HelixCore

# P4GT for Maya User Guide

2020.1

*June 2021*

PERFORCE

www.perforce.com

Copyright © 2017-2021 Perforce Software, Inc..

All rights reserved.

All software and documentation of Perforce Software, Inc. is available from www.perforce.com. You can download and use Perforce programs, but you can not sell or redistribute them. You can download, print, copy, edit, and redistribute the documentation, but you can not sell it, or sell any documentation derived from it. You can 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.

Perforce assumes no responsibility or liability for any errors or inaccuracies that might 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.

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

Any additional software included within Perforce is listed in License Statements.

# Contents


**About Perforce Helix Core server and P4GT for Maya** ..... **4**

Syntax conventions ..... 4

Feedback ..... 5

Other documentation ..... 5

**Load the Perforce Helix Core menu in Maya** ..... **6**

**Get started with Perforce Helix Core server** ..... **7**

**Use P4GT** ..... **8**

Perforce Helix Core menu overview ..... 8

Add files to Perforce Helix server ..... 9

Retrieve files from the depot ..... 9

Check files out of Perforce Helix server ..... 10

Check files into Perforce Helix server ..... 10

Work offline ..... 11

Configure P4GT ..... 11

Check In/Check Out tab ..... 11

Open/Close tab ..... 12

**Troubleshoot P4GT** ..... **14**

Troubleshoot adding files ..... 14

Troubleshoot checking out files ..... 14

**Glossary** ..... **15**

# About Perforce Helix Core server and P4GT for Maya

Perforce Helix Core server, also referred to as Helix server, is a software configuration management tool. The files managed by Perforce Helix server reside in a depot. To work on files, you open and edit them in your client workspace. When you're done, you check in changed files, using the available Perforce Helix server menu options. The depot keeps track of all the current and previous revisions of a file.

This guide tells you how to use P4GT, the Helix Plugin for Graphical Tools. P4GT integrates Helix Core server into drawing tools, including Autodesk 3ds Max, Autodesk Maya, Autodesk Softimage, and Adobe Photoshop. It is intended for anyone using P4GT to perform version control tasks with Helix server.

## Syntax conventions

Helix documentation uses the following syntax conventions to describe command line syntax.

| Notation | Meaning |
|---|---|
| `literal` | Must be used in the command exactly as shown. |
| *italics* | A parameter for which you must supply specific information. For example, for a *serverid* parameter, supply the ID of the server. |
| `-a -b` | Both *a* or *b* are required. |
| `{-a \| -b}` | Either *a* or *b* is required. Omit the curly braces when you compose the command. |
| `[-a -b]` | Any combination of the enclosed elements is optional. None is also optional. Omit the brackets when you compose the command. |
| `[-a \| -b]` | Any one of the enclosed elements is optional. None is also optional. Omit the brackets when you compose the command. |
| `...` | Previous argument can be repeated.<br>▪ `p4 [g-opts] streamlog [ -l -L -t -m max ]` ***`stream1`*** `...`<br>means `1` or more stream arguments separated by a space<br>▪ See also the use on `...` in Command alias syntax in the *Helix Core P4 Command Reference*<br>**Tip**<br>`...` has a different meaning for directories. See Wildcards in the *Helix Core P4 Command Reference*. |

# Feedback

How can we improve this manual? Email us at manual@perforce.com.

# Other documentation

See https://www.perforce.com/support/self-service-resources/documentation.

> **Tip**
> You can also search for Support articles in the Perforce Knowledgebase.

# Load the Perforce Helix Core menu in Maya

After installing P4GT, you must configure Maya to load the Helix Core menu. You do this by enabling the P4GT plug-in.

**To load the P4GT plug-in in Maya:**

1. In Maya, select **Windows > Settings/Preferences > Plug-in Manager**.
2. In the **Plug-in Manager** dialog, scroll down to locate **P4GT-Maya-2020_x64.mll**.
3. Select **Loaded**; then select **Auto-Load**.
4. Save your settings.

# Get started with Perforce Helix Core server

To start using Perforce Helix Core server, you need to configure your client computer to access the Helix server. For more information, see the Configure clients chapter in the *Helix Core Server User Guide.*

To view Helix Core menus in Maya, you first need to enable the P4GT plugin.

