



HelixCore

P4Thumb User Guide

2020.1
April 2020

PERFORCE

www.perforce.com



Copyright © 2008-2020 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.

Contents

How to use this guide	4
Syntax conventions	4
Feedback	4
Other documentation	4
Introduction	5
Installation	6
Supported platforms	6
Disk space and performance	6
Installation steps	6
Windows	6
Linux	8
Configuration	10
Built-in conversions of Qt-supported image formats	10
Overwriting or extending of built-in formats	11
Configuration blocks	12
Testing conversion definitions	13
Startup and shutdown	15

How to use this guide

This guide tells you how to install and configure P4Thumb, the Helix Thumbnail Generator. It is intended for anyone tasked with adding thumbnail views to P4V.

Syntax conventions

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

Notation	Meaning
<code>literal</code>	Must be used in the command exactly as shown.
<i>italics</i>	A parameter for which you must supply specific information. For example, for a <i>serverid</i> parameter, supply the ID of the server.
<code>[-f]</code>	The enclosed elements are optional. Omit the brackets when you compose the command.
<code>...</code>	Previous argument can be repeated. <ul style="list-style-type: none">▪ <code>p4 [g-opts] streamlog [-l -L -t -m max] stream1 ...</code> means <code>1</code> or more stream arguments separated by a space▪ See also the use on <code>...</code> in Command alias syntax in the <i>Helix Core P4 Command Reference</i>
<code>element1 element2</code>	Either <i>element1</i> or <i>element2</i> is required.

Tip

`...` 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>.

Introduction

P4Thumb, the Helix Thumbnail Generator, creates thumbnails of graphic files managed by Helix Core and stores them in the Helix Core server, also called the Helix server. A thumbnail is a 160x160 pixel PNG representation of the image. P4V, the Helix Visual Client, displays the thumbnails in the right pane when you choose **View > Show Files As > (Large/Medium/Small) Thumbnail**.

P4V does not communicate directly with P4Thumb; instead, it obtains thumbnails from the server.

By default, P4V displays thumbnails for image files located in the workspace. If you install and configure P4Thumb, P4V can also display thumbnails for any revisions of graphic files stored in the server.

Before running P4Thumb, you need to create a dedicated client (workspace). This client only maps directories containing image files that require thumbnails. A thumbnail is stored as a Revision attribute named `thumb`. P4V recognizes this attribute. You can read this attribute with the `fstat` function (for example: `p4 fstat -Oae -A thumb //depot/images/luftballon.jpg#1`).

When you start P4Thumb against a server for the first time, it creates thumbnails for all graphic file versions in its client view and then adds thumbnails for any new image versions as they get added to the depot.

Note

Make sure to start only one P4Thumb server. Starting two servers that map the same directories in their respective clients would result in two competing P4Thumb servers managing the same set of attributes.

Installation

This section provides information on supported platforms, disk space, performance, and the individual installation steps.

Supported platforms

- Windows 7, 8.1, 10 (x86, x86_64)
- Linux (x86_64)
 - RHEL 6.6+
 - CentOS 6.6+
 - Ubuntu 16.04+

P4Thumb 2017.3 works with Helix Core version 2016.1 or later.

Disk space and performance

The thumbnails are stored in both the journal and checkpoint files in hexadecimal format. Ensure that the server machine running P4Thumb has adequate disk space because checkpointing, restoring from checkpoints, and journaling run more slowly as a result, and the resulting files are larger.

Running P4Thumb does not noticeably slow down Helix server, but the retrieval and display of thumbnails does have some effect on P4V performance.

Installation steps

You can install P4Thumb on the same machine as Helix server. However, to avoid affecting the performance of Helix server, it is recommended that you run P4Thumb on a separate machine. If P4Thumb runs on a separate machine, it requires network access to the machine where the Helix server runs.

Windows

1. Download the `p4thumb.zip` file from:
<https://www.perforce.com/downloads/helix-thumbnail-generator-p4thumb>
2. Unzip the file on the machine where you intend to run it.
3. Make sure that the vs11 and vs12 redistributables are installed on the P4Thumb machine.
If they are not present, install them by running `installredist.bat`.

