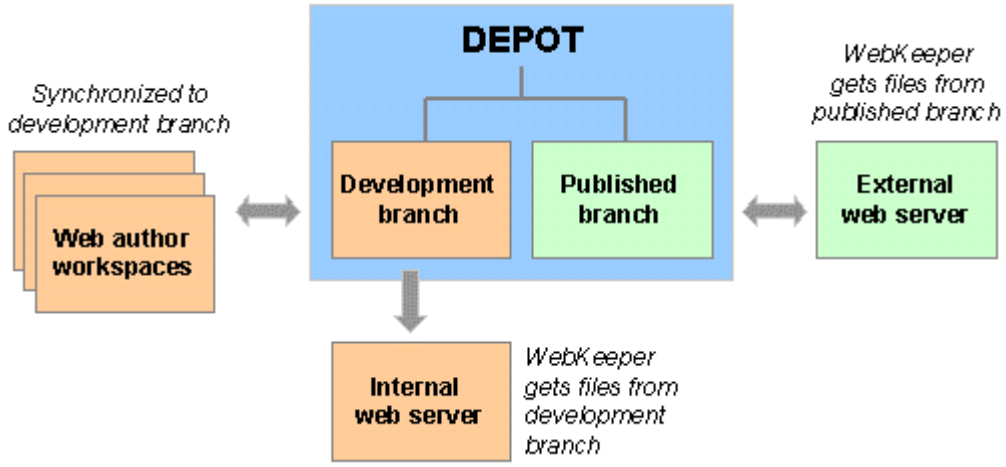

Perforce uses an innovative underlying mechanism called InterFile Branching™, which has these characteristics:

- Perforce branches look like directories -- there is no mysterious branching on version numbers or deferred per-file branching. Authors and web masters can clearly see the two branches in the depot, and examine the contents and metadata of each.
- Perforce tracks propagation of changelists from one branch to another. So an author who has submitted a particular set of files as one changelist can tell with a simple reporting command if her changelist has been propagated to the published branch. A web master can tell with a simple command which files still need propagating to the published branch.
- Files will always be propagated as virtual copies to the published branch. (Because no development takes place there, no merging is necessary). In other words, the published branch occupies no space in the Perforce depot.

## 4. Live Depot Access

The models described so far depend on a service or daemon that periodically synchronizes the dedicated web site workspace with depot files. An alternative to using a web site workspace is to install the Perforce WebKeeper module in the web server itself. This allows the web server to fetch files directly from the Perforce depot. WebKeeper is available as a plug-in to the Apache HTTP server.

As authors submit files into the depot, the files become available to browsers immediately. The live depot access provided by WebKeeper works well with the pre-publication review branch model described above. WebKeeper aliases can be used to select files in either the development branch when accessed by the internal web server or in the published branched when accessed by the external web server.




For web sites where "7x24" availability is not a requirement, WebKeeper saves a small amount of setup and administration by eliminating the need for web site workspaces and synchronization programs. However, WebKeeper does rely on the Perforce server in order to access Perforce-managed files. So, while the Perforce server is generally quite reliable, high-availability web sites are better off using the dedicated workspace model instead of WebKeeper. With a dedicated web site workspace, pauses in the Perforce server (for checkpoints or upgrades, for example) do not impact web site availability.

## 5. Compatibility with Web Tools

Because Perforce client workspaces can be mapped to any filesystem, there is no restriction on the tools or environments authors can use to create web files. Perforce has some features that integrate well with certain web tools:

- Tools which provide menu bar customization can use Perforce's command line interface for source control operations.
- Some Windows tools support the Microsoft Source Code Control (SCC) interface. Perforce provides a Windows SCC DLL that allows tools to be configured to use Perforce for source control. This permits authors to access Perforce operations from within the authoring environment itself.
- When authoring tools do not provide a source control interface, Perforce client commands can be used directly to manage files in a workspace. An author uses Perforce commands to synchronize files to a workspace, open files to work on them, and then uses the authoring tool to make modifications to the files. When modifications are complete in the workspace, the author uses Perforce commands to submit the files to the depot.
- Many web authoring tools provide facilities to put files on directly onto web sites, and are actually designed to do this by default. This facility is not needed when Perforce is used for WCM -- in fact, Perforce saves the administrative overhead of giving authors network access to the web server machine. Authors simply use

Perforce to fill their workspaces, then use the authoring tool in "local" mode to modify workspace files. When they submit files to the Perforce depot, their files become available to the web site.

- For the case where authoring tools create or modify files unexpectedly, Perforce client commands can be used to detect new and changed files in the workspace. This allows authors to put *all* files involved in the web site under source control.
- When web files are generated from source files, Perforce can be used to track files that need regeneration. Perforce's InterFile Branching™ allows files to be associated by filename; HTML files, for example, can be associated with XML files from which they are generated. As new XML files are submitted, Perforce reporting commands can be used to identify which HTML files need regenerating.
- Perforce can handle a variety of file types (text, binary, etc.). Perforce's three-way merge tool can be used to integrate changes when more than one author has modified the same text file. For non-text files, Perforce can be configured to use third-party merge tools.

## 6. Conclusion

By far, the majority of Perforce customers are software development organizations. An added benefit of Perforce is that once it is installed for software development it can be used for web content management at no additional cost, and with very little administrative effort. However, Perforce is just as suitable for web content management independent of software development. As the models described above illustrate, Perforce can provide complete, unobtrusive control of web content evolution and distribution in a variety of implementations.

---

## Appendix: Relevant Perforce Features

Some additional features of Perforce are particularly relevant for WCM purposes:

- The operation that synchronizes the workspace with the depot (called "sync") only transfers files when needed. That is, the sync operation can be invoked as often as desired, but the server only sends files to the workspace if the workspace versions are no longer current. Therefore, an automated service or daemon that synchronizes a web site workspace every few minutes is very efficient -- if nothing has changed in the depot, no data transfer takes place.
- When a user submits a collection of changed files, Perforce numbers and records an atomic transaction called a *changelist*. Perforce enforces the integrity of each changelist so that even in the face of network or system problems, a user can never unintentionally end up with a partially submitted unit of work. Each changelist number can subsequently be used to identify not only the files and revisions that were affected, but the state of *any* depot file at the moment the changelist was submitted.

- File versions can be represented literally or symbolically. E.g., "foo.html#12" is literally the twelfth version of foo.html, whereas "foo.html@reviewed" is the foo.html version identified by the symbol "reviewed." Labels are typically used as symbols, but client workspace names and changelist numbers can be used as well. For example, if a web site workspace is named "website", then "foo.html@website" indicates the version of foo.html that is on the web site.
- Perforce uses TCP/IP for communication between client and server (that is, it does not rely on NFS or any other file sharing system), and provides optional data compression for file transfer and depot storage.
- Perforce supports clients on a wide range of platforms, including Macintosh, Windows 95/NT, Windows NT Alpha, IBM OS/2, IBM OS/390OE, Alpha VMS, VAX OpenVMS, and nearly all Unixes.