P4GT offers access to a number of Helix server commands. Use the Helix Core menu to:

- Add files to the Helix server depot.
- Check files into and out of the depot.
- Lock a file while you have it checked out, preventing others from modifying it.
- Undo or revert your add or check out.

  > **Note**
  > When you revert a checked-out file, all changes you made after check-out are discarded and the file is restored to the last checked-in version (which is not necessarily the last version you saved).

- View detailed Helix server and file information, including revision history, file status, and connection information.

To learn more about these commands, see the Helix Core menu overview.

**To uninstall P4GT**

Do one of the following:

- Run the P4GT installer and select **Uninstall**.
- Use the appropriate Windows control panel (for example, **Add or Remove Programs**), select Helix Plugin for Graphical Tools, and uninstall it.

# Use P4GT

This section includes the following topics:


**Perforce Helix Core menu overview** ..... **8**
**Add files to Perforce Helix server** ..... **9**
**Retrieve files from the depot** ..... **9**
**Check files out of Perforce Helix server** ..... **10**
**Check files into Perforce Helix server** ..... **10**
**Work offline** ..... **11**
**Configure P4GT** ..... **11**
Check In/Check Out tab ..... 11
Open/Close tab ..... 12


## Perforce Helix Core menu overview

The Perforce Helix Core menu contains the following items:

| Perforce Helix Core menu option | Description |
|---|---|
| **Add to Perforce** | Adds the file that you are currently working on to Perforce Helix Core. For more information, see Adding files to Perforce. |
| **Check Out** | Checks a file out of Perforce Helix Core so you can edit it. For more information, see Checking files out of Perforce. |
| **Check In** | Checks files into Perforce Helix Core, to store your changes in the depot. For more information, see Checking files in to Perforce. |
| **Lock/Unlock File** | Prevents other users from checking in a file while you have it open, to prevent conflicts. |
| **Undo Add/Check Out** | Discards any changes that you have made to the files in your workspace. This operation is known as reverting a file. |
| **Results of Last Command** | Displays the results of the last Perforce Helix Core command that you performed. |
| **File Status** | Displays the status of the current file, including the file name and path, as well as Perforce information. |
| **File History** | Displays the current file's name and path, as well as the past revisions of the file, and the changelists associated with it. |

| Perforce Helix Core menu option | Description |
|---|---|
| **Checked-out Files** | Displays a list of all of the files that you currently have checked out of Perforce Helix Core. |
| **Perforce Information** | Displays your Perforce connection information. For more information about these fields, see the Perforce documentation portal. |
| **Get Latest Revision from Perforce** | Enables you to browse and open files that are stored in the depot (as opposed to your workspace). |
| **Options** | Enables you to set P4GT options. For more information on these options, see Configuring P4GT. |
| **Help** | Displays the online help. |
| **About P4GT** | Displays information about the version of P4GT that you are running. |
| **Disconnect from Server/Connect to Server** | Deactivate/activate Perforce Helix Core operations. |

## Add files to Perforce Helix server

To add files to Perforce Helix server, the files that you want to add must be located in your client workspace. If the files are not in your client workspace, P4GT displays an error message and the files are not added.

**To add files to Perforce Helix server:**

1. Select **Perforce > Add to** Perforce.
2. If prompted to check the files into Perforce Helix server, enter a description of your changes.
3. Click **Submit**. P4GT confirms that your files have been successfully added.
4. Click **OK**.

For troubleshooting help, see Troubleshooting: Adding files.

## Retrieve files from the depot

The depot is the central repository for files under Helix server control. Files that you and your coworkers create and modify are stored in the depot when you check the files in.

**To retrieve a copy of a file in the depot:**

1. Select **Perforce > Get Latest Revision from** Perforce, browse to the desired file, and click **Open**. The file is copied to your workspace and opened.
2. If you want to change the file, choose **Perforce > Check out**.

To view an older version of an open file, select **Perforce > File History**, click the desired version, and then click **View**. To copy the older version to your workspace, click **Sync**.