4. Create a workspace for P4Thumb that maps only the files for which you want thumbnails to be created.

For example, this view maps all of the image formats that Qt supports, plus formats supported by plugins to a client named `<p4thumb>` (substitute this with your client name):

```
//depot/....ppm //p4thumb/....ppm
//depot/....PPM //p4thumb/....PPM
//depot/....bmp //p4thumb/....bmp
//depot/....BMP //p4thumb/....BMP
//depot/....jpg //p4thumb/....jpg
//depot/....JPG //p4thumb/....JPG
//depot/....bpm //p4thumb/....bpm
//depot/....BPM //p4thumb/....BPM
//depot/....gif //p4thumb/....gif
//depot/....GIF //p4thumb/....GIF
//depot/....pgm //p4thumb/....pgm
//depot/....PGM //p4thumb/....PGM
//depot/....png //p4thumb/....png
//depot/....PNG //p4thumb/....PNG
//depot/....mng //p4thumb/....mng
//depot/....MNG //p4thumb/....MNG
//depot/....xbm //p4thumb/....xbm
//depot/....XBM //p4thumb/....XBM
//depot/....xpm //p4thumb/....xpm
//depot/....XPM //p4thumb/....XPM
//depot/....max //p4thumb/....max
//depot/....MAX //p4thumb/....MAX
//depot/....ma //p4thumb/....ma
//depot/....MA //p4thumb/....MA
//depot/....mb //p4thumb/....mb
//depot/....MB //p4thumb/....MB
//depot/....tga //p4thumb/....tga
//depot/....TGA //p4thumb/....TGA
//depot/....psd //p4thumb/....psd
//depot/....PSD //p4thumb/....PSD
```

Linux

1. Download the `p4thumb.tgz` file from:

<https://www.perforce.com/downloads>

2. Uncompress the file with `gzip` and extract the runtime files into the directory of your choice.

You can create a symbolic link that points to the `p4thumb` executable script. For example, to create the link where the distribution files were untarred, issue the following commands:

```
$ cd /foo
$ tar xvf p4thumb.tgz
$ ln -s foo/p4thumb/bin/p4thumb /usr/local/bin/p4thumb
```

3. Create a workspace for P4Thumb that maps only the files for which you want thumbnails to be created.

For example, this view maps all of the image formats that Qt supports, plus formats supported by plugins to a client named `<p4thumb>` (substitute this with your client name):

```
//depot/....ppm //p4thumb/....ppm
//depot/....PPM //p4thumb/....PPM
//depot/....bmp //p4thumb/....bmp
//depot/....BMP //p4thumb/....BMP
//depot/....jpg //p4thumb/....jpg
//depot/....JPG //p4thumb/....JPG
//depot/....bpm //p4thumb/....bpm
//depot/....BPM //p4thumb/....BPM
//depot/....gif //p4thumb/....gif
//depot/....GIF //p4thumb/....GIF
//depot/....pgm //p4thumb/....pgm
//depot/....PGM //p4thumb/....PGM
//depot/....png //p4thumb/....png
//depot/....PNG //p4thumb/....PNG
//depot/....mng //p4thumb/....mng
//depot/....MNG //p4thumb/....MNG
//depot/....xbm //p4thumb/....xbm
//depot/....XBM //p4thumb/....XBM
//depot/....xpm //p4thumb/....xpm
//depot/....XPM //p4thumb/....XPM
//depot/....max //p4thumb/....max
```



```
//depot/....MAX //p4thumb/....MAX
//depot/....ma //p4thumb/....ma
//depot/....MA //p4thumb/....MA
//depot/....mb //p4thumb/....mb
//depot/....MB //p4thumb/....MB
//depot/....tga //p4thumb/....tga
//depot/....TGA //p4thumb/....TGA
//depot/....psd //p4thumb/....psd
//depot/....PSD //p4thumb/....PSD
```

4. Create the configuration file for startup of P4Thumb.

A configuration file is located under: `./doc/thumbconfig.json`

For the default QT configuration, only the setting section is required, as follows:

```
{
  "settings": {
    "connection": {
      "client": "<workspace name>",
      "port": "<server:port>",
      "user": "<thumbnail user>"
    },
    "logFile": "<path to log file>",
    "nativeSize": {
      "height": <number of pixels default=160>,
      "width": <number of pixels default=160>
    },
    "pollInterval": <number of seconds default=30>,
    "maxFileSize" : <size in bytes default no-limit>
  }
}
```

For details, see ["Configuration" on page 10](#).