## Check files out of Perforce Helix server

To work on a file that has been checked into Perforce Helix server, you must check the file out. You can do this automatically or manually.

When you open a file that is checked into Helix server, you are prompted to check the file out. To check the file out at this time, click **OK**.

If you click **Cancel**, the file opens in read-only mode. You can edit this file, but you cannot save it (unless you use **Save As** and specify a different file name). To edit and save a file that is in Helix server, you must first check it out.

**To check a file out of Perforce Helix server:**

1. Open the file.
2. Select **Perforce > Check Out**.

> **Note**
> By default, you are prompted to check files out after you open them. You can configure P4GT not to prompt by disabling the **When opening a file, check out from Perforce** option.

For troubleshooting help, see Troubleshooting: Checking files out. If you are working offline (without network access), see Working offline for details about checking out files offline.

## Check files into Perforce Helix server

When you are finished working on a file that has been checked out of Perforce Helix server, you need to check it back in, to ensure that your changes are stored in the depot.

**To check a file into Perforce Helix server:**

1. Select **Perforce > Check in**. The **Submit** dialog opens.
2. In the **Description** field, enter in a brief description of the work you have done on the file. Note that you must enter something into this field before submitting.
3. If prompted, choose the files that you want to check in. The file that you are working on is automatically selected.
4. If you want to check in your changes and immediately check the file out again (for example, to

save a interim version), check **Keep checked out**.

5. Click **Submit**.

> **Note**
> You can change the status of Perforce Helix server jobs (which track work to be done) when you submit a file. To close jobs using P4GT, you must either set a jobview or attach jobs to changelists. These tasks must be done outside of P4GT. Consult your Helix server administrator for details.

**To discard changes:**

To discard your changes and restore the version you checked out, select **Perforce> Undo Add/Checkout**.

# Work offline

Typically, your computer is connected to the network and has access to Helix server. However, if you work remotely, you might need to be able to work without network access. P4GT supports working offline.

To work offline, select **Perforce > Disconnect from Server**. To restore the connection, select **Perforce > Connect to Server**. Note that if you inadvertently lose network access to Helix server, P4GT displays a warning dialog and automatically disconnects. When the connection is restored, select **Perforce > Connect toServer** to resume the connection.

When working offline, you cannot check files in or out using the Perforce menu entries. If you want to work on a file that you did not check out while connected, perform the following steps:

- Before editing a file: Using Windows Explorer, browse to the file, right-click it, and select **Properties**. Clear the read-only attribute and click **OK**.
- After editing the file: Reset the read-only attribute.
- After reconnecting: Open the file and select **Perforce > Check out**. At this point, you can make more changes or check the file in.

# Configure P4GT

To configure P4GT, select **Helix Core > Options**, configure settings as desired, and then click **OK**. You can configure the following settings.

## *Check In/Check Out tab*

**When adding a file to** Perforce**, check in immediately:**

- To automatically check in a file that you have just added to Helix server, select **Always**.
- To disable automatic check-in, select **Never**. Files are not checked in until you select **Perforce > Check In**.
- To display a "Check In?" prompt after you add a file, select **Prompt**.

**When checking out a file, automatically lock it:** When you lock a file, other users can check the file out, but cannot check it in until you unlock it, check it in, or revert it.

- To lock files when you check them out, select **Always**.
- To disable automatic locking, select **Never**.
- To display a "Lock?" prompt, select **Prompt**.

**When checking in a file:** To list all checked-out files when you check a file in, select **View other files in changelist**. Selecting this option enables you to check in multiple files at a time.

**Informational prompts:** Enables you to configure the level of verbosity for P4GT. Select to display confirmation of successful operations (errors are always displayed, regardless of this setting).

## *Open/Close tab*

**When opening a file, check out from** Perforce**:**

- To automatically check out files when you open them, select **Always**.
- To disable automatic check-out, select **Never**.
- To display a "Check Out?" prompt, select **Prompt**.

**When reopening a file after undo (reverting), check out from Perforce:**

- To automatically check out files when you reopen them after reverting, select **Always**.
- To disable automatic check-out, select **Never**.
- To display a "Check Out?" prompt, select **Prompt**.

**When closing a file, check in to Perforce:**

- To automatically check in files when you close them, select **Always**.
- To disable automatic check-in, select **Never**.
- To display a "Check In?" prompt, select **Prompt**.

**When opening a file, update external file references:** Automatically syncs the head revision of any external files referenced by the main file. If other users are working on files that are referenced by your files, you can ensure that you always have the most recent version of their files by enabling this option.

- To automatically sync any external files when you open a file, select **Always**.
- To disable automatic syncing, select **Never**.
- To display a "Sync?" prompt, select **Prompt**.

**When closing a file, prompt to revert file from** Perforce**:** This option lets you discard any changes you have made and restore the file from the depot to your workspace.

> **Note**
> Most of these options can also be configured through the **Do not show this dialog again** checkbox that is displayed on dialogs in P4GT.

# Troubleshoot P4GT

This section includes the following topics:

**Troubleshoot adding files** ..... **14**
**Troubleshoot checking out files** ..... **14**

## Troubleshoot adding files

**When I try to add a file to Perforce Helix server, I get an error stating that the file is not "under client's root." What does this mean?**

To add a file to Perforce Helix server successfully, the file must be saved in your client workspace. If you do not know what this means, see your Perforce Helix server administrator.

## Troubleshoot checking out files

**When I try to check out a file from Perforce Helix server,I get an error stating that the "newest version in the Helix Core depot has been deleted." What does this mean?**

The file that you are trying to check out has been deleted from Perforce Helix server. If you are familiar with Helix server, you can use P4V or the Helix server command-line client to undo the delete. If you are not familiar with Helix server, contact your Helix server administrator.

You cannot access this file in Helix server until the file has been recovered.

# Glossary

## A

### access level

A permission assigned to a user to control which commands the user can execute. See also the 'protections' entry in this glossary and the 'p4 protect' command in the P4 Command Reference.

### admin access

An access level that gives the user permission to privileged commands, usually super privileges.

### APC

The Alternative PHP Cache, a free, open, and robust framework for caching and optimizing PHP intermediate code.

### archive

1. For replication, versioned files (as opposed to database metadata). 2. For the 'p4 archive' command, a special depot in which to copy the server data (versioned files and metadata).

### atomic change transaction

Grouping operations affecting a number of files in a single transaction. If all operations in the transaction succeed, all the files are updated. If any operation in the transaction fails, none of the files are updated.

### avatar

A visual representation of a Swarm user or group. Avatars are used in Swarm to show involvement in or ownership of projects, groups, changelists, reviews, comments, etc. See also the "Gravatar" entry in this glossary.

## B

### base

For files: The file revision, in conjunction with the source revision, used to help determine what integration changes should be applied to the target revision. For checked out streams: The public have version from which the checked out version is derived.

### binary file type

A Helix server file type assigned to a non-text file. By default, the contents of each revision are stored in full, and file revision is stored in compressed format.

### branch

(noun) A set of related files that exist at a specific location in the Perforce depot as a result of being copied to that location, as opposed to being added to that location. A group of related files is often referred to as a codeline. (verb) To create a codeline by copying another codeline with the 'p4 integrate', 'p4 copy', or 'p4 populate' command.

### branch form

The form that appears when you use the 'p4 branch' command to create or modify a branch specification.

### branch mapping

Specifies how a branch is to be created or integrated by defining the location, the files, and the exclusions of the original codeline and the target codeline. The branch mapping is used by the integration process to create and update branches.

### branch view

A specification of the branching relationship between two codelines in the depot. Each branch view has a unique name and defines how files are mapped from the originating codeline to the target codeline. This is the same as branch mapping.

### broker

Helix Broker, a server process that intercepts commands to the Helix server and is able to run scripts on the commands before sending them to the Helix server.

## C

### change review

The process of sending email to users who have registered their interest in changelists that include specified files in the depot.

### changelist

A list of files, their version numbers, the changes made to the files, and a description of the changes made. A changelist is the basic unit of versioned work in Helix server. The changes specified in the changelist are not stored in the depot until the changelist is submitted to the depot. See also atomic change transaction and changelist number.