5. Start P4Thumb using the configuration file, as follows:

```
p4thumb -c <config files>
```

Configuration

P4Thumb supports the following modes:

- Built-in conversions of Qt-supported image formats
- Overwriting or extending of built-in formats with custom conversions

The following sections describe these formats.

Built-in conversions of Qt-supported image formats

To support this mode, you need to provide a configuration file that defines all settings needed to run P4Thumb.

The following code snippet is a configuration template for supporting built-in formats only:

```
{
  "settings": {
    "connection": {
      "client": "<workspace name>",
      "port": "<server:port>",
      "user": "<thumbnail user>"
    },
    "logFile": "<path to log file>",
    "nativeSize": {
      "height": <number of pixels default=160>,
      "width": <number of pixels default=160>
    },
    "pollInterval": <number of seconds default=30>,
    "maxFileSize" : <size in bytes default no-limit>
  }
}
```

This configuration only includes the mandatory 'settings' block. You find an explanation of the settings after the following example of a configuration file that adds customized conversions using 3rd party tools of your choice.

Overwriting or extending of built-in formats

To create thumbnails, you can use external tools, such as ImageMagick, XnConvert, TinyPng, Imagine, or Image Tuner, provided the generated thumbnails are in a format supported by Qt and stored as PNG files in Helix Core server.

Using an external tool requires that you register a **conversion** for a set of file extensions. A **conversion** defines the executable to be called and the parameters required to perform the transformation/resizing. If a revision's file extension matches a registered file extension, P4Thumb uses the **conversion** to generate a thumbnail.

P4Thumb delivers the following example configuration files:

- **thumbconfig.json**: Uses ImageMagick and Ghostscript to extend P4Thumb for chosen extensions on Linux
- **winconfig.json**: Uses XnView on Windows

You can also use your own tool to do the conversion for you. The only requirement is that you provide **\$FILEIN** and **\$FILEOUT** as parameters. P4Thumb provides the value for **\$FILEIN** (revision) and **\$FILEOUT** (thumb).

The following code snippet is a configuration template:

```
{
  "settings": {
    "connection": {
      "client": "<workspace name>",
      "port": "<server:port>",
      "user": "<thumbnail user>"
    },
    "logFile": "<path to log file>",
    "nativeSize": {
      "height": <number of pixels default=160>,
      "width": <number of pixels default=160>
    },
    "pollInterval": <number of seconds default=30>,
    "maxFileSize" : <size in bytes default no-limit>
  },
  "conversions": [
    {
      "extensions": [
        <file extension>,
        <file extension>
      ],
    }
  ]
}
```

```

        "execPath": "<CONVERSION EXECUTABLE>",
        "arguments": [
            "<argument>",
            "<argument>",
            "$FILEIN",
            "$FILEOUT"
        ],
        "convertToPng": <true/false default=true>,
        "thumbnailExtension": <file extension>
    }
}

```

Configuration blocks

A configuration includes the following blocks:

- **Settings**: Mandatory.
- **Conversions**: Optional. Only to be used when using P4Thumb for non-Qt formats.

The following table describes the available parameters.

Parameter	Description
<code>settings::connection</code>	The Helix Core connection and client used by P4Thumb.
<code>settings::logFile</code>	Optional. The log file used to output P4Thumb operations.
<code>settings::nativeSize</code>	Optional. Specifies the size of thumbnails. By default (if not set), the thumbnail size of Qt formats is 160x160 pixels.
<code>settings::pollinterval</code>	Specifies how often P4Thumb polls the server for changes. By default (if not set), polling happens every 30 seconds.
<code>settings::maxFileSize</code>	If set, P4Thumb only creates thumbnails for revisions below the specified maximum size (in bytes). If not set, P4Thumb creates a thumbnail for every file, assuming it presents a supported format.