### changelist form

The form that appears when you modify a changelist using the 'p4 change' command.

### changelist number

An integer that identifies a changelist. Submitted changelist numbers are ordinal (increasing), but not necessarily consecutive. For example, 103, 105, 108, 109. A pending changelist number might be assigned a different value upon submission.

### check in

To submit a file to the Helix server depot.

### check out

To designate one or more files, or a stream, for edit.

### checkpoint

A backup copy of the underlying metadata at a particular moment in time. A checkpoint can recreate db.user, db.protect, and other db.* files. See also metadata.

### classic depot

A repository of Helix server files that is not streams-based. Uses the Perforce file revision model, not the graph model. The default depot name is depot. See also default depot, stream depot, and graph depot.

### client form

The form you use to define a client workspace, such as with the 'p4 client' or 'p4 workspace' commands.

### client name

A name that uniquely identifies the current client workspace. Client workspaces, labels, and branch specifications cannot share the same name.

### client root

The topmost (root) directory of a client workspace. If two or more client workspaces are located on one machine, they should not share a client root directory.

### client side

The right-hand side of a mapping within a client view, specifying where the corresponding depot files are located in the client workspace.

### client workspace

Directories on your machine where you work on file revisions that are managed by Helix server. By default, this name is set to the name of the machine on which your client workspace is located, but it can be overridden. Client workspaces, labels, and branch specifications cannot share the same name.

### code review

A process in Helix Swarm by which other developers can see your code, provide feedback, and approve or reject your changes.

### codeline

A set of files that evolve collectively. One codeline can be branched from another, allowing each set of files to evolve separately.

### comment

Feedback provided in Helix Swarm on a changelist, review, job, or a file within a changelist or review.

### commit server

A server that is part of an edge/commit system that processes submitted files (checkins), global workspaces, and promoted shelves.

### conflict

1. A situation where two users open the same file for edit. One user submits the file, after which the other user cannot submit unless the file is resolved. 2. A resolve where the same line is changed when merging one file into another. This type of conflict occurs when the comparison of two files to a base yields different results, indicating that the files have been changed in different ways. In this case, the merge cannot be done automatically and must be resolved manually. See file conflict.

### copy up

A Helix server best practice to copy (and not merge) changes from less stable lines to more stable lines. See also merge.

### counter

A numeric variable used to track variables such as changelists, checkpoints, and reviews.

### CSRF

Cross-Site Request Forgery, a form of web-based attack that exploits the trust that a site has in a user's web browser.

## D

### default changelist

The changelist used by a file add, edit, or delete, unless a numbered changelist is specified. A default pending changelist is created automatically when a file is opened for edit.

### deleted file

In Helix server, a file with its head revision marked as deleted. Older revisions of the file are still available. in Helix server, a deleted file is simply another revision of the file.

### delta

The differences between two files.

### depot

A file repository hosted on the server. A depot is the top-level unit of storage for versioned files (depot files or source files) within a Helix Core server. It contains all versions of all files ever submitted to the depot. There can be multiple depots on a single installation.

### depot root

The topmost (root) directory for a depot.

### depot side

The left side of any client view mapping, specifying the location of files in a depot.

### depot syntax

Helix server syntax for specifying the location of files in the depot. Depot syntax begins with: //depot/

### diff

(noun) A set of lines that do not match when two files, or stream versions, are compared. A conflict is a pair of unequal diffs between each of two files and a base, or between two versions of a stream. (verb) To compare the contents of files or file revisions, or of stream versions. See also conflict.

### donor file

The file from which changes are taken when propagating changes from one file to another.

## E

### edge server

A replica server that is part of an edge/commit system that is able to process most read/write commands, including 'p4 integrate', and also deliver versioned files (depot files).

### exclusionary access

A permission that denies access to the specified files.

### exclusionary mapping

A view mapping that excludes specific files or directories.

### extension

Similar to a trigger, but more modern. See "Helix Core Server Administrator Guide" on "Extensions".

## F

### file conflict

In a three-way file merge, a situation in which two revisions of a file differ from each other and from their base file. Also, an attempt to submit a file that is not an edit of the head revision of the file in the depot, which typically occurs when another user opens the file for edit after you have opened the file for edit.

### file pattern

Helix server command line syntax that enables you to specify files using wildcards.

### file repository

The master copy of all files, which is shared by all users. In Helix server, this is called the depot.

### file revision

A specific version of a file within the depot. Each revision is assigned a number, in sequence. Any revision can be accessed in the depot by its revision number, preceded by a pound sign (#), for example testfile#3.

### file tree

All the subdirectories and files under a given root directory.

### file type

An attribute that determines how Helix server stores and diffs a particular file. Examples of file types are text and binary.

### fix

A job that has been closed in a changelist.

### form

A screen displayed by certain Helix server commands. For example, you use the change form to enter comments about a particular changelist to verify the affected files.

### forwarding replica

A replica server that can process read-only commands and deliver versioned files (depot files). One or more replicate servers can significantly improve performance by offloading some of the master server load. In many cases, a forwarding replica can become a disaster recovery server.

## G

### Git Fusion

A Perforce product that integrates Git with Helix, offering enterprise-ready Git repository management, and workflows that allow Git and Helix server users to collaborate on the same projects using their preferred tools.

### graph depot

A depot of type graph that is used to store Git repos in the Helix server. See also Helix4Git and classic depot.

### group

A feature in Helix server that makes it easier to manage permissions for multiple users.

## H

### have list

The list of file revisions currently in the client workspace.

### head revision

The most recent revision of a file within the depot. Because file revisions are numbered sequentially, this revision is the highest-numbered revision of that file.

### heartbeat

A process that allows one server to monitor another server, such as a standby server monitoring the master server (see the p4 heartbeat command).

### Helix server

The Helix server depot and metadata; also, the program that manages the depot and metadata, also called Helix Core server.

### Helix TeamHub

A Perforce management platform for code and artifact repository. TeamHub offers built-in support for Git, SVN, Mercurial, Maven, and more.

### Helix4Git

Perforce solution for teams using Git. Helix4Git offers both speed and scalability and supports hybrid environments consisting of Git repositories and 'classic' Helix server depots.

### hybrid workspace

A workspace that maps to files stored in a depot of the classic Perforce file revision model as well as to files stored in a repo of the graph model associated with git.

## I

### iconv

A PHP extension that performs character set conversion, and is an interface to the GNU libiconv library.

### integrate

To compare two sets of files (for example, two codeline branches) and determine which changes in one set apply to the other, determine if the changes have already been propagated, and propagate any outstanding changes from one set to another.

## J

### job

A user-defined unit of work tracked by Helix server. The job template determines what information is tracked. The template can be modified by the Helix server system administrator. A job describes work to be done, such as a bug fix. Associating a job with a changelist records which changes fixed the bug.

### job daemon

A program that checks the Helix server machine daily to determine if any jobs are open. If so, the daemon sends an email message to interested users, informing them the number of jobs in each category, the severity of each job, and more.

### job specification

A form describing the fields and possible values for each job stored in the Helix server machine.

### job view

A syntax used for searching Helix server jobs.

### journal

A file containing a record of every change made to the Helix server's metadata since the time of the last checkpoint. This file grows as each Helix server transaction is logged. The file should be automatically truncated and renamed into a numbered journal when a checkpoint is taken.

### journal rotation

The process of renaming the current journal to a numbered journal file.

### journaling

The process of recording changes made to the Helix server's metadata.

## L

### label

A named list of user-specified file revisions.

### label view

The view that specifies which filenames in the depot can be stored in a particular label.

### lazy copy

A method used by Helix server to make internal copies of files without duplicating file content in the depot. A lazy copy points to the original versioned file (depot file). Lazy copies minimize the consumption of disk space by storing references to the original file instead of copies of the file.

### license file

A file that ensures that the number of Helix server users on your site does not exceed the number for which you have paid.

### list access

A protection level that enables you to run reporting commands but prevents access to the contents of files.

### local depot

Any depot located on the currently specified Helix server.

### local syntax

The syntax for specifying a filename that is specific to an operating system.