Parameter	Description
<code>conversions:extensions</code>	The file extensions handled by this conversion.
<code>conversions::execPath</code>	The executable being called.
<code>conversion::arguments</code>	The arguments needed. <code>\$FILEIN</code> and <code>\$FILEOUT</code> are required. P4Thumb uses them as variables.
<code>conversion::convertToPng</code>	Optional. If set to false (default), the thumbnail extension is <code>.png</code> . If set to true because the conversion program cannot convert to <code>.png</code> , P4Thumb uses the thumbnail extension specified.
<code>conversion::thumbnailExtension</code>	Optional. When <code>convertToPng</code> is set to true and this parameter is not set, the thumbnail extension is the source extension. Otherwise, if this parameter is set, the thumbnail extension is used. The generated <code>%FILEOUT%</code> now has the specified extension, such as <code>.jpg</code> .

Note
The thumbnail extension needs to be a format supported by Qt. P4Thumb loads the file, converts it to `.png`, and saves it as a `.png` thumbnail.

Testing conversion definitions

It is a good idea to test conversion definitions first, without adding thumbnail attributes to file revisions. The following command executes the conversion and writes the result to a local file:

```
$ p4thumb -c <config file> -test <file name>
```

For example, to test the following conversion:

```
{
  "extensions": [
    "gif",
    "tif",
    "tiff"
  ],
  "execPath": "/usr/bin/convert",
  "arguments": [
```

```
    "-resize",  
    "160x160",  
    "$FILEIN[0]",  
    "$FILEOUT"  
  ]  
},
```

You can run the following command:

```
$ p4thumb -c config.json -test images/gmarbles.tif
```

If the conversion succeeds, the result is similar to the following:

```
== executing command:  
"/usr/bin/convert" -resize 160x160 "/home/thumbuser/images/gmarbles.tif  
[0]" "/home/thumbuser/images/p4thumbs/gmarbles.png"  
== Conversion successful
```

The output file, **p4thumbs/gmarbles.png**, indicates the accuracy of the thumbnail result when P4Thumb uses this conversion.

If the conversion fails, it results in the following:

```
'== Conversion successful' state '== error'
```

You can use the executing command to test and then correct conversions until all conversions succeed. At that point, the configuration file is ready to be used by P4Thumb to create thumbnails.

Startup and shutdown

Run one instance of P4Thumb for each server in which you want to enable thumbnails. Do not run multiple instances of P4Thumb using the same client view and server. (You might improve performance by running multiple instances with differing client views, for example one instance for each large project.)

The user associated with P4Thumb:

- Does not need to be dedicated to P4Thumb, but the workspace must be dedicated for use only by P4Thumb.
- Must have `admin` privileges on the server to run the daemon.

Note

The first time you start P4Thumb for a specified Helix server, it creates thumbnails for all images in its client view, which is likely to affect server performance. After the initial thumbnails are created, P4Thumb does not have a significant effect on server performance.

To start P4Thumb:

Issue the following command specifying the server containing the images files, the workspace for P4Thumb to use, and the Helix Core username associated with P4Thumb activity:

```
p4thumb -c <config files> [options]
```

The following table describes the available switches.

Switch	Description
<code>-v</code>	Writes verbose status output to console. Default value is <code>silent</code> .
<code>-d</code>	Deletes thumbnails from server. Must use with <code>-n</code> (a <code>ChangeListRange</code> in the configuration file).
<code>-f</code>	Forces to always write thumbnail. Default skips if set.
<code>-v</code>	Prints application version and exits.
<code>-test</code> <code><testFile></code>	Tests thumbnail conversion of <code>testFile</code> and exits.

To stop P4Thumb:

In the command window used to start P4Thumb, type `CTRL-C`.

By default, when you restart P4Thumb, it resumes processing starting with the last changelist that was processed. P4Thumb maintains a counter for this purpose on the specified server. Starting P4Thumb from a specified changelist using the `-n` flag resets this counter if it exceeds the specified value. Deleting thumbnails does not affect the counter.