### lock

1. A file lock that prevents other clients from submitting the locked file. Files are unlocked with the 'p4 unlock' command or by submitting the changelist that contains the locked file. 2. A database lock that prevents another process from modifying the database db.* file.

### log

Error output from the Helix server. To specify a log file, set the P4LOG environment variable or use the p4d -L flag when starting the service.

## M

### mapping

A single line in a view, consisting of a left side and a right side that specify the correspondences between files in the depot and files in a client, label, or branch. See also workspace view, branch view, and label view.

### MDS checksum

The method used by Helix server to verify the integrity of versioned files (depot files).

### merge

1. To create new files from existing files, preserving their ancestry (branching). 2. To propagate changes from one set of files to another. 3. The process of combining the contents of two conflicting file revisions into a single file, typically using a merge tool like P4Merge.

### merge file

A file generated by the Helix server from two conflicting file revisions.

### metadata

The data stored by the Helix server that describes the files in the depot, the current state of client workspaces, protections, users, labels, and branches. Metadata is stored in the Perforce database and is separate from the archive files that users submit.

### modification time or modtime

The time a file was last changed.

### MPM

Multi-Processing Module, a component of the Apache web server that is responsible for binding to network ports, accepting requests, and dispatch operations to handle the request.

## N

### nonexistent revision

A completely empty revision of any file. Syncing to a nonexistent revision of a file removes it from your workspace. An empty file revision created by deleting a file and the #none revision specifier are examples of nonexistent file revisions.

### numbered changelist

A pending changelist to which Helix server has assigned a number.

## O

### opened file

A file you have checked out in your client workspace as a result of a Helix Core server operation (such as an edit, add, delete, integrate). Opening a file from your operating system file browser is not tracked by Helix Core server.

### owner

The Helix server user who created a particular client, branch, or label.

## P

### p4

1. The Helix Core server command line program. 2. The command you issue to execute commands from the operating system command line.

### p4d

The program that runs the Helix server; p4d manages depot files and metadata.

### P4PHP

The PHP interface to the Helix API, which enables you to write PHP code that interacts with a Helix server machine.

### PECL

PHP Extension Community Library, a library of extensions that can be added to PHP to improve and extend its functionality.

### pending changelist

A changelist that has not been submitted.

### Perforce

Perforce Software, Inc., a leading provider of enterprise-scale software solutions to technology developers and development operations (“DevOps”) teams requiring productivity, visibility, and scale during all phases of the development lifecycle.

### project

In Helix Swarm, a group of Helix server users who are working together on a specific codebase, defined by one or more branches of code, along with options for a job filter, automated test integration, and automated deployment.

### protections

The permissions stored in the Helix server’s protections table.

### proxy server

A Helix server that stores versioned files. A proxy server does not perform any commands. It serves versioned files to Helix server clients.

## R

### RCS format

Revision Control System format. Used for storing revisions of text files in versioned files (depot files). RCS format uses reverse delta encoding for file storage. Helix server uses RCS format to store text files. See also reverse delta storage.

### read access

A protection level that enables you to read the contents of files managed by Helix server but not make any changes.

### remote depot

A depot located on another Helix server accessed by the current Helix server.

### replica

A Helix server that contains a full or partial copy of metadata from a master Helix server. Replica servers are typically updated every second to stay synchronized with the master server.

### repo

A graph depot contains one or more repos, and each repo contains files from Git users.

### reresolve

The process of resolving a file after the file is resolved and before it is submitted.

### resolve

The process you use to manage the differences between two revisions of a file, or two versions of a stream. You can choose to resolve file conflicts by selecting the source or target file to be submitted, by merging the contents of conflicting files, or by making additional changes. To resolve stream conflicts, you can choose to accept the public source, accept the checked out target, manually accept changes, or combine path fields of the public and checked out version while accepting all other changes made in the checked out version.

**reverse delta storage**

The method that Helix server uses to store revisions of text files. Helix server stores the changes between each revision and its previous revision, plus the full text of the head revision.

**revert**

To discard the changes you have made to a file in the client workspace before a submit.

**review access**

A special protections level that includes read and list accesses and grants permission to run the p4 review command.

**review daemon**

A program that periodically checks the Helix server machine to determine if any changelists have been submitted. If so, the daemon sends an email message to users who have subscribed to any of the files included in those changelists, informing them of changes in files they are interested in.

**revision number**

A number indicating which revision of the file is being referred to, typically designated with a pound sign (#).

**revision range**

A range of revision numbers for a specified file, specified as the low and high end of the range. For example, myfile#5,7 specifies revisions 5 through 7 of myfile.

**revision specification**

A suffix to a filename that specifies a particular revision of that file. Revision specifiers can be revision numbers, a revision range, change numbers, label names, date/time specifications, or client names.

**RPM**

RPM Package Manager. A tool, and package format, for managing the installation, updates, and removal of software packages for Linux distributions such as Red Hat Enterprise Linux, the Fedora Project, and the CentOS Project.

## S

### server data

The combination of server metadata (the Helix server database) and the depot files (your organization's versioned source code and binary assets).

### server root

The topmost directory in which p4d stores its metadata (db.* files) and all versioned files (depot files or source files). To specify the server root, set the P4ROOT environment variable or use the p4d -r flag.

### service

In the Helix Core server, the shared versioning service that responds to requests from Helix server client applications. The Helix server (p4d) maintains depot files and metadata describing the files and also tracks the state of client workspaces.

### shelve

The process of temporarily storing files in the Helix server without checking in a changelist.

### status

For a changelist, a value that indicates whether the changelist is new, pending, or submitted. For a job, a value that indicates whether the job is open, closed, or suspended. You can customize job statuses. For the 'p4 status' command, by default the files opened and the files that need to be reconciled.

### storage record

An entry within the db.storage table to track references to an archive file.

### stream

A "branch" with built-in rules that determines what changes should be propagated and in what order they should be propagated.

### stream depot

A depot used with streams and stream clients. Has structured branching, unlike the free-form branching of a "classic" depot. Uses the Perforce file revision model, not the graph model. See also classic depot and graph depot.

### submit

To send a pending changelist into the Helix server depot for processing.

### super access

An access level that gives the user permission to run every Helix server command, including commands that set protections, install triggers, or shut down the service for maintenance.

### symlink file type

A Helix server file type assigned to symbolic links. On platforms that do not support symbolic links, symlink files appear as small text files.

### sync

To copy a file revision (or set of file revisions) from the Helix server depot to a client workspace.

## T

### target file

The file that receives the changes from the donor file when you integrate changes between two codelines.

### text file type

Helix server file type assigned to a file that contains only ASCII text, including Unicode text. See also binary file type.

### theirs

The revision in the depot with which the client file (your file) is merged when you resolve a file conflict. When you are working with branched files, theirs is the donor file.

### three-way merge

The process of combining three file revisions. During a three-way merge, you can identify where conflicting changes have occurred and specify how you want to resolve the conflicts.

### trigger

A script that is automatically invoked by Helix server when various conditions are met. (See "Helix Core Server Administrator Guide" on "Triggers".)

### two-way merge

The process of combining two file revisions. In a two-way merge, you can see differences between the files.

### typemap

A table in Helix server in which you assign file types to files.

## U

### user

The identifier that Helix server uses to determine who is performing an operation. The three types of users are standard, service, and operator.

## V

### versioned file

Source files stored in the Helix server depot, including one or more revisions. Also known as an archive file. Versioned files typically use the naming convention 'filenamev' or '1.changelist.gz'.

### view

A description of the relationship between two sets of files. See workspace view, label view, branch view.

## W

### wildcard

A special character used to match other characters in strings. The following wildcards are available in Helix server: * matches anything except a slash; ... matches anything including slashes; %%0 through %%9 is used for parameter substitution in views.

### workspace

See client workspace.

### workspace view

A set of mappings that specifies the correspondence between file locations in the depot and the client workspace.

### write access

A protection level that enables you to run commands that alter the contents of files in the depot. Write access includes read and list accesses.

## X

### XSS

Cross-Site Scripting, a form of web-based attack that injects malicious code into a user's web browser.

## Y

### yours

The edited version of a file in your client workspace when you resolve a file. Also, the target file when you integrate a branched